

Szolgáltatás orientált architektúrák információs rendszerekben

A szervezeti architektúra nézetei, nézőpontjai és tervezési módszerei

Molnár Bálint

2013

ISBN 978-963-284-469-5

TARTALOMJEGYZÉK

TARTALOMJEGYZÉK.....	1
ÁBRAJEGYZÉK.....	8
TÁBLÁZAT JEGYZÉK.....	12
DEFINÍCIÓ JEGYZÉK.....	13
1 TÖRTÉNETI ÁTTEKINTÉS.....	15
1.1 AZ INFORMATIKAI, MŰSZAKI ARCHITEKTÚRA RÖVID TÖRTÉNETE	18
1.1.1 A vaskorszak.....	19
1.1.2 Reneszánsz	20
1.1.3 Az ipari forradalom	21
1.1.4 A kozmikus felvilágosodás.....	23
2 SZERVEZETI ARCHITEKTÚRA.....	27
2.1 INFORMÁCIÓ ARCHITEKTÚRA	28
2.2 SZERVEZETI (VÁLLALATI, ÜZLETI) INFORMÁCIÓRENDSZER ARCHITEKTÚRA.....	29
2.3 MŰSZAKI, INFORMATIKAI ARCHITEKTÚRA	30
2.4 AZ ALKALMAZÁSI ARCHITEKTÚRA	30
2.5 A SZERVEZET (VÁLLALATI, ÜZLETI) ARCHITEKTÚRA ALTERNATÍV DEFINÍCIÓI.....	31
2.6 A SZERVEZET IRÁNYÍTÁS, IGAZGATÁS PARADIGMÁJA	32
2.7 A SZOFTVER ARCHITEKTÚRA.....	34
2.7.1 Az architektúra meghatározza a struktúrát.....	36
2.7.2 Az architektúra meghatározza a komponensek kommunikációját.....	36
2.7.3 Az architektúra és a nem-funkcionális követelmények	37
2.7.4 Az architektúra, mint absztrakció	37
2.7.5 Architektúrális nézet.....	38
3 A SZOLGÁLTATÁS ORIENTÁLT ARCHITEKTÚRA.....	40
3.1 A WEB SZOLGÁLTATÁS	41
3.2 A FOLYAMATMENEDZSMENT RENDSZER ELEMEI	45
3.3 SZERVEZETI (VÁLLALATI, ÜZLETI) FOLYAMAT MENEDZSMENT	46
3.4 A FOLYAMATOK KATEGORIZÁLÁSA	47
3.5 SZERVEZETI (VÁLLALATI, ÜZLETI) FOLYAMATMENEDZSMENT ÉS A SZOA.....	47
3.6 A SZOA ÉS AZ INFORMATIKA VISZONYA	48
GYORSAN VÁLTOZÓ ÜZLETI ELVÁRÁSOK KIELÉGÍTÉSÉRE IDEÁLIS	49
3.7 A SZOA INFORMATIKAI STRATÉGIA	49
3.8 A SZOLGÁLTATÁS ORIENTÁLT ARCHITEKTÚRA ALTERNATÍV DEFINÍCIÓI.....	50

3.9	SZOLGÁLTATÁSI INFRASTRUKTÚRA	50
3.9.1	A SzOA infrastruktúra részei.....	51
3.9.2	Szolgáltatási sín (Enterpris Service Bus, ESB)	52
3.9.3	SOAP és az üzenetküldés	53
3.9.4	UDDI	56
3.9.5	WSDL modellezésének alapfogalmai.....	57
3.10	SzOA IRÁNYÍTÁSA (GOVERNANCE)	62
3.11	AZ ÉRETT SzOA	62
3.12	AZ SzOA IRÁNYÍTÁS FŐ ALKOTÓELEMEI	65
3.13	A SZOLGÁLTATÁS ORIENTÁLTSAÉG ALAPELVEI	67
3.14	SZOLGÁLTATÁS-ORIENTÁLT RENDSZEREK	69
3.14.1	Szolgáltatásorientált rendszerek	70
3.14.2	Autonóm szolgáltatások	71
3.14.3	Sémák, szerződések megosztása	72
3.14.4	Irányelveken alapuló szolgáltatás kompatibilitás	73
3.15	WEB SZOLGÁLTATÁSOK MODELLEZÉSE	74
3.16	WEB SZOLGÁLTATÁS MODELLEZÉSE	75
3.16.1	Web szolgáltatás kommunikációs protokoll: SOAP.....	75
3.16.2	WSDL és SOAP közti kapcsolat.....	75
3.16.3	Web szolgáltatás publikálása.....	76
3.16.4	Stateful Web szolgáltatás modellezés.....	76
3.17	ÖSSZETETT WEB SZOLGÁLTATÁS MODELLEZÉS	78
3.17.1	BPEL alapfogalmak.....	78
3.17.2	Szolgáltatás leírás létrehozása	78
3.17.3	Üzleti folyamatok létrehozása.....	78
3.17.4	BPEL kulcselemei	79
3.17.5	Partner link	79
3.17.6	Üzleti partner.....	79
3.17.7	Végpont hivatkozás	80
3.17.8	Tevékenység	80
3.17.9	Adatkezelés.....	80
3.17.10	Összefüggés (korreláció).....	80
3.17.11	Hatókör.....	80
3.17.12	3D-s Web szolgáltatás modellezés	81
3.18	WEB SZOLGÁLTATÁSOK MODELLEZÉSÉNEK ÉRTÉKELÉSE.....	81
3.19	A SZEMANTIKUS HÁLÓ/WEB	82
3.20	METAADATOK LÉTREHOZÁSA ÉS HASZNÁLATA A SZEMANTIKUS WEBEN	83
3.21	WEB SERVICES INSPECTION LANGUAGE – WSIL.....	85

3.22	RESOURCE DESCRIPTION FRAMEWORK -RDF.....	86
3.23	WEB ONTOLOGY LANGUAGE -OWL	87
3.24	SZEMANTIKUS WEB SOLGÁLTATÁSOK	88
4	SZERVEZETI ARCHITEKTÚRA MÓDSZEREK, MEGKÖZELÍTÉSEK.....	90
4.1	ZACHMAN FÉLE SZERVEZETI ARCHITEKTÚRA KERETRENDSZER (THE ZACHMAN ENTERPRISE FRAMEWORK).....	90
4.1.1	<i>A keretrendszer oszlopainak jelentése</i>	<i>91</i>
4.1.2	<i>A keretrendszer sorainak jelentése:</i>	<i>92</i>
4.2	A CLINGER-COHEN TÖRVÉNY AZ AMERIKAI EGYESÜLT ÁLLAMOKBAN	96
4.3	C4ISRAF	96
4.4	DoDAF	97
4.5	MoDAF	97
4.6	ISO/IEC 42010 (IEEE STD 1471-2000) SZABVÁNY	97
4.7	E-KORMÁNYZATI ARCHITEKTÚRA - FEAF (FEDERAL ENTERPRISE ARCHITECTURE FRAMEWORK).....	99
5	AZ ARCHITEKTÚRA NÉZETEI ÉS NÉZŐPONTJAI A TOGAF SZERINT	102
5.1	TOGAF: ARCHITEKTÚRA FEJLESZTÉSI MÓDSZER	104
5.1.1	<i>Kezdeti szakasz céljai.....</i>	<i>104</i>
5.1.2	<i>A- Architektúra jövőképe és célkitűzései</i>	<i>106</i>
5.1.3	<i>B-Szervezeti Architektúra</i>	<i>107</i>
5.1.4	<i>C - Információ-rendszer Architektúra</i>	<i>108</i>
5.1.5	<i>D- Informatikai/ műszaki architektúra – Kimenetek (Outputs)</i>	<i>109</i>
5.1.6	<i>E- Lehetőségek és megoldások.....</i>	<i>109</i>
5.1.7	<i>F - Áttérésre tervekészítés (migrálásra)</i>	<i>110</i>
5.1.8	<i>G- Kivitelezés / megvalósítás irányítása.....</i>	<i>110</i>
5.1.9	<i>H- Architektúra változáskezelés</i>	<i>110</i>
5.1.10	<i>Architektúra fejlesztési módszer (Architecture Development Method (ADM)) –Architektúra követelmény kezelés.....</i>	<i>111</i>
5.2	ARCHITEKTÚRA NÉZETEI.....	112
5.3	A TOGAF SZERINTI ARCHITEKTÚRA MODELLEZÉS ALAPÚ MEGKÖZELÍTÉS.....	115
5.3.1	<i>Fogalmi struktúra legfontosabb elemei</i>	<i>115</i>
5.4	NÉZETEK ÉS NÉZŐPONTOK.....	124
5.4.1	<i>A legfontosabb alapfogalmak meghatározása</i>	<i>125</i>
6	MDA.....	146
6.1	MIÉRT HASZNÁLJUNK MDA-T?	147
6.1.1	<i>Hordozhatóság.....</i>	<i>147</i>
6.1.2	<i>Átjárhatóság.....</i>	<i>147</i>
6.1.3	<i>Újrafelhasználhatóság</i>	<i>148</i>

6.1.4	<i>Gyakorlat és eszközök</i>	148
6.1.5	<i>MDA és szoftver architektúra</i>	148
6.1.6	<i>MDA és nem-funkcionális követelmények</i>	149
6.1.7	<i>Model Transzformációk és Szoftver architektúra</i>	149
6.1.8	<i>A SzOA és az MDA</i>	149
6.1.9	<i>Az analitikus modellek is modellek</i>	150
7	SZÁMÍTÁSI FELHŐ (CLOUD COMPUTING)	151
7.1	A HÁLÓZATI ÉS SZERVEZETI ARCHITEKTÚRA FEJLŐDÉSE	152
7.2	SZOLGÁLTATÓVAL SZEMBEN SZABOTT KÖVETELMÉNYEK	155
7.3	A SZOLGÁLTATÓ SZOLGÁLTATÁS-NYÚJTÁS MODELLJE	155
7.4	SZOLGÁLTATÁS-KÖZPONTÚSÁG ÜGYEI	157
7.5	EGYÜTTMŰKÖDÉSI KÉPESSÉG (INTEROPERABILITÁS).....	158
7.6	SZOLGÁLTATÁS MINŐSÉGE (QOS).....	158
7.7	HIBA-TÚRÓ KÉPESSÉG	159
7.8	ADATKEZELÉS, TÁROLÁS ÉS ADATFELDOLGOZÁS	160
7.9	VIRTUALIZÁCIÓ KEZELÉSE	162
7.10	SKÁLÁZHATÓSÁG.....	163
7.11	TERHELÉS KIEGYENSÚLYOZÁS	164
7.12	A MAGÁN FELHŐ ÉS A SZERVEZETI ARCHITEKTÚRÁK KAPCSOLATA.....	164
7.13	SZERVEZET/ VÁLLALAT ÁLTAL TÁMASZTOTT KÖVETELMÉNYEK	165
7.13.1	<i>Felhő-számítástechnika üzemeltetési megoldásai</i>	165
7.13.2	<i>Biztonság</i>	167
7.13.3	<i>Biztonsági szempontok</i>	168
7.13.4	<i>Felhő-számítástechnika üzemgazdaságtana</i>	169
7.13.5	<i>Adat-migráció</i>	170
7.13.6	<i>Szervezeti, üzleti folyamatok szervezése, kezelése (Folyamatmenedzsment)</i>	172
7.13.7	<i>Külső fél bevonása és kezelése</i>	173
7.13.8	<i>Szaktudás, gyakorlati tapasztalatok átvihetősége számítási felhő környezetbe</i>	174
7.14	VÉGFELHASZNÁLÓ ÁLTAL TÁMASZTOTT KÖVETELMÉNYEK.....	174
7.14.1	<i>Felhasználó fogyasztásának mérésén alapuló számlázás</i>	174
7.14.2	<i>Felhasználó-központú személyes adatok védelme</i>	175
7.14.3	<i>Szolgáltatás szint megállapodás (Service Level Agreements (SLAs))</i>	176
7.14.4	<i>Adaptálhatóság és megtanulhatóság</i>	179
7.14.5	<i>Végfelhasználói tapasztalatok („élmény” User Experience (UX))</i>	180
7.15	A FUNKCIONÁLIS ÉS A NEM FUNKCIONÁLIS KÖVETELMÉNYEK ELEMZÉSE.....	182
7.16	A SZÁMÍTÁSI FELHŐ ÉS AZ INFORMATIKAI TUDOMÁNY JELENLEGI ÁLLÁSA.....	183
7.17	SZINERGIÁK KIMUTATÁSA ÉS KIAKNÁZÁSA.....	184

7.18	SZÁMÍTÁSI FELHŐ ARCHITEKTURÁLIS ÉPÍTŐELEMEI.....	184
7.19	A SZÁMÍTÁSI FELHŐ MODULJAI, FUNKCIONÁLIS BLOKKOK.....	185
7.19.1	<i>Szerver (kiszolgáló) gépek modul</i>	185
7.19.2	<i>Háttértároló modul (Storage module)</i>	186
7.19.3	<i>SAN kiterjesztés (extension)</i>	186
7.19.4	<i>A hálózati kapcsolatok szövődéke (Fabric modul)</i>	186
7.19.5	<i>WAN modul</i>	186
7.19.6	<i>Hálózati virtualizáció</i>	187
7.19.7	<i>1. típusú végfelhasználó - hivatali helyiségben, irodában ügyintéző</i>	187
7.19.8	<i>2. típusú végfelhasználó – mobil, külső helyszíneken dolgozó ügyintéző</i>	187
7.20	SZÁMÍTÁSI FELHŐ ÁLTAL NYÚJTHATÓ SZOLGÁLTATÁSOK TÍPUSAI.....	188
7.20.1	<i>Adattároló mint szolgáltatás</i>	189
7.20.2	<i>Adatbázis mint szolgáltatás</i>	189
7.20.3	<i>Információ mint szolgáltatás (Information-as-a-service)</i>	189
7.20.4	<i>Folyamat mint szolgáltatás (Process-as-a-service);</i>	189
7.20.5	<i>Alkalmazás mint szolgáltatás</i>	190
7.20.6	<i>Platform mint szolgáltatás (Platform-as-a-service)</i>	191
7.20.7	<i>Integrálás mint szolgáltatás (Integration-as-a-service)</i>	191
7.20.8	<i>Biztonság mint szolgáltatás (Security-as-a-service)</i>	191
7.20.9	<i>Vezetés / irányítás mint szolgáltatás (Management/governance-as-a-service MaaS and GaaS)</i> 192	
7.20.10	<i>Tesztelés mint szolgáltatás (Testing-as-a-service, TaaS)</i>	192
7.20.11	<i>Infrastruktúra mint szolgáltatás (Infrastructure-as-a-service, IaaS)</i>	192
7.21	A SZÁMÍTÁSI FELHŐ POTENCIÁLIS ELŐNYEI	193
7.22	A SZÁMÍTÁSI FELHŐ FELHASZNÁLÓI SZEMSZÖGBŐL	193
7.23	AKADÁLYOK ÉS HÁTRÁNYOK A SZÁMÍTÁSI FELHŐ ALKALMAZÁSÁVAL KAPCSOLATBAN	195
7.24	VÁLLALATI IRÁNYÍTÁSI RENDSZEREK (ERP) ÉS A SZÁMÍTÁSI FELHŐ	196
7.25	VÁLLALAT IRÁNYÍTÁSI RENDSZEREK (ERP) KIVÁLASZTÁSI KRITÉRIUMAINAK VÁLTOZÁSA A SZÁMÍTÁSI FELHŐ KONTEXTUSÁBAN.....	198
7.26	SZÁMÍTÁSI FELHŐ JÖVŐJE.....	200
8	INFORMÁCIÓ BIZTONSÁGI ARCHITEKTÚRA (INFORMATION SECURITY ARCHITECTURE)	202
8.1	PKI (PUBLIKUS KULCSÚ INFRASTRUKTÚRA) SZOLGÁLTATÁSOK ÉS JELLEMZŐIK	204
8.2	TANÚSÍTVÁNY ALAPÚ SZEMÉLYI AZONOSÍTÁS	208
8.3	IDŐPECSÉT (IDŐBÉLYEG) SZOLGÁLTATÁS.....	209
8.4	BIZTONSÁGOS VONALI KOMMUNIKÁCIÓ ÁLTAL IGÉNYELT MEGOLDÁSOK.....	209
8.5	KOMMUNIKÁCIÓS ÉS HÁLÓZATBIZTONSÁGI SZEMPONTOK	209
8.6	A SZOFTVER ARCHITEKTÚRA SZOLGÁLTATÁSAIVAL SZEMBEN SZABOTT ÁLTALÁNOS KÖVETELMÉNYEK	211

8.7	BIZTONSÁGI ARCHITEKTÚRA KÖVETELMÉNYEK	214
8.8	WEB SZOLGÁLTATÁSOK BIZTONSÁGI KÉRDÉSEI	215
8.8.1	<i>WS-Security</i>	216
8.8.2	<i>WS-Trust</i>	217
8.8.3	<i>XACML</i>	219
8.8.4	<i>Logikai következtetések levonása a biztonsági irányelvekkel kapcsolatban</i>	220
8.9	RENDSZEREK SZABVÁNYOS INFORMÁCIÓ-ARCHITEKTÚRA ALAPJAI	221
8.10	EGYSZERI BEJELENTKEZÉS ÉS FÖDERATÍV SZEMÉLY AZONOSÍTÁSI PROTOKOLL	225
8.10.1	<i>Alapműködési modell</i>	225
8.10.2	<i>Előnyei</i>	225
8.10.3	<i>SSO az Interneten keresztül és föderatív azonosítás</i>	226
8.10.4	<i>Modellek: Internet szolgáltatások használata egyszeri bejelentkezéssel</i>	226
8.11	SAML ALAPÚ SSO HASZNÁLATI ESETEI	231
8.11.1	<i>SAML párbeszédet részt vevői</i>	231
8.11.2	<i>Interneten/ Weben/ Világhálón keresztüli egyszeri bejelentkezés használati esete</i>	232
8.11.3	<i>A föderatív azonosítás használati esete</i>	234
8.11.4	<i>Föderáció</i>	237
8.12	JOGOSULTSÁGI VISZONYOK ÁBRÁZOLÁSÁRA SZOLGÁLÓ ARCHITEKTÚRA MEGOLDÁSOK	238
8.12.1	<i>Szerepkör és jogosultság tervezés</i>	239
8.12.2	<i>Munkafolyamat modell forgatókönyv felfogásban</i>	241
8.12.3	<i>Szerepkör tervezés, szervezés</i>	242
8.12.4	<i>A hozzáférési és kezelési jogosultsági modell tovább finomítása</i>	244
9	A SZERVEZETI (VÁLLALATI, ÜZLETI) ARCHITEKTÚRA INTEGRÁCIÓJÁNAK KÉRDÉSEI	249
9.1	INFORMÁCIÓ ARCHITEKTÚRA ÉS SZERVEZETI (VÁLLALATI, ÜZLETI) INTEGRÁCIÓ	249
9.1.1	<i>Az adatgazdálkodás/adatmenedzsment területei</i>	249
9.1.2	<i>Törzsadat-kezelés</i>	252
9.2	ALKALMAZÁSINTEGRÁCIÓ	262
9.2.1	<i>Alkalmazási rendszerek integrációja</i>	264
9.2.2	<i>Alkalmazás-integráció és szervezeti ellenállás</i>	268
9.2.3	<i>Alkalmazás-integráció és technológiai architektúra</i>	268
9.2.4	<i>Az alkalmazás-integráció tradicionális módszerei</i>	269
9.2.5	<i>Alkalmazásintegrációs köztesszoftver-technológiákkal (Middle-ware)</i>	271
9.2.6	<i>Az alkalmazásintegráció köztesszoftverek típusai</i>	276
9.2.7	<i>Alkalmazás kiszolgálók (Application Servers)</i>	287
9.2.8	<i>Szervezeti (vállalati, üzleti) folyamat vezénylő/vezérlő (Business Process Orchestrator)</i>	289
9.2.9	<i>Vállalati architektúra integráció SzOA segítségével</i>	296
10	A FOLYAMAT FOGALMA	302

10.1	SZERVEZETI (VÁLLALATI, ÜZLETI) FOLYAMATMENEDZSMENT – BUSINESS PROCESS MANAGEMENT.....	304
10.2	ÜZLETI FOLYAMATOK ÚJRASZERVEZÉSE – BUSINESS PROCESS REENGINEERING.....	308
SZERVEZETI (VÁLLALATI, ÜZLETI) FOLYAMATOK ÚJRASZERVEZÉSE (BPR)		308
10.3	BPM ÉS BPR INFORMATIKAI TÁMOGATÁSA	309
10.4	A FOLYAMATMODELLEZÉS	311
10.5	FOLYAMATMODELLEZÉSI MÓDSZERTANOK	312
10.6	PETRI HÁLÓ.....	313
10.7	ESEMÉNYVEZÉRELT FOLYAMATLÁNC – EVENT-DRIVEN PROCESS CHAIN (EPC)	315
10.7.1	Az esemény.....	317
10.7.2	A funkció.....	320
10.7.3	Szervezeti egység.....	321
10.7.4	Információobjektum	322
10.7.5	Logikai műveletek és vezérlés.....	324
10.8	SZERVEZETI FOLYAMAT MODELLEZÉS JELÖLÉSRENDSZERE- BUSINESS PROCESS MODELING NOTATION (BPMN)	325
10.8.1	Folyamat objektumai.....	327
10.9	SOA ALAPÚ BPM	337
10.9.1	Felülről-lefele BPM	338
10.9.2	Alulról-felfele BPM.....	338
10.9.3	A vállalati folyamatok és az IT implementáció közti rés áthidalása.....	339
11	FOLYAMATMODELLEZÉSI MÓDSZERTANOKAT ALKALMAZÓ ESZKÖZÖK	343
11.1	ADONIS CE 3.9	343
11.2	ARIS TOOLSET 7.0.2.....	343
11.3	MICROSOFT OFFICE VISIO PROFESSIONAL 2007	346
12	IRODALOMJEGYZÉK	347
13	ÁBRA ÉS TÁBLÁZAT MELLÉKLETEK.....	357
14	FÜGGELÉK – ALAPFOGALMAK.....	362
15	FÜGGELÉK: KIS ANGOL – MAGYAR SZÓTÁR	363
KONKRÉT FELADATRA LÉTREHOZOTT MUNKACSOPORT		394
15.1	RÖVIDÍTÉSEK	397
LÁBJEGYZETEK/VÉGJEGYZETEK		399

ÁBRAJEGYZÉK

1. ábra Műszaki konstrukciók, mint architektúrák fejlődése.....	16
2. ábra Az informatikai, műszaki architektúra fejlődése.....	20
3. ábra Stratégiai tervezés és a szervezeti informatikai architektúra tervezés folyamata.....	28
4. ábra A szervezeti (vállalati, üzleti) architektúra szerepe.....	33
5. ábra Szoftver architektúra komponensei.....	35
6. ábra Felső szintű architektúra terv.....	38
7. ábra Szoftver architektúra példák.....	38
8. ábra A szolgáltatás orientált architektúra alap sémája.....	40
9. ábra WEB szolgáltatások leírása és megtalálása.....	42
10. ábra Egyszerű üzenet sorozat.....	44
11. ábra Kommunikáció a szolgáltatások között a szabványok segítségével.....	53
12. ábra SOAP üzenet struktúra.....	54
13. ábra A WSDL alapelemei.....	58
14. ábra Web szolgáltatások szoftver architektúrája – szolgáltatás platform.....	60
15. ábra Az érett/fejlett SzOA modellje- nem minden rétegre van szükség az egyes konkrét alkalmazásokhoz.....	64
16. ábra Egy e-kormányzati SzOA referencia architektúra (Hivatkozási alap).....	65
17. ábra A Web szolgáltatások három dimenziós modellje (3 D).....	81
18. ábra A Web szolgáltatások szintaktikai (formai) és tartalmi (szemantikai) viszonyrendszere.....	83
19. ábra XML példa adatábrázolásra.....	84
20. ábra Szemantikus Web technológiák.....	85
21. ábra RDF hármasok (triplet).....	86
22. ábra Egyszerű ontológia pénzügyi tranzakciókra.....	88
23. ábra Szabályok a pénzügyi ontológiában.....	89
24. ábra Szervezeti (üzleti vállalkozás) architektúra Zachman és TOGAF keretrendszere. A TOGAF a színezett részt feedi le.....	94
25. ábra The Zachman Framework ² ™ Standards, 2008.....	95
26. ábra IEEE 1471 szabvány szerint a szoftver architektúra kulcs fontosságú fogalmi.....	99
27. ábra Az architektúra keretrendszerek kaotikus fejlődésének rendszer az Amerikai Egyesült Államok-ban.....	101
28. ábra TOGAF Szerkezete és alkotórészei.....	102
29. ábra TOGAF Irányítási keretrendszer.....	104
30. ábra TOGAF ADM módszere.....	105
31. ábra Az alap architektúra építőelemek (ABB) és szabványos építőelemek (SBB) formájában történő használata.....	107
32. ábra A szervezeti architektúra kontinuum.....	126

33. ábra Műszaki modell hivatkozási alapja (Technical Reference Model (TRM))	127
34. ábra Oldalnézet: Műszaki/technológiai hivatkozási modell — a technológiai szintű szolgáltatások	130
35. ábra Felül nézet: Műszaki/technológiai hivatkozási modell — a technológiai szintű szolgáltatások	137
36. ábra III-RM Integrált infrastruktúra hivatkozási modell szempontjából lényegtelen elemek kiszűrítve	138
37. ábra Az integrált információ infrastruktúra modell részletes ábrázolása	139
38. ábra III-RM Integrált infrastruktúra hivatkozási modell szempontjából lényegtelen elemek kiszűrítve	142
39. ábra A szervezeti (vállalati, üzleti) architektúra szervezeti (vállalati, üzleti) szolgáltatáshoz kapcsolódó komponensei.....	142
40. ábra A TOGAF szervezeti (vállalati, üzleti) architektúra fogalmi szerkezetének ábrázolása	143
41. ábra A szervezeti (vállalati, üzleti) szolgáltatásokról gondoskodó adat, alkalmazás és műszaki/informatikai architektúra komponensek.....	144
42. ábra A szervezeti (vállalati, üzleti) szolgáltatásokról gondoskodó adat, alkalmazás és műszaki/informatikai architektúra komponensek.....	144
43. ábra Szolgáltatás orientált architektúra szervezeti architektúra megközelítésben.....	145
44. ábra Modell transzformáció MDA-ban	146
45. ábra Adatfeldolgozási módok fejlődési típusai ([18] Fig.1 alapján)	153
46. ábra NIST számítási felhő modellje	154
47. ábra A számítási felhő architektúrájának 3 szintű modellje	156
48. ábra A magánfelhő szervezeti architekturális komponensei	166
49. ábra Számítási felhő szolgáltatás-orientált infrastruktúrája (SOI)	182
50. ábra A számítási felhő architekturális építőelemei.....	184
51. ábra A magánfelhő architekturális építő elemei.....	187
52. ábra A számítási felhő komponensei	190
53. ábra A számítási felhő három nagyvonalú szolgáltatás kategóriája	192
54. ábra Biztonsági architektúra	204
55. ábra A fokozott szintű elektronikus aláírás követelményeinek megfelelő aláírás szerkezete a ETSI szabvány szerint.....	205
56. ábra A nyilvános kulcsú (PKI) infrastruktúra elemei.....	206
57. ábra Személy azonosítás logikai/technológiai architektúra komponensek.....	208
58. ábra Azonosítási és hitelesítési szolgáltatások adatfolyamai	210
60. ábra Példa: Az információ-architektúra biztonsági elemzésére	224
61. ábra Föderatív azonosítás és egyszeri bejelentkezés alap architektúrája.....	226
62. ábra A szervezetnél nyilvántartott azonosságokban bekövetkezett változásokat megduplikálja a szolgáltatónál létező nyilvántartásokban.	227
63. ábra Az egyszeri bejelentkezés használati esetének általános sémája.....	233
64. ábra Föderatív azonosítás forgatókönyve.....	235
65. ábra A jogosultságok megadásának alapsémája egy elektronikus információrendszer környezetben	239

66. ábra A jogosultsági engedélyek és forgatókönyvek lépéseinek összekapcsolása munkafolyamat munkafeladatain belül.....	240
67. ábra A forgatókönyv és az engedély modell közti kapcsolatok	243
68. ábra Példa engedély katalógusra	244
69. ábra Az ember-gép párbeszéd dinamikusán változó jellegének figyelembe vétele egy elektronikus információrendszer környezetben.....	245
70. ábra Hozzáférési jogosultság ellenőrzési rendszer	246
71. ábra Szerepkör hierarchiák- Általános és korlátozott.....	247
72. ábra Attribútum alapú jogosultság kezelés.....	248
73. ábra A törzsadat-kezelés szolgáltatás- és komponens architektúra.....	256
74. ábra Integráció szintjei	263
75. ábra A platform jellegű köztesszoftver architektúra.....	272
76. ábra Köztesszoftver és elosztott rendszerek architektúrája	273
77. ábra A köztesszoftverek (Middleware) egyik fajta osztályozása	273
78. ábra Elosztott objektumok CORBA architektúra alkalmazásával.....	278
79. ábra Üzenet továbbító rendszer modell	280
80. ábra Az üzenettovábbító/üzenetcsatolt köztesszoftverek szerepe az alkalmazások integrálásában.....	281
81. ábra Az üzenetközpontú köztesszoftver kiszolgáló architektúra- logikai architektúra komponens.....	283
82. ábra Üzenetközpontú köztesszoftver esetében az üzenettovábbítás garانتálása.....	284
83. ábra Üzenetközpontú köztesszoftver esetében tranzakció jellegű feldolgozás	285
84. ábra A kiszolgáló gépek klaszterezése (fürtözése) megbízhatóság növelése és a skálázhatóság érdekében.....	286
85. ábra Kérelem válasz üzenetváltás.....	287
86. ábra N-szintű (N-tier) Web alkalmazás architektúra.....	288
87. ábra Egy üzleti folyamat levelezési platform felépítése.....	290
88. ábra Az integráció brókerek helye az integráció architektúrában	291
89. ábra Integráció köztesszoftver használata nélkül és köztesszoftver használatával	293
90. ábra A folyamat és kapcsolatai.....	303
91. ábra: A szervezeti, vállalati folyamatok teljes környezetükben [82].....	304
92. ábra A folyamatokat befolyásoló hatások – A klasszikus projekt háromszög	305
93. ábra A folyamatmenedzsment rendszer legfontosabb elemei	307
94. ábra Egyszerű Petri-háló tokenekkel- klasszikus tankönyvi példa.....	314
95. ábra A XOR (logikai kizáró vagy) leképezése Petri hálóba.....	315
96. ábra Az AND (logikai és kapcsolat) leképezése Petri hálóba	315
97. ábra Eseményvezérelt folyamatlánc objektumai	317
98. ábra Sematikus példa egyszerű és kibővített eseményvezérelt folyamatláncra.....	318
99. ábra Az eseményvezérelt folyamatlánc és azt információrendszer nézetek.....	319

100. ábra Funkció és szervezeti egység összekapcsolása.....	322
101. ábra Információáramlás a funkcióba	324
102. ábra A BPMN alap diagram technikai elemei.....	327
103. ábra BPMN-ben az események típusai [116].....	329
104. ábra Kibontott részfolyamat, összevont folyamat (collapsed subprocess) és feladat (task).....	330
106. ábra A logikai kapuk típusai.....	331
107. ábra Esemény alapú.....	332
108. ábra Elágazó és csatlakozó VAGY kapuk.....	333
109. ábra Sávokon belüli tevékenységek kapcsolatai [116].....	334
110. ábra A BPMN jelölésrendszer összefoglalása.....	336
111. ábra Folyamatszervezés SzOA segítségével felülnézetből.....	337
112. ábra Alulnézetből folyamatszervezés a SzOA segítségével.....	338
113. ábra BPMN objektumok ARIS-ban.....	344
114. ábra ARIS ház	345
115. ábra A szabályalapú folyamatmenedzsment elemeinek metamodellje	357
116. ábra Folyamat automatizálás főbb komponensei	357
117. ábra Folyamat automatizálás szabály és szolgáltatás központú alapon.....	358
118. ábra Általános kommunikációs infrastruktúra.....	359
119. ábra E-kormányzati célok elérése.....	360
120. ábra Egy referencia architektúra.....	360
121. ábra Közigazgatási szolgáltatások és jogosultságok kezelése megállapodásokon keresztül.....	361

TÁBLÁZAT JEGYZÉK

1. Táblázat A jelentősebb architektúra megközelítések és szintjeik	33
2. Táblázat Web szolgáltatások leírása.....	43
3. Táblázat Az objektum-orientált és szolgáltatás-orientált paradigma összehasonlítása	49
4. Táblázat Az architektúra nézetek hierarchikus rendje, taxonómiája.....	112
5. Táblázat Az architektúra meta-modellel, ontológiával és kiegészítéseivel kapcsolatos architektúra nézőpontok.....	115
6. Táblázat Egy alkalmazási rendszer architekturális leírásának példa sémája. Információrendszer architektúra szempontjából történő jellemzése.....	131
7. Táblázat Jelentős felhő-szolgáltatások üzemszünetei	159
8. Táblázat A számítási felhő egy olyan szolgáltatás, amely az információ-technológiai infrastruktúra használatát jelenti	161
9. Táblázat Magánfelhő magas szintű architektúra építő elemei (ld. még 48. ábra)	165
10. Táblázat A felhő-szolgáltatás üzemeltetési módjai és követelmények.....	167
11. Táblázat A funkcionális és nem-funkcionális követelmények összehasonlítása számítási felhő esetében	177
12. Táblázat Nem funkcionális követelmények és Funkcionális követelmények leképezése	179
13. Táblázat A követelmények összefoglaló táblázata.....	181
14. Táblázat Tanúsítványokkal és elektronikus aláírással kapcsolatos fogalmak	206
15. Táblázat Szoftver architektúra képességeivel szemben támasztandó igényfelmérési táblázat.....	212
16. Táblázat Példa a biztonsági igények megfogalmazására.....	221
17. Táblázat Jogosultsági engedélyek	241
18. Táblázat - Példa egy folyamat RACI mátrixára	259
19. Táblázat A szocio-technológiai rendszerszemléletű és az információrendszerek informatikai szemlélet leképezése.....	265
20. Táblázat A SzOA befogadásának szintjei	296
21. Táblázat SzOA kivitelezésének hat lehetséges megközelítése.....	296
22. Táblázat SzOA kivitelezésének lehetséges mintázatai és hasznai.....	297
23. Táblázat A szervezet szolgáltatás integrációra való érettségének felmérésére szolgáló modell táblázata	299
24. Táblázat A folyamat menedzsment négy szintje.....	358

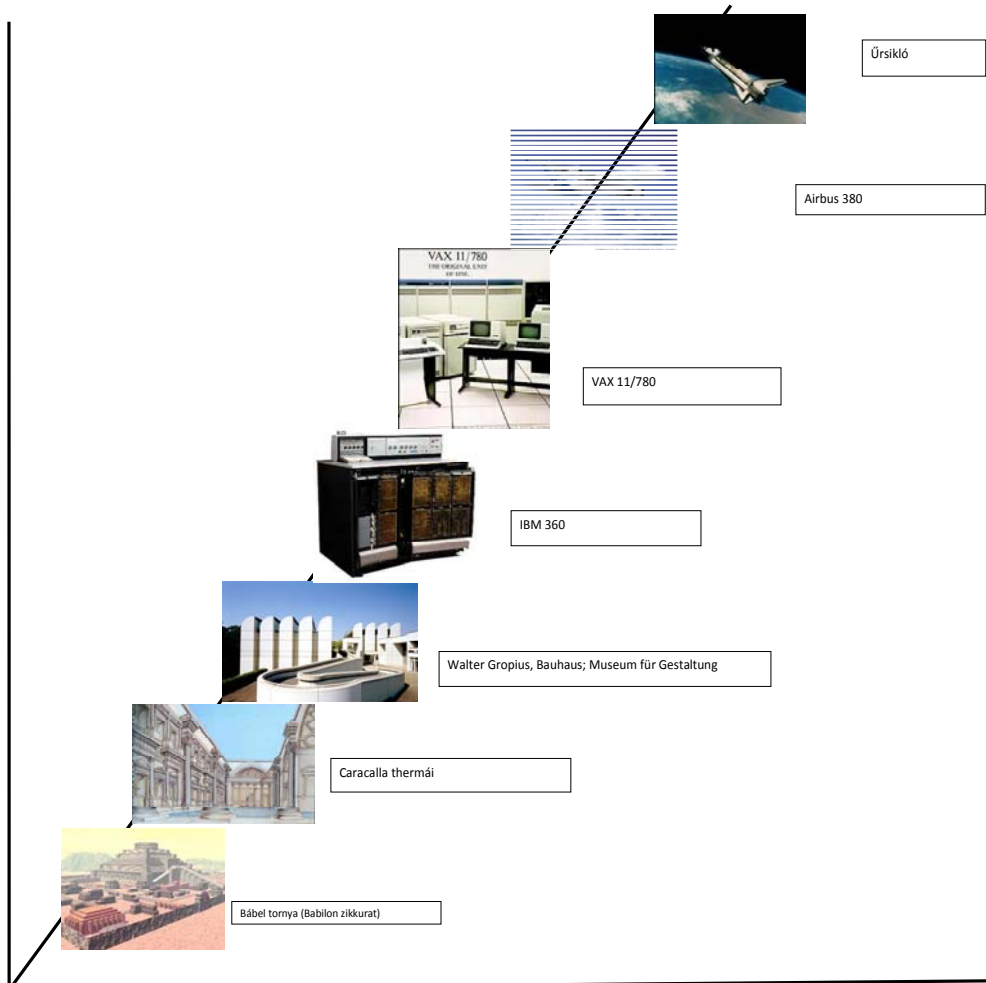
DEFINÍCIÓ JEGYZÉK

A klasszikus architektúra (görög).....	16
Architektúra Blaauw szerint.....	17
Szoftver architektúra	17
Informatikai, műszaki architektúra.....	18
SzOA, Szolgáltatás Orientált Architektúra.....	25
Szervezeti (vállalati) informatikai architektúra	27
Információ architektúra	28
Szervezeti (vállalati, üzleti) információrendszer architektúra	29
A platform	30
Alkalmazási architektúra	30
IEEE architektúra	31
TOGAF architektúra.....	31
Clinger–Cohen Act információ-technológia architektúra	31
Holland Architektúra Fórum	31
ArchiMate Foundation szervezeti (vállalati, üzleti) architektúra	32
Capgemini szervezeti (vállalati, üzleti) architektúra	32
Gartner Group szervezeti (vállalati, üzleti) architektúra	32
Fogalmi (konceptcionális) integritás	32
A szervezet irányítása („governance”)).....	33
Szolgáltatás.....	41
Web szolgáltatás (Web service)	41
Szolgáltatás-orientált architektúra	50
Web Szolgáltatások:	51
Szolgáltatási sín (Enterprise Service Bus, ESB):	52
SOAP (Simple Object Access Protocol):.....	53
UDDI (Universal Description, Discovery and Integration Business Registry):.....	56
WSDL (Web Services Description Language):	58
Elérési szolgáltatások:	60
Informatikai szolgáltatás menedzsment:	60
Szervezet nyomon követése, vezetői műszerfal (dashboard):	60
SzOA OASIS (the Organization for the Advancement of Structured Information Standards):.....	64
Számítási felhő	153
AAA, Authentication, Authorization, Accounting/Access Control	202
Azonosítás	204
Web szolgáltatás biztonsága WS-Security	216

Funkcionális szerepkör („role”)	239
Strukturális szerepkör	240
Törzsadat	253
Kulcs adatentitások	254
Létfontosságú adatalemek	254
Integrált rendszer	262
A vállalati alkalmazások integrációja (Enterprise Application Integration, röviden EAI)	265
Vállalati alkalmazások integrációja	266
Köztesszofrtver (Middle-ware)	271
Köztesszofrtver	273
Szervezeti (vállalati, üzleti) folyamat	302
Munkafolyamat („workflow”)	303
Informatikai (adattranszformáló) folyamat	303
Szervezeti (vállalati, üzleti) folyamatok újraszervezése (BPR)	308
Szervezeti (vállalati, üzleti) esemény	317
Vállalati, üzleti esemény	318
Eseménytér	319
Funkció	320
Szabály a műveletekre	325
Informatika (Informatics)	362
Információtechnológia	362

1 TÖRTÉNETI ÁTTEKINTÉS

A klasszikus **architektúráról**, az építőművészetről a történeti feljegyzések több mint 4000 évvel ezelőtől szólnak, kezdve az egyiptomi piramisok felépítésétől, amelyeknek a bonyolultsága mind a tervezőket mind az építőket nagyon komoly feladat elé állította. Ez a bonyolultsági probléma abból a jelenségből ered, hogy amilyen mértékben a létrehozandó rendszerek egyre egyre törőbbek lettek, az elemeik közötti kapcsolatok sokkal jobban növekedtek, mint maguknak az elemeknek a száma. A piramisok többé nem egyszerűen temetkezési helyek voltak, hanem a világi és vallási hatalom érzékeltetői, istennek tekintett uralkodók és kincseik védett temetkezési helyei, ugyanakkor kiemelkedő mérnöki teljesítmények. Az elemek közötti bonyolult kapcsolatrendszer túl volt azon a határon, amelyet a mérnökök és építők hagyományos eszköz készletükkel kezelni tudtak. Ez vezetett el az architektúra fogalmának kialakulásához, mint egy olyan eszköz létrejöttéhez, amely lehetővé teszi ezeknek a komplex kapcsolatoknak az átlátását és kézben tartását. Ez a probléma megközelítés napjainkban is él, itt maradt velünk. A társadalom fejlődését követve az architektúra fogalmát arra használtuk, hogy sok és nagy változatosságú területeken különböző műszaki konstrukciók átlátását és kézben tarthatóságát elérjük.



1. ábra Műszaki konstrukciók, mint architektúrák fejlődése

Nagyjából ebben az időben kezdte munkáját Dijkstra [40] a strukturált programozás tudományos igényű megalapozásán. Noha Dijkstra nem használta az architektúra szót, de ismételten hangsúlyozta a szoftver szerkezetének, felépítésének fontosságát, és ezen keresztül bizonyos alapozását végezte el a **szoftver architektúra** fogalmának.

A klasszikus architektúra (görög)

- Építészet, építőművészet;
- egy épület művészi jellege, stílusa.

Informatika területén az architektúra fogalma az 1960-as években jelent meg; talán Blaauw lehetett az első, aki az **információ-technológiában (IT)** az architektúra fogalmát használta a számítógép felépítésével kapcsolatban, az IBM-nél korábban használt „gép szervezése” („*machine organization*”) fogalma helyett. Blaauw az egyik társ konstruktőre volt az IBM 360-as számítógépnek („mainframe”).

Architektúra Blaauw szerint

- Az **architektúra** kifejezést arra használjuk, hogy leírjuk a *rendszer* tulajdonságait olyan módon, ahogy azt egy programozó látja, vagyis a *rendszer* fogalmi szerkezetét és funkcionális viselkedését, amelyek elkülönülnek az adatok áramlásától, vezérlésétől, a logikai tervtől és a fizikai megvalósítástól.¹ [20]

Blaauw-ék publikációjukban több lényeges fogalmat tárgyalnak: a számítógép architektúrát mint a számítógép tervét, továbbá a modularitást, a megbízhatóságot és a strukturált programozás paradigmájának egyik téziséét; nevezetesen azt, hogy a szoftver tervnek szerkezete van és rétegekből áll. Ez a fejlődés vezetett el a **szoftver tervezés**, **szoftver technológia** („software engineering”) és a **strukturált programozás** fogalmához.

Perry és Wolfe 1992-ben modernizálta Dijkstra eredeti megközelítését és **szoftver architektúra**, szoftver terv **elemek** egy halmazát definiálta.

Ahogy a szoftver alkalmazási rendszerek egyre nagyobbak és nagyobbak lettek, Shaw és Garlan [104] létrehozta a **szoftver architektúra** fogalmát. A *szoftver architektúra* megközelítés a szoftver tervezési termékeivel kapcsolatos kulcs fontosságú tervezési elveket foglalta össze. Booch, Rumbaugh, and Jacobson 1999-ben leírt egy olyan *architektúra* fogalmat írt le, amely a szoftver rendszerre vonatkozó lényeges döntések összességét testesíti meg; nevezetesen a szoftver szerkezetét alkotó elemek kijelölése, valamint a kiválasztott alkotó elemek közötti kapcsoló felületek („interface”) halmaza, amelyek az együttműködésüket leíró specifikációban megadott viselkedésükkel együttesen alkotják a rendszert. Ezeknek a szerkezeti és viselkedési elemeknek az összeépítése fokozatosan egyre nagyobb rendszerekké és az építés stílusa az, amely vezérli ezt a rendszerszervezési megközelítést.

Szoftver architektúra

- **Szoftver architektúra:** Az alkotó elemek olyan pragmatikus és koherens gyűjteménye, amely ezeken az alkotórészekon keresztül a rendszer egészét átfogóan látó „végfelhasználó” rendszerről alkotott elképzeléseinek megvalósulását elegáns módon támogatja.

Az 1980-as és az 1990-es években egyre világosabbá vált, hogy az *információ-technológia* fejlesztése csak azzal a környezettel összhangban történhet, amelyben az információ-technológiát alkalmazzák, ez vezetett el annak a problémának a felismeréséhez, amelyet az

információ-technológia és a **szervezeti környezet illesztése**² (vállalati, üzleti környezet) fogalmával ragadhatunk meg. Az információ-technológia és a szervezeti környezet illesztése problémájának megoldása azt igényli a szervezetektől, hogy a szervezet mint rendszer különböző oldalainak illesztését is elvégezzék, nevezetesen a humán erőforrás, a szervezési, az információ kezelési és műszaki, informatikai nézőpontjai között teremtsenek összhangot. Ez a fogalmi és gondolati fejlődés elvezetett annak a felismeréséhez, hogy az *információ-technológia és a szervezeti környezet illesztése* mint megközelítés sem elegendő a fennálló és felmerülő problémák orvoslására. Ezért alakult ki az architektúra fogalmának egy nagyvállalat szervezetének megfelelő fogalma, amit **vállalati architektúrának** vagy **szervezeti architektúrának**³ nevezhetünk.

1.1 Az informatikai, műszaki architektúra rövid története

Az informatikai, műszaki architektúra tulajdonképpen már akkor létezett, amikor még a számítógépipart nem nevezték információ-technológiai iparnak, sőt már akkor is létezett mielőtt a számítástechnikát elnevezték volna *elektronikus adat feldolgozásnak*. Manapság több ezer informatikai termék, specifikáció, szabvány, nemzetközileg bevált gyakorlat, útmutatók és stratégiák léteznek, amelyeket át kell ahhoz fésülni, hogy megvalósítható megoldásokat lehessen építeni. Egyetlen „rendszer” felépítéséhez hihetetlen mennyiségű különböző alkotó részre van szükség. Ebben a környezetben egyáltalán nem meglepő folyamatosan lehet hallani az „architektúra tervezés” kifejezést az informatikában. Az informatika architektúra tervező szerepe egyre jelentősebb (mind információ-technológiai rendszerek architektúrája mind a szervezeti architektúra tekintetében).

Informatikai, műszaki architektúra

- **Informatikai, műszaki architektúra:** Azt a műszaki és irányítási, igazgatási platformot határozza meg, amelyre építkezve úgy alakítja ki egy szervezet az információ-technológiai rendszereit, hogy abból haszna származzon (üzleti nyereség, stb.).

Az informatikai architektúrák fejlődését a következőképpen kategorizálhatjuk (ld. [88]) :

- Vaskorszak
- Reneszánsz
- Ipari forradalom

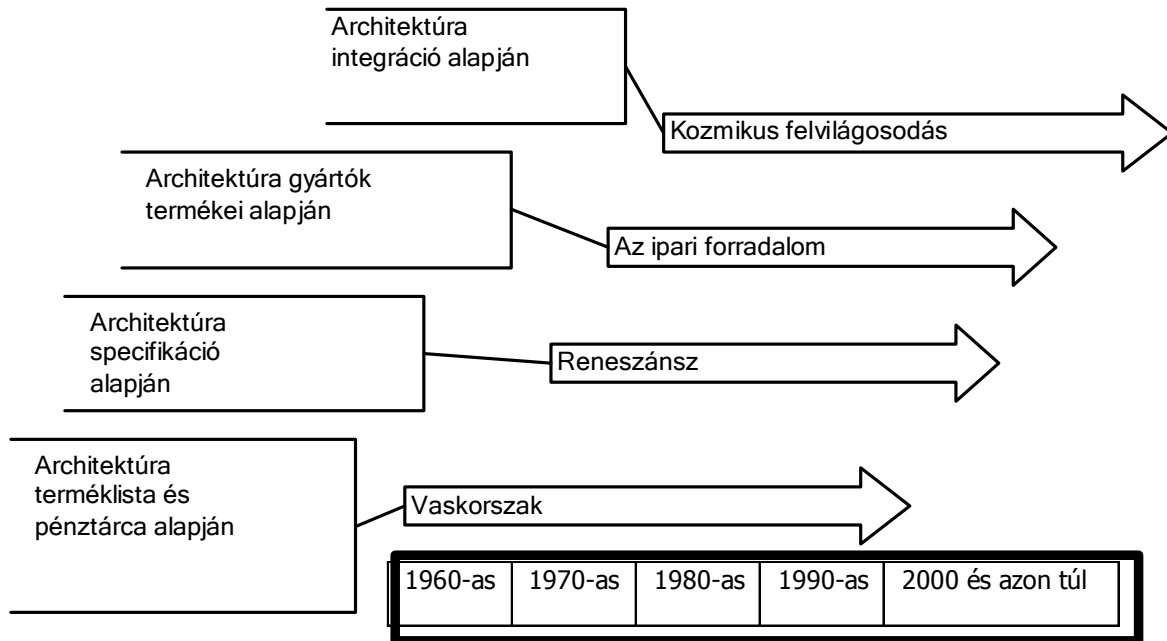
- Kozmikus felvilágosodás (talán)

1.1.1 A vaskorszak

Ez volt az információ-technológiai ipar kezdete. Ezt az időszakot a nagy, gyártó függő és mindenek felett nagyon drága nagy számítógépek („mainframe”) jellemezték. Ezek a nagy számítógépek – összehasonlítva a mai technológiával – ormótlanok, bonyolultak voltak és rengeteg tehetséges műszaki szakemberre és tervezőre volt szükség csak ahhoz is, hogy a legegyszerűbb információrendszert létrehozzák, üzleti, vállalati vagy egyéb célokra. Ebben a korszakban a technológia célja az volt – és ez lényegében változatlan maradt napjainkig - , hogy megbízható, masszív, nem túl bonyolultan használható és skálázható rendszereket nyújtson azoknak, akiknek elég sok pénzük van ilyen célokra.

Az *architektúra* fogalma ebben az időben került használatba az információ-technológiában (ld. Blaauw [20].) a számítógép szerkezet, szervezés helyett. Ebben a korszakban kezdtek elválni a tisztán szoftver architektúra megközelítések és az információ-technológia egyéb infrastrukturális és környezeti elemeinek architektúrája.

A szervezetek szempontjából a valóságban ekkor az architektúra a cég *pénztárcájától* függött. Ekkoriban a szervezetek tipikusan egy szállítótól vásárolták meg a komponenseket, és a szállító felelőssége volt az, hogy az alkotóelemek együttműködjenek. A szállító oldaláról pedig a *terméklista* alapján egy ún. konfiguráció tervezési feladat előtt álltak, amely egyszerűbb mint egy teljesen új tervekészítésének feladata (kreatív tervezés) de egyáltalán nem magától értetődő, nem egyszerű feladat (ld. McDermott [75]). Az architektúrának és rendszertervezésnek ez az egyszerűsített felfogása az elsődleges oka annak, hogy ez a technológiai megközelítés, a technológia alkalmazásának ez a formája ma is sikeres az információ-technológia iparban. Természetesen ennek a megközelítésnek az oka ebben a korszakban azért az volt, hogy a számítástechnikai alkotó részek – hardver, periféria és a szoftver is – egyedi gyártók termékei voltak, a nagy számítógép gyártók nem voltak érdekeltek az együttműködés – **interoperabilitás** – megvalósításában.



2. ábra Az informatikai, műszaki architektúra fejlődése

1.1.2 Reneszánsz

A *vaskorszak* gyártói, szállítói és architektúrái ebben a korszakban is tovább léteznek, noha egyre növekvő versennyel néznek szembe egy állandóan változó információ-technológia környezetben. A „nagy számítógépekkel” a gondolati szakítás e korszak elején megtörténik, a hardver miniaturizálása, valamint új piaci szereplők megjelenése (az egyik legfigyelemre méltóbb a Digital Equipment Corporation, DEC) egy jelentős információ-technológia kulturális elmozdulást jelentettek költség kímélőbb megoldások irányába (természetesen ezt is relatíve a korszakhoz értendő). A reneszánsz korszak hajtóereje a mini-számítógépek és a UNIX operációs rendszer volt. A UNIX terjedésével együtt nőtt meg az érdeklődés a nyílt szabványú rendszerek irányába. A UNIX, amely egyetemi környezetben keletkezett, a kutatás világából átkerült a gyakorlati információ-technológiába és sok leszármazottja keletkezett. A UNIX hívők korán felismerték, hogy a UNIX csak akkor tud „vereséget mérni” a nagy számítógépekre, ha a különböző szállítók platformjai között a rendszerek hordozhatóságát fenn tudja tartani. Ezért a UNIX hívők és támogatók tábora elkezdett egy önszabályozást. Tulajdonképpen a UNIX harcba hívott a nyílt (**open**) és együttműködésre (**interoperable**) képes számítástechnika érdekében.

Szabványosítási szervezetek komoly erőforrásokat fektettek szabványok kifejlesztésébe, nevezetesen az X/Open, Open Software Foundation (OSF), IEEE és ISO. A probléma ott je-

lentkezett, hogy a szabványosítási testületek egymással kezdtek versenyezni, aminek a következménye rivalizáló szabványok lettek. A szervezetek számára pedig kritikus fontosságú lett, hogy megértsék, melyik szabványt válasszák ki saját maguk számára. Ebben az időben manapság egy olyan egyszerű döntés, hogy melyik hálózati protokollt válasszák ki – alap infrastruktúra döntés –, rendkívül bonyolult volt az egymással versengő technológiák és szállítók miatt.

A műszaki architektúra e korszakát úgy lehet jellemezni, hogy *architektúra specifikáció*⁴ alapján.

Ebben az időszakban történt meg az „architektúra” mint tervezési megközelítés lényegre törő megalapozása. Az IEEE és az ISO szabványosítási testületek az elsők között voltak azok között, amelyek **műszaki hivatkozási architektúra** (*technical reference models (TRM)*) és **architektúra keretrendszer** szabványokat hoztak létre (*architectural framework*). A *műszaki hivatkozási architektúra* az architektúra fejlesztés fontos eszköze. A Government Open Systems Interconnection Profile⁵ (GOSIP) az egyik példa egy olyan szervezeti szintű műszaki architektúrára, amely az Open Systems Interconnection (OSI) specifikációira támaszkodva a kormányzati informatika területét célozta meg. Noha ezek az architektúrák széleskörű, átfogó fogalmi rendszerrel dolgoztak, de ugyanakkor meglehetősen nehezen voltak megérthetőek, és talán még lényegesebb az, hogy nehezen voltak megvalósíthatók. A szervezetek szempontjából pedig ezek az architektúra szabványosítási törekvések elfogultak voltak ez egyes szabványosítási testületek saját specifikációi irányába, egyes technológiai megoldásokat részesítették előnyben. Röviden a szervezeti szintű, szervezetet átfogó szemlélet teljesen hiányzott ezekből a megközelítésekből.

Érdemes megjegyezni, hogy az információ-technológia e boldog békeidőnek tekinthető periódusában egy kis, az Amerikai Egyesült Államokbeli hálózat, amely elsősorban a katonai és egyetemi kutatóhelyeket kapcsolta össze és „The Internet” nevet viselte, kifejlődött és lassan növekedésnek indult.

1.1.3 Az ipari forradalom

Az ipari forradalom korszaka az információ-technológiában olcsóbb és jobb teljesítményű rendszerekkel és a személyi számítógép megjelenésével köszöntött be. A piacot elárasztották az új gyártók és szállítók, olyan nagyszámú piaci és technológiai réseket töltöttek be, amelyekről a korábbi korszakban addig senkinek sem volt fogalma. A Microsoft újradefiniálta az

ipar hozzáállását az információ-technológiához; nagyon drága információ-technológia rendszerek kis darabszámú értékesítését felváltotta egy drámai árzuhanás nagy volumenű értékesítéssel. Ebben a korszakban az információ-technológiai termékek közönséges árucikké. A *vaskorszak* legtöbb jelentős gyártója komoly küzdelmet folytatott az életben maradásért az *ipari forradalom* korszakában, a kérdés az volt, hogy vajon alkalmazkodnak-e vagy eltűnnek-e a piacról.

A korszak érdekessége az, hogy noha az általános vélekedés az volt, hogy a *szabványok* és a (szabványok) *specifikációi*, a részletes műszaki előírások egyre fontosabb szerepet fognak játszani, ezt a korszakot az egyedi, gyártói technológiák uralkodása és előnyben részesítése jellemezte. Több tényező összjátéka vezetett el ehhez az eredményhez. Egyrészt a szabvány készítés egyre fáradságosabb folyamat lett, a szabványok és specifikációk kialakítás hosszú időt igényelt, a szabványosítási bizottságok tevékenységére ránehezedtek az ipar jelentős szereplői, amelynek következménye lassan kialakuló konszenzus volt. Természetesen ha valamikor létre is jött egy szabvány, a gyártóknak akkor is időre volt szükségük a termékeikben megvalósításához és utána a piacra történő bevezetéshez. A technológiai fejlődés gyorsulásával azok a szervezetek, amelyek a szabványok létrejöttére vártak, elkerülhetetlenül vesztek piaci részesedésükből azokhoz képest, akik pedig saját, egyedi technológiai termékeket hoztak ki a piacra. Microsoft, Cisco, Oracle és más gyártók a saját piaci szegmensükben fokozatosan piacvezetővé váltak saját, egyedi termékeik bevezetésével a piacra (gyakran ezt a technológiai kizárólagosságot a „**piaci szabvány**” elnevezéssel igyekeztek leplezni), ezeket a termékeket - nem várva a szabványosítási folyamatok lezárulására - sokkal korábban piacra dobtak mielőtt a szabványok publikálásra kerültek volna. Ebben az időszakban az olyan szabványosítási testületek mint pl. az ISO, az X/Open és az OSF elkeseredetten küzdöttek az ismertség megőrzéséért a technológia világában mint a nyílt számítástechnika elveinek i és ápolói. Sok OSI specifikációt csak piaci és technológiai résekben valósítottak meg, és egyre kisebb hatást gyakoroltak ezek a specifikációk a szervezeti környezetre.

Architektúra fogalmakkal ez a korszak úgy jellemezhető, hogy **architektúra a gyártók termékei alapján**. A szervezetek stratégiai döntéseket hozhattak olyan termékek beszerzésére, amelyet egyes néhány gyártóból álló szövetségek tudtak szállítani. A Microsoft dominanciája az asztali számítógépek - majd később a kiszolgáló gépek piacán – segítette elő ennek az architekturális megközelítésnek az érvényesülését.

Ebben a korszakban következett be az **Internet** robbanás, ekkor „fedezték” fel végre. Az egyetemi és kutató közösségek már ekkor évtizedek óta használták az *Internetet*, de a **Világháló (World Wide Web)** megérkezése tette mindennapi, közönséges informatikai szolgáltatássá, amelynek révén az egyéb informatikai hozzáférhetősége drámaian megnövekedett. Ez az esemény vissza lendítette az ingát az ipari szabványosítás irányába. Az IETF⁶ az Internet használat legtöbb területére folyamatosan alakított ki szabványokat - valójában az Internet sikerének ez volt a kulcsa. Általánosan elfogadott architektúrák és specifikációk nélkül nem lett volna lehetséges egy ilyen globális hálózatot felépíteni. The IETF más szemszögből közelítette meg a szabványosítás kérdését, amely talán segítette a munkáját. Mielőtt egy specifikáció szabvánnyá vált volna (**Request for Comment**, **RFC**⁷) az előfeltétel az volt, hogy legalább két együttműködésre képes (interoperábilis) megvalósítása létezzen. Ugyanakkor az IETF szabvány felfogása azt jellette, hogy ha a szabvány „elegendően jó” akkor már el lehetett fogadni és ki lehetett bocsátani, nem kellett tökéletességre törekedni. Talán ez lehetett az OSI szabványosítási megközelítésének alapvető hibája.

1.1.4 A kozmikus felvilágosodás

A kérdés az, hogy mi fog történni a jövőben? Többféle jövőkép is létezik. Az Internet alapú számítástechnikai további terjedése várható, amint az a **SOA / SZOA (Service Oriented Architecture / Szolgáltatás Orientált Architektúra)** és **Web szolgáltatások (Web services)** intimit koegzisztenciájából látszik a különböző vállalati, üzleti és közszolgálati információrendszerekben, bizonyos problémákra megoldásokat nyújtva. Tulajdonképpen egy másik dimenzió, bizonyos értelemben ortogonális architektúra megközelítés a **számítási felhő (Cloud Computing)**, amely megint az Internet szolgáltatásaira, támaszkodva terjed. Az architektúrának tulajdonképpen egy végfelhasználó nézete a **mobil számítástechnika**, a távoli elérése mindenféle információrendszer szolgáltatásnak mindenféle információ- és kommunikációtechnológiai eszközökkel: hordozható számítógép, tenyérgép, mobil telefon, iPhone, iPad stb. A technológia gyors fejlődése miatt a felsorolás nem teljes, de nem is lehet az a különböző technikai eszközök tekintetében.

A technológia fejlődése a „világháló a számítógép” közhelyszerű kifejezés megvalósítása irányában halad? A világ olyan megállíthatatlan mértékben átszövődött az *Internet technológiával*, a világgazdaságnak és az egyes nemzeti gazdaságoknak olyan lényeges tényezője lett az *Internet*, hogy a gazdasági teljesítőképessége egy országnak már nemcsak kizárólag az or-

szág gazdaságának teljesítményétől függ, hanem a gazdasági szereplők hálózatától, amelynek egyik leképezése, technológiai megjelenése az *Internet*. Az alap infrastruktúra elemekben bizonyos konszolidáció látható: hálózatok, operációs rendszerek és szoftver architektúrák olyan néhány meghatározó technológia megoldásokban olvadnak össze, amelyek mindegyike szükségszerűen támogatja a Web alkalmazásokat. Eközben az *ipari forradalom* megállíthatatlanul, egyre gyorsuló iramban folytatódik. A piaci verseny és csata az alap infrastruktúra platformok fölötti szolgáltatások szintjén dúl: tudás menedzsment, szórakoztatás, portálok, köztes rendszerek / brókerek, és szemantikus megoldások.

Architekturális szemszögből nagyon különböző és változatos technológiák integrálása válik szükségessé és egyre fontosabbá annak az érdekében, hogy a kívánatos szolgáltatási portfóliót az adott szervezet számára biztosítsák. A szervezeteknek nemcsak a szervezeten belüli bonyolult alkotóelemeket kell egy működő rendszerbe összeilleszteni, hanem a beszállítók és az ügyfelek rendszereit mind hazai mind nemzetközi vonatkozásban össze kell kapcsolni a saját rendszereikkel. Más szavakkal a nagyvállalati, szervezeti architektúra többé nem tekinthető kizárólag belügyként, hanem tekintettel kell lenni a partner szervezetek, ügyfelek és egyre növekvő számú, megnevezhetetlen társadalmi, gazdasági szereplő technológiájára, ez a képesség gazdasági versenyben élet vagy halál kérdése lehet.

Az emberiség történetében gyakran fedezhetünk fel ciklusokat, jó okkal gondolhatjuk, hogy az informatika története is valamilyen ciklikus viselkedést mutat. A kozmikus felvilágosodás felé történő megállíthatatlan menetelés helyett a jövőben bekövetkezhet információ-technológia érő gazdasági megszorítások, egy második vaskorszak, vagy jelentős, piacvezető szállítók saját egyedi technológiáinak ráerőltetése a vásárlókra, egyre központosítottabb rendszer megoldásokkal. Azonban a *műszaki, informatikiai architektúra* fogalma megmarad, de folyamatosan alakul a változó körülményekhez igazodva.

1.1.4.1 Mi is ez a számítási felhő a és SzOA?

A SzOA fogalma egyáltalán nem új. Azok a próbálkozások, amelyek a közösen használt informatikai folyamatokat, információkat és szolgáltatásokat megoszthatóvá és elérhetővé kívánták tenni, jelentős múltra tekintenek vissza. A több szintű ügyfél-kiszolgáló (client / server) architektúra –a közös kiszolgáló gépen a megosztott szolgáltatások egy halmaza volt található, amely a szervezet számára az adott infrastruktúrán újra felhasználhatóságról és integrációról gondoskodott – és az elosztott objektumokkal kapcsolatos fejlődés. Az *újra fel-*

használhatóság egy értékes célkitűzés, a SzOA esetében ez a szolgáltatások és a hozzájuk kötődő információk újra hasznosítását jelenti. A közösen használható és rendelkezésre álló szolgáltatások halmaza ösztönzi az *újra felhasználhatóságot*, és lényegesen csökkenti a redundáns alkalmazások és szolgáltatásaik iránti igényt.

SzOA, Szolgáltatás Orientált Architektúra

- A **SzOA** egy olyan stratégiai technológiai keretrendszer, amely a szervezet összes érintett rendszerei számára – akár a szervezeten kívül akár belül vannak - felajánljon és elérhetővé tegyen jól-definiált szolgáltatásokat, valamint a szolgáltatásokhoz kötődő információkat. Ezek a szolgáltatások tovább absztrahálhatók folyamat rétegekbe és összetett alkalmazásokba megoldások kifejlesztése végett. A SzOA kibővíti az architektúra fogalmát az **agilitással**, lehetővé téve azt, hogy a rendszer változtatások egy konfigurációs rétegben történjenek ahelyett, hogy ezeket a rendszereket folyamatosan újra kelljen fejleszteni.

Mi köze van a SzOA-nak felhő számítástechnikához? A számítási felhő tulajdonképpen bármilyen olyan információ-technológiai erőforrás - beleértve a háttértároló kapacitást, adatbázist, alkalmazás fejlesztést, alkalmazási rendszer szolgáltatást stb. -, amely a szervezet tűzfalain keresztül létezik és a szervezet az Interneten keresztül hasznosítja, kiaknázza. Az alapötlet a számítási felhő mögött az, hogy sokkal olcsóbb ezeket az erőforrásokat, mint szolgáltatásokat hasznosítani, és akkor fizetni értük, amikor rájuk szükség van, ahelyett, hogy a számító központba még több hardvert és szoftvert vásárolnának. Ezenkívül a cég gazdasági alkalmazkodóképességét is megnöveli a számítási felhő, lehetővé téve a költségek növelését és csökkentését közvetlenül a cég szükségleteinek megfelelően. Továbbá az információ-technológia erőforrások bővítésével kapcsolatos kockázatok egy részét is áthárítja a cég a számítási felhő szolgáltatójára.

1.1.4.2 Mi a kapcsolat a SzOA és számítási felhő között?

A SzOA és számítási *felhő* között az a kapcsolat, hogy a számítási felhő gondoskodik az információ-technológiai erőforrásokról, amelyeket a szervezet igény szerint hasznosít, beleértve a hosztadatokat, szolgáltatásokat, és folyamatokat. A szervezet SzOA-ja kiterjeszthető a szervezet tűzfalain kívülre a számítási felhő szolgáltatóig, az előbb leírt előnyök kiaknázása végett.

A szervezeti SzOA maga nagyon lényeges a számítási felhő szempontjából:

- 1) A SzOA a szervezeti architektúra szempontjából egy jó megoldás, amely az információrendszerek korrekt kialakításával foglalkozik és olyan mechanizmusokat alkalmaz, amelyek révén az információrendszerek jól együtt tudnak működni mind a szervezeten belül mind kívül.
- 2) A számítási felhő előnyeinek kihasználása érdekében olyan kapcsolófelületekre és architektúrára van szükség, amelyek kinyúlnak a számítási felhő erőforrásáig és elérik azokat. A számítási felhő eredményes és hatékony kihasználása végett olyan szervezeti architektúrára van szükség, amely a legtöbbet hozza ki a felhő számítástechnikából, ilyen lehet pl. a SzOA.

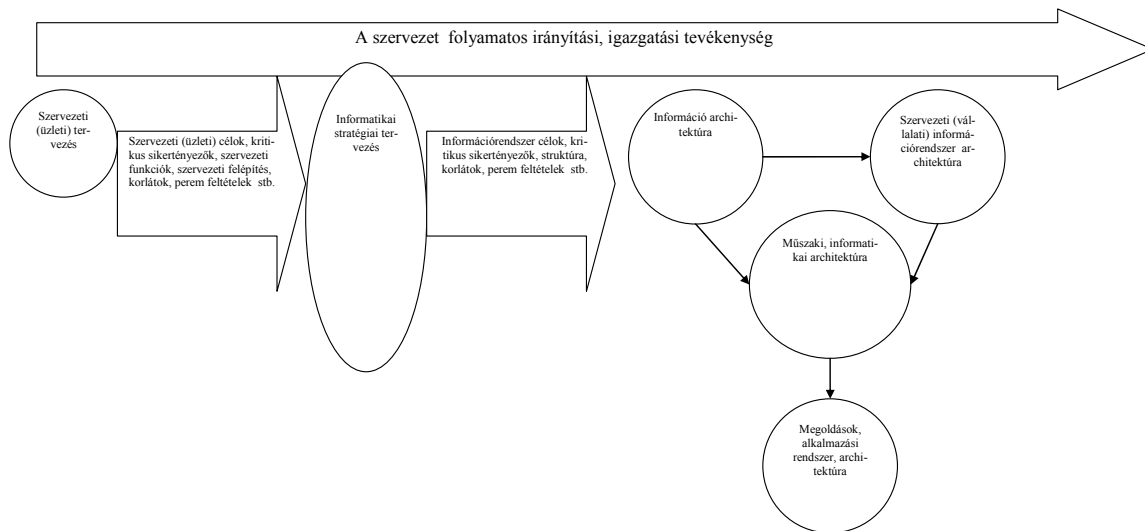
2 SZERVEZETI ARCHITEKTÚRA

A korábbi szakaszokban az *architektúra* kifejezés mögött meghúzódó különböző jelentésekkel találkozhattunk – jellemzően az informatikai alkalmazásokkal és rendszerekkel kapcsolatban. Az *architektúra* kifejezésnek több mellékjelentése van a szervezeti stratégia szintjén. Az általános architektúra tervezési elvek alkalmazhatók az informatika olyan felépítése és kialakítása céljára, amely a *szervezeti, üzleti stratégiához történő illeszkedést* valósítja meg.

Szervezeti (vállalati) informatikai architektúra

- **Szervezeti (vállalati) informatikai architektúra:** Azoknak a stratégiai és architekturális tervezési elveknek a gyűjteménye. Amely magában foglalja az **információ, szervezeti (vállalati) információrendszer** és a **műszaki, informatikai architektúrát**.

Az **információrendszer vagy informatikai stratégiai tervezés (IS/IST Strategic Planning, ISSP)** és a *szervezeti informatikai architektúra* tervezés bensőséges kapcsolatban áll egymással (Ld. 3. ábra). A szervezeti informatikai architektúra készítés sok területen segíteni tudja a stratégiai tervezést, az informatikai fejlesztések, a projekt portfólió alapjait tudja megteremteni. *Szervezeti architektúra* (Ld.[86]) két formában jelenik meg, egyrészt az informatikai stratégiai tervezés folyamatának kísérője, másrészt mint egy élő organizmus, amelyik ésszerű módon alkalmazkodik a környezetéhez. Az informatikai stratégiai tervezés fekteti le a ciklikus szervezeti architektúra tervezés alapelveit. Az *információ, a szervezeti és a műszaki, informatika architektúra* az informatikai környezet és annak irányítási folyamatainak egészén átívelő, a részek közötti együttműködést megteremtő szinten létezik, és ezek az architektúra szintek kölcsönösen támogatják egymást. Az **alkalmazási (szoftver) architektúra** akkor lép működésbe, amikor egy egyedi információrendszer beszerzéséről vagy kifejlesztéséről van szó.



3. ábra Stratégiai tervezés és a szervezeti informatikai architektúra tervezés folyamata

2.1 Információ architektúra

Az információ architektúra nézőpont a szervezet által végzett tevékenységekkel és a hozzájuk szükséges információkkal. Az adatok és tevékenységek magas szintű nézete teremti meg a szervezeti szintű követelményelemzés alapjait a szervezeti (vállalati) információrendszer architektúra kialakítása során.

Információ architektúra

- **Információ architektúra:** A szervezet által igényelt és használatban lévő információ szerkezete.

Az információ architektúra kialakításával kapcsolatos leglényegesebb tevékenységek:

- *A szervezet információ szükségleteinek feltárása.* A legfontosabb szervezeti (vállalati, üzleti) tények, adatok felismerése és ezeknek megfogalmazása a szervezet (vállalati, üzleti) információ szükségleteinek formájában.
- *Az információ architektúra meghatározása.* Az architektúra meghatározása a következő módszereket használja: párhuzamos lebontás (dekompozíció), értéklánc elemzés, és eseményelemzés.
- *A funkciók közötti függőségek elemzése.* A funkciók részekre bontása (dekompozíció) érvényességének ellenőrzése, továbbfinomítása, és a függőségek feltárása.

2.2 Szervezeti (vállalati, üzleti) információrendszer architektúra

- *Az entitások és köztük fennálló kapcsolatok meghatározása.* Az információ architektúra adat oldalát pontosítja – lényegében ez egy szervezeti szintű adatmodellezési tevékenység.
- *Az információ szükségletek leképezése.* Az entitás típusok teljes listáján szereplő elemek és a feltárt információ szükségletek listáján szereplő elemek összerendelése.
- *Az entitás típusok használatának elemzése.* A szervezeti (vállalati, üzleti) funkciók által az entitás típusokon kiváltott hatások feltárása, helyességének ellenőrzése, a tevékenység hierarchia és entitás kapcsolat diagram tovább finomítása, pontosítása.
- *A szervezeti funkciók és az entitás típusok leképezése a szervezeti egységekre.* A szervezeti egységek és az adatok összekapcsolása, a szervezeti tevékenység hierarchia elemzésével és annak feltárásával, hogy az *információ architektúra* hogyan használja az adatelemeket.

2.2 Szervezeti (vállalati, üzleti) információrendszer architektúra

Szervezeti (vállalati, üzleti) információrendszer architektúra leírja szervezeti (vállalati, üzleti) információrendszereket és azokat az információkat, amelyeket a rendszerek elsődlegesen tárolnak az *információ architektúra* támogatása érdekében. *Szervezeti (vállalati, üzleti) információrendszer architektúra* tudja jelezni a kezdetekben, hogy vajon a jelenlegi rendszerek (gyakran megörökölt, elavult vagy régi rendszerek – **legacy systems**) újra hasznosítására, továbbfejlesztésére, illetve új szervezeti (vállalati, üzleti) információrendszerek kifejlesztésére van szükség. *Szervezeti (vállalati, üzleti) információrendszer architektúra* kialakítása vezérli általában a jelenlegi információrendszerek és a szervezeti tevékenységek, a szervezeti információrendszerek és a kapcsolófelületek közötti összerendelést. A szervezet és az informatika irányítása szempontjából a *szervezeti (vállalati, üzleti) információrendszer architektúra* adja meg azt a környezetet, amelyben az új rendszerek kialakítására irányuló kezdeményezések mérlegelhetők.

Szervezeti (vállalati, üzleti) információrendszer architektúra

- **Szervezeti (vállalati, üzleti) információrendszer architektúra:** A szervezet összes információrendszerének struktúráját és tartalmát (információ és funkció értelmében) definiálja.

2.3 Műszaki, informatikai architektúra

A *műszaki, informatikai architektúra* elsősorban azt a **platformot** írja le, amely *szervezeti* (vállalati, üzleti) *információrendszer architektúra* és az *információ architektúra* támogatásához szükséges hardver, szoftver és infrastruktúra elemeket tartalmazza. Ezen kívül a *műszaki, informatikai architektúra* a *platform* elemeinek integritását, összhangját tartja fenn. Az információ-technológia megoldások kivitelezése, a műszaki, technológiai környezet tervezése, az információ-technológia világának változására adandó reagálások nagy mértékben függenek *műszaki, informatikai architektúra* definíciójától. A *műszaki, informatikai architektúra* kialakítása nemcsak azért felelős, hogy a technológiai környezet korrekt módon működjön, hanem a működéshez szükséges funkciók struktúráját is meghatározza.

A platform

- A **platform**: Az architektúra központi fogalma a platform, amely a szervezet stratégiai vagyona. A platform magában foglalja az architektúrában definiált összes olyan kulcs fontosságú szolgáltatást, amely rendelkezésre áll az egyes szervezeti (vállalati, üzleti) információrendszerek számára az infrastruktúrán belül.

2.4 Az alkalmazási architektúra

Az *alkalmazási architektúra* az egyes alkalmazásokkal és rendszerekkel foglalkozik. Természetesen átfedés van általában a *műszaki, informatikai architektúra* és az *alkalmazási architektúra* között. A műszaki, informatikai architektúra a szabványokkal, technikákkal és az alkalmazási architektúra szabályozási és irányelvekkel kapcsolatos oldalaival foglalkozik. Az alkalmazási architektúra koncepcionális megalapozását és keretrendszerait a műszaki, informatikai architektúra szolgáltatja.

Alkalmazási architektúra

- **Alkalmazási architektúra**: A szoftver rendszer szerkezetére, szervezésére vonatkozó lényeges döntések halmaza, valamint azoknak az architektúra tervezési elveknek a gyűjteménye, amely irányítja a szoftver szerkezet kialakítását.

2.5 A szervezet (vállalati, üzleti) architektúra alternatív definíciói

Az előbbieken láttuk, hogy miután az informatikában megjelent az architektúra fogalma az 1960-as években, fokozatosan alakultak ki részletesebb architektúra fogalmak és azok definíciós kísérletei az informatika, az információ-technológia fejlődésével párhuzamosan. A szervezet (vállalati, üzleti) architektúra fogalma is több mint 20 éves múltra tekint vissza, de ennek ellenére nem mondható el az, hogy a fogalom kikristályosodott volna. A különböző definíciók segítenek körül járni és megérteni a probléma kört.

Néhány létező és szakmai közösségekben forgalomban levő definíció :

IEEE architektúra

- Architektúra (Ld. [93]) Az architektúra egy rendszer alapvető szervezése, amely az alkotó elemeiben, azok egymáshoz és a környezethez való viszonyában, valamint a tervezését és evolúciós továbbfejlődését irányító elvekben testesül meg.

TOGAF architektúra

- Open Group's Architectural Framework (TOGAF) meghatározása: Az architektúrának két jelentése van a szövegkörnyezettől függően: (1) egy rendszer formális leírása, vagy az alkotó elemeinek szintjéig kibontott részletes kivitelezési terv; (2) Az alkotó elemek struktúrája, egymáshoz való kapcsolatuk, és a tervezésüket és evolúciós tovább fejlesztésüket vezérlő elvek és útmutatók összessége. [113]

Clinger–Cohen Act információ-technológia architektúra

- Az információtechnológiai architektúra egy közigazgatási szervezet szempontjából egy olyan integrált keretrendszer, amely a létező információ-technológia fenntartását és fejlesztését, új információ-technológia beszerzését jelenti a közigazgatási szervezet stratégiai és információ erőforrás kezelései céljainak elérése végett. [114]⁸

Holland Architektúra Fórum⁹

- Az architektúra egy olyan normatív előírás, amely korlátozza a tervezési szabadságot, és a tervezés folyamatában a tervezési elvek egy gyűjteménye, amely előírás-ként jelenik meg.
- Általában a tervező tervezési szabadsága - nem kívánatosan - túl nagy. Az architektúra fogalma ennek a helyzetnek a kezelésére szolgál. Ezért, az architektúrát a tervezési szabadság normatív korlátozásaként lehet meghatározni. Ez a megközelítés az architektúra fogalmát egyértelműen előíró jellegűvé teszi és szigorúan elhatárolja a le-

író jellegű megközelítésektől. [123]

ArchiMate Foundation szervezeti (vállalati, üzleti) architektúra

- Elveknek, módszereknek és modelleknek koherens egésze, amelyet a szervezet felépítési struktúrájának, szervezeti folyamatainak, információ rendszereinek és infrastruktúrájának megtervezésére és kivitelezésére használnak. [68]

Capgemini szervezeti (vállalati, üzleti) architektúra

- Az architektúra elveknek, szabályoknak, szabványoknak és útmutatóknak a gyűjteménye, amelyek a szervezet jövőképét (vízióját) fejezik ki és teszik láthatóvá, továbbá a szervezeti koncepciókat, elképzeléseket valósítják meg, valamint tervezési stílusok, mérnöki módszerek, konstrukciós és kivitelezési elvek gyűjteményét tartalmazza. [63]

Gartner Group szervezeti (vállalati, üzleti) architektúra

- A szervezet jövőképe és stratégiája lefordításának folyamata a szervezet eredményes megváltoztatására, olyan modellek és kulcs fontosságú elvek kialakítása, és a szervezeten belüli terjesztése és folyamatos továbbfejlesztése révén, amelyek a szervezet jövőbeli állapotát írják le és lehetővé teszik tovább fejlődését.

A fentebbi definíciók nagy változatossága annak a jele, hogy a *szervezeti (vállalati, üzleti) architektúra* fogalma, vagy mint önálló műszaki, tudományos részterület még gyerek cipőben jár. Ugyanakkor a *szervezeti (vállalati, üzleti) architektúra* iránti széles érdeklődés azt jelzi, hogy a szervezetek érzik annak alapvető szükségletét, hogy saját fejlődésüket irányítsák (beleértve a szervezetüket, tevékenységüket és az informatikai területet) és úgy gondolják hogy a *szervezeti (vállalati, üzleti) architektúra* egy olyan eszköz volna, amely ezt az igényt ki tudja elégíteni.

Fogalmi (konceptcionális) integritás

- A rendszer tervének minden szintet átfogó nézete, és a meghatározó motívuma, amely egységes keretbe egyesíti az egész tervet.

2.6 A szervezet irányítás, igazgatás paradigmája

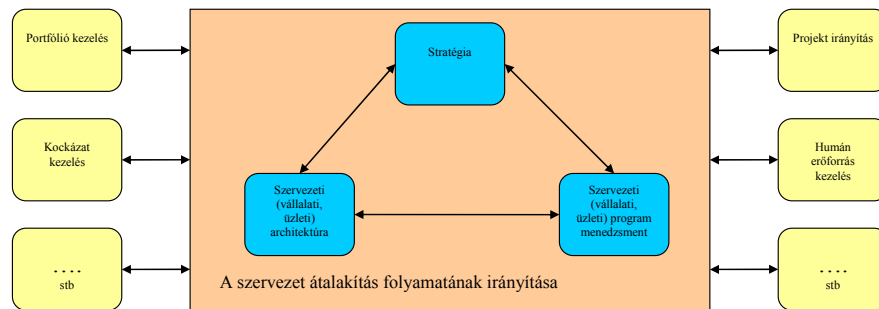
A szervezet és vezetés illetve a gazdálkodás tudományban, elsősorban angolszász területen, a bevett fogalmak mellett – *szervezetek vezetése, irányítása, ellenőrzése, menedzsment, kontrolling* –, megjelent egy újabb fogalom. Ezt a szót szó szerinti fordítással **kormányzásnak**

2.6 A szervezet irányítás, igazgatás paradigmája

lehetne vissza adni; de akár a szervezetek, vállalkozások, de akár az informatikai részleg, vagy akár egy összetett technológia (pl. SzOA, szervezeti architektúrák) esetében magyarul furcsán hangzik, ezért inkább az *irányítás* és / vagy *igazgatás* szót célszerű használni.

A szervezet irányítása („governance”)¹⁰)

- alatt egy szervezett vagy cég ellenőrzését, felügyeletét értjük, illetve az ellenőrzés, felügyelet módját.[87]



4. ábra A szervezeti (vállalati, üzleti) architektúra szerepe

Más szavakkal a szabályok betartásának felügyeletét, a szabályszerűséget, a szabályokkal összhangban történő működés számonkérését jelenti. Ebben a tekintetben a szervezeti architektúrával kapcsolatos tevékenység a *szervezet irányításának* és *átalakításának*, **transzformációjának** a része.

1. Táblázat A jelentősebb architektúra megközelítések és szintjeik

Architektúra rétegek	Számítógép architektúra	OSI modell Hálózati szoftver architektúra	Elosztott rendszerek architektúrája	Szervezeti (vállalati) architektúra
0. szint	Digitális logika	Fizikai	Fizikai réteg / hardvererőforrások	Műszaki, informatikai, technológiai architektúra (hardver, szoftver, ICT kommunikáció)
1. szint	Mikro-architektúra	Adatkapcsolati réteg		
2. szint	Utastásrendszer architektúra	Hálózati réteg		
3. szint	Operációs rendszer	Szállítási réteg	Operációs rendszer	Alkalmazási rendszer (szoftver) architektúra
4. szint	Assembly	Viszony réteg (Session)	Hálózati réteg	Információrendszer architektúra
5. szint	Magas szintű programozási nyelvek	Megjelenítési réteg	Köztes réteg (Middleware)	Információ architektúra
6. szint		Alkalmazási réteg	Alkalmazási réteg	Szervezeti (vállalati, üzleti) architektúra

A szervezeti (vállalati, üzleti) architektúrával kapcsolatos tevékenység a szervezet átalakítás, transzformáció *irányítása* részének tekinthető.

A 4. ábra a szervezet átalakítás folyamat *irányításának* egy olyan részletezett képét mutatja, amely három részterületet tartalmaz: a stratégiát, az architektúrát és a szervezeti (vállalati, üzleti) **programok** irányítását, amely bizonyos projektek egye-egy programba történő csoportosítását jelenti.

2.7 A szoftver architektúra

Az utóbbi 15 évben a szoftver architektúra ágazata egyre nagyobb hangsúlyt kapott a szoftverfejlesztésen belül. Napjainkban már túlságosan sokan használják a szoftver architektúra terminológiáját. Így ez a fogalom bekerült a nemzetközi üzleti nyelvezetbe.

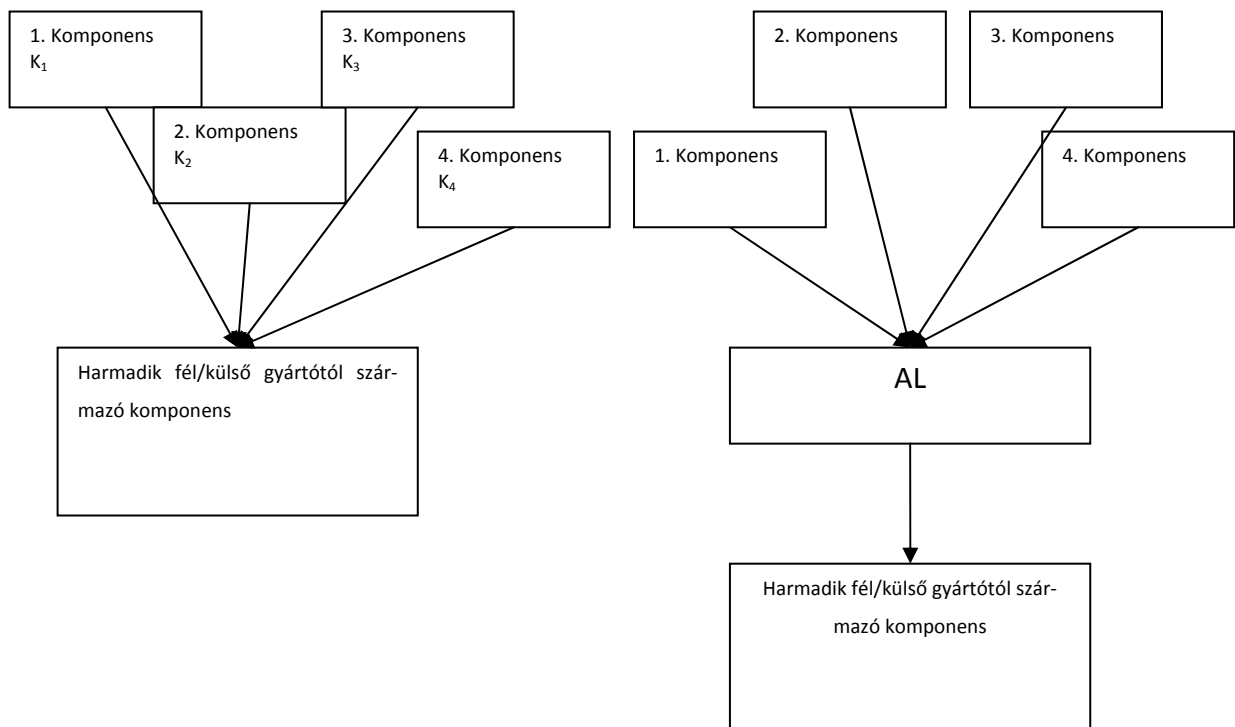
Az IEEE szervezet által adott definíciójával találkoztunk. (2.5).

- Az architektúra a rendszer alapszerkezete mely magába foglalja a rendszer komponenteit és egymáshoz illetve a környezethez való kapcsolatukat és azokat az alapelveket melyek a rendszer tervezését és fejlesztését irányítják.

A szakirodalomban egyik legújabb definíciója:

- Egy program vagy egy számítástechnikai rendszer szoftver architektúrája a rendszer struktúrája vagy struktúráinak összessége, mely tartalmazza a szoftver részeit, e részek külsőleg látható tulajdonságait és kapcsolatait.

Garlan és Shaw korai munkái alapján adott definíció ([1]):



5. ábra Szoftver architektúra komponensei

A szoftver architektúra túlmutat az algoritmusokon és adatstruktúrákon, a rendszer tervezése és általános struktúrájának meghatározása újfajta problémaként merül fel. A strukturális kérdések a következő fogalmak meghatározására keresik a választ:

- durva szerkezeti leírás illetve a központi vezérlő szerkezet megadása;
- kommunikációs protokoll, szinkronizálás és adathozzáférés megtervezése;
- funkcionalitás megadása az egyes részegységek tervezéséhez;
- fizikai elosztás;
- elemek összetétele;

- skálázási és teljesítménybeli tulajdonságok;
- az egyes tervezési alternatívák megfelelő kiválasztása.

2.7.1 Az architektúra meghatározza a struktúrát

Egy architektúra megtervezésénél az egyik legfontosabb feladat az alkalmazás ügyes felbontása egymással együttműködő egységekre, például komponensekre, modulokra vagy objektumokra. Az „ügyes” felbontást az alkalmazás követelményeinek és megszorításainak megfelelően kell elvégeznünk. Erre láthatunk két példát az ábrán (5. ábra).

A bal oldali megközelítés esetén, ha a harmadik féltől/külső gyártótól származó komponens megváltozik, akkor a K_1, \dots, K_n direkt módon hozzá kapcsolódó komponenseket is a változáshoz kell igazítanunk. A második ügyesebb megvalósításnál beépítésre került egy AL köztes (absztrakt) komponens, így ebben az esetben 4 helyett csak 1 egységet kell lecserélnünk/megváltoztatnunk. Az egyes egységek közötti függőségek minimalizálása fontos tervezési cél: ha egy komponensünk megváltozik, akkor a változás hatása és korrigálása lehetőleg ne terjedjen tovább a többi komponensre és így akár az egész architektúrára, maradjon egy lokális esemény, ahogy azt az előbbi példában is láthattuk.

2.7.2 Az architektúra meghatározza a komponensek kommunikációját

Nem csak az egyes komponensek fontosak, hanem az is, hogy ezek hogyan kommunikálnak egymással. Egy széles körben ismert gyűjteménye van ezeknek a kommunikációs módoknak, az úgynevezett **tervminták** vagy **mintázatok**. Ezek a minták lényegében újra és újra felhasználható tervrajzok, amelyek leírják a struktúrát és a résztvevő elemek közötti interakciókat.

Minden mintának ismert karakterisztikája van, ezáltal képesek megfelelni bizonyos típusú követelményeknek. Például kliens-szerver vagy ügyfél-kiszolgáló tervminta, mely támogatja a távoli szinkron kommunikációt, a szervereket a kliensek egy vagy több nyilvános interfészen tudják elérni vagy a szerver hozzáféréshez opcionálisan hozzárendelhetőek biztonsági beállítások, stb. Tervminták használatakor korábbi tudást és tapasztalatot használunk fel. Jelentősebb, nagyobb rendszerek esetén, a már meglévő tervminták kombinálhatóak egymással így elégítve ki a rendszer igényeit.

2.7.3 Az architektúra és a nem-funkcionális követelmények

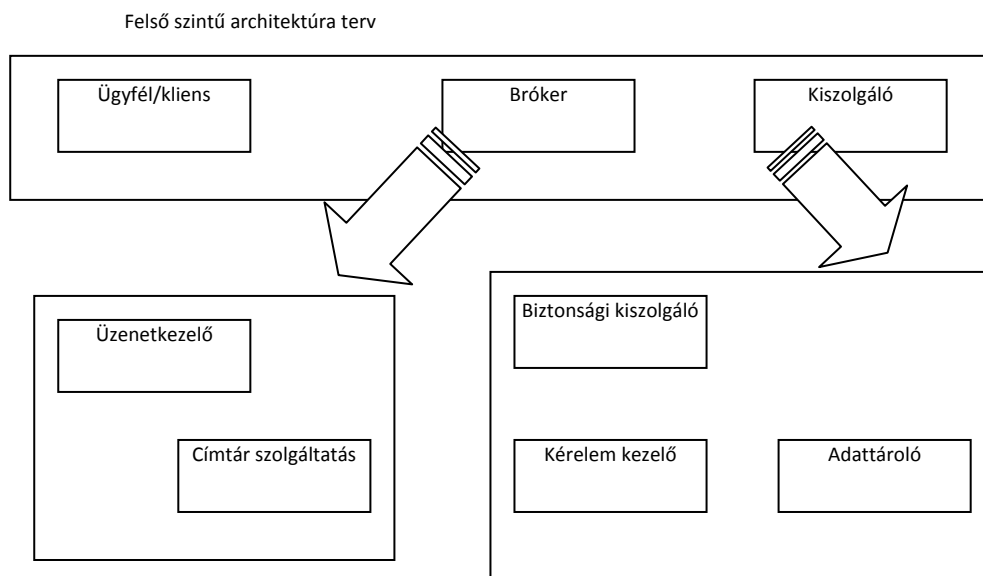
Ezek a követelmények nem a működésben jelennek meg. Nem azt mutatja meg, hogy mit csinál az alkalmazás, hanem azt, hogy a megkövetelt funkcionalitást hogyan éri el.

Három lényeges típusuk létezik:

- Technológiai peremfeltételek, korlátok
 - Azok a technológiák, amiket az alkalmazás használhat. Például nekünk csak Java fejlesztőink vannak, ezért Javában fogunk fejleszteni.
- Szervezeti (vállalati, üzleti) peremfeltételek, korlátok
 - Azok az üzleti szempontok, amelyeket figyelembe kell venni, általában ezek is adottak, már nem tudunk változtatni rajta. Például a rendszerünkhöz használt közteszoftver réteg ára megemelkedett, ezért nyílt forrású rendszerre állunk át.
- Minőségi jellemzők
 - Skálázhatóság, elérhetőség, egyszerű változtathatóság, hordozhatóság, használhatóság, teljesítmény, stb. (ld. ISO 9126). Ezek a felhasználók, vagy a fejlesztők igényeitől függenek.

2.7.4 Az architektúra, mint absztrakció

Az egyik leghasznosabb – absztrakt - leírása az architektúrának az úgynevezett *marketecture*. Ez az ábrázolás megmutatja néhány találónan megválasztott címkével és szöveggel ellátott „dobozokkal” a rendszer alapvető elemeit és azok kapcsolatát. Ez a tervezők és felhasználók közti vita tárgyát képezheti és a későbbi mélyebb analízis alapjául szolgál. A felhasználó számára lényegtelen részleteket nem tartalmazza (vagy csak fekete dobozként, *black box-ként*), csak a külsőleg látható komponensekre fókuszál. Az architektúrát nagyon gyakran hierarchikus felbontás szerint ábrázolják, a részletes kifejtést pedig a dokumentációban találjuk meg.



6. ábra Felső szintű architektúra terv

- Az egyes részekkel más-más fejlesztői csapatok foglalkoznak – felelősségeik és más csapatokkal való függőségeik külön-külön kerülnek definiálásra.
- Az ábrán a Bróker és a Kiszolgáló komponensek kerültek kirészletezésre, hiszen valószínűleg ezek rendelkeznek olyan peremfeltételekkel és korlátokkal, amelyeket kifejezetten figyelembe kell venni.
- Az ügyfelet/klienst nem részletezi az ábra – a nem fontos információk nem szerepelnek.

- Elosztott
- Keret-alapú (Frame-based)
- Kötegelt (Batch)
- Adatközpontú
 - Repozitórium, adatszótár
 - „Fekete tábla” („Black Board”)
- Interpreter
- Szabály-alapú (Rule-based)
- Réteges (Layered)
- Eldobható
- Adatfolyam központú
 - Kötegelt szekvenciális
 - Adatfolyam háló
 - Pipes and filters
- Modul rutin hívás és visszatérés
 - Főprogram / szubrutin
 - Absztrakt adattípusok
 - Objektumok, objektum-orientált
 - program hívásra alapuló ügyfél / kiszolgáló architektúra
 - hívási rétegek
- Független komponensek
 - Kommunikációra támaszkodó architektúra
 - Esemény vezérelt

7. ábra Szoftver architektúra példák

2.7.5 Architektúrális nézet

Megadja, hogy mi alapján szeretnénk leírni és megérteni egy architektúrát, az IEEE szoftver architektúra szabvány is használja ezt a fogalmat. A szakmai terminológiát Philippe Krutchen vezette be 1995-ben a „4+1 View Model” című írásában. A négy nézet a következő:

- *Logikai nézet*

2.7 A szoftver architektúra

- Az architektúra fontos elemeit és a köztük lévő kapcsolatot ábrázolja. Osztálydiagramokat vagy ennek megfelelőeket használ.
- *Folyamat nézet*
 - Az egyidejűség és a kommunikációs elemek leírására fókuszál. Az IT alkalmazásokban a legfontosabb a többszálú (multithreaded) vagy másolt (replicated) komponensek és a szinkron/aszinkron kommunikációs mechanizmusok leírása.
- *Fizikai nézet*
 - Leképezi a fő folyamatokat és komponensek megjelenítését az alkalmazás hardvereire. Más szavakkal megadja a felhasznált hardver állományt és azok kapcsolatait.
- *Fejlesztési nézet*
 - A szoftver belső szerkezetét adja meg, ahogyan a fejlesztői környezetekben is szerepel.

Egy másik felosztás szerint – SEI: „*Views and beyond*” felosztás – három nézet adja meg az architektúrát:

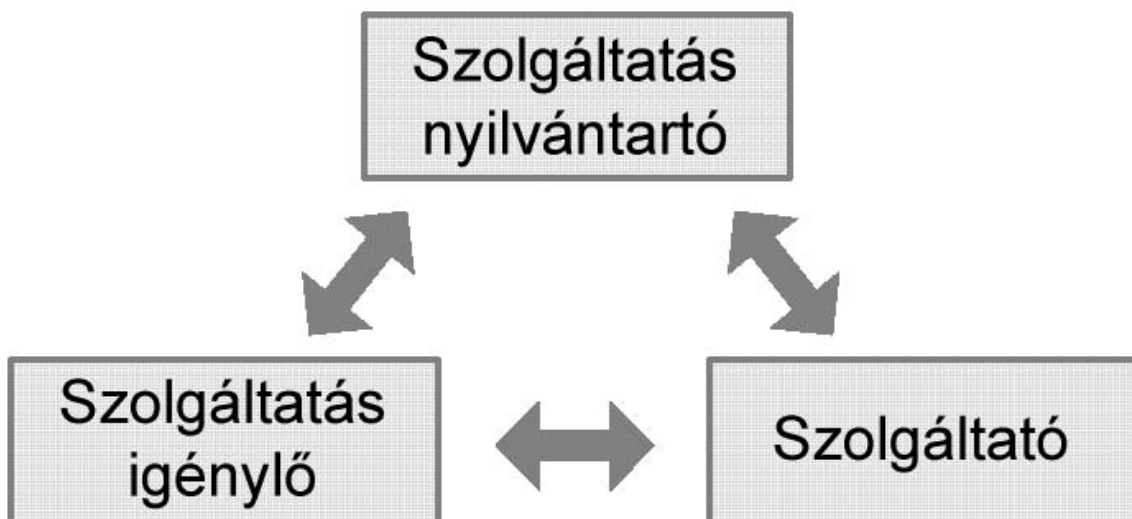
- *Modul*
 - Az architektúra strukturális szerkezete – *objektum osztályok, alrendszerek, illetve hierarchikus felosztások, asszociációk, aggregációk.*
- *Komponensek és csatolók*
 - Az architektúra viselkedési vonatkozásai. A komponensek általában valamilyen *objektumok, szálak vagy folyamatok.* A csatolók (connector) megadják az előbbi komponensek kommunikációs módját.
- *Elosztás (telepítési szerkezet)*
 - Az architektúra leírása hardver szinten – a kommunikáció milyen hálózatokon keresztül történik, mely adatbázisok milyen módon kerülnek felhasználásra, illetve ki a felelős az egyes modulokért.

3 A SZOLGÁLTATÁS ORIENTÁLT ARCHITEKTÚRA

Ahogy fejlődött az IT úgy lettek egyre több szolgáltatásai a vállalatoknak, amelyek egy része idővel helyet követelt magának az Interneten is. Az Internet alapú szolgáltatások, - a weboldalak - mellett egyre jobban terjedtek a vállalatok belül is (Intranet). A növekvő számú vállalatok belüli szolgáltatás, a fogyasztók elvárásai, arra kényszerítették a vállalatokat, hogy egyre több informatikai eszköz, program kerüljön bevezetésre az igények kielégítése miatt.

A SOA nem új keletű megoldás. A szolgáltatásorientált architektúra előtt történtek már próbálkozások a vállalatok alkalmazásfejlesztési problémáinak orvoslására. Ilyen előzmények voltak: a strukturált programozás, az objektumorientált, programozás, a komponens alapú fejlesztés stb. Az OO programozás megjelenésével lehetőség nyílt például a kód újrafelhasználhatóságára, ami egyfajta gondolati előzménye volt a SOA szolgáltatásai újrafelhasználhatóságának. Azonban a SzOA nem egy programozási modell, hanem egy üzleti gondolkodásmód, egy informatikai architektúra.

A Szolgáltatás Orientált Architektúra tulajdonképpen tervezési irányelvek gyűjteménye, illetve egy módszertan az integrációra, rugalmas informatikai infrastruktúrák kialakításához. Kiemelkedik az integrációs módszerek közül azzal, hogy nyílt szabványokon alapul. A módszertan nem foglalkozik az implementációs részletekkel. Az architektúráis elvekre koncentrálna, szabályokat és tervezési mintákat definiál. Az architektúra működésének elemi modelljét az alábbi ábra szemlélteti:



8. ábra A szolgáltatás orientált architektúra alap sémája

3.1 A Web szolgáltatás

A fenti ábra (8. ábra) a SOA architektúra legelemibb komponenseit és összefüggéseit szemlélteti. A legalapvetőbb SOA implementációnak minimum az alábbi funkciókat meg kell valósítania:

- Az üzenetküldés és fogadás módját;
- A szolgáltatások leírásának rendszerét;

A szolgáltatás leírók publikálását és feltárását, megkereshetőségét.

Ezen felül 3 alapvető szerepkört definiál:

- Szolgáltatás nyújtó
- Szolgáltatás hívó
- Szolgáltatás regiszter

Egy rendszer egyszerre több szerepkört is betölthet. Lehetséges, hogy ugyanaz a rendszer az egyik folyamatban szolgáltatás nyújtó, míg egy másikban szolgáltatás hívó.

A teljes SzOA használatához az alábbi 3 elem, az architektúra alap komponenseinek kialakítása szükséges:

- Szolgáltatás **nyilvánosságra hozatala**, publikálása;
- Szolgáltatás felderítés („discovery”), megtalálhatóság, **megkereshetőség**.
- Szolgáltatások közötti **üzenetváltás**, adatcsere, kommunikáció (szolgáltatás hívás és válasz).

A SzOA egyik központi fogalma a szolgáltatások, pontosabban a **Web szolgáltatások**.

3.1 A Web szolgáltatás

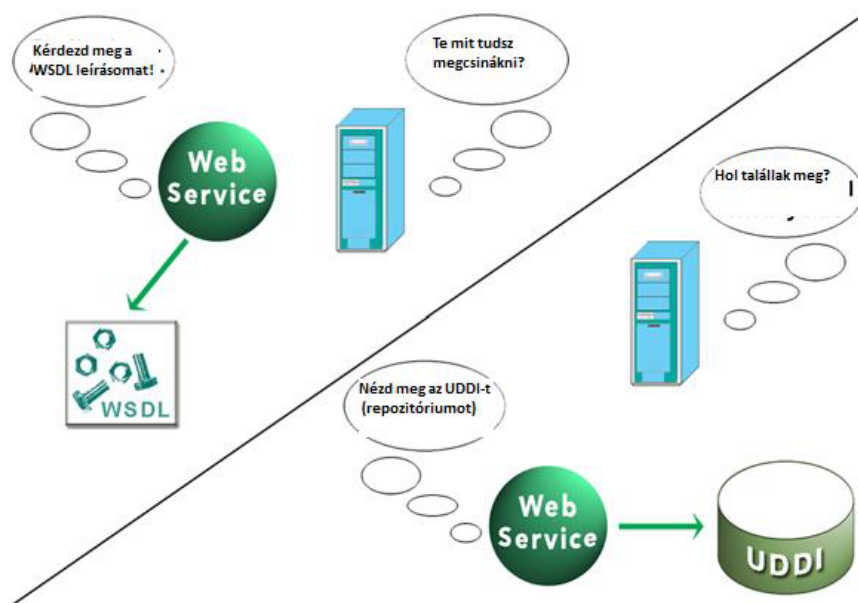
SzolgáltatásA **szolgáltatások** egy-egy jól meghatározott funkciót megvalósító távolról elérhető funkciók, objektum-orientált programozási körben metódusoknak is nevezhetnénk.

Ezek a funkciók jellemzően szervezeti (vállalati, üzleti) folyamatokhoz kapcsolódnak. A SzOA architektúra részeként definiált szolgáltatások újrafelhasználhatóak, kombinálhatók és a meglévő szolgáltatásokból új szolgáltatások építhetők. A szolgáltatások megvalósíthatók szabványos technológiákkal, így bármely a szabványt támogató alkalmazás képes csatlakozni a szolgáltatáshoz. Lényeges tulajdonságuk, hogy a megvalósítás részleteinek elrejtésével segítik a lazán csatolt architektúra kialakítását.

Web szolgáltatás (Web service)

A **Web szolgáltatás** technológia alkalmas a SzOA architektúrában definiált szolgáltatások implementálására. Bár más technológiával is megvalósítható, ez a legelterjedtebb. Egy SzOA szolgáltatásnak a következő követelményeknek kell megfelelnie:

- Leírható egy szolgáltatás-leíró nyelv segítségével;
- Közzétehető egy szolgáltatásjegyzékben;
- Megkereshető szabványos módszerekkel (akár futási időben, akár tervezés során);
- Meghívható egy jól meghatározott programozási felületen keresztül;
- Összekapcsolható más szolgáltatásokkal.



9. ábra WEB szolgáltatások leírása és megtalálása

A web szolgáltatások olyan szoftverek, amelyeket arra a célra terveztek, hogy gép-gép információcserét a világhálón keresztül interoperábilis formában megvalósítsák. A web szolgáltatásoknak van egy kapcsoló-felülete (*interface*), amely számítógép által feldolgozható írja le a Web szolgáltatást, ezt a leírást WSDL-el jelölik. A Web szolgáltatásokkal az más rendszerek a WSDL leírásban szereplő előírás alapján tudnak kapcsolatba lépni, tipikusan SOAP üzenetek keresztül, amelyeket a http protokoll közvetít.

A Web szolgáltatások az e-szolgáltatások megvalósításának alapkövei. A Web szolgáltatások definíciójukból adódóan alkalmazás, megvalósítás függetlenek, alapvetően semlegesek az egyéb technológiák viszonylatában, lehetővé teszik a szabványosítás különböző szintjeit és formáit.

3.1 A Web szolgáltatás

A Web szolgáltatások egy olyan mechanizmust nyújtanak, amely lehetővé teszi a szolgáltatások egyértelmű leírását, valamint számítógépek révén, automatizált módon történő elérésüket. Mind humán mind nem humán szereplők igénybe vehetik a Web szolgáltatásokat, továbbá a Web szolgáltatásról minden lényeges információt tartalmaz a Web a szolgáltatás felületének leírása (*interface*).

2. Táblázat Web szolgáltatások leírása

e-szolgáltatás alapfogalmai (elektronikus szolgáltatás)	Web szolgáltatás
Az e-szolgáltatás leírásának nyelve	XML, XML Schema
A szolgáltatás és a „szerződés” formális leírása	WSDL
A szolgáltatás felhasználásához szükséges kommunikációs csatornák	Portok, port típusok
A szolgáltatás nyilvánosság számára történő publikációja („meghirdetése”)	UDDI, ebXML, repozitóriumok, adat-szótárak, címtárak
Üzenetváltás, kapcsolattartás más szolgáltatásokkal	SOAP

Web szolgáltatások WSDL leírásból és a hozzája kapcsolódó SOAP üzenetekből áll össze tulajdonképpen.

Web szolgáltatás úgy definiálható, mint olyan funkcióknak az összecsomagolása, amelyek együttesen egységet alkotnak és a világhálón keresztül más szoftverek számára rendelkezésre bocsájtják. A Web szolgáltatások be tudnak csomagolni más szolgáltatásokat (*wrapper*), szolgáltatás nyújtási módokat és kommunikációs csatornákat, amelynek révén meg lehet őrizni a korábbi beruházások értékét.

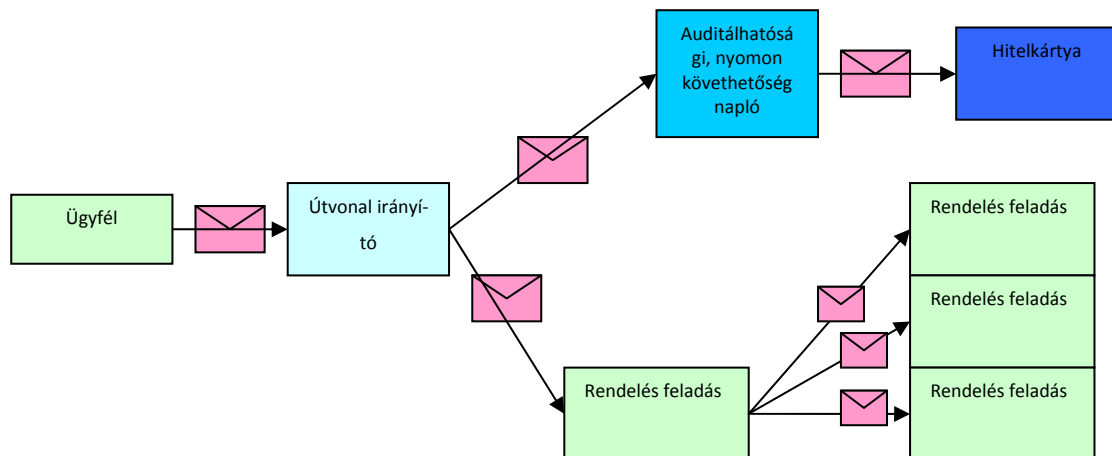
A SOAP, WSDL és UDDI voltak a legelső Web szolgáltatás szabványok melyeket publikáltak, de ezek csupán a legalapvetőbb követelményeknek feleltek meg az alkalmazásintegráció kapcsán. Hiányos volt a támogatottsága a *biztonságnak, tranzakcióknak, megbízhatóságnak* és számos más fontos funkciónak. Ez a hiányosság fokozatosan javításra került egy sor szabvány megalkotásával (közismerten „WS-*”), melyeket az IBM és a Microsoft dolgozott ki 2001-ben a W3C workshop-on. Ezeknek a szabványoknak a megalkotása, valamint egy általánosan elfogadott megegyezés létrehozása nem könnyű feladat. A specifikációk néha komplementerei egymásnak vagy egyéb esetekben átfedik egymást. Részletek a specifikációkkal kapcsolatban: <http://www.w3.org/2002/ws/>.

A **Web szolgáltatások XML szabványok**. A szolgáltatások XML-ben vannak definiál-

va és az alkalmazások ezekhez a szolgáltatásokhoz XML kéréseket küldenek. A Web szolgáltatás szabványok – ott ahol csak ahol azt csak lehet –, kihasználják a más XML szabványok nyújtotta előnyöket. Számos Web szolgáltatás szabvány létezik. Ezek kategorizálhatók, ahogy az ábra is mutatja (14. ábra).

A szabványok száma inkább bonyolultságot sugall, nem pedig az elérni kívánt egyszerűséget és sok alkalmazás csupán az alapvető szabványokat használja. Manapság már sok eszköz létezik (*library, framework*) amely támogatja a fejlesztés menetét, így a fejlesztőknek elég az adott szabványokat megérteni, nem kell feltétlenül ismerniük a pontos XML szerkezetet.

Az egyik „egyszerűsítő” irányelv, amely a Web szolgáltatások mögött van az, hogy számos üzenet mező és attribútum, melyet arra használnak, hogy olyan funkciókat támogassanak, mint a *biztonságosság* és *megbízhatóság* teljesen független egymástól. Az alkalmazásoknak csupán az a néhány adatot kell elhelyezniük az üzenetekben, amire tényleg szükségük van, és figyelmen kívül hagyhatnak minden más szabványt. Például egy SOAP kérés azonosíthatja a szolgáltatást kérőt az által, hogy tartalmazza a felhasználónevet és jelszót abban a formában, ahogy specifikálva van a *WS-Security UsernameToken* profilban. Ez a felhasználónév/jelszó információ az egyetlen biztonság specifikus fejléc elem, amelyet az üzenet tartalmaz. A *WS-Security* támogat más típusú felhasználó hitelesítést és azonosítást is (ld. 8.8), ahogy titkosítást és aláírást is, de mivel ezek nem szükségesek a szolgáltatás használatához ezért egyáltalán nem jelennek meg a SOAP kérés üzenetekben.



10. ábra Egyszerű üzenet sorozat

Egy másik célja a Web szolgáltatás szabványoknak, hogy magas színvonalú támogatottságot biztosítsanak az olyan rendszerarchitektúrák számára, melyek „közvetítőket” használnak. Ahelyett, hogy azt feltételeznék, hogy a kliensek közvetlenül küldenek üzeneteket a szolgál-

3.2 A folyamatmenedzsment rendszer elemei

tatókhoz, a közvetítő modell azt feltételezi, hogy az üzenetek számos alkalmazáson keresztülhaladnak, mire elérik céljukat. Ezek a közvetítők bármilyen módon felhasználhatják a beérkező üzeneteket: pl. útvonal irányítás, választás, naplózás, biztonsági ellenőrzés vagy akár transzformálás.

A Web szolgáltatások számos módon támogatják a közvetítő alapú architektúrákat. Ebbe beletartozik a fejléc elemek megjelölése a címzettek, valamint az „end-to-end” irányelvek támogatása biztonsági funkciókat, így biztosítják, hogy a szolgáltatás akkor is működőképes maradjon, ha az üzenetek nem közvetlenül a klientsztől a szolgáltatóhoz jutnak. Például az ábrán (10. ábra) szereplő kliens a WS-Security által biztosított biztonsági mechanizmust használja, hogy megvédje azokat a kényes információkat, melyeket csak a hitelkártya alkalmazás kell, hogy lásson az *útvonal irányító (router)* nem, melyen az üzenet időközben áthalad.

Ezt a modellt az ábra (10. ábra) szemlélteti, ahol a közvetítők útvonal irányító és naplózási szolgáltatásokat is biztosítanak. A Web szolgáltatások számos módon támogatják a *közvetítő alapú architektúrákat*. Ebbe beletartozik a fejléc elemek megjelölése (*tag*) a címzettől elvárt szerepkörrel, valamint a faltól-falig („end-to-end”) elv támogatása például a biztonsági funkciók tekintetében, így biztosítják, hogy a szolgáltatás akkor is működőképes maradjon, ha az üzenetek nem közvetlenül jutnak el a klientsztől a szolgáltatóhoz. Például az ábrán szereplő kliens a *WS-Security* által nyújtott biztonsági mechanizmust használja, hogy megvédje azokat a kényes információkat, melyeket csak a hitelkártya alkalmazás kell, hogy lásson az útvonalirányító (*router*) nem, amelyen az üzenet időközben áthalad.

3.2 A folyamatmenedzsment rendszer elemei

A folyamatmenedzsment, folyamatszervezés, folyamatkezelés öt olyan tevékenységből állítható össze, amelyek egymással integrálhatóak:

- A **folyamat-stratégia** a folyamatmenedzsment kialakításának kiindulóponja. Elemezni és értékelni kell a szervezetet, az üzleti világban, a céget, a piacait, a termékeit. A legfontosabb, hogy meg kell határozni a folyamatcélokat.
- A második lépés a **folyamatelemzés és optimalizálás**. Ebben a fázisban írják le a jelenlegi folyamatokat és tervezik meg a jövőbelieket, illetve az adat-, információ-, szervezet struktúráját, architektúráját. Folyamatokat kiértékelik az elemzések alapján és optimalizálják a minőségüket. Erre alkalmas eszköz az **összemérés, összehasonlítás** („benchmarking”). A költség és időráfordítás jelentős mértékben csökkenthető **referencia mo-**

dellek alkalmazásával, mivel a további fejlesztések számára egy stabil kiinduló pontot jelentenek.

- A következő szakasz a **megvalósítás, implementáció**, amely a szervezet felkészítését, a technikai feltételeket biztosító szoftverek, infrastruktúra bevezetését, a beavatkozások nyomon követését jelenti.
- Az egész folyamatot végigkíséri a **változtatáskezelés**. A teljes szervezetet, vállalatot fel kell készíteni a változásra, illetve a munkatársakat meg kell győzni a változás szükségességéről, el kell velük fogadtatni, és el kell érni a motiváltságukat.
- A végső szakasz a **folymat-kontrolling**, amely során az ellenőrzési és jelentési rendszereket kell kialakítani, a teljesítménymérés rendszerét meg kell tervezni. A **nyomon követés** (monitoring) napra kész információkkal látja el a szervezetet, a vállalatot a folyamatok aktuális állapotáról, költségeiről, idő- és teljesítménybeli adatairól. Ezek az információk visszacsatolásként szolgálnak a stratégiai tervezési szakasz számára, így könnyen felismerhetik, ha valami nem az előírtaknak megfelelően működik, és még időben korrigálhatják az eltéréseket.

3.3 Szervezeti (vállalati, üzleti) folyamat menedzsment

Business Process Management (BPM) összehangolja a szervezetek, és az ügyfelek igényeit. Segíti a szervezetek, vállalkozások eredményességét és hatékonyságát. A *folymatmenedzsment* támogatja a szervezetek törekvését az innovációra, a rugalmasságra és a technológiák integrációjára. Az szervezeti (vállalati, üzleti) folyamatmenedzsment már több évtizedes múltra tekint vissza, mint a szervezet igazgatás része. A fő célja, hogy javítsa az vezetők gondolkodását és a szervezet (vállalat, üzlet) irányítását.

Folyamatmodelleket több indokból is készíthet egy szervezet (vállalat), mint például teljesítményértékelő rendszerek bevezetésekor, minőségirányítási rendszer bevezetése vagy fejlesztésekor, informatikai vagy szervezetfejlesztés esetén. Ám a folyamatok teljesítményének visszamérésére, a meghozott vezetői intézkedések értékelésére is alkalmas.

Az üzleti szférában a versenyelőny megtartása miatt van szükség a jól szervezett üzleti folyamatokra, ami stabil profitot, gyors, hatékony ügyfélszolgálatot, a változó környezethez való gyors alkalmazkodást segíti.

Lényeges azt is átgondolni, a *BPM* kapcsán hogy valójában szükség van egy új informatikai eszköz bevezetésére, vagy elég a meglévő működési folyamatokat átalakítani, módosítani.

Tehát a *BPM* meghatározza az üzleti folyamatok életciklus-modelljét, amely négy alapvető elemet tartalmaz: az (1) elemzés / szimuláció, (2) megvalósítás, (3) folyamat-végrehajtás, valamint a (4) nyomon követés / monitorozás / visszamérés lépéseit. Annak érdekében felmérést kell végezni, hogy melyik az a folyamat, amelyik kevésbé támogatott, vagy ahol szervezeti átalakításra van szükség, és ott be kell avatkozni. A legtöbb informatikai szállító biztosítja mind a négy lépés meghatározását és felügyeletét, de nem minden esetben kell az összesre koncentrálni, elég csak azokra a területekre fókuszálni, amelyekre mindenképp szükség van.

3.4 A folyamatok kategorizálása

A folyamatokat tipizálhatjuk szervezeti szintek szerint. Az első kategória a szervezeten belüli folyamat, amely különböző szervezetek között lép fel. A funkcióközi már szervezeteken belül, a különböző funkcionális területek közötti folyamat. A harmadik típus a funkción belüli, ami a funkció határát nem lépi át, és végül az egy adott munkahelyhez kötődő folyamat, amelyben tartalmazzott feladatokat egy munkatárs is teljes mértékben el tudja végezni.

Az üzleti folyamatokat tudásalapú és működési folyamatokra lehet osztani. A tudásalapú folyamatok nem szabványosítottak és főleg a résztvevő személyek kreativitására, tudására támaszkodnak. Ezzel ellentétben a működési, üzemi („operational”) folyamatok szabványosítottak és többször ismétlődnek. Ezen belül még megkülönböztetünk kulcsfolyamatokat (elsődleges folyamatokat) illetve másodlagos folyamatokat is, mégpedig annak megfelelően, hogy a vállalati stratégiát és küldetést („mission”) mennyire támogatják.

Ezenkívül csoportosíthatjuk a folyamatokat úgy, hogy külső vagy belső ügyfelet, partnert szolgálnak-e ki, illetve a strukturáltság alapján, azaz vajon mennyire strukturált - strukturált, részben strukturált, strukturálatlan -, vagy automatizált - automatizált, részben automatizált, automatizálatlan.

3.5 Szervezeti (vállalati, üzleti) folyamatmenedzsment és a SzOA

A folyamatok vezérlését, automatizálását megvalósító szolgáltatásokat nevezzük folyamatszolgáltatásoknak. A SzOA architektúrában szervezeti (vállalati, üzleti) folyamatvezérlő eszközök biztosítják a folyamatszolgáltatásokat, amelyek lehetővé teszik a humán feladatok támogatását („workflow”) és a folyamatelemeket megvalósító szolgáltatások láncolásával a folyamatok automatizálását, továbbá biztosítják a folyamatok állandó monitorozását. Mivel a SzOA megközelítés legényegesebb célja a könnyedén módosítható és rugalmas üzleti folyamatok megvalósításának a segítése, ezért a SzOA architektúra központi elemei a BPM szoft-

verek és a szervezeti (vállalati, üzleti) tevékenységet nyomon követő (monitorozó) modulok (**Business Activity Monitoring, BAM**).

3.6 A SzOA és az informatika viszonya

A vállalatok sokszor követik el azt a hibát, hogy egy informatikai probléma megoldását csak abban látják, hogy még több pénzt fordítanak informatikai beruházásokra, remélve azt, hogy így ki tudják küszöbölni a problémát. Az egyes részlegek számára vásárolt, általuk beszerzett rendszerek többnyire alapjában véve eltértek egymástól, így jöttek létre a nagyobb szervezetekben a silók, az egymástól elszigetelten létező, decentralizált alkalmazások. Az informatikai részleg, funkció nem tekintette ezt a helyzetet problémának, mivel ezeket a rendszereket csak üzemeltette. Így jöhetett létre a rendszerek közötti nagy fokú inkompatibilitás, sőt együttműködési képtelenség, az interoperabilitás teljes hiánya. Az egyes rendszerek az egyes platformokra eltérő technológiával készültek, és nem támogatták a többi területet, azaz nem lehetett a folyamatokat módosítani és más szervezeti (vállalati, üzleti) folyamatokhoz kapcsolni.

Az alkalmazásintegrációt **pont-pont alapon**, különböző technikával oldották meg, ennek a következménye a rendszerek bonyolultságának növekedése volt. A szervezeti (vállalati, üzleti) egységesség, mint ilyen fogalom a gyakorlatban az ilyen típusú szervezeteknél nem létezik, mivel a funkciók, az adatok, a folyamatok elkülönülnek egymástól, az adatok heterogének és nem konzisztensek. Ebben a környezetben a költségek nagy része a karbantartásra és üzemeltetésre megy el, és nem marad anyagi forrás a fejlesztésekre, a stratégiai célokra.

Ezzel párhuzamosan az szervezeti (vállalati, üzleti) elvárások nagymértékben átalakultak az informatikával szemben.

A piaci változásokra való gyors és rugalmas reagálás előnyt jelent a vállalat számára. A vállalaton belül, a vállalat és a vásárlók, a vállalat és a partnerek közötti folyamatok automatizálása és informatikai támogatása szükséges a megfelelő kommunikációhoz. Például hogy képesek legyenek az elektronikus értékesítési csatornák kiszolgálására. A munkamenet könnyebbé tétele végett átlátható, mérhető és rugalmasan alakítható üzleti folyamatokat kell kialakítani. A szervezet egységesítése érdekében célszerű egységes vállalati **adatszabványokat** és **törzsdokumentumokat** létrehozni. A szemléletváltás segíti az informatikai költségek szinten tartását és ésszerűbb felhasználását.

3.7 A SzOA informatikai stratégia

Tehát a problémák főleg a *folyamatok, alkalmazások és adatok integrációja* tekintetében, valamint a fejlesztés és üzemeltetés hatékonysága körül jelennek meg.

A fent említett elvárások jelentősen függenek az informatika részlegi rugalmasságától, hatékonyságától, és jelzi, hogy a már meglévő architektúrával, illetve irányítási megközelítéssel a kitűzött üzleti célok nem érhetőek el.

Erre a problémára a SzOA a válasza informatikai iparágnak. Egy olyan megoldást kerestek, amely képes az általános és sürgős üzleti igények kielégítésére. A SOA (Service Oriented Architecture), magyar megfelelője Szolgáltatás Orientált Architektúra, az a megközelítés, amely képes a fenti elvárásokat kielégíteni.

3. Táblázat Az objektum-orientált és szolgáltatás-orientált paradigma összehasonlítása

Objektum	Szolgáltatás
Önálló tulajdonságokkal rendelkezik	Önálló tulajdonságokkal rendelkezik
Lazán kapcsolt	Rugalmas, lazán kapcsolt
Van pillanatnyi állapota	Állapotnélküliség
Objektumosztály (egyforma belső tulajdonságú objektumok)	Szolgáltatásplatformok (szolgáltatások összekapcsolhatósága)
Objektumosztályon belül példányok	A szolgáltatásplatformon belül több rész-szolgáltatás összekapcsolhatósága
Objektum elrejt a belső tulajdonságait a használat során	Elrejt a belső tulajdonságait.
Absztrakció	Absztrakció
Öröklődés	Nem jellemző
Dinamikus feladatok megoldására ideális	Gyorsan változó üzleti elvárások kielégítésére ideális
Az objektumok újrafelhasználhatók	Újrafelhasználható komponens

3.7 A SzOA informatikai stratégia

A SzOA fogalmát sokféleképpen lehet definiálni. A szűkebb értelemben vett megközelítés egy technológiai megközelítés, mely szerint a SzOA egy architektúra, egy technológiai modell és tervezési irányelvek, útmutatók gyűjteménye, amelynek alapvető célja az, hogy az egy-

mástól eltérő, különálló, erősen decentralizált alkalmazásokból egy szervezeti (vállalati, üzleti) folyamatokat egységes keretben kezelő alkalmazásokat alakítsanak ki.

A szervezési, vezetési illetve gazdálkodási szempontból a SzOA egy **informatikai stratégia**, amely szabályozza az informatikai részleg működési és fejlesztési folyamatait és előírásait, valamint vezérli az egységes SzOA alapú vállalati architektúra kialakítását.

A SzOA stratégiát két részre lehet bontani, a szolgáltatás infrastruktúrára (SzOA architektúra) és a SzOA irányításra (*SOA Governance*). A szolgáltatás infrastruktúrája alatt a szervezet (vállalat, üzlet) rendszereit működtető technológia értendő.

3.8 A Szolgáltatás Orientált Architektúra alternatív definíciói

A Szolgáltatás Orientált Architektúra definíciójának könnyebb megértése érdekében először meg kell határozni a szolgáltatás és a **szolgáltatás-orientált megoldás** fogalmát.

Egy **szolgáltatás** meghívásakor egy meghatározott tevékenység kerül elvégzésre. Ezt a funkciót bármikor meg lehet ismételni. Például az operációs rendszer funkció, saját fejlesztésű szervezeti (vállalati, üzleti) logika / művelet vagy „dobozos” alkalmazás egy modulja.

Egy *szolgáltatás-orientált megoldás*, egy olyan alkalmazási rendszer, amely szolgáltatási komponensek összeépítéséből, kompozíciójából áll. A hálózaton keresztül hozzáférhető független szolgáltatásokat egy SzOA környezetben anélkül lehet használni, hogy ismernénk a működési sajátosságait, platformjukat.

Szolgáltatás-orientált architektúra

- *A szolgáltatás-orientált architektúra az üzleti folyamatok integrációját támogató és az informatikai infrastruktúrát kihasználó biztonságos, szabványos komponensek (szolgáltatások) keretrendszere, amelyben a szolgáltatások a változó szervezeti (vállalati, üzleti) prioritásoknak megfelelően kombinálhatók és újra felhasználhatók. [23]*

3.9 Szolgáltatási infrastruktúra

A SzOA technológia alapja a szolgáltatások, a *Web szolgáltatások*. Az informatikától igényelt szolgáltatások, információrendszer funkciók, amelyek Web szolgáltatások formájában is megfogalmazhatók.

Egy SzOA környezetben a szervezeti (vállalati, üzleti) igények, információrendszer szolgáltatásokkal szemben támasztott követelmények Web szolgáltatások formájában elégíthetők ki. Ezek a szolgáltatások önállóan is működőképeseek, platform- és eszköz függetlenek, szab-

ványos, jól definiált kapcsoló felülettel („interfészsel”) rendelkeznek, és az elosztott hálózatokban szabványos adatcsere és kommunikációs protokollokkal érhetőek el.

A szolgáltatások két csoportba sorolhatók aszerint, hogy rész vagy teljes szervezeti (vállalati, üzleti) folyamatokat valósítanak-e meg. Az szervezeti (vállalati, üzleti) szolgáltatások megvalósításához elengedhetetlenek az informatikai szolgáltatások. Az informatikai szolgáltatások révén a Web szolgáltatások informatikai folyamatai, adatfeldolgozási folyamatok egységesebbé és újrahasznosíthatóvá válnak. Ezek az informatikai szolgáltatások adják az alapját a szolgáltatásoknak, és az szervezeti (vállalati, üzleti) végfelhasználók számára láthatatlanok maradnak. Informatikai szolgáltatások például az archiválás, a naplózás, a dokumentumtárolás.

3.9.1 A SzOA infrastruktúra részei

Web Szolgáltatások:

- Webszolgáltatások olyan protokoll és szabványgyűjtemények, amelyek segítik az alkalmazások közötti kommunikációt és információcserét. „*A szolgáltatás olyan ismételtelhető funkció, amely meghívásakor elvégez valamilyen meghatározott tevékenységet.*” [115]

A fejlesztők a nagy, bonyolult, monolitikus alkalmazások helyett olyan szervezeti (vállalati, üzleti) és informatikai *szolgáltatásokat* hoznak létre, amelyek teljesen függetlenek egymástól, önmagukban megállnak, meghatározott korlátozott, de mégis használható funkcionalitást nyújtanak. Kialakításuk után tulajdonságaik, sajátosságaik leírását szabványos felületen („interfészen) keresztül publikálják. Az elemibb szervezeti (vállalati, üzleti) és informatikai szolgáltatások összekapcsolásával hozható létre egy összetett, szervezeti (vállalati, üzleti) folyamat centrikus a tényleges szervezeti folyamatokat támogató szolgáltatás.

A *Web szolgáltatások* biztosítják, hogy a SzOA architektúra megvalósításánál ne találjunk ki mindent újra, hogy a komponenseket akár többször is újra felhasználhassuk, mindezeket szabványosított környezetben, szabványos építőelemek felhasználásával lehet megtenni.

A *Web szolgáltatások* XML-ben leírt dokumentumok segítségével tudnak kommunikálni egymással.

3.9.2 Szolgáltatási sín (Enterprise Service Bus, ESB) A szerviz orientált architektúra irodalmában gyakran használt kifejezés az ESB.

Az egyik angol szakember amikor először meglátta az **ESB** rövidítést, nem tudta, hogy mi az összefüggés az „Extra Special Bitter” („Extra Speciális (angol) Barna Sör”) és a szoftver-integráció architektúra egyik építőeleme között. Amikor megtudta, hogy a kifejezés (a magyar szakmai kereszttségben) szervezeti (vállalati, üzleti) „**Szolgáltatási sít**” (*Enterprise Service Bus*) takar, akkor egy kicsit az érdeklődése lelohadt és alábbhagyott

Álljon itt néhány sor a történeti előzményekről, hogy vajon honnét is ered az ESB fogalma. A 2004-2005 közötti időszakban, az SzOA-t kikiáltották az informatikai iparban a szállítók, gyártók és tanácsadók "következő nagy technológiai lépésnek" a vállalati integráció körében. A szoftver eladóknak szükségük volt valami újra, ami segített nekik, hogy el tudják adni a SzOA-t támogató integrációs technológiájukat. Ezért egyik szoftver gyártó megalkotta a *ESB* kifejezést. Mostanság, minden eladó saját *ESB*-vel rendelkezik, amely alapjában véve egy saját *üzenetbróker* és *üzleti folyamat vezénylő* technológia keveréke, amely természetesen rendelkezik azzal a sajátossággal, hogy képes integrálni a Web a szolgáltatási végpontokat. Nagyon sok definíció látott napvilágot az ESB-ről. Mindegyik többé-kevésbé megegyezik

Szolgáltatási sín (Enterprise Service Bus, ESB):

- A **szolgáltatási sín** a SzOA alapú IT-infrastruktúra gerinceként működik, egységes, központilag kezelt, kommunikációs és biztonsági szolgáltatásokat kínál a szolgáltatások és a SzOA infrastruktúra elemeinek az integrálására, és a szolgáltatáshoz kapcsolódó szabályzatok és nem funkcionális követelmények betarttatására.

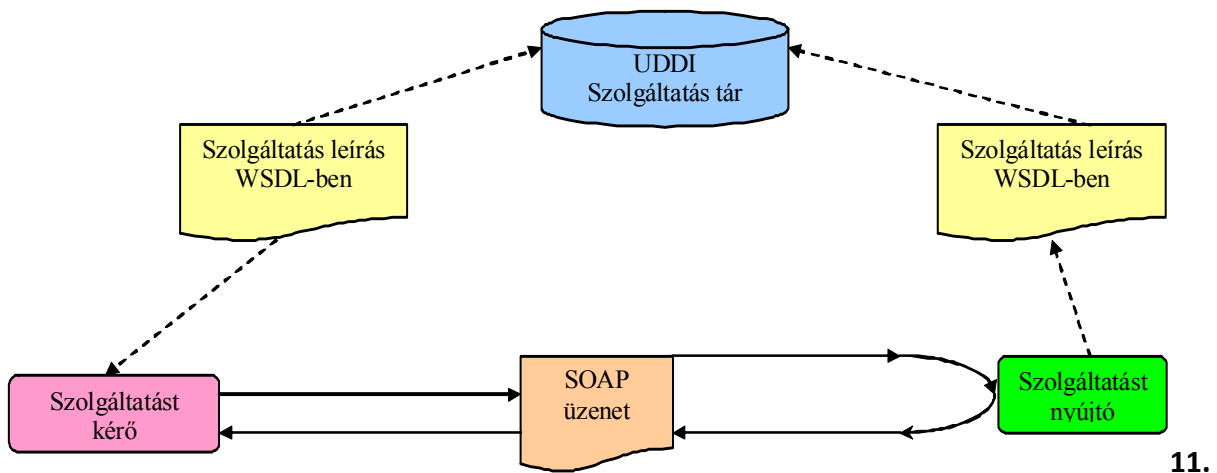
A piacon lévő ESB-khez, különösen a nyílt forráskódúakhoz bárki hozzáférhet. Ezek tipikusan egy *üzenetközpontú köztesszoftver* réteg architektúrát nyújtanak és rendelkeznek azzal a tulajdonsággal, hogy képesek egy külső végponthoz csatlakozni TCP/IP, SOAP, JMS, FTP vagy más protokollokon keresztül.

A *Web szolgáltatás* gyakran eltérő platformúak, eltérő programozási nyelven íródtak, különböző szabványok alapján készültek. Ahhoz, hogy ezeket a Web szolgáltatásokat megfelelően össze tudjuk kapcsolni, szükségünk van egy **köztes rétegre** („middleware”), a szervezeti szolgáltatás sínre. A szolgáltatások is és az alkalmazások is, csak a vállalati szolgáltatás sínrel tudnak kommunikálni, egymással csak a szolgáltatási sínen keresztül. A szolgáltatási sín a szolgáltatásorientált architektúra egyik legfontosabb része.

A szolgáltatási sín feladatai [115]:

- Üzenetek *továbbítása* a szolgáltatások között;
- Protokollok *konverziója* a hívó és szolgáltatás között;
- Üzenetformátumok *átalakítása* a hívó és a szolgáltatás között;
- Különböző forrásokból származó események *kezelése*.

Egyik, talán legfontosabb előnye a szolgáltatási sínnek, a szolgáltatások, és azok kapcsolatainak kezelése, kézben tartása. Továbbá, segítségével valósul meg a laza csatolás a szolgáltatást igénybevevő (szolgáltatás, vagy alkalmazás) és a szolgáltatást nyújtó között, ezáltal a hívó szolgáltatásnak nem kell tudni az őt kiszolgáló szolgáltatás helyéről.



11. **ábra Kommunikáció a szolgáltatások között a szabványok segítségével**

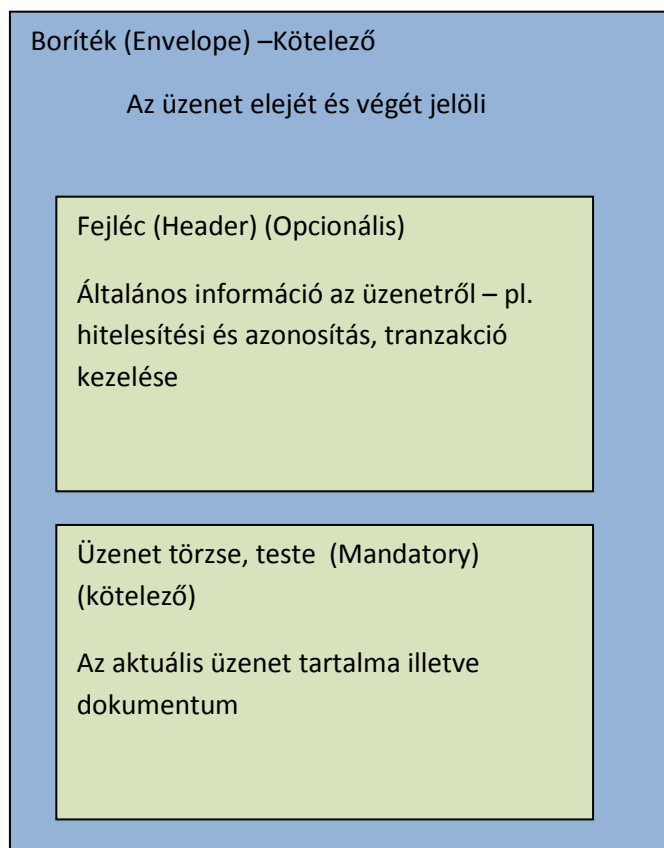
3.9.3 SOAP és az üzenetküldés

A SOAP volt az eredeti *Web szolgáltatás szabvány*. Ez manapság is az egyik legfontosabb, ráadásul széles körben elterjedt. Egy egyszerű, de kiterjeszthető XML alapú alkalmazás-alkalmazás kommunikációs protokollt specifikál, amely nagyjából ekvivalens az RPC-vel vagy a JAVA RMI-vel, de sokkal kevésbé komplex és emiatt sokkal könnyebben implementálható. Ez az egyszerűség annak köszönhető, hogy elkerüli az olyan komplex problémák kezelését, mint az elosztott szemétyűjtés és az objektumreferenciák, hivatkozások átadása. A SOAP egy egyszerű, de kiterjeszthető XML alapú üzenet orientált protokollt definiál a távoli szolgáltatások meghívásához, olyan transzport protokollokon keresztül, mint például a *HTTP, SMTP* vagy az *UDP*.

SOAP (Simple Object Access Protocol):

- XML-alapú üzenetküldő protokoll. Használható a *Web szolgáltatás* kérések és válasz-üzenetek elküldés előtti kódolására.
- Hálózati protokoll független, bár az Interneten való felhasználásnak megfelelően általában a HTTP-TCP/IP hálózati protokollokat használják, de RPC (Remote Procedure Call) üzenetcsere protokollal együtt is használható.

A SOAP üzeneteknek egyszerű a struktúrájuk, ahogy az ábra (12) mutatja. Az üzenet fejléc információkat tartalmaz az üzenet tartalmáról, valamint általában olyan elemeket, mint például a biztonsági zsetonok (security tokenek) és a tranzakció értelmezési környezet leírását. Az üzenet teste, törzse (*body*) tartalmazza magát az üzenet tartalmát, amely az alkalmazás specifikus információkat tartalmazza. A SOAP szabvány nem határozza meg, hogy mit tartalmazhat egy üzenet fejléce, így bármikor kiterjeszthető a SOAP szabvány anélkül, hogy a specifikáción változtatni kellene.



12. ábra SOAP üzenet struktúra.

A SOAP üzenet egy közös XML dokumentum mely 4 részből áll. A boríték elem képviseli a gyökér elemet. Van egy opcionális header elem melyen feltételeket lehet megszabni, pél-

dául a hitelesítés és azonosítás formájára. Van egy kötelező törzs, test, amely tartalmazza az üzenet célpontját. Valamit lehet még egy opcionális hibakezelő (*Fault*) elem, amely a hiba jelzésére szolgál.

A SOAP kifejezés eredetileg jelentett az Egyszerű Objektum Hozzáférési Protokolt (Simple Object Access Protocol) jelentette, de ma már csupán nem is csak egy betűszó, hanem csak egy egyszerű szó, hiszen nyilvánvalóan semmi köze a távoli objektumokhoz való hozzáféréshez. A kliensek SOAP XML kéréseket küldenek a szolgáltatókhoz, akikről szintén egy SOAP XML válaszüzenetet kapnak. A kérés általában a fejlécben hordoz egy felhasználónevet és hash-elt jelszót, hogy a szolgáltatás be tudja azonosítani a felhasználót, aki a lekérdezést végzi.

Két fő információcsere formátumot támogat az **RPC**, és a **dokumentum orientált** mintázatot. A **RPC szinkron**, kérelem-válasz jellegű megközelítés. A dokumentum orientált minta egy **aszinkron** megközelítés.

A SOAP három tulajdonságot határoz meg az alapértelmezett **névtérben** (<http://www.w3.org/2001/12/soap-envelope>), *actor*, *mustUnderstand*, és *encodingStyle*. A megfelelő SOAP fejléc (*header*) határozza meg, hogy hogyan kell feldolgozni az üzenetet. Az *actor* határozza meg, hogy a címzetnek hogyan kell feldolgoznia az üzenetet. A *mustUnderstand* azt határozza meg, hogy kötelező-e a *header* használata. Az *encodingStyle*-t a dokumentum adatainak a meghatározására használják.

A másik nagyon fontos sajátossága a SOAP üzeneteknek, hogy a valóságban a *kérés*, és a *válasz* teljesen külön üzenetek. Egy üzenetet úgy kell elképzelni mit ha csak az egyik – kérelem vagy válasz – üzenetet tartalmazná.

Számos más szabvány is szerepel a *Web services Messaging* kategóriában, beleértve a *WS-Addressing*-et és a *WS-Eventing*-et. *WS-Addressing* azért létezik, mert a Web szolgáltatások nem függenek a HTTP protokoll használatától. SOAP üzeneteket küldhetünk számos transzport protokollon keresztül, mint pl.: TCP/IP, UDP, e-mail (SMTP), üzenet sorok és a *WS-Addressing* biztosítja a szállítás-semleges mechanizmust a szolgáltatások címzéséhez és üzenetek azonosításához. A *WS-Event* támogatást nyújt a publikál, előfizet (publish-subscribe) modellhez, az által, hogy definiálja a feliratkozás, előfizetés típusát.

A SOAP szolgáltatások általában a WSDL (Web Service Description Language) segítségével kerülnek leírásra és felkutatásuk az UDDI (Universal Description Discovery and Integration)

könyvtár, címtár segítségével történik. A szolgáltatások leírhatják azokat a követelményeket, amelyek például a biztonságosságra és megbízhatóságra vonatkoznak a policy állítások segítségével. Ehhez nyújt segítséget a *WS-Policy* keretrendszer és olyan specifikus policy szabványok, mint pl. a *WS-SecurityPolicy*. Ezek az irányelvek hozzacsatolhatóak a WSDL-hez vagy külön irányelv tárban is elhelyezhetőek, amihez a *WS-MetadataExchange* biztosít hozzáférést.

3.9.4 UDDI

UDDI-ban regisztrálják a Web szolgáltatásokat, mint repozitóriumban, címtárban és itt lehet keresni a Web szolgáltatásokat. Az UDDI egyik fő szolgáltatása a Web szolgáltatások megtalálhatósága. A Web szolgáltatást nyújtó ebben a repozitóriumban tárolja el a szolgáltatásról az alábbiakat:

- ki;
- mit,
- hol;
- és hogyan;
- vagyis a szolgáltatás leírását, a szolgáltatás helyét (URL) és a szolgáltatás eléréséhez szükséges felület leírását (*interface*).

A szolgáltatás potenciális felhasználója az UDDI-ban keresheti és találhatja meg az igényelt szolgáltatást. Ezt a humán felhasználó böngészővel, a számítógép valamilyen programmal teheti meg.

UDDI (Universal Description, Discovery and Integration Business Registry):

- Az UDDI egy XML alapú adatbázis, amelyen keresztül nyilvánosságra lehet hozni az elkészített Web szolgáltatásokat. Segítségével lehetővé válik a szervezetek számára egymás megtalálása, illetve a közös Internetes kommunikáció kiépítése.

Három részből tevődik össze:

- *Fehér lapok*: cím, elérés, azonosítók (hasonlítható egy telefonkönyvhöz).
- *Sárga lapok*: ipari besorolás taxonómiák alapján (hasonlítható a szaknévsorhoz).
- *Zöld lapok*: műszaki, informatikai információk a szolgáltatásokról.

Az UDDI valójában a **szolgáltatástár (Registry-Repository)**: itt tárolódnak a szolgáltatásdefiniciók és a hozzájuk kapcsolódó követelmények. Erre a funkcióra nagy szükség van, hiszen biztosítja a szolgáltatás tervezési és fejlesztési ciklusának hatékony irányítását, a szolgáltatások publikálását, újrafelhasználását, a függőségek, verziók és egyéb követelmények kezelését. A nyilvános *szolgáltatástárak* az szolgáltatást nyújtó szervezetről, szervezeti egységről tartalmazznak bejegyzést, leírva a szervezet legfontosabb jellemzőit. Ez a bejegyzés egy vagy

több olyan szervezeti (vállalati, üzleti) terület leírását tartalmazza, amelyek a szervezet által nyújtandó szolgáltatásról gondoskodnak. Azonban ezek a szolgáltatásokat - szervezeti, vállalati, üzleti – megvalósíthatják Web szolgáltatások vagy más alkalmazások is, amelyek nem feltétlenül Web szolgáltatások. A Web szolgáltatások használatához nemcsak adatokat, hanem utasításokat is továbbítani kell.

A UDDI két fontos utasítás-hozzáférési felületet - API-t (Application Programmable Interface) - tartalmaz: egy *publikálási (publish)* és egy *lekérdező (inquiry)* felületet. A publikálási felület írja le a szolgáltatások regisztrációját ebben a szolgáltatás címtárban, a lekérdező felület pedig a szolgáltatások kereshetőségére nyújt funkcionális szolgáltatásokat. Az adott szolgáltatás címtárban annak a szabályait rögzíti, hogy a szolgáltatást kérők hogyan kereshetnek benne.

A Web szolgáltatások leírását megtestesítő XML WSDL leírások számára az UDDI egy tárolási mechanizmust definiál az adatbázisban (címtárban). Az UDDI rekordok többféle olyan információt tartalmaznak a szolgáltatásokról, amelyek segítenek keresni a kérdésekre adandó válaszok megtalálása végett. A következő kérdésekre kaphatunk választ belőlük: *Ki? Mit? Hol? Hogyan?*

- A *Ki* az szolgáltatás üzemeltetőjével kapcsolatos adatokat adja meg. A *Mit* a szolgáltatás besorolását, és leírását jelenti.
- A *Hol* a szolgáltatás igénybevételenek az URL címe.
- A *Hogyan* az interfész és egyéb tudnivaló leírása.

Az UDDI modellje négy részből áll:

1. *businessEntity*: leírás és a szolgáltató adatai.
2. *businessService*: A nyújtott szolgáltatások.
3. *bindingTemplate*:e Hogyan hívható a szolgáltatás.
4. *Technical Model:s* A szolgáltatás működése.

Az UDDI XML használ az adatok leírására.

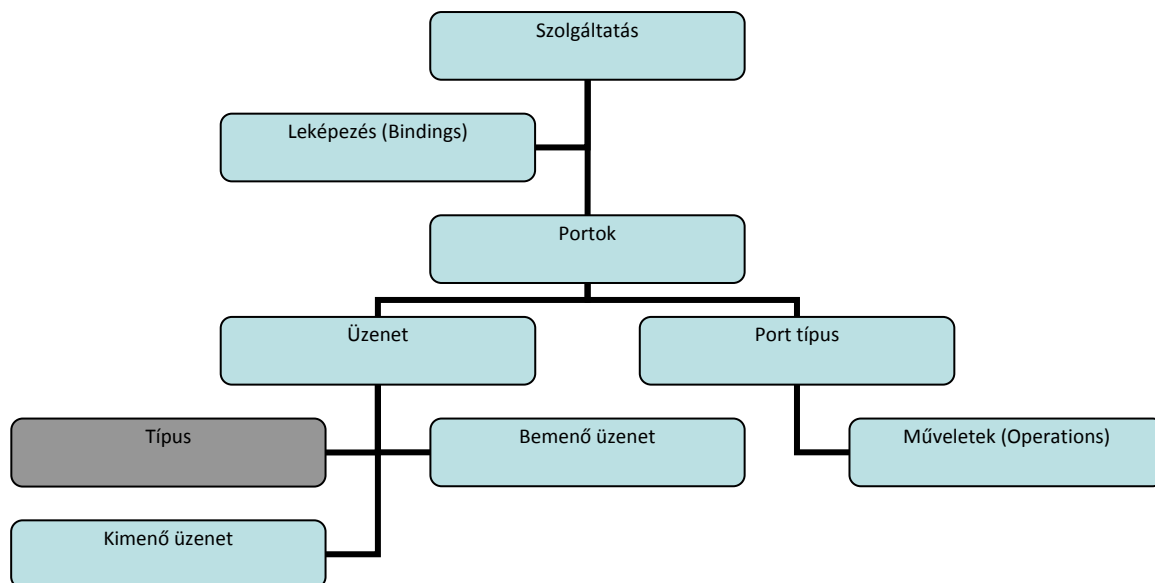
3.9.5 WSDL modellezésének alapfogalmai

A **WSDL** egy XML dokumentum, amely leírja a szolgáltatásokat. Meghatározza, hogy milyen szolgáltatások vannak, és hogyan lehet meghívni őket. Meghatározza az üzenet formátumát, és a kapcsolódási pontot, hogyan érjük el a szolgáltatást. Ezen kívül meghatározza, hogy szinkron, vagy aszinkron kommunikáció legyen-e. Röviden a WSDL válaszol a *Mit?, Hol?, Ho-*

gyan? kérdésekre. Milyen szolgáltatások vannak? Hol érhető el? Hogyan érhető el? Az adat formátumok gyakran be vannak ágyazva a WSDL-be XML formátumban. Ezért sokszor a WSDL és a SOAP együtt határozzák meg a web szolgáltatást az interneten. Egy program beolvassa a WSDL, hogy megtudja szolgáltatásokat, majd a SOAP segítségével felveszi a kapcsolatot a WSDL leírás alapján. Ez a folyamat automatizálható.

3.9.5.1 WSDL konstrukciója

A web szolgáltatás definiálja azokat a kapcsolódási pontokat, melyek leírják, hogyan lehet kapcsolódni a szolgáltatáshoz. Közzéteszi a műveleteket melyek tartalmazzák a bemeneti paraméterek leírását, és a válasz értelmezéséhez szükséges leírást. Minden üzenet adatelemekből áll, XML formátumban vannak leírva. Ez lehet egyszerű XSD, vagy akár összetett is. De a WSDL szabvány lehetővé teszi más adatok közvetítését is mint például a CORBA *Interface Definition Language (IDL)* adatait.



13. ábra A WSDL alapelemei

WSDL adattípusok:

A WSDL adattípusok széles körének definiálását teszi lehetővé. A szabvány megengedi, a típus definícióját a WSDL fájlban, vagy akár egy külön fájlban is.

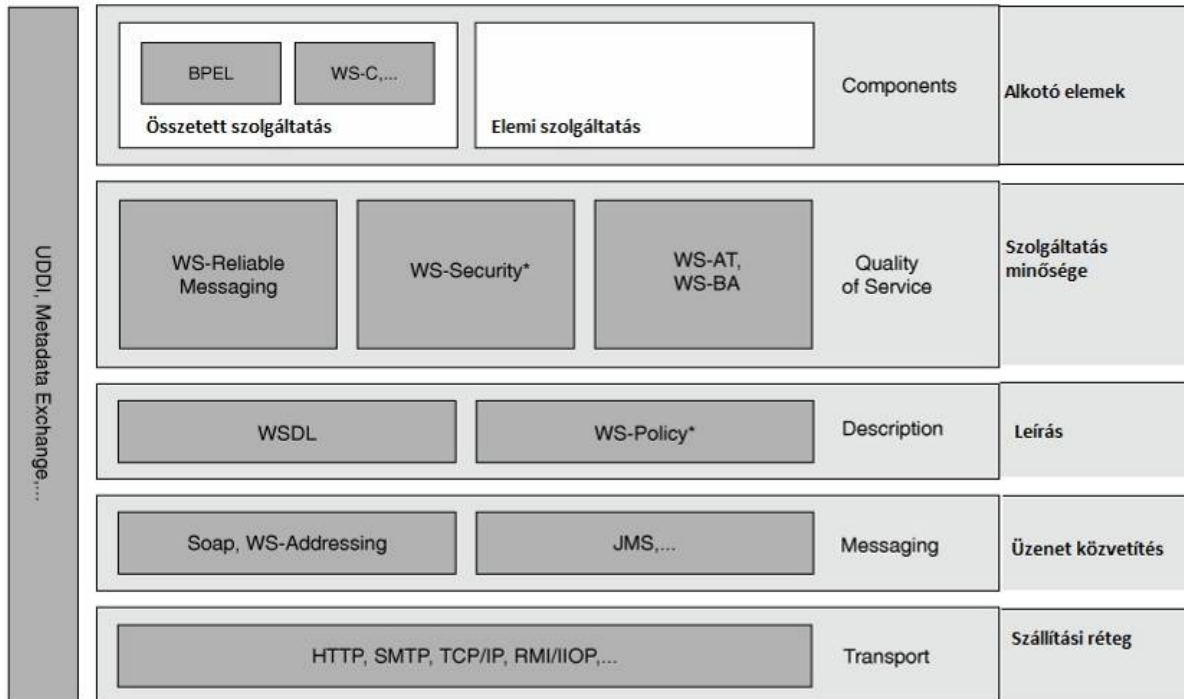
WSDL (Web Services Description Language):

- A **WSDL** a *Web szolgáltatások* egyik alapeleme. *Web szolgáltatások* leírására használt szabványos, XML alapú nyelv. Meghatározza a Web szolgáltatás adatainak típusait, a

használandó protokollokat, a Web szolgáltatás üzenetét stb. Deklaratív programozási stílusban írja le a szolgáltatást.

- A WDSL állomány olyan XML dokumentum, amely részletesen leírja a szolgáltatást nyújtó kommunikációs végpontokon a szolgáltatás sajátosságait, igényeit. Azt, hogy hogyan és milyen formátumban kell megkapniuk a szolgáltatásoknak az adatokat és utasításokat ahhoz, hogy a szóban forgó feladatot elvégezzék.

A közös nyelv az **XML (Extensible Markup Language)**; az XML egy sokkal átfogóbb szabványon, az **SGML (Standard Generalized Markup Language)** alapul, s annak bizonyos fokig szűkített változata. A *meta-nyelvnek*, tag nyelvnek is nevezik az SGML-t és az XML-t; e nyelvek sajátossága az, hogy lehetőséget ad maguknak a szabályoknak, az alkalmazott struktúrának a definiálására is, a konkrét szöveg/dokumentum értelmezését lehetővé tevő nyelvtan megfogalmazására, a szóban forgó területhez illeszkedve. Vagyis egy XML-dokumentum szerkezete az XML-ben megfogalmazott dokumentumot felhasználó alkalmazás igényei szerint alakítható ki. Az XML-dokumentum ráadásul humán erőforrás részéről is "olvasható", szöveges formátumú, alapértelmezésben nem bináris, hanem karakter alapú szöveg formátum. Noha a nyelvtanát nem ismerők számára nem könnyen olvasható, de a szabályrendszerre elsajátítható. (A weboldalakot leíró HTML [HyperText Markup Language] szintén a szöveges formátumú leírónyelvek (tag nyelvek) közé tartozik, de mivel rögzített a struktúrája, ezért nem tekinthető *metanyelvnek*, azaz nem fejleszthető ki egy adott területhez illeszkedő *ad-hoc* nyelvtan.)



14. ábra Web szolgáltatások szoftver architektúrája – szolgáltatás platform

Elérési szolgáltatások:

- azokat a technikai szolgáltatásokat soroljuk ebbe a kategóriába, amelyek segítségével biztosíthatjuk a meglévő alkalmazáscsomagokban (ERP, CRM, SCM stb.) és egyedi fejlesztésekben található funkciók *szolgáltatásként* történő publikációját. Ide tartoznak az olyan módszerek mint például a becsomagolás („wrapping”), a közvetítés („mediation”) stb.

Informatikai szolgáltatás menedzsment:

- Gondoskodik a Web szolgáltatásokról, alkalmazásokról és erőforrásokról, valamint nyomon követi, kezeli és felügyeli ezeket az elemeket.

A SzOA alapú működés tekintetében elengedhetetlen, hogy mérni tudjuk a szolgáltatások különböző sajátosságait, beleértve az adatforgalmat, a szolgáltatás felhasználásának mértékét, a nem funkcionális követelmények teljesülését, a SzOA infrastruktúra komponenseinek a rendelkezésre állását stb. E folyamatos mérések segítségével a hibák gyorsan kivédhetőek, hatékony üzemeltetés valósítható meg, mi több az szervezeti (vállalati, üzleti) folyamatok optimalizálhatók.

Szervezet nyomon követése, vezetői műszerfal (dashboard):

3.9 Szolgáltatási infrastruktúra

- SOA architektúra a szervezeti (vállalati, üzleti) vezetés számára is biztosít ü monitorozási és beavatkozási felületeket, ahol a szervezeti teljesítménymutatók figyelemmel kísérhetők, és lehetőség van a beavatkozásra. Ez a funkció alapkövetelmény ezen a területen, mivel a SzOA megközelítés egyik alapfeltétele az átlátható, rugalmas és mérhető szervezeti (vállalati, üzleti) folyamatok biztosítása

Közmű szolgáltatások a Web szolgáltatások számára: A gyártói, de a nyílt forráskódú SzOA megoldások is, sok beépített alap, kiegészítő, közmű szolgáltatást („utility”) tartalmaznak. Általában ide tartoznak a naplózási, jogosultság kezelési, egyéb biztonsági, szolgáltatások, tevékenységek nyomon követés, monitorozása stb.

Előtt Front-end) rendszerek, a szolgáltatások felhasználói: pár példa a legfontosabb alkalmazásokra, amelyek a Web szolgáltatásokat használhatják.

- A *folyamatmenedzsment, folyamatszervezés (Business Process Management)* és a *folyamat modellezés (Business Process Modelling)* segítségével szolgáltatásokból szervezeti (vállalati, üzleti) folyamatokat lehet felépíteni akár úgy is, hogy nincs szükség informatikai szakemberek bevonására, hanem szervezeti (vállalati, üzleti) elemzők, rendszer-szervezők, tanácsadók az informatikai szakemberek és részleg bevonása nélkül is meg tudják oldani. Így ezek a folyamatok rugalmasan, gyorsabban és kisebb költséggel módosíthatók, mivel a központi folyamatvezérlés megszünteti a fejlesztés és működtetés redundanciáját.
- Az alkalmazottaknak és az ügyfeleknek a felhasználói felületet egy egységes vállalati Intranet / Intranet portál biztosítja, annak ellenére, hogy egységes személyre és szerepkörre is szabható valamint több csatornán is elérhető.
- Az üzleti partnereknek és ügyfeleknek a szolgáltatások biztonságosan elérhetővé tehető a B2B / B2C megoldásokkal. A felhasználói felületek gyorsabb kialakítását és egységesebbé tételét további felhasználói felület megjelenítését, tervezését és az munkatársak közti együttműködést segítő komponensekkel támogatják.
- **BPEL (Business Process Execution Language)** XML nyílt szabványon alapuló, a folyamatok modellezésének végrehajtó nyelve.

Amikor egy szervezet (vállalat) honlapján más cégek szolgáltatásait kívánja felhasználni, akkor a BPEL nyelv segítségével integrálhatja a szolgáltatásait a másik vállalat szolgáltatásaival.

3.10 SzOA irányítása (governance)

A SzOA a szervezeti (vállalati, üzleti) architektúra része, vagy esetleg annak egésze. Ebben az értelemben beleilleszkedik a szervezeti architektúra irányításának kérdéseibe a SzOA irányítása (Ld. 2.6)

A SzOA előnyei csak akkor valósulnak meg, ha megfelelő az irányítás. Teljesen új megközelítést igényel a SOA infrastruktúra, a szolgáltatások és az ezekre épülő üzleti alkalmazások megtervezése, megvalósítása és természetesen a működtetése is. A SzOA irányítás alatt az informatikai irányítás („IT governance”) kiterjesztését értjük a szolgáltatások teljes életciklusára.

A SzOA irányítása fogalma magában foglalja a tervezést, a megvalósítást, a tesztelést és az üzemeltetést is. A SzOA irányítás legfőbb feladata, hogy a szervezet gondolkodásmódja teljes egészében áttérjen a sziget szemléletről a globális, a teljes infrastruktúra átfogó szemléletre, szolgáltatások és alkalmazások egységes portfólióban való felfogására. Ebből adódóan a cél nem a különálló egységek, hanem a szolgáltatások és a szolgáltatásokból épülő alkalmazások fejlesztése és üzemeltetése. Az szervezeti (vállalati, üzleti) folyamatok esetében is szükséges a hasonló gondolkodás, hogy könnyebb legyen az együttműködés a két fél számára, a szervezet és informatika, a szervezéssel és az informatikával foglalkozó szakemberek között.

3.11 Az érett SzOA

A szolgáltatásorientált architektúrák és a Web szolgáltatások az utolsó lépést jelentik az *alkalmazás integráció köztesszoftver (middleware)* fejlesztése során. A hordozhatósági problémákat hivatottak feloldani, amelyek már régebben felmerültek, valamint alapjául szolgálnak a jövőben az Internetet átölelő elosztott alkalmazásoknak. Ezen kívül megkísérelnek (kisebbségi sikereket elérve) véget vetni a „köztes réteg háborúknak” és célul tűzték ki azt, hogy az összes nagyobb gyártó végül megegyezzen a technológiai szabványok egy egységes, lehetőleg egyetlen jól használható gyűjteményében, amely alkalmas lesz az *alkalmazás-integráció* és elosztott számítástechnika problémáinak orvoslására.

Az alkalmazás integráció *köztesszoftver* (ld. 9.2) számos célra felhasználható. Kezdve a helyi, lokális szoftver/hardver, logikai/fizikai komponensek összekapcsolásától az egyszerű asztali vagy Web szerver alkalmazások létrehozásáig, vagy egy olyan infrastruktúra kialakításáig mely az egész Interneten átívelhet. A tradicionális technológiák ebben a közegben, mint például a *JEE* alkalmazáserverek és üzenetátvitel tökéletes megoldást jelenthetnek egy

adott szervezeten belül. Ezek a megoldások azonban meglehetően távol esnek attól, amire szükség van, ha egymástól független szervezetek szervezeti (vállalati, üzleti) folyamatait szeretnénk összekapcsolni, globálisan, az Interneten keresztül. A Web szolgáltatások és a szolgáltatásorientált architektúrákat pontosan arra tervezték, hogy az ilyen szükségleteket kielégítsék.

Sok szempontból a szolgáltatásorientált megközelítés és a Web szolgáltatások nem jelennek semmi újat. A korábbi elosztott technológiákhoz és architektúrákhoz hasonlóan, a fő céljuk az, hogy lehetővé tegyék az alkalmazások számára azt, hogy olyan funkcionalitásokhoz férjenek hozzá, melyeket más alkalmazások szolgáltatnak. Ezt hasonlóképp teszik, mint a *JEE köztesszoftver*, ami lehetőséget ad a Java kliensek számára, hogy olyan metódusokat hívjanak meg, melyeket JEE komponensek szolgáltatnak.

A legnagyobb különbség az, hogy ez a technológia a *szolgáltatás alapú modellt* helyezi előtérbe, valamint az azt támogató egyéb információtechnológiákatazért, hogy a hordozhatóságot és a különböző platformok és programozási nyelvek miatt fellépő problémákat feloldja. Habár lehetőség van egy „*szolgáltatásorientált rendszer*” megtervezésére és megvalósítására bármilyen elosztott technológiával, illetve integráció *köztesszoftverrel*, azonban csupán a *Web szolgáltatás technológiák* rendelkeznek azokkal a kulcsfontosságú tulajdonságokkal, amelyek manapság szükségesek a hordozhatóság zökkenőmentes megvalósításához. Ez pedig a legsarkalatosabb pontja a szolgáltatásorientált nézeteknek.

A hordozhatóság hangsúlyozása annak eredményeképp vált ilyen fontossá, hogy elfogadtuk azt a sokféleséget, amely napjainkban a vállalatokat (szervezeteket) jellemzi és rájöttünk arra, hogy ez a sokféleség vélhetőleg nem fog megváltozni a jövőben. Napjainkban majdnem minden szervezet támogatja a vegyes platformokat, programozási nyelveket és szoftver csomagokat (beleértve az üzleti szempontból kritikus régebbi alkalmazásokat). Bármely integráció *köztesszoftver*, amely feltételezi annak a szükségességét, hogy az alkalmazásokat jelentős mértékben át kell írni vagy a már meglévő és működő alkalmazásokat hordozni kell migrálni kell újabb platformokra, meg fog bukni az első olyan akadálnál, ahol a költségek túl nagyok vagy a kockázat túl magas lesz. Az igazság az, hogy a nagyméretű vállalati alkalmazások egyre nagyobb mértékben tevődnek össze olyan alkalmazásokból, csomagokból, illetve komponensekből, melyeket soha nem terveztek úgy, hogy képesek legyenek együttműködni vagy egyáltalán inkompatibilis platformokon üzemelni. Ez a tény egyre inkább a hordozható-

ság szükségességét hangsúlyozza, amely egyre fontosabbá válik. Ily módon a vállalatok elkezdtek kiépíteni olyan új generációs, széles körben integrált alkalmazásokat, melyek közvetlen együttműködésre képesek üzleti partnereikkel és bizonyos szolgáltatókkal.

A Web szolgáltatások és a szolgáltatás orientált architektúrák az informatikai ipar válasza a felmerült igényekre az *integráció* és *hordozhatóság* terén.

SzOA OASIS (the Organization for the Advancement of Structured Information Standards):

SzOA egy olyan informatikai paradigma, amelyik megszervezi és kiaknázza az elosztott informatikai szolgáltatási képességeket, és ezek a szolgáltatási képességek különböző felelősök felügyelete alá tartozhatnak. SzOA-ban a szolgáltatások azok a mechanizmusok, amelyek a szükségletek és képességek egymásra találását elősegítik.

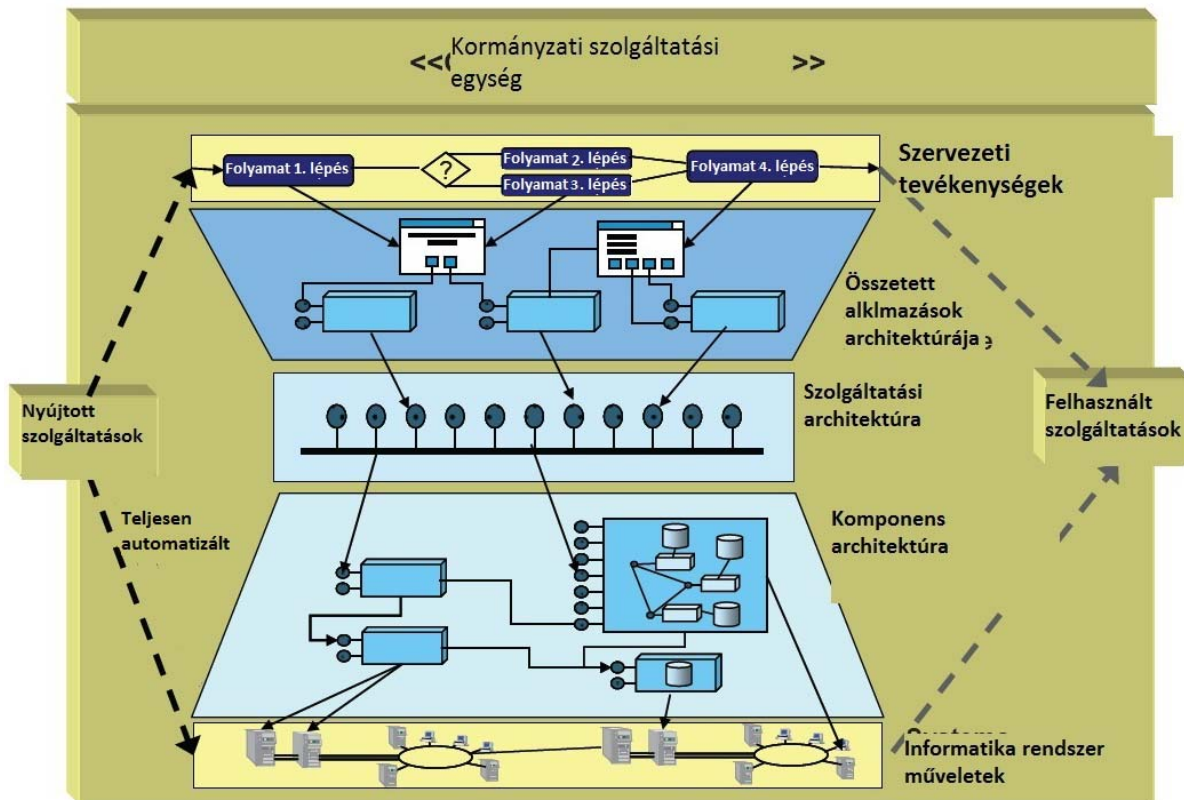


15. ábra Az érett/fejlett SzOA modellje- nem minden rétegre van szükség az egyes konkrét alkalmazásokhoz.

A SzOA referencia architektúra legfelső szintje a *szervezeti tevékenységek* és a legalsó szintje az *informatika rendszer műveletek* alkotják azt a két réteget, amelyek a szervezet működését, működtetését jelenítik meg. A többi réteg azt a tükrözi vissza, hogy szolgáltatás központú megközelítésben hogyan szervezik (architektúrát építenek és megtervezik) meg a képességeket, amelyek a szervezet tevékenységeit és folyamatait támogatják. Az összetett alkalmazási architektúra rakja össze és vezényli a szolgáltatásokat oly módon, hogy a *szervezeti tevékenységek* igényeit kielégítsék, a *szolgáltatási architektúra* pedig ajánlja fel azokat hasz-

3.12 Az SzOA irányítás fő alkotóelemei

nálható szolgáltatásokat, amelyekből az összetett szolgáltatások építkezése megtörténhet. A *komponens architektúra* olyan erőforrásokat használ fel, amelyek lehetővé teszik a szervezeti szolgáltatások megvalósítását.



16. ábra Egy e-kormányzati SzOA referenci architektúra (Hivatkozási alap)

3.12 Az SzOA irányítás fő alkotóelemei

A **SzOA irányítási modell** a szerepköröket, a döntési jogokat, a kapcsolódó mérési és ellenőrzési mechanizmusokat és folyamatokat határozza meg. Mindezek az IT-döntések előkészítéséhez, meghozatalához és végrehajtásához szükségesek. A döntési és végrehajtási mechanizmusok szabályzatokon és irányelveken („policy”) alapulnak, melyek meghatározzák például az alkalmazások architekturális szabályait vagy a csomagalkalmazások kiválasztási szempontjait. Az irányítási modellnek lényeges eleme a projektfinanszírozási szabályzat, ezenkívül a folyamatfelelősök és a „szponzorok” (a szervezeten belül a projekt költségeit vállalók, finanszírozók) kijelölése is. Az irányítási modellben célszerű egy SzOA egy kompetencia-központ létrehozni, feladatait, hatás és felelősségi körét kijelölni.

SzOA referencia-architektúra: amennyiben a vállalat a SzOA teljes kiépítését igényli, a referencia-architektúra tartalmazza a teljes szervezet SzOA architektúra tervét. A referencia-modell határozza meg az alkotóelemeket, azon belül is a modulok kapcsolódását és elhatárolását. Mindez a vállalat számára elengedhetetlenül szükséges, mivel ez teszi lehetővé a SzOA infrastruktúra lépcsőzetes, inkrementális fejlesztését az átfogó szervezeti architektúra koncepciójának megfelelően. Az szolgáltatási sínen keresztül megy végbe a szolgáltatások összekapcsolódása, illetve egymástól történő elszigetelése, elhatárolása. A SzOA paradigma alapú megközelítés esetén is a nyílt szabványok alkalmazása a célszerű, mivel a referencia-architektúra kialakítása hosszú távra szól. Erre annak ellenére figyelni kell, hogy „elvileg” a SzOA maga nyílt szabvány, azonban az egyes gyártói megvalósítások sajátos, egyedi megoldásokat tartalmaznak, amelyek miatt a különböző gyártóktól származó SzOA elemek összekapcsolása, csak fejlesztési erőfeszítések árán valósítható meg. Nyílt szabványokhoz ragaszkodás révén a beruházás értékálló marad, és a későbbi módosítások sem jelentenek gondot, esetleges többlet költséget. A SzOA architektúra alkalmazásával tulajdonképpen komponens alapú fejlesztési megközelítést követünk, amely lényegében lehetővé teszi, hogy a folyamatvezérlési, biztonsági elemeket, valamint az integrációs és kommunikációs funkciókat ne kódoljuk be az egyes Web szolgáltatásokba, hiszen azokat egységes, újrafelhasználható módon a SzOA infrastruktúra biztosítja.

SzOA kivitelezési ütemterv (Roadmap): a referencia-architektúra teljes kiépítéséhez szükséges lépéseket és azok várható ütemezését tartalmazó tervezet, mely során meghatározzák a SzOA infrastruktúra kialakítását.

Fő területei:

- alap infrastruktúra, például BPM bevezetése;
- szervezeti (vállalati, üzleti) funkció típusú szolgáltatások például egy megrendelés vagy ügyfélszolgálati folyamatok elemei;
- információelérési szolgáltatások például központi dokumentumkezelő szolgáltatásainak publikálása;
- közös informatikai szolgáltatások például felhasználó azonosítás.

A SzOA infrastruktúra bevezetése során a referencia-architektúra és az ütemterv folyamatos felülvizsgálata és napra készen tartása szükséges, természetesen az szervezeti célok megvalósításának szem előtt tartásával és az idő során felhalmozódott tapasztalatok alapján.

3.13 A szolgáltatás orientáltság alapelvei

Mérőszámok: szervezetenként változó a SzOA metrikák, mérőszámok és siker kritériumok felállítása és rendszeresítése. Egyedi, szervezetre szabott és nem minden esetben könnyű feladat. A SzOA bevezetésének sikerességét mutatja, ha a szervezet üzleti előnyre, informatikai előnyre tesz szert, továbbá a szolgáltatásokat és a kapcsolódó erőforrásokat képes lesz újrafelhasználni a működése során.

Az irányítási modell és az ütemterv teljessége és megfelelő minősége érdekében, még a bevezetés megkezdése előtt, a tervezéshez szükség van az aktuális informatikai környezet és szervezeti (vállalati, üzleti) folyamatok felmérésére, illetve az aktuális informatikai irányítási képességek és irányítási struktúra részletes felmérésére, és nagy figyelmet kell fordítani a szervezet stratégiai terveire és prioritásaira.

A SzOA bevezetésének két jelentős előnye az újrafelhasználhatóság és az **agilitás**. Abban az esetben beszélünk újrafelhasználhatóságról, ha a meglévő szolgáltatásokat újrahasznosítják, illetve az újonnan létrehozott szolgáltatásokat, alapadatokat ismételten felhasználják az elkövetkezendő projektekben. A fent említett esetek bármelyikére igaz, hogy jól mérhetőek. A következő mérőszámokat alkalmazhatják: hány projektben lett hasznosítva egy adott szolgáltatás, hány meglévő szolgáltatást sikerült a SzOA rendszerbe integrálni és felhasználni. Az újrafelhasználás viszont nem minden vállalat számára jelent versenyelőnyt a többivel szemben.

Az **agilitás** sokkal nehezebben mérhető fogalom. Több szempontot vehetünk figyelembe a mérőszámok meghatározásánál: A szervezeti vagy informatikai igények, követelmények megvalósításához szükséges idő lehet egy mérték. A SzOA rendszer stabilitásának valamilyen mértékét, amely a környezet változásából származó olyan szervezeti igények szintjét jelenti, amelyeket a SzOA, illetve az informatikai architektúra jelentős változtatása nélkül meg lehet valósítani. Az olyan szervezeti igények száma, amelyeket a SzOA rendszer képes viszonylag rövid időn belül kielégíteni, szintén lehet, egy mérőszám.

3.13 A szolgáltatás orientáltság alapelvei

A szolgáltatás orientáltságnak a fejlődés során kialakult alapelvei [355]:

- **Szolgáltatási szerződések:** A szolgáltatások között úgynevezett szerződések állnak fenn, amelyek meghatározzák, hogy hogyan működjön a szolgáltatás. Tulajdonképpen a szolgáltatás meta-adatainak leírása. Azokat a szabályokat és feltételeket adja meg egységes

formában, amelyeket a szóban forgó szolgáltatás igénybevételét kérelmezőnek ki kell elégítenie.

- **Laza csatolás:** A szolgáltatások kialakításánál a laza csatolásra kell törekedni programozási illetve szoftver fejlesztési értelemben, ami azt jelenti, hogy tudnak ugyan egymásról, de nincs közöttük függőség. A laza csatolás mértékének meghatározása a szerződésben kerül rögzítésre.
- **Újrafelhasználhatóság:** A szolgáltatások ismételt felhasználhatósága alapkövetelmény, függetlenül attól, hogy épp szükség van-e a szolgáltatásra. A szolgáltatások újrafelhasználásával kiiktathatók a redundáns szolgáltatások, ezáltal csökkentve az informatikai költségeket.
- **Absztrakció:** Az absztrakció a dolgok, leegyszerűsítését jelenti. Csak a lényegre történő koncentráció. Az absztrakció segít gondolataink rendezésében, abban, hogy ne vesszünk el a dolgok részletezésében. A szolgáltatásokra is jellemző az absztrakció, ún. fekete dobozoknak tekinthetők. Csak annyi információ érhető el róluk, amennyi szerepel a szolgáltatási szerződésben. A szolgáltatás maga bonyolult adatfeldolgozási logikát valósíthat meg, de a szerződésben csak egy általános, leíró jellegű információk jelennek meg.
- **Autonómia:** a szolgáltatás teljes autonómiát élvez a saját erőforrásai és az algoritmusai fölött a használat ideje alatt. Önmagát irányítja, ezáltal kiküszöbölhető, az a probléma, hogy más szolgáltatások gátolják a telepítését és továbbfejlesztését.
- **Összeépíthetőség:** a szolgáltatások más szolgáltatásokból felépíthetők. A **szolgáltatásplatformok** egy olyan egységet alkotnak, amelyek lebonthatók részsolgáltatásokra, de ennek ellenére megtartják a laza csatolásukat a *szolgáltatásplatformon* belül, illetve az egyes rész vagy elemibb szolgáltatásokból jelentősebb összetett szolgáltatási egységek rakhatók össze dinamikusan.
- **Állapotnélküliség:** a szolgáltatásoknak a lehető legkevesebb állapotinformációt kell kezelniük, és le kell rövidíteni azt az időtartamot, amíg ezt az állapotot tárolhatják. Az állapotok hosszú ideig történő tárolása megakadályozza, hogy a szolgáltatások más tranzakciókat hajtsanak végre.
- **Felfedezhetőség (megkereshetőség):** a szolgáltatásokat úgy kell megtervezni, hogy azok megtalálhatóak legyenek különböző keresési eljárásokkal. Ha egy szolgáltatás könnyedén megtalálható, akkor elkerülhetők az újrafelhasználásával kapcsolatos idővesztések.

3.14 Szolgáltatásorientált rendszerek

A szolgáltatásorientált rendszerek irányába történő elmozdulás hátterében az áll, hogy szükségessé vált az *alkalmazások és az szervezeti (vállalati, üzleti) rendszerek* (amelyeket támogatnak) egyidejű integrációja. A legtöbb jelenlegi integrációs technológia zárt vagy szabaddalmazott és kizárólag azokat az alkalmazásokat képes integrálni, melyek azonos technológián alapulnak. Ez szállító, gyártó függőséget jelent, persze csak abban az esetben, ha a vállalatok nem hajlandóak viselni a költségeit egy komplex adapter réteg előállításának, amely képes különböző technológiák és platformok összehangolására. Ezek a megszorítások elfogadhatóak lennének egyetlen vállalat esetében, de a valóságban még így is elég csekély lenne a valószínűsége a számítógépes rendszerek ilyen nagyfokú *homogenitásának*.

Azóta szükség van a vállalati rendszerek integrációjára, amióta egyáltalán léteznek vállalati rendszerek. Ezt az integrációt régen a hagyományos papír alapú dokumentumok nyújtották árajánlatok, számlák és megrendelések formájában. Ezek a hagyományos dokumentumok a mai napig használatban vannak, de majdnem minden esetben valamilyen informatikai rendszer állítja őket elő. A feladat, hogy ezeket az szervezeti (vállalati, üzleti) rendszereket integráljuk még ma is élő feladat; és gyakran még most is papír alapú dokumentumok postázásával vagy faxolásával valósul meg.

Költségcsökkenés és hatékonyságnövekedés jellemezte évekig azokat az üzleti informatikai rendszereket, melyeket közvetlenül összekapcsoltak ezzel megszabadulva a felesleges papírmunkától. Viszont ennek megvalósítása sok esetben majdnem ugyanennyi időt emésztett fel. Az **EDI (Electronic Data Interchange)** volt ez egyik legjelentősebb kezdeti próbálkozás, amely azt célozta meg, hogy felismerjük a potenciális előnyöket. Sok szempontból megelőzte a korát, viszont a legnagyobb vállalatokat leszámítva kevesen engedhették meg maguknak az EDI hálózatok használatát annak zárt természete, valamint a költséges szoftvere miatt.

Az Internet és a Web szolgáltatások megjelenésével sok minden gyökeresen megváltozott. Az Internet feltehetőleg összekapcsolja valamennyi számítógépet egy globális hálózaton, így lehetőséget biztosítva a vállalatok számára, hogy elektronikusan továbbítsák dokumentumaikat a partnereik és ügyfeleik számára az egész világon. Mindezt meglehetősen gyorsan és alacsony költségek árán. A Web szolgáltatások a probléma másik oldalát célozzák meg az által, hogy az integrációs sztenderdek egy olyan gyűjteményét biztosítják, melyeket a legtöbb nagy gyártó implementált. Ezen kívül a szerver platformok szerves részét képezik. Mindezek eredményeképp elmondható, hogy az üzleti szintű integráció hama-

rosan viszonylag könnyű, olcsó és hétköznapi feladattá válik.

A *Web szolgáltatások* csupán egy újabb *alkalmazás integrációs technológia*, szemléletében kicsit más, mint a CORBA, JEE, DCOM vagy ezek bármely versenytársa. Ezek a technológiák meglehetősen hasonlóak: a kliens alkalmazások felderítik a szerveret, meghatározzák, hogy azok milyen szolgáltatásokat ajánlanak ki és meghívják e funkciók egyikét. Ami ezektől különbözik a *szolgáltatás orientált architektúrák* és az ezeket támogató *Web szolgáltatások* tekintetében az az, hogy ezek az alkalmazások és szerverek most már hozzáférhetőek lesznek más vállalatok és egyének számára a nyilvános Interneten. Ennek a nézőpontbeli változásnak az eredményeképpen létrejöttek szabványok, architekturális irányelvek, amelyek a hordozhatóságot állítják középpontba az által, hogy a lehető legkevesebb feltételezéssel élnek a szolgáltatók és a végfelhasználók rendszerei belső működését illetően.

3.14.1 *Szolgáltatásorientált rendszerek*

A rendszertervező bármely rendszerről is legyen szó, komoly kihívásokkal nézne szembe, ha biztosítani szeretné a rendszer *robosztusságát* és *hordozhatóságát*. Kifejezetten ezeket a sarkalatos pontokat célozzák meg a szolgáltatás orientált architektúrák és a Web szolgáltatás technológiák.

Az irányelvek, amelyek a szolgáltatás orientált architektúrák alapjait képzik, egyáltalán nem jelentenek nagy újdonságot a rendszertervezés szempontjából. Sokkal inkább hosszú évek tervezési tapasztalatai tükrözik, amelyek az olyan nagy rendszereket jellemezték, amik végül működőképesnek és jól karbantarthatónak bizonyultak. Ezeket az irányelveket általában a következő 4 pontnak szokták megfeleltetni:

1. Egyértelmű határok;
2. Autonóm szolgáltatások;
3. Sémák, szerződések megosztása (nem a megvalósításoké);
4. Irányelveken alapuló szolgáltatás kompatibilitás.

Lássuk, ezek mit is jelentenek.

Az első pont annak a ténynek a felismerése, hogy a szolgáltatások független alkalmazásokat jelentenek, nem pedig olyan egy olyan programkódot, ami gyakorlatilag bármilyen plusz költség nélkül meghívható. Egy szolgáltatáshoz való hozzáférés legalább egy olyan határátlé-

3.14 Szolgáltatásorientált rendszerek

pését jelenti, amely elválasztja egymástól a különböző folyamatokat, valamint feltehetőleg több hálózaton átívelhet és tartományok közötti hitelesítéssel (autentikációval) jár. Minden egyes logikai, informatikai határ (folyamat, gép, bizalmi tartomány (process, machine, trust)) amelyet át kell lépni, csökkenti a teljesítményt, növeli a komplexitást és annak az esélyét, hogy valamilyen hiba fog fellépni.

A fejlesztők és a szolgáltatók elképzelhető, hogy földrajzilag is távol kerülnek egymástól, így más határokat is át kell lépni, ami tovább növeli a fejlesztési időt, valamint csökkenti a robusztusságot. A megoldás erre a problémára az, ha az egyszerűséget tartjuk szem előtt, mind a szolgáltatások specifikációjánál, mind pedig a támogatást jelentő Web szolgáltatások esetében. A jó szolgáltatásoknak egyszerű kapcsolófelületei (interfészei) vannak és a lehető legkevesebb feltételezéssel élnek és minimálisan szükséges, absztrahált sajátosságokat osztják meg klienseikkel, a szolgáltatás felhasználóival. Ez könnyen érthetővé teszi őket és nagyobb sikerrel kerülnek felhasználásra a távoli fejlesztők által.

3.14.2 Autonóm szolgáltatások

A szolgáltatások független, *autonóm* alkalmazások, melyek nem osztályok vagy komponensek, amelyek szorosan kapcsolódnak a kliens alkalmazáshoz. A szolgáltatások célja az, hogy egy hálózaton keresztül elérhető legyenek, nagy valószínűség szerint az interneten, ahol könnyen integrálhatóvá válnak bármely olyan alkalmazáshoz, amely számára hasznos lehet a szolgáltatása. A szolgáltatásnak semmilyen információval nem kell rendelkezni a kliens alkalmazásokról, viszont feltehetőleg ki kell szolgálnia bármilyen beérkező kérést, amely megfelelően van formázva és megfelel bizonyos biztonsági követelményeknek.

A szolgáltatások kihelyezésre kerülhetnek és teljes mértékben menedzselhetőek, valamint a szolgáltatások tulajdonosai bármikor változtathatnak a definíciókon és az implementáción. A verzió kompatibilitás már egy régóta fennálló probléma az elosztott rendszereknél, amely egyre súlyosabbá vált a szolgáltatások nyílt természete miatt. Hogyan fejleszthető egy szolgáltatás úgy, hogy valószínűleg nagy (pontosan ismeretlen) számú felhasználó függ tőle?

Például egy bank, ami egy olyan szerverkomponenst üzemeltet, amely csupán egy belső pénzkidó alkalmazás által kerül meghívásra, pontosan ismeri a kliensek pozícióját és személyazonosságát. Ebből kifolyólag egy esetleges szolgáltatás frissítés technikailag viszonylag könnyen kivitelezhető. Viszont egy hitelkártya feldolgozó komponens, amely az internetről beérkező engedélyezési kéréseket kezel, nem fogja tudni meghatározni a kliensek helyét,

elérhetőségét és nem fogja tudni rávenni a klienseket arra, hogy frissítsék a már meglévő hívó alkalmazásaikat, annak érdekében, hogy megfeleljenek az új szolgáltatásdefinícióknak.

A megoldás erre a problémára részben a szándékos *egyszerűségben* rejlik, valamint abban, hogy a szolgáltatásmodell kiterjeszhető. A kliensek csupán azt tudják a szolgáltatásokról, hogy milyen üzeneteket fogadnak és küldenek vissza. Ez az egyetlen egy függőség, amely egy kliens és egy szolgáltatás között fennáll. A szolgáltatások tulajdonosai igény szerint változtathatnak a megvalósításon mindaddig, amíg a jelenleg érvényben lévő üzeneteket elfogadják. Ezen felül tetszés szerint kibővíthetik a kérés/válasz üzeneteket, amennyiben ezek visszamenőleg kompatibilisek maradnak. A mi esetünkben a hitelkártya feldolgozó komponens implementációja akár teljes mértékben megváltozhat, például a CISC/COBOL rendszerünket C#/.NET-re cseréljük, hiszen ez a változtatás a kliensek számára mindaddig láthatatlan marad, amíg inkompatibilis változtatásokat nem vezetünk be a „fizetés engedélyezése” üzenetbe.

Mivel a szolgáltatások *autonómok*, ezért felelnek a saját *biztonságukért* és meg kell tudniuk védeni magukat a rosszindulatú kliensekkel szemben. Azok a rendszerek, melyek fizikailag egyetlen rendszeren vagy egy zárt hálózaton találhatóak figyelmen kívül hagyhatnak bizonyos biztonsági intézkedéseket és támaszkodhatnak a tűzfalak vagy biztonságos hálózati protokollok nyújtotta védelemre (SSL). Viszont azok a szolgáltatások, melyek az internetről is hozzáférhetőek sokkal komolyabb védelmet igényelnek.

3.14.3 *Sémák, szerződések megosztása*

A sok éves tapasztalat azt mutatja, hogy egy nagyméretű robosztus és megbízható integrált rendszerek kiépítése nehéz feladat. Ezeknek a rendszereknek a megalkotása olyan komponensekből, amelyek eltérő programozási modelleken alapulnak, és eltérő platformokon futnak még nehezebb feladat. A szolgáltatás orientált technológiák úgy próbálják meg feloldani ezt a problémát, hogy az *egyszerűséget* tartják szem előtt. A szolgáltatások ***nem távoli objektumok***, öröklődéssel, *metódusokkal* és komplex, futási idejű *viselkedéssel*, mint a CORBA esetében és nem ***komponensek***, melyek támogatják az eseményeket, tulajdonságokat vagy állapotörző metódushívásokat. A szolgáltatások csupán alkalmazások, melyek üzeneteket küldenek és fogadnak. A kliensek és a szolgáltatások nem osztanak meg semmit egymással, leszámítva az üzenetek definícióját és egész biztosan nem osztoznak semmilyen program kódon vagy komplex futtatókörnyezeten.

Minden, amit az alkalmazásnak tudnia kell a szolgáltatásról az a **szervződés** (*contract*): az üzenetek struktúrája (sémája), amelyeket befogad és visszaküld, valamint a sorrendiség, amennyiben az üzeneteket egy meghatározott sorrendben kell elküldeni. A kliens alkalmazás arra használja ezt a sémát, hogy kérés üzeneteket állítson össze és ezeket küldje a szolgáltatásnak. A szolgáltatás a séma alapján ellenőrzi és érvényesíti (*validálja*) a kienstől érkező üzeneteket.

3.14.4 Irányelveken alapuló szolgáltatás kompatibilitás

A klienseknek teljes mértékben *kompatibiliseknek* kell lenniük a szolgáltatásokkal, amelyeket használni szeretnének. A kompatibilitás nem csupán azt jelenti, hogy a kliensek betartják a meghatározott üzenetformátumot, hanem azt is, hogy más fontos követelményeknek is eleget tesznek. Egy ilyen követelmény lehet például, hogy az **üzenetek titkosítva** legyenek vagy nyomon követhetők legyenek, annak érdekében, hogy az átvitel során ne veszíthessük el őket. A szolgáltatás orientált modellben, ezeket a nem funkcionális követelményeket az irányelvek definiálják, ily módon nem csak egyszerűen a szolgáltatás dokumentációba kerülnek leírásra.

Például elképzelhető, hogy a hitelkártya feldolgozónk úgy dönt, hogy csak azoktól a kereskedőktől fogad el fizetési engedélyezési kéréseket, akik egy X.509-es *autentikációs zsetonnal* (*token*) azonosítják magukat. Ez a biztonsági feltétel könnyedén reprezentálható egy szabállyal a kiadott biztonsági irányelvekben, az jogosultságok megadását, az engedélyek kibocsátást végző szolgáltatás számára (*authorization*). A irányelvek számítógépes feldolgozásra alkalmas állítások, szabályok gyűjteménye, amelyek lehetővé teszik a szolgáltatások számára, hogy definiálják a *biztonságra* és *megbízhatóságra* vonatkozó követelményeket. Ezeket a irányelveket tartalmazhatja a **szolgáltatási szerződés**, lehetővé téve ezzel, hogy a szolgáltatások viselkedése és elvárásai teljes körűen legyenek specifikálva a szerződésben vagy különállóan tárolhatók egy irányelvek tárban, ahonnan dinamikusán, futási időben hozzáférhetőek.

A szerződés alapú irányelvekre tekinthetünk úgy, mint a szolgáltatás dokumentáció részére, de felhasználhatók a fejlesztőeszközökben is azért, hogy automatikusan a szerződésnek megfelelő kompatibilis kliens vagy szolgáltatás program kódot hozzanak létre. Például egy szerver oldali biztonsági irányelv használható arra, hogy olyan kódot generáljanak, amely ellenőrzi, hogy a beérkezett üzenet bizonyos részei megfelelően vannak e titkosítva, majd de-

kódolja az üzenetet, és így bocsátja a szolgáltatás rendelkezésére. Mindez véghezvihető anélkül, hogy a fejlesztő akár egyetlen kódsort is begépelne.

A irányelvek különválasztása a szerződéstől lehetőséget ad a kliens alkalmazásoknak arra, hogy dinamikusan alkalmazkodjanak egy adott szolgáltató által elvárt követelményekhez. Ez kifejezetten hasznos lehet akkor, ha szolgáltatások szabványosításra kerülnek és több konkurens szolgáltató is kiejánlja őket. Például egy online kereskedő két szállítót is alkalmazhat, akik megegyező szolgáltatásokat biztosítanak, melyek azonos üzenetsémával működnek, de eltérő hitelesítési és azonosítási szolgáltatással (*authentication*). A dinamikus irányelvek akkor alkalmazhatók, ha a fejlesztő egy olyan alkalmazást szeretne létrehozni, amely támogatja mindkét hitelesítési és azonosítási módszert és képes dinamikusan kiválasztani a szolgáltatásnak megfelelőt.

3.15 Web szolgáltatások modellezése

Web szolgáltatások alapvető fogalmát a 3.1 fejezet tárgyalja.

A Web szolgáltatás egy programozható modul mely szabványos interfész leírással rendelkezik. Ez a szabványos kommunikációs protokoll egyetemleges elérést biztosít a szolgáltatáshoz. Így különböző programozási nyelven íródott, és különböző platformon futó szolgáltatásokból tudják felépíteni a terület specifikus alkalmazásokat.

Párhuzam húzható a **Web szolgáltatás**, és az **objektum orientált programozás** között. Egy objektum is elrejti a megoldás részleteit, és csak egy felületen keresztül érhető el. A Web szolgáltatás is magában foglalja a funkciókat, és csak az csatlakozási felületen (interfész) keresztül érhető el. A különbség a kettő között az, hogy a Web szolgáltatás **üzleti logikát hordoz, kereshető, és elérhető** az internetes környezetben. A szolgáltatás szabványos nyelven érhető el. A Web szolgáltatásnak három felhasználási területe van:

1. Elsősorban arra használják, hogy szervezetek *elérhetővé* teszik *szolgáltatásaikat*, üzleti alkalmazásaikat. Ezzel segítve az üzletvitelüket, és kiterjesztik a vállalat határait, kialakítják ökoszisztémüket.
2. Másik felhasználása olyan *elosztott rendszerek* építése, amelynek részei különböző programozási nyelveken íródtak, és különböző platformokon futnak, és az interneten

3.16 Web szolgáltatás modellezése

keresztül kommunikálnak egymással. Általában az elosztott erőforrások megosztást használják, megosztásával dolgoznak.

3. A harmadik terület, gyors szoftver építés. Itt a meglévő szolgáltatások felhasználásával építenek programokat.

3.16 Web szolgáltatás modellezése

Ahhoz, hogy webes szolgáltatást nyújthassunk, meg kell határozni, egy szabványos hozzáférési felületet, hogy az érdeklődő felhasználók tudják használni azt. Erre szolgál a Web Services Description Language (WSDL) és a Simple Object Access Protocol (SOAP). Ezek váltak szabvánnyá a szolgáltatás leírása, és hozzáférése terén. Mindegyik a World Wide Web Consortium (W3C) szabványa. Mivel ezek a technológiák jelenleg még fejlődnek, ezért ez a dokumentum a 1.1-es szabványt mutatja be a továbbiakban. A webes szolgáltatásokat nem kell külön fejleszteni. Bármilyen programból készíthető szolgáltatás, ha be van csomagolva web szolgáltatás felülettel (WSDL), és elérhetővé tesszük egy adatbázisban. Szolgáltatás kéréssel le lehet kérdezni a szolgáltatások listáját, és használni is lehet őket távoli process hívásokkal (RPC) amik a SOAP üzenetbe vannak ágyazva. Web szolgáltatás készítésénél célszerű először az eléréseket kifejleszteni, és csak utána a funkcionalitást.

3.16.1 Web szolgáltatás kommunikációs protokoll: SOAP

Miután meghatároztuk a WSDL segítségével, hogy *mit?*, *hol?*, és *hogyan?* tudunk igényelni. Ezek után már csak a szabványos kommunikációs protokollal meg kell szólítani a szolgáltatást. Ennek leírására szolgál a Simple Object Access Protocol (SOAP), ez írja le a webes szolgáltatások szabványos kommunikációját. (ld. 3.9.3)

3.16.2 WSDL és SOAP közti kapcsolat

Amellett, hogy tudjuk azt, hogy milyen szolgáltatásokat vehetünk igénybe azt is tudni kell, hogy hol igényelhetjük azokat. Azt kell megadni, hogy milyen legyen az üzenet formátuma, és stílusa. A kapcsolat felvételéhez szükséges adatokat a *binding tag*-ben kell meghatározni (leképezés, lekötés). Ennek két attribútuma van a *name* ami a leképezés nevét határozza meg, míg a *type* a portot a kapcsolathoz. Ezt követi a *body* ami további két további kötelező részt tartalmaz. A *soap:binding* írja le a kapcsolódás módját. Két attribútuma van *style-nak*, amely azt határozza meg, hogy *RPC*, vagy *dokumentum orientált kapcsolódás* legyen. A *transport* pedig, hogy milyen protokoll használható a szolgáltatások kérésére. Például stílus *RPC*, a protokoll *HTTP*. A másik kötelező rész a *soap:operation*, amely a művelet részleteit ad-

ja meg. Minden művelethez kötelező megadni egy *soap:operation* tevékenységet. Minden bemeneti, és kimeneti adathoz meg kell határozni, egy kódolási módot is. Példáulez *literal* kódolást.

3.16.3 Web szolgáltatás publikálása

Ha készen vagyunk egy web szolgáltatással, akkor közzé kell tenni azt mások számára. Ahhoz, hogy ezt mások is elérjék regisztrálni kell szolgáltatás adatbázisba. Az adatbázis nem tartalmazza a végrehajtható szolgáltatást, csak annak leírását, *WSDL* fájlt, *URL*-t. A tényleges program, és *WSDL* fájl a szolgáltatónál található, és ő tartja napra készen. **Universal Description, Discovery, and Integration (UDDI)** (Id.3.9.4) szolgál a Web szolgáltatások nyilvántartására, és ezek közötti keresésre. Az UDDI koncepció használja a sárga lapok (Yellow Page) fogalmát, és címtár megközelítést, könyvtárat.

3.16.4 Stateful Web szolgáltatás modellezés

A Web szolgáltatások kialakítását állapot megőrző (*Stateful*) vagy állapottartás, (állapot megőrzés) nélküli (*Stateless*) stílusban is meg lehet fogalmazni.

3.16.4.1 Stateless kontra Stateful Web Szolgáltatás

A *WSDL stateless* Web szolgáltatások leírására alkalmas. Ez azt jeleneti, hogy nem jegyzi meg az állapotokat, így csak olvasható szolgáltatások megvalósítására alkalmas. De sajnos a felhasználók jobb kiszolgálása érdekében bizonyos adatokat meg kell jegyezni a szolgáltatásokban. A *stateless* szolgáltatások jobban paraméterezhetők, és hibátűrőbbek. De sokszor van szükség az adatok megjegyzésére. Ekkor *stateful* módszert kell használni. Mindig a feladatnak megfelelően kell megválasztani a megvalósítást. De az üzleti alkalmazásokban jellemzően szükség van a *stateful* technikára, bár ez munkaigényesebb, és nem olyan jól skálázható.

3.16.4.2 Stateful Web szolgáltatás modellezés WSRF

IBM, Computer Associates (CA), az Oracle és mások kezdeményezése a **Services Resource Framework (WSRF)**, az amely specifikálja egy rendszerhez az állapottartó web szolgáltatás elérését, és irányítását. A WSRF is *XML* alapú. Négy részből áll: **WS-ResourceProperties**, **WS-Resource-Lifetime**, **WS-BaseFaults**, és **WS-ServiceGroup**. Ezek teszik elérhetővé a belső állapotot. A *WSRF* támogatja az erőforrások dinamikus létrehozását, és ehhez értékek rendelé-

sét. Tehát a *WSRF WS-Resource* hozzáférhetővé teszi az állapotokat a szolgáltatás felületén, meghatározza a kapcsolódó mechanizmusok csoportosítását, kezelését.

3.16.4.3 Granularity Enablement állapot felügyelet

Egy másik lehetőség az állapot megjegyzésére a tagolás vezérlés. Ekkor több kérdésre kell válaszolnunk. Milyen tagoltság fogja az állapotokat leírni, és követni? Milyen kölcsönhatásoknak, és tevékenységeknek kell jelentkezniük? Mik kerülnek rögzítésre, és tárolásra?

Három megoldás lehetséges:

1. Előre meghatározott állapot irányítás, újrakonfigurálható állapot irányítás, és hibrid megoldás.
2. Előre meghatározottnál megváltoztathatatlan a szolgáltatás.
3. Konfigurálhatónál az adminisztrátor az igények szerint változtathatja a részletességet.

A részletezettség állításával lehet befolyásolni a szolgáltatás paramétereit. Kis részletezettség esetén gyors, és olcsó az üzemeltetés. Nagy részletezettség esetében viszont pontosabb eredményt, jobb paraméterezhetőséget lehet kapni. A tervezésnél mindig figyelembe kell venni a cég, és a szolgáltatás érdekeit is.

3.16.4.4 Web Szolgáltatás interoperabilitás

Egy web szolgáltatás úgy áll össze, hogy **WSDL+SOAP+UDDI**. A WSDL meghatározza a szolgáltatást, a SOAP kapcsolja az internethez, és a UDDI kereshetővé teszi mások számára. De a valóságban vannak különbségek a gyártók megoldásai között. **Web Services Interoperability (WS-I)** szervezet azért alakult, hogy megoldást találjanak a különbségek áthidalására.

Azzal foglalkoznak, hogy különböző platformokra, programozási nyelveken írt szolgáltatások együttműködésére megoldásokat dolgozzanak ki, és ezeket megpróbálják általánosan használttá tenni. A **Basic Profile 1.0** már megjelent mint az egyik kulcsfontosságú mérföldkő. Ennek alapján a szolgáltatásokról lehet **profil** (arculatot) készíteni, aminek segítségével másik szolgáltatással könnyebb az együttműködés. Vannak általános *profilok* is amiket a WS-I dolgozott ki, például **biztonsági profil**. A munkacsoport minta profilokat tesztelő eszközöket is készített a profilok gyártásához. Már fejlesztő eszközök is léteznek a különböző szoftver gyártókhoz.

3.17 Összetett web szolgáltatás modellezés

Általában egy üzleti szolgáltatás nem egy szolgáltatásból áll, hanem több különálló szolgáltatás dolgozik együtt. Például egy utazási iroda szolgáltatása minimum három részből áll: Repülőgép foglalás, hotel foglalás, autó foglalás.

Ezek együtt alkotnak egy rendszert. A **Business Process Execution Language for Web Services (BPEL4WS)** könnyebbé teszi az ilyen szolgáltatások fejlesztését.

3.17.1 BPEL alapfogalmak

- A **BPEL** egy kiterjesztett információcsere modell *Web szolgáltatásokhoz*. Lehetővé tesz a vállalati *tranzakciók kezelését*. A BPEL megkönnyíti az *automatizált munkafolyamat* integrációt (workflow) vállalaton belül, és a vállalatok között is. A lehetőséget ad a szervezeti (vállalati, üzleti) folyamatok közötti kölcsönhatások, és a folyamatban részt vevő partnerek leírására. Az kapcsolófelület (interfész) a kölcsönös információcserén, kölcsönhatáson alapul, valamint a partner kapcsolat a felület szintjén van beleágyazva. A BPEL *koordinálja* a szolgáltatásokat a közös üzleti cél megvalósítása felé.

A gazdag folyamat leíró nyelvten miatt képes a szolgáltatások pontos viselkedését számítógép számára érthető formában ábrázolni. A BPEL elválasztja a külső viselkedést, és a tényleges megvalósítást (az implementációt). Az szervezeti (vállalati, üzleti) folyamat két módon modellezhető: (1) futtatható módon, és (2) absztrakt módon. A végrehajtható modellek részt vesznek a tényleges együttműködésben, míg az absztrakt modell az üzenetek cseréjét határozza meg a belső viselkedés ismerete nélkül.

3.17.2 Szolgáltatás leírás létrehozása

Először meg kell határozni a felek közti kicserélendő adatokat. Az üzenetek meghatározásánál a BPEL támaszkodik WSDL-re. Nincsen kötelező információ meghatározva a WSDL-ben a BPEL tekintetében. Ezt majd a BPEL absztrakt része határozza meg azért azt, így több helyen is fel lehet majd használni.

3.17.3 Üzleti folyamatok létrehozása

Ha megvannak a szolgáltatások leírásai akkor az üzleti folyamatot kell kialakítani. A BPEL meghatározás a *névtérrel* kezdődik, amely lehetővé teszi, hogy elolvassa a WSDL-eket. A BPEL leírás három részből áll:

3.17 Összetett web szolgáltatás modellezés

1. A partner kapcsolat definícióból,
2. a változó definícióból,
3. és a folyamat definícióból.

A partner kapcsolatok csoportosítva vannak *partner linkekkel*. Minden partnerkapcsolatot jellemzi a tag linkje. A *myRole* meghatározza a folyamat szerepét a *serviceLinkType*-en keesztül, míg a *partnerRole* a partner szerepét határozza meg.

A változók meghatározása egyszerű *XML típusok*, vagy *XML elemek*. Természetesen figyelembe véve a szolgáltatás WSDL dokumentumait az üzenetek váltásához. Állapot megőrzésére is van lehetőség, és a változók története is megjegyezhető.

3.17.4 BPEL kulcselemei

Kilenc fontos eleme van a BPEL-nek: *partnerek*, *partner link típus*, *partner kapcsolatok*, *üzleti partnerek*, *végpont hivatkozások*, *tevékenység*, *adatkezelés*, *korreláció*, és *hatókör*.

3.17.4.1 Partner

Ez határozza meg a kapcsolatot az üzleti folyamat és a partner folyamat között. A folyamat tekinthető partner folyamatnak, ha a három közül valamelyik igaz rá: fogyasztó/végfelhasználó által nyújtott szolgáltatás, üzleti folyamat, amit a szolgáltató használ, vagy olyan szolgáltatás, ami aktivál, egy üzleti folyamatot.

3.17.4.2 Partner link típus

Ez határozza meg a szolgáltatások közti szerepeket. Minden szerepet pontosan meghatároz egy *portType* a *WSDL* dokumentumban.

3.17.5 Partner link

Ez a modell határozza meg azt, hogy a partner szolgáltatások hogyan működnek együtt az szervezeti (vállalati, üzleti) folyamatokkal. Minden partnerkapcsolatot jellemez a partner kapcsolat típus. Több kapcsolatnak lehet azonos típusa. Ez statikus formát határoz meg a partner kapcsolatok, és az üzleti folyamatok között. Két szerep adható meg *myRole* az üzleti folyamat szerepe, és *partnerRole* a partner szolgáltatás szerepe.

3.17.6 Üzleti partner

A modellben megadhatók *szervezeti (vállalati, üzleti) partnerek* is, akikkel több kapcsolat is létesíthető. A *BPEL* ezek tulajdonságainak a leírására is a partnerek részt használja az *XML* le-

írásban. Ezek a partner kapcsolatok részhalmazát jelentik. Egy partner csak egy kapcsolatban jelenhet meg.

3.17.7 Végpont hivatkozás

Ez egy dinamikus *kötés* (leképezés, *binding*) a port specifikus adatok, és a szolgáltatások között. Lehetővé teszi a BPEL, hogy működés közben egy adott szolgáltató legmegfelelőbb szolgáltatását válasszuk, ha több hasonló van. Minden partner szerepét a *partner linkben* egy végpont hivatkozás, vagy egy dinamikus tevékenység határozza meg.

3.17.8 Tevékenység

Két tevékenységet támogat: **alaptevékenységet**, és **strukturált tevékenységet**. Az *alaptevékenység* nem tartalmaz más tevékenységet, például ilyen lehet *üzenet fogadás, válasz, indítás*. A *strukturált tevékenység* egy előre meghatározott *tevékenység sorozat*. A strukturált tevékenységnek további *három alete* van: *sequence* szekvenciális végrehajtás, *flow* párhuzamos végrehajtás, *pick* nem determinisztikus.

3.17.9 Adatkezelés

Az üzleti folyamatok kölcsönhatásba lépnek egymással, üzenetet küldenek, fogadnak. Lehetőség van *állapot megtartásra* a BPEL-ben. Ennek három formája van: *állapot változó, kifejezés, és feladat*. Az üzleti folyamat állapotát az állapot változó tárolja. A kifejezés bontja ki, és ellenőrzi az adatokat. Feladatokkal lehet frissíteni, az adatokat. Ehhez a BPEL XML típusokat, és WSDL üzeneteket is biztosít.

3.17.10 Összefüggés (korreláció)

A korreláció képes kezelni tartósan fent maradó adatokat (napok, évek). Ezt arra használják, hogy visszatérő üzletfeleknek is megfeleljen a rendszer.

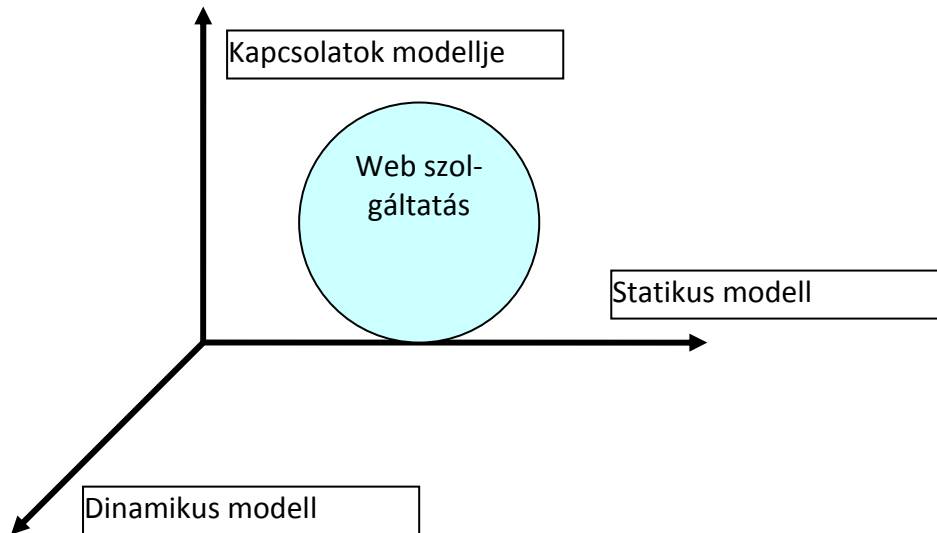
3.17.11 Hatókör

Tevékenységek összefüggését lehet vele jelezni. Ezt öt esetben szokás használni: *hiba kezelés, esemény kezelés, kompenzáció kezelés, adat változó, összefüggő halmazok*. Minden hatás kört meghatároz a *fő tevékenység*. Az elsődleges tevékenység lehet, egy alapvető tevékenység, vagy strukturált tevékenység, amiben még vannak beágyazott tevékenységek. A hatókör a beágyazottakra is vonatkozik.

3.17.12 3D-s Web szolgáltatás modellezés

WSDL és BPEL segítségével lehet összetett szolgáltatásokat modellezni. De mindkettő elsősorban csak statikus információkat ír le. A WSDL a szolgáltatás absztrakt felületét és felépítését írja le. A BPEL a kapcsolatokat írja le az üzleti folyamatban. Tehát mindegyik statikus modell.

De a web szolgáltatásnak vannak dinamikus paramétere is. Mint az az ábrán is látható.



17. ábra A Web szolgáltatások három dimenziós modellje (3 D)

Tehát a Web szolgáltatást három modellel lehet jól leírni. A **statikussal**, amely megadja a szolgáltatást. A **kapcsolatokkal**, amely megadja a szolgáltatások közötti kapcsolatot. Valamint a **dinamikus** modell, amely működés közben írja le a szolgáltatást. Ide tartoznak a minőségre vonatkozó értékek is (QoS, Quality of Service).

A háromdimenziós modell szerepe kettős. Segíti a szolgáltatás információinak tárolását. Valamint automatikusan elemezhetővé teszi azt. Nem csak az alapvető koncepciót, a szemantikus leírást tárolja, hanem a szolgáltatások kapcsolatait is. A szolgáltatás adatbázisának építésénél is segít ez a módszer.

3.18 Web szolgáltatások modellezésének értékelése

A Web szolgáltatások architektúra építőelemeit áttekintettük: hogyan néz ki a *WSDL* az hogyan kapcsolódik a *SOAP*-hoz, és hogyan kell szervezeti (vállalati, üzleti) *folyamatot* építeni szolgáltatásokból *BPEL* segítségével. Ezt követte a **3D modell** bemutatása. Eza modellezési keretrendszer a statikus modellek mellett a dinamikus, és kapcsolati modelleket is használja.

Bevezetésre került a *WSRF* fogalma is. Ez azt mutatta meg hogyan lehet új szolgáltatásokat írni meglévők segítségével.

A *WSDL* meghatározza a szolgáltatás *alap funkcióit*. Manapság a Webes szolgáltatások terjednek, így naponta sok új szolgáltatás kerül fel az internetre. A szakemberek azt próbálják megoldani, hogy olyan adatbázisokat üzemeltessenek, amik segítik a megfelelő kiválasztását. Próbálnak minőség i adatokat, és használati példát is a szolgáltatás mellet mutatni. De a jó adatbázis megalkotása még kutatások tárgyát képezi.

A webes szolgáltatások előnye, hogy nincsenek protokollhoz kötve. A fejlesztő határozza meg, hogy HTTP, IETF (*Internet Engineering Task Force*), BEEP (*Blocks Extensible Exchange Protocol*), vagy bármely protokollt használhat. Ahhoz hogy valódi B2B szolgáltatást készítsünk, sok további technológiát igénybe kell venni például *ebXML*-t azért, mert ez jobban kedvez az üzleti folyamatoknak, és használható többek között a SOAP-ban is. Nagyobb biztonságot nyújt.

3.19 A szemantikus háló/Web

A szemantikus háló/Web célja, hogy a hálózaton közzétett információkat meta adatokkal kiegészítve a hálón létező alkalmazások számára értelmezhető tegye, amelynek révén az alkalmazások közötti információcsere, kapcsolat felvétel, együttműködés automatizálható.

Alapvető célja, hogy lehetővé tegye az alkalmazások közötti minél magasabb szintű kapcsolatok létrejöttét és magas szintű információk feldolgozása megvalósítható legyen. A kellően magas szintű szemantikai együttműködés megvalósítására a *Web szolgáltatások* jó alapot biztosítanak.

A Szemantikus Web nyújtotta szolgáltatások alapelvei:

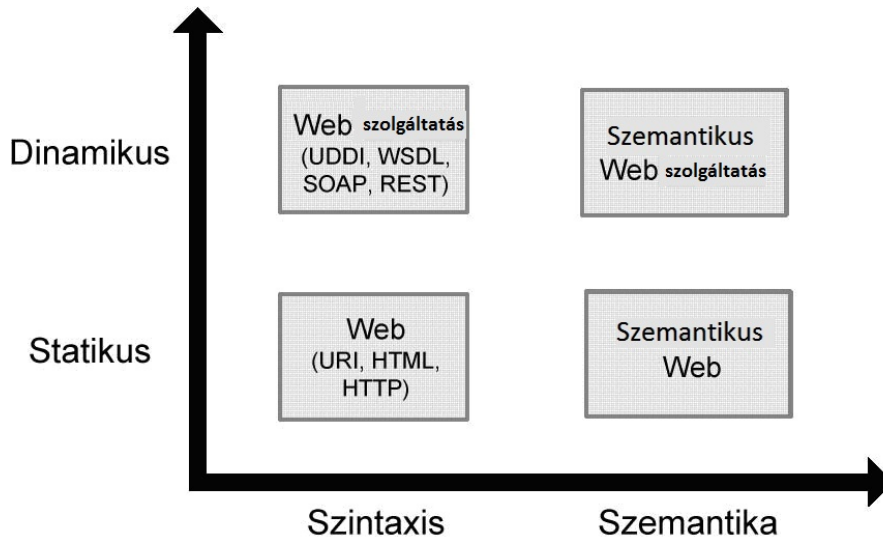
1. Metaadat reprezentáció formalizálása;
2. Az ismeretbázis (tudásbázis) reprezentáció folyamatos fejlesztése;
3. Logikai érvelő és következtető technikák, melyek segítségével kiaknázhatók a metaadatokban és az ismeretbázisban reprezentált tudás.

A fent említett szolgáltatásokhoz szükséges rugalmas metaadat és kapcsolat reprezentációt **ontológiák** valósítják meg. Lehetővé teszik a különböző metaadatok közötti fordítást, valamint a metaadat entitásaival kapcsolatos következtetést .

A szemantikus Web lényege az **ismeret reprezentáció**. A felhasznált **ontológia** hasonló szerepet tölt be, mint az *XSD* az adatstruktúra leírásában, azzal a különbséggel, hogy az onto-

lógia az adatok mellé még az adatok közötti összefüggések törvényszerűségeit is leírja. Az **RDF** (Resource Description Framework), valamint a kapcsolódó **OWL** (Web Ontology Language) szabványok a szemantikus Web egyik legfontosabb leíró nyelvivé váltak. Az **OWL** nyelv a **DAML** (Darpa Agent Markup Language) és az **OIL** (Ontology Inference Layer) nyelvekre épül.

Ugyan a szemantikus web mögött nincs lényegi technológiai újdonság, azonban a meglévő technológiák ilyen irányú felhasználásai sok, még kihasználatlan lehetőséget rejtnek.



18. ábra A Web szolgáltatások szintaktikai (formai) és tartalmi (szemantikai) viszonyrendszere

3.20 Metaadatok létrehozása és használata a Szemantikus Weben

Az XML egy strukturált, rugalmas módszert biztosít az adatok leírására, azonban segítségével az entitások közötti kapcsolatok szemantikája nem fejezhető ki a gép számára is feldolgozható módon.

```

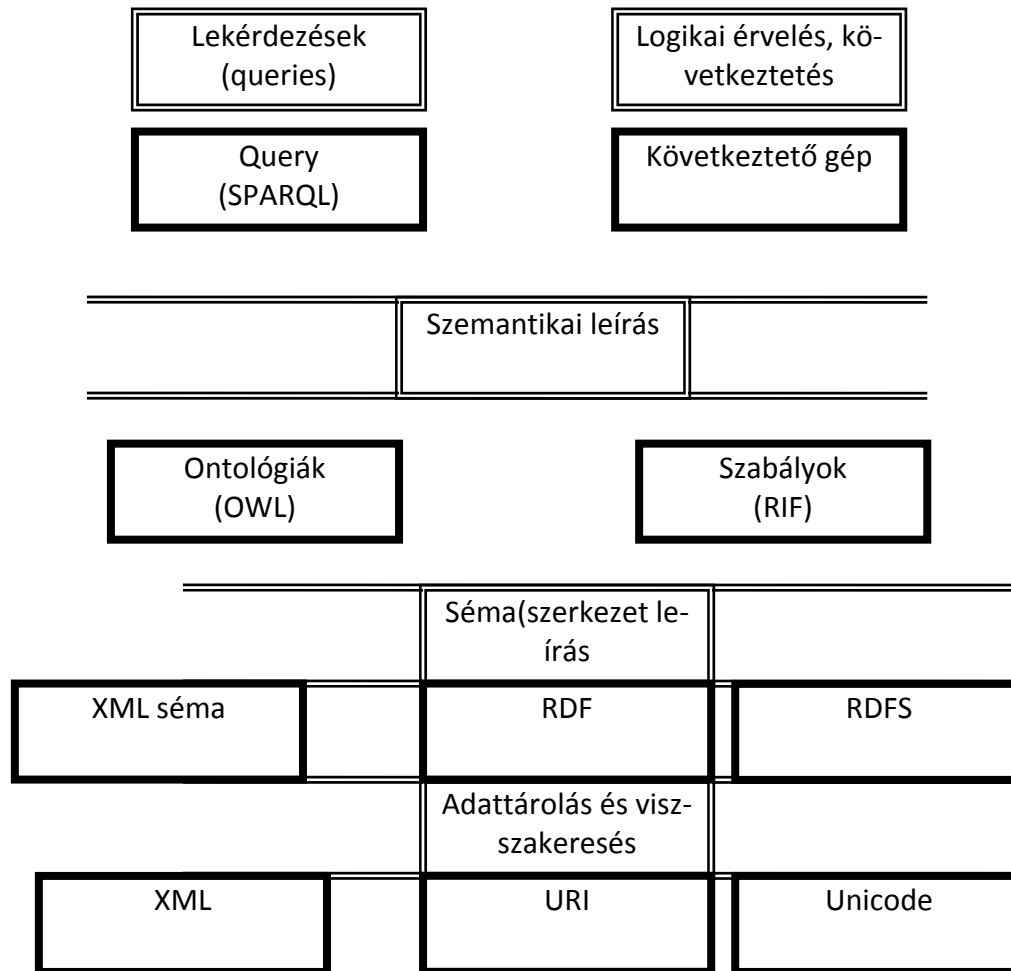
<example>
  <Person id="0332123">
    <Name>J Doe< !Name>
    <Email_Address>nemo@my.place.edu<!Email>
    <Phone Number>123 456 7890<!Phone Number>
  </Person>
  <Transaction transID="123">
    <Downtick>800</ Downtick>
    <Client>Captain Nemo</Client>
    <AccountNumber>333222111</AccountNurnber>
  </Transaction>
</example>

```

19. ábra XML példa adatábrázolásra

Az ember számára világos, hogy az ügyfelek (*Client*) a nevükkel vannak azonosítva, mely így megfeleltethető a *Name* mezőnek (19. ábra). A gép számára azonban ezt az információt a reprezentáció nem tartalmazza.

E probléma kiküszöbölésére fejlesztették ki az RDF-et, amely az entitások közötti kapcsolatok számítógép számára érthető reprezentációjának leírását biztosítja. Minden *entitást* egy *URI* azonosít. Az URI-k felhasználásával fogalmazhatók meg az RDF állítások, az úgynevezett "tripletek" vagy RDF hármassok: {*Alany*, *Állítmány*, *Tárgy*}.



20. ábra Szemantikus Web technológiák

3.21 Web Services Inspection Language – WSIL

A WS-Inspection specifikáció egy olyan keresést megkönnyítő XML formátumot biztosít, amellyel egyszerűbben feltérképezhetők a rendelkezésre álló szolgáltatások, valamint azok a *szabályok*, amelyek segítségével a kereséssel kapcsolatos információk felhasználhatóak. Egy WS-Inspection dokumentum egy már létező szolgáltatás leíráshoz tartalmaz kapcsolódási adatokat. Ezek a dokumentumok a szolgáltatás végpontján kerülnek publikálásra, akár a szolgáltatáshoz kapcsolva, akár valamilyen kapcsolódó csatornán, például HTML-ként.

A specifikációkat egy Web szolgáltatás különböző szintű és megközelítési módú leírására használjuk. A *WSDL* célja a szolgáltatás funkcionális szintű leírása, Az *UDDI* séma viszont egy inkább szervezeti (vállalati, üzleti), szervezési megközelítéssel él (Ld. 3.9.1, 51 o.). Egyelőre még nem sikerült elérniük a szabványoknak azt, hogy azokon a logikai helyeken, ahol a szolgáltatás leírásai megjelennek, egy összekapcsolható, egyszerűen létrehozható és könnyen

használható információk leírását alkossanak. A szabványosítási folyamatok, kísérletek, kutatók folynak ezen a területen.

A WS-Inspection specifikáció ezen problémát megoldandó egy olyan XML struktúrát definiál, amely a különböző típusú szolgáltatás-leírások hivatkozásait összegyűjti egy összevont (aggregált) formában, és egy jól definiált felhasználási mintázatot hoz létre a szolgáltatási struktúra példányaihoz. Ezáltal a WS-Inspection specifikáció egy lehetőséget nyújt egy szolgáltatás-kiszolgáló, szolgáltatást nyújtó szolgáltatásainak feltérképezéséhez.

Már léteznek szolgáltatástárak, amelyekben *Web szolgáltatások* leírásainak gyűjteménye található. A WS-Inspection specifikáció olyan mechanizmusokat biztosít, melyekkel ezekre a szolgáltatástárakra hivatkozni lehet, és értelmezhetővé válnak, így a leírásokban található redundáns információk kiszűrhetők.

3.22 Resource Description Framework -RDF

Az RDF egy adatleíró nyelv, amellyel XM formátumban erőforrásokkal kapcsolatos meta-adatokat ábrázolhatunk. Az RDF e meta-adatokat hierarchikusan felépíthető URI-k segítségével azonosítja, így lehetőség van az egyes erőforrások tulajdonságainak és összefüggéseinek gráfszerű ábrázolására.

```
<http://example.net/transaction/id567><http://example.net/conducted_by><http://different.example.net/person/id123>
```

21. ábra RDF hármások (triplet)

Az RDF létrehozásának egyik fő célja az volt, hogy formális szemantikát és egy egyszerű, gráf alapú adatmodellt alkalmazva létrejöjjön egy olyan leíró nyelv, mellyel az internetes erőforrásokat oly módon ábrázolhatjuk, hogy egyrészt azok bármely később készített alkalmazásban felhasználhatók legyenek, másrészt ezek információk hozzáférhetősége, kereshetősége és felhasználhatósága Internetes mértékben is jól működjön.

A fenti cél megvalósítása érdekében az RDF korlátlanul bővíthető, URI hivatkozásokra épülő szókészletet használ. Az alkalmazások közötti kommunikáció (adatcsere) azért egyszerű, mert egyrészt az RDF az XML Schema adattípusait használja, másrészt rendelkezik egy ajánlott XML szintaxissal, amely lehetővé teszi az adatmodell egységes kódolását az alkalmazások számára.

A relációkra való megszorításokhoz, peremfeltételek leírásához sémanyelveket fejlesztettek ki az XML-hez és az RDF-hez is. Egy ember könnyen felismeri azt, hogy pl.: egy tranzakciót egy személy hajtja végre, de egy tranzakció nem vezérelhet egy személyt. Az RDF az XML és a sémanyelvek együttes alkalmazása robusztus, jól használható eszközt képez a meta-adatok lekódolására és az entitások közötti relációk automatikus kinyerésére. Noha az RDF teljes körűen nem tudja megoldani az értelmezési, szemantikai problémát, mégis tud valami segítséget nyújtani. Jelenleg még nem létezik olyan mechanizmus, amelynek segítségével a kapcsolatokat automatikusan azonosítani lehessen, illetve a résztvevő felekre vonatkozó peremfeltételeket kellő részletezettséggel meg lehessen fogalmazni.

Az entitások közötti implicit kapcsolatok felderítésére logikai szabályok bevezetésére van szükség, melyek segítségével automatikus következtetés valósítható meg. Ezek a logikai szabályok a kapcsolódó meta-adatokkal képezik a formális ontológiák alapját, melyekről a későbbiekben lesz szó.

3.23 Web Ontology Language -OWL

Az OWL **ontológiák** [79, 95] leírására alkalmas nyelvek családja. Az OWL három résznyelvet specifikál, ahol minden egyes szint az alatta levő szint szintaktikai bővítése:

- -OWL Full (legbővebb)
- -OWL DL
- -OWL Lite (legsűkebb)

Az OWL segítségével definiálhatunk osztályokat, egyedeket, tulajdonságokat, valamint kapcsolatokat ezek között. Ahhoz, hogy az ontológiákat maximális hatékonysággal használhassuk, fontos, hogy újrafelhasználhatóak legyenek (ezáltal kevesebb erőforrást kell a fejlesztésre fordítani). Ha megfelelő módon kapcsolunk össze osztályokat és tulajdonságokat, akkor sokféle információ kinyerésére van lehetőség, így ontológiáink flexibilisebben felhasználhatóak lesznek.

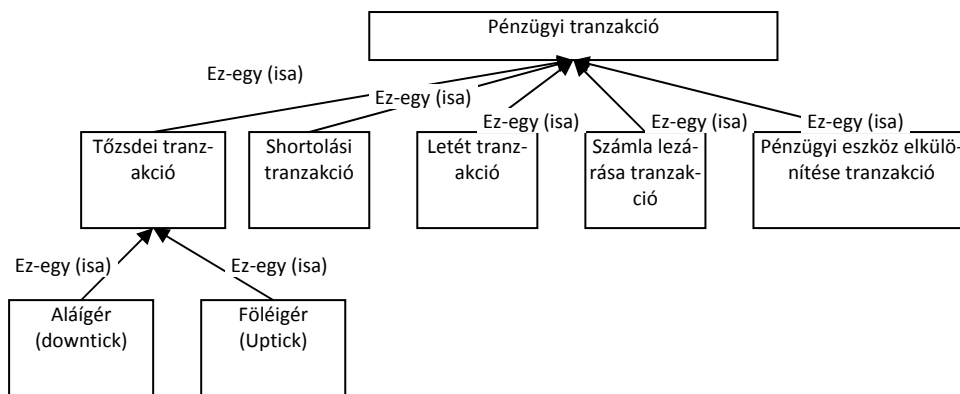
A fentiek alapján látszik, hogy a legnagyobb kihívás egy jól használható ontológia megalkotásánál az entitások (osztály, egyed) és a köztük lévő kapcsolatok megfelelő felépítése. Az OWL ehhez számos segítséget nyújt. Lehetőségünk van például meghatározni:

- entitások azonosíthatóságának a feltételeit;
- tulajdonságok/osztályok közötti egyenértékűséget, egyenlőséget;
- az entitások sajátosságaira, tulajdonságaira vonatkozó korlátozásokat;

- entitások megkülönböztethetőségének feltételeit.

Fontos, hogy egy ontológia verzió nyomon követhetők legyenek, változás és verzió kezelés alkalmazható legyen, hiszen normál esetben egy ontológia az életciklusa során számos fejlesztésen esik át. Az OWL specifikációja erre is nyújt megoldást, lehetőségünk van az éppen módosított ontológiát az előző verziójához hozzákapcsolni. Az OWL könnyű használhatóságát garantálja az XML alapú felépítés.

A szemantika egy széles körben elterjedt reprezentációja az ontológia. Az ontológia fogalmakat, köztük lévő relációkat valamint a fogalmakkal kapcsolatos következtetésekhez szükséges szabályokat tárol, strukturált formában.



22. ábra Egyszerű ontológia pénzügyi tranzakciókra

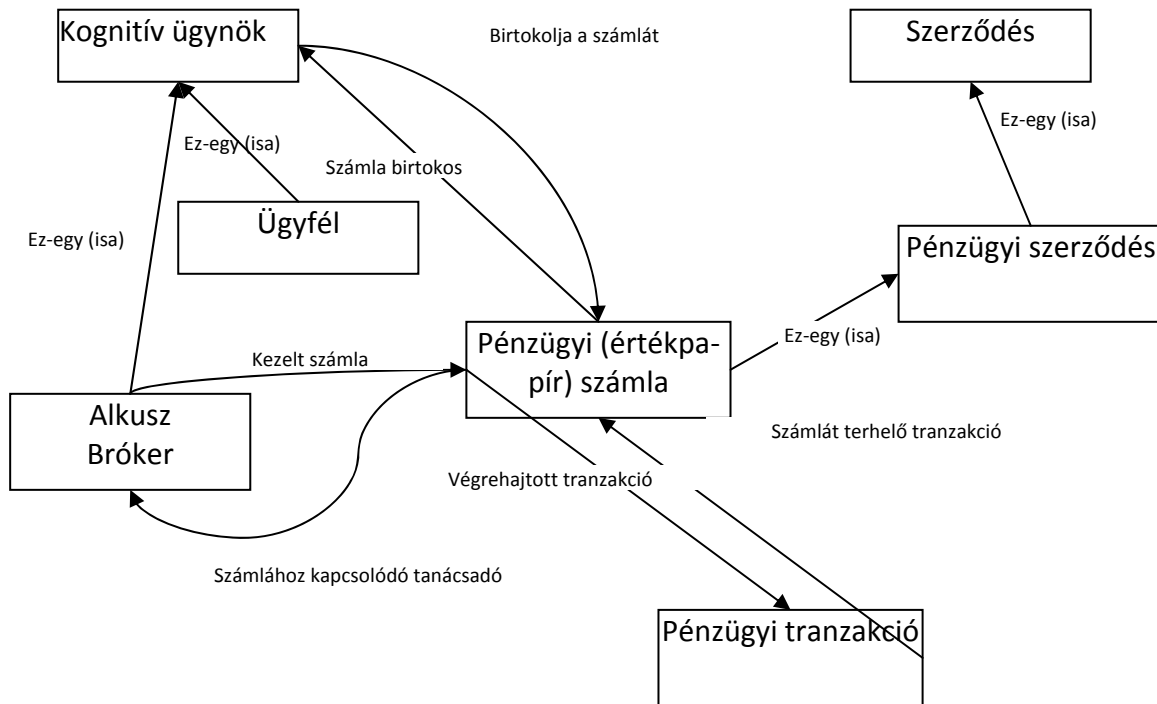
3.24 Szemantikus Web szolgáltatások

Jelenleg is fontos kutatási terület az ontológiák és egyéb szemantikus web technológiák alkalmazása, a szemantikus Web szolgáltatások használata. A terület egyik legfontosabb feladata a Web szolgáltatások szemantikus leírásának szabványosítása.

Az egyik legismertebb Web szolgáltatás szemantikus annotációjára kifejlesztett nyelv az SAWSDL (Semantic Annotations for Web Services Description Language). Az SAWSDL segítségével egy szolgáltatás leíráson (WSDL) belül a Web szolgáltatásokkal kapcsolatos ontológia fogalmakat, azonosíthatjuk és definiálhatjuk. Ez a nyelv nem alkalmas teljes ontológia leírásra, de a WSDL lehetséges komponenseinek egy részhalmazára lehetővé teszi a fogalmak annotálását, megjegyzésekkel történő ellátását. SAWSDL ontológia **agnosztikus**, azaz nem részesít előnyben egyetlen egy ontológia leíró nyelvet sem., vagy kódolási konvenciót.

Egyéb web szolgáltatásokhoz tartozó ontológiákat leíró nyelvek pl.: OWL-S (OWL Web szolgáltatás specifikus változata) és a WSMO (Web Services Modeling Ontology). Ezek a nyelvek lehetővé teszik az ontológiák WSDL-lel történő integrációját.

Sok szemantikus Web szolgáltatás alkalmazás született, azonban az automatikus szolgáltatás integráció feladata még megoldásra vár.



23. ábra Szabályok a pénzügyi ontológiában

A szemantikus web az elmúlt években népszerűsége tett szert. Bár a kezdeti lelkesedés egyfajta mérséklődésen ment keresztül, a területen történő mind egyetemi, akadémiai, mind ipari fejlesztések továbbra is folytatódnak. A terület bevált mesterséges/számítógépes intelligencia beli módszerek újjáélesztését, refaktorálását igényli, valamint az adatkezelés területén is rengeteg kihívást jelent, különösen a nagy méretű adathalmazok feldolgozása terén (**Big Data**). További információ technológiai, adatintegritási, jogi és kiértékeléssel kapcsolatos kérdések merülnek fel, melyeket a szemantikus webet képező módszerek kidolgozásakor és azokat felhasználó alkalmazások készítésekor elsődleges szempontként kell figyelembe venni. A szemantikus Web technológia fontos területeiként azonosították: a mediációt (*mediation*), a biztonságot, a folyamatok összeépítését, tárgyalásos egyeztetése, szerződéses megállapodás létrehozását, üzenet megformázását. Ezeket a kérdéseket kísérleti fejlesztésekkel és kutatásokkal, prototípusokkal vizsgálják a dinamikus, Internet környezetben.

4 SZERVEZETI ARCHITEKTÚRA MÓDSZEREK, MEGKÖZELÍTÉSEK

A szervezeti (vállalati) működés egyre komplexebbé válásával egyre fontosabbá vált az architektúrais megközelítés, mely egyfajta rendszerezett, strukturált gondolkodásmódot jelent. Ez a fajta megközelítés alapot nyújt – ahogy azt korábban már láttuk (2.5) - a szervezet átalakításánál informatikai rendszerfejlesztési programok és projektek irányítására és megszervezésére.

A gazdaság és társadalom fejlődésével, a globalizációval párhuzamosan a szervezeteknek (vállalatoknak) egyre komplexebb és szerteágazóbb feladatokat kellett megoldani, a hatékonyabb vállalaton belüli és vállalatok közötti együttműködés érdekében. Az architektúra alapú megközelítése először a műszaki, információ-technológiai, informatikai területeken nyert létjogosultságot, de lassan a szervezetek (vállalatok) vezetésével foglalkozó szakemberek is felismerték előnyeit. Fokozatosan a szervezeti (vállalati, üzleti) stratégia alkotás, és szervezet fejlesztés integráns részévé vált a *szervezeti (vállalati, üzleti) architektúra* alapú megközelítés.

A gyakorlati megoldások, a műszaki-tudományos megközelítések, valamint a szervezés-vezetés tudomány fejlődése során nemcsak különböző definíciók és meghatározások, hanem különböző megközelítések, módszerek, keretrendszerek jöttek létre. Az alábbiakban nem teljes körű ismertetését adjuk a legjelentősebb megközelítéseknek, röviden összefoglalva a jellemzőiket. Egy szélesebb körű feldolgozást a következő munkában lehet találni [41].

4.1 Zachman féle szervezeti architektúra keretrendszer (The Zachman Enterprise Framework)

A keretrendszer valójában egy olyan **ontológia** a szervezeti architektúra leírására, fejlesztésére, amely az idők folyamán (pontosan 1987 óta, [124,106]) komoly evolúción ment keresztül. A keretrendszer nem módszertan, mivel nem ad folyamat leírást, előírást a szervezeti architektúra kialakítására, hanem egy deklaratív leírása a szervezetnek (vállalatnak, üzletnek), egy struktúra a szervezeti (vállalati, üzleti) információrendszerek fejlesztésére. A későbbiekben több szervezet, több iparágban – a szerzők és a keretrendszer eredeti szándékai szerint - egy szervezeti szintű architektúra szemléletben a szervezetek folyamatainak fejlesztésére, javítására kezdték el széles körben használni. A keretrendszer tehát nem módszer, nem módszertan, nem tartalmaz architektúra kialakítására lépéseket, technikákat, nyelveket és folyamat leírást, folyamat-központú megközelítést. (Ld. [351]). Más informatikai terminoló-

giát használva: **meta-modellként** szolgál, a *szervezeti (vállalati, üzleti) architektúra* kialakításához.

4.1.1 A keretrendszer oszlopainak jelentése

A szervezeti (vállalati, üzleti) architektúra keretrendszer elemeit, egy két-dimenziós mátrixban, táblázatban adta meg Zachman. A táblázat oszlopainak megfelelő első dimenzió elemei, a vizsgált dolog egy-egy oldalának („*aspect*”) felelnek meg. Rudyard Kipling egy versében ezt a hat kérdést „A hat derék szolgaként” fogalmazta meg¹¹. Az oszlopok fejlécében a következő kérdések találhatóak: **Mit?**, **Hogyan?**, **Hol?**, **Ki?**, **Mikor?** és **Miért?**. (Ld.

24. ábra). A másik dimenzió, a táblázat sorainak megfelelő dimenzió, bizonyos szempontokat, perspektívát („*perspective*”), a keretrendszer egy-egy nézőpontját testesíti meg. Nevezetesen a következőket: a **szervezeti (vállalati, üzleti) tervező** („*planner*”), a **tulajdonos** („*owner*”), a **tervező** („*designer*”), a **kivitelező / építő** („*builder*”), **alvállalkozó / beszállító, készítő** („*subcontractor*”), és végül a **működő szervezet** („*functioning enterprise*”).

A Zachman féle keretrendszer általános ábrázolása egy 6x6-os mátrix, aminek oszlopait a kérdések és sorait a nézőpontok határozzák meg. A keretrendszer elemeinek ábrázolása a táblázat celláival történik, amik a kérdő szavak és a nézőpontok metszéspontjai.

Amióta John A. Zachman publikálta módszertanát, azóta az épületek, repülőgépek, és más komplex ipari termékek leíró ábrázolásának, architektúrájának meta-modellezéséhez is felhasználják ezt a megközelítést. Ez bizonyítéka a Zachman féle keretrendszer létjogosultságának.

A keretrendszer segítséget ad a szervezet céljainak, működésének feltételeinek, környezetének vizsgálatára, a szervezet és az informatikai funkció vezetési szintjein elhelyezkedő vezetőkkel, kezdve a felső vezetéssel a stratégiai szinten.

1. oszlop: **Adat (mit)** : Ez az oszlop a szervezet (vállalat, üzlet) működéséhez szükséges adatokkal foglalkozik, az adatok szerkezetével, hogyan tárolják az adatokat stb.
2. oszlop **Funkció (hogyan)**: Ez az oszlop a szervezet (vállalat, üzlet) működésével foglalkozik. A szervezet stratégiai célkitűzéseit, „küldetését” fordítja le a szervezeti működés egyre részletesebb leírásaira.
3. oszlop **Hálózat (hol)**: Ez az oszlop a szervezet (vállalat, üzlet) elemeinek, tevékenységeinek földrajzi elhelyezkedésével, szétosztottságával foglalkozik.

4. oszlop **Személyek (ki)**: Ebben az oszlopban a szervezetben munkát végzőkkel, a munkafeladatok személyekhez rendelésével, az emberek közötti kapcsolatokkal foglalkoznak.
5. oszlop **Idő (mikor)**: A valóságos idő egy absztrakciójával foglalkozik azért, hogy az események közötti kapcsolatokat meg lehessen tervezni, amelyek meghatározzák a teljesítmény kritériumokat és számszerűsíthető szolgáltatási szinteket szabnak meg a szervezet erőforrásai tekintetében.
6. oszlop **Motiváció (miért)**: A szervezet (vállalat, üzlet) motivációjának meghatározásával leíró formában foglalkozik, és általában célok és elérésükhöz szükséges eszközök megfogalmazásán keresztül jeleníti meg a szervezet motivációit. Az eszközök itt általában a cél eléréséhez használt módszert jelentik.

4.1.2 A keretrendszer sorainak jelentése:

1. sor: **Kiterjedés, terjedelem** : Az első architektúra vázlatot jelenti, amely egy olyan „gombócokat” tartalmazó ábra, amely nagyvonalakban leírja az elképzelt végső struktúrának a kiterjedését, határait, nagyságát, körvonalait, térbeli kapcsolatrendszerét és alapvető céljait. Tulajdonképpen egy *szervezeti (vállalati, üzleti) tervező* vagy **befektető** számára szóló vezetői összefoglalónak felel meg, akinek a leendő rendszer kiterjedéséről, a várható költségekről és a nyújtandó szolgáltatásokról kellene egy előzetes képet kapnia.
2. sor: **Szervezeti (vállalati) modell** : A következő ábrázolás, az architektúra tervezőé, aki a rendszer végső felépítését annak a tulajdonosnak a szemszögéből, perspektívájából írja le, akinek majd naponta együtt kell élnie ezzel a szervezeti (vállalati, üzleti) környezettel. Megfelel a szervezeti (vállalati, üzleti) modellnek, amely a szervezeti (vállalati, üzleti) működés tervét testesíti meg és a legfontosabb szervezeti (vállalati, üzleti) egységeket, elemeket, folyamatokat és köztük fennálló kölcsönhatásokat jeleníti meg.
3. sor: **Rendszer modell**: Ez az architektúra tervekészítő tervezete, amely az eddigi modell ábrázolásokat lefordítja a rendszertervező szemszögének, perspektívájának megfelelő részletes műszaki leírásokra, specifikációkra.
4. sor: **Technológia modell**: (műszaki, informatikai modell) A szállítónak általában az architektúra tervekészítő tervezetét át kell dolgoznia azért, hogy a kivitelező szemszögé-

ből, perspektívájából értelmezhető legyen, visszatükrözze azokat a korlátokat és peremfeltételeket, amelyek a rendelkezésre álló eszközökből, technológiákból, műszaki lehetőségekből, anyagokból állnak. A kivitelező (rendszerkészítő) tervezete a technológiai modellnek felel meg, amelynek a feladata az, hogy az információrendszer modellt a programozási nyelvek, a számítógép perifériák és más technológiai elemekhez illessze, adaptálja.

5. sor: **Részletes specifikáció:** Ez a modell megfelel annak a részletes specifikációnak, amelyet azoknak a programozóknak adnak át, akik egyes egyedi modulok programozását végzik el tekintet nélkül arra a környezetre illetve annak szerkezetére, amiben dolgoznak, és / vagy a folyamat tervezők kapják meg ezeket a terveket azért, hogy a munkafolyamatok részletes tervét elkészítsék. (a rendszer tényleges megvalósítása, telepítés, üzembe helyezés).
6. sor: **Működő szervezet** (vállalat) : Végül, a rendszert elkészítik és a szervezet részévé válik. Ebből a szemszögből a program listák, az adatbázis specifikációk, a hálózatok és így tovább jelennek meg, amelyek mindegyike egy bizonyos rendszert alkot. Ezeket a leírásokat az adott rendszerhez illeszkedő, specifikus műszaki, informatikai terminológiával írják le.

J. A. Zachman S. H. Spewak	Tervezési csillagzatok/terjedés (összhangok)	Entitások = mit? adat architektúra	Tevékenységek = hogyan? alkalmazási architektúra	Helyek = hol? műszaki architektúra	Személyek = ki? egységek	Idő = mikor? események listája	Motiváció = miért? célkitartások listája	Kifejedés
Tulajdonos Szervezeti modell (konceptuális)	A szervezeti feladatok listája Entitás = A szervezeti feladatok osztálya Séma modell Entitás = Szervezeti egység Kapcsolat = Szervezeti kapcsolatok	A szervezeti folyamatok listája Folyamat = a szervezeti folyamatok osztálya Szervezeti folyamatmodell Folyamat = szervezeti folyamat Kibemenet = Szervezeti erőforrások	A szervezet telephelyének listája Csomópontok = Főbb szervezeti telephelyek Szervezeti logisztikai rendszere Csomópont = Szervezeti telephely Kapcsolódás = Szervezeti kapcsolódások	A szervezet legfontosabb egységeinek listája Emberi erőforrás = Nagy szervezeti egységek Munkafolyamat modell Emberi erőforrás = szervezeti egység Munkavégzés = A munka terméke	A szervezetek fontos események listája Idő = Jelentős szervezeti események Központi munkaterv Idő = Szervezeti esemény ciklus Ciklus = Szervezeti ciklus	A szervezeti célkitartások listája Eredmény / eszköz = Főbb szervezeti célok / Kritikus sikertényezők Üzleti szervezeti terv Eredmény = Szervezeti célkitartások Eszköz = Szervezeti stratégia	Szervezeti szabályok Eredmény = Strukturális utasítás Eszköz = Működési utasítás Szabályzat tervezés Eredmény = feladatok Eszközök = tevékenységek	Szervezeti modell
Felhasználó Információsrendszer modell (Logikai)	Logikai adat modell Entitás = adat entitás Kapcsolat = adat kapcsolat	Alkalmazási architektúra Folyamat = alkalmazási funkció Kibemenet = Felhasználói szempontok	A rendszer földrajzi elhelyezkedésének architektúrája, pl. Elosztott rendszerszervezet Csomópont = Inf. rendszer funkció. (Processzo, tároló, stb.) Kapcsolódás = Vonal jellemzők	Ember-gép kapcsolati architektúra Emberi erőforrás = szerep Munkavégzés = Lezajló tevékenység Mégjelentési architektúra Emberi erőforrás = felhasználó Munkavégzés = Képzőanyag formátum	Feldolgozási struktúra Idő = rendszer esemény ciklus Ciklus = Adatfeldolgozási ciklus Ellenőrzési struktúra Idő = Végrehajtási ciklus Ciklus = Egység ciklus	Szervezeti szabályok Eredmény = Strukturális utasítás Eszköz = Működési utasítás Szabályzat tervezés Eredmény = feladatok Eszközök = tevékenységek	Rendszer modell	
Kivitelező, rendszerfejlesztő technológiai modell (fizikai)	Fizikai adatmodell Entitás = Személy / tábla/ stb. Kapcsolat = pointer/ kulcs/stb.	Rendszerterv Folyamat = Számítógép művelet Kibemenet = adat elemek/halmiszok	Rendszer architektúra technológiai architektúra Csomópont = Hardver/rendszer szoftver Kapcsolódás = Vonal specifikációk	Biztonságtechnikai architektúra Emberi erőforrás = személy azonosítás Munkavégzés = feladat	Feldolgozási struktúra Idő = rendszer esemény ciklus Ciklus = Adatfeldolgozási ciklus Ellenőrzési struktúra Idő = Végrehajtási ciklus Ciklus = Egység ciklus	Szervezeti szabályok Eredmény = Strukturális utasítás Eszköz = Működési utasítás Szabályzat tervezés Eredmény = feladatok Eszközök = tevékenységek	Technológiai modell	
Végrehajtó (operatív) Rendszerespecifikáció (szükségletek nélkül)	Adat definíció szótar vagy könyvtár Entitás =mező Kapcsolat =cím Adatok	Programok támogató szoftver elemek Folyamat = programnyelvi definíció Kibemenet =Ellenőrzési blokk Funkciók	Hálózati architektúra Csomópont = címzés Kapcsolódás =protokoll	Emberi erőforrás = személy azonosítás Munkavégzés = feladat	Feldolgozási struktúra Idő = rendszer esemény ciklus Ciklus = Adatfeldolgozási ciklus Ellenőrzési struktúra Idő = Végrehajtási ciklus Ciklus = Egység ciklus	Szervezeti szabályok Eredmény = feladatok Eszközök = tevékenységek	Elemek	
Működő vállalat/ szervezeti intézmény	Adatok	Funkciók	Hálózat	Szervezet	Munkaterv	Stratégia		

24. ábra Szervezeti (üzleti vállalkozás) architektúra Zachman és TOGAF keretrendszere. A TOGAF a színezett részt fedi le.

4.1 Zachman féle szervezeti architektúra keretrendszer (The Zachman Enterprise Framework)

	WHAT	HOW	WHERE	WHO	WHEN	WHY	
SCOPE CONTEXTS	Inventory Identification Inventory Types	Process Identification Process Types	Network Identification Network Types	Organization Identification Organization Types	Timing Identification Timing Types	Motivation Identification Motivation Types	STRATEGISTS AS THEORISTS
BUSINESS CONCEPTS	Inventory Definition Business Entity Business Relationship	Process Definition Business Transform Business Input	Network Definition Business Location Business Connection	Organization Definition Business Role Business Work	Timing Definition Business Cycle Business Moment	Motivation Definition Business End Business Means	EXECUTIVE LEADERS AS OWNERS
SYSTEM LOGIC	Inventory Representation System Entity System Relationship	Process Representation System Transform System Input	Network Representation System Location System Connection	Organization Representation System Role System Work	Timing Representation System Cycle System Moment	Motivation Representation System End System Means	ARCHITECTS AS DESIGNERS
TECHNOLOGY PHYSICS	Inventory Specification Technology Entity Technology Relationship	Process Specification Technology Transform Technology Input	Network Specification Technology Location Technology Connection	Organization Specification Technology Role Technology Work	Timing Specification Technology Cycle Technology Moment	Motivation Specification Technology End Technology Means	ENGINEERS AS BUILDERS
COMPONENT ASSEMBLIES	Inventory Configuration Component Entity Component Relationship	Process Configuration Component Transform Component Input	Network Configuration Component Location Component Connection	Organization Configuration Component Role Component Work	Timing Configuration Component Cycle Component Moment	Motivation Configuration Component End Component Means	TECHNICIANS AS IMPLEMENTERS
OPERATIONS CLASSES	Inventory Instantiation Operations Entity Operations Relationship	Process Instantiation Operations Transform Operations Input	Network Instantiation Operations Location Operations Connection	Organization Instantiation Operations Role Operations Work	Timing Instantiation Operations Cycle Operations Moment	Motivation Instantiation Operations End Operations Means	WORKERS AS PARTICIPANTS
Released October 2008	INVENTORY SETS	PROCESS TRANSFORMATIONS	NETWORK MODES	ORGANIZATION GROUPS	TIMING PERIODS	MOTIVATION REASONS	Normative Projection on Version 2.01

25. ábra The Zachman Framework²™ Standards, 2008¹²

4.2 A Clinger-Cohen törvény az Amerikai Egyesült Államokban

A Zachman féle keretrendszer kialakítása után sokféle architektúra keretrendszer sok különböző szándékkal jött létre. Az Amerikai Egyesült Államok kormányzata jelentős szerepet játszott a szervezeti architektúrák fejlesztése területén. A sikertelen kormányzati szoftverfejlesztési projektek okát feltáró 1996-os kongresszusi vizsgálat eredményeként elfogadott információtechnológia irányításának átalakításáról szóló törvény (ITMRA The Information Technology Management Reform Act) - amit az 1996-os Clinger-Cohen törvénynek is neveznek - kimondja, hogy az Amerikai Egyesült Államok minden szövetségi állami szervezete **Informatikai igazgatójának** (Chief Information Officer, CIO) felelőssége: az adott szövetségi állami szervezet jó minőségű és integrált informatikai architektúrája megvalósításának elősegítése, fejlesztése és fenntartása. Az ITMRA törvény alapján, az USA összes szövetségi állami szervezeténél kötelező létrehozni szervezeti architektúrát.

A Clinger-Cohen törvény következtében jöttek létre további szervezeti architektúra keretrendszerek: a C4ISR AF (Command, Control, Communications, Computers, Intelligence, Surveillance and Reconnaissance Architecture Framework) és GERAM (Generalised Enterprise Reference Architecture and Methodology) keretrendszerek, amelyek alapját képezték a később 2002-ben kifejlesztett DoDAF modellnek.

4.3 C4ISRAF

C4ISRAF (Command, Control, Communications, Computers, Intelligence, Surveillance and Reconnaissance Architecture Framework) architektúra keretrendszert az első Öböl háború (Irak, Kuvait) során felismert informatikai, információtechnológiai problémák kiküszöbölésére hozták létre. Az elnevezése mutatja a katonai alkalmazás szándékát – parancsnoki, irányítási, távközlési, számítógépes, hírszerzési, megfigyelési és felderítési architektúra keretrendszer.

Az Amerikai Egyesült Államok Védelmi Minisztériuma (Defence of Department, DoD) 1995-ben bocsátott ki egy rendeletet, melynek értelmében, a különböző harcászati egységek – vadászrepülőgépek, tengerészeti, és szárazföldi egységek - számára szükséges C4I (parancs, irányítás, kommunikáció, hírszerzés) rendszerek számára a katonai képességek iránti igényt előírják, és a képesség igényeknek megfelelő harcászati és informatikai eszközöket hozzanak létre.

4.4 DoDAF

A keretrendszert az Amerikai Egyesült Államok Védelmi Minisztériuma fejlesztette ki, mind az amerikai honvédelmi erők mind a szövetségi tevékenységgel kapcsolatos szervezeti és informatikai feladatok kezelésére. A DoDAF (*Department of Defense Architecture Framework*) egy szabvány, mely USA-ban minden védelmi tevékenységgel kapcsolatban kötelező betartani. A DoDAF keretrendszer segítségével modellezhető a szervezet felépítése, az informatikai rendszerek, és a részletesebb műszaki alkotó elemek.

A DoDAF célja, hogy biztosítsa az architektúra-leírások összehasonlíthatóak és összekapcsolhatóak legyenek egymással, a szervezetek között, beleértve a különböző katonai parancsnokságokat is.

A DoDAF felépítésének a filozófiai alapja az, hogy az architektúrákat egy konkrét céllal kell építeni, az architektúráknak segítenie és nem megakadályoznia kell az emberek közötti kommunikációt, az architektúráknak olvashatónak, érthetőnek, összehasonlíthatónak kell lenni és az embereknek képesnek kell lenniük, több architektúra integrálására. Minden architektúrának meg kell annyira felelnie az architektúra keretrendszernek, hogy az előbb felsorolt elvek teljesüljenek. [43]

4.5 MoDAF

Az Egyesült Királyság (Nagy-Britannia) Védelmi Minisztériumának architektúra keretrendszerre, amely szigorú módszertan alapján szabványos módot ad a minisztériumi architektúra ábrázolására, és a MOD megoldásai lehetővé teszik az elosztott, hálózati környezetben történő alkalmazásra (NEC, Network Enabled Capability), amivel elősegíti azt, hogy Nagy Britannia Védelmi Minisztériuma a védelmi feladatok megszervezését ellássa. Előír kulcsfontosságú szervezeti, műszaki és informatikai területeket, amelyek alapján kell dokumentálniuk a jelentősebb harcászati eszközöknek és információtechnológiai rendszereknek az architektúrájukat a MODAF által előírt módon. Könnyen alkalmazható nagy rendszerekben és a rendszerek rendszereiben is (SoSs, Systems of Systems), amelyekben komplex integrációs és együttműködési problémák merülnek fel. [20]

4.6 ISO/IEC 42010 (IEEE Std 1471-2000) szabvány

Ez a szabvány elsősorban a szoftver architektúra fogalmi szintű megragadása érdekében készült. Ezért az un. szoftverintenzív rendszerekre tekintettel dolgozták ki ISO/IEC 42010 szabványt, de a szoftverfejlesztés és a komplex rendszerek integrációjának kérdéseit is figyelem-

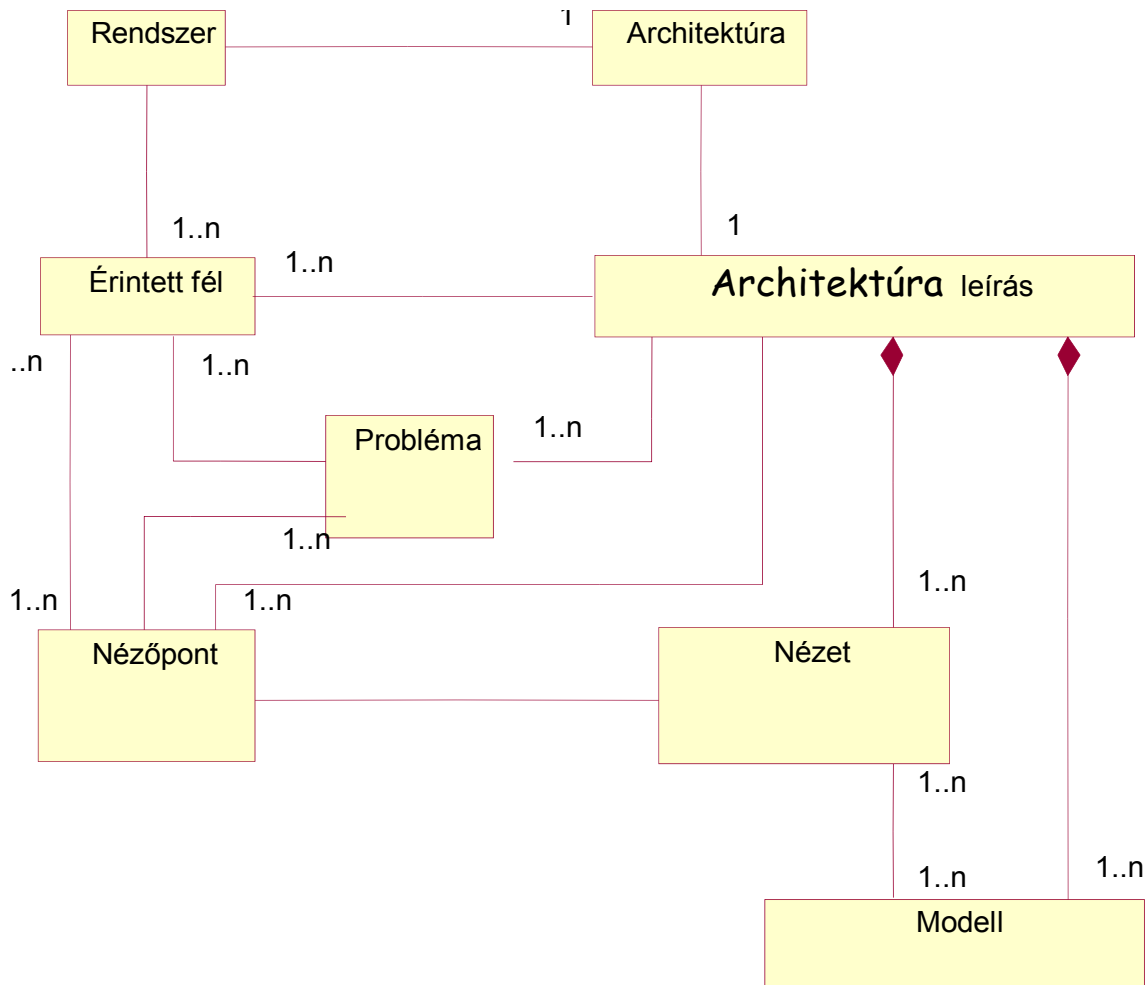
be vették. A nagy bonyolult rendszerek körébe értendők a beágyazott rendszerek, a rendszerek rendszerei, valamint az információrendszerek. A szabvány célja az előbbieken felsorolt rendszerek szabványos formátumú leírására az alapok megteremtése.

Ez a keretrendszer egy architektúra leírás tartalmát szabványosítja két fő szemszögből történő megközelítést használ fel erre a célra, a **nézetet** („view”) illetve a **nézőpontot** („viewpoint”). A *nézet* azt jeleníti meg, amit a kiválasztott *nézőpontból* látni lehet.

A modell legfontosabb alkotórészeinek leírása:

- **Rendszer**
 - Alkotórészek, komponensek halmaza, amelyeket egy adott cél elérésére szerveztek meg.
- **Architektúra**
 - Egy *rendszer* alapvető szervezési módja, amely megtestesül az alkotórészeiben, a köztük és a környezettel fennálló kapcsolataiban, továbbá a tervezését és továbbfejlesztését irányító elvekben.
- **Architektúra leírása**
 - Az architektúra dokumentálására szolgáló termékek halmaza, amelyeket különböző *nézetekbe* vannak szervezve.
- **Nézet**
 - Egy előre definiált *nézőponttal* összhangban, azaz bizonyos *problémák*, *kérdések* és *ügyek* perspektívájából a *teljes rendszer* ábrázolása, reprezentációja.
- **Nézőpont**
 - Azoknak a konvencióknak a specifikációja (műszaki leírása), amelyeket egy *nézet* felépítésére és annak a felhasználására alakítanak ki. Olyan mintázat vagy sablon, amelyből az egyedi *nézetek* kifejleszthetők, meghatározva a *nézetre* vonatkozó célokat és a célközönséget (*az érintett feleket*), továbbá a *nézet* létrehozásához és elemzéséhez szükséges technikákat.
- **Modell**
 - A *nézet* ábrázolása, reprezentációja., megjelenítése .
- **Érintett fél (Stakeholder)**

- Egyének, csoportok vagy szervezetek, akiknek valamilyen érdeke fűződik a *rendszerhez* vagy valamilyen közös ügye van a *rendszerrel*. Az érdekeket illetve a kapcsolatos ügyeket egy vagy esetleg több *nézettel* lehet reprezentálni.



26. ábra IEEE 1471 szabvány szerint a szoftver architektúra kulcs fontosságú fogalmai

4.7 E-kormányzati architektúra - FEAF (Federal Enterprise Architecture Framework)

A fejlett informatikával rendelkező országokban a szervezeti architektúra, szabványok, keretrendszer alapú megközelítések egyik vezérlő ereje az informatikai tanácsadási ipar mellett a közigazgatás volt. Ahogy azt már az előbbiekben láttuk az Amerikai Egyesült Államokban és Nagy-Britanniában a védelmi igazgatás volt az első, amely az informatika e területén is szabványosítási és szabályozási lépéseket tett, annak megfelelően, hogy több mint kétezer éves tapasztalat az, hogy műszaki újdonságok kifejlesztésének döntő része a katonai alkalmazások érdekében történik meg először.

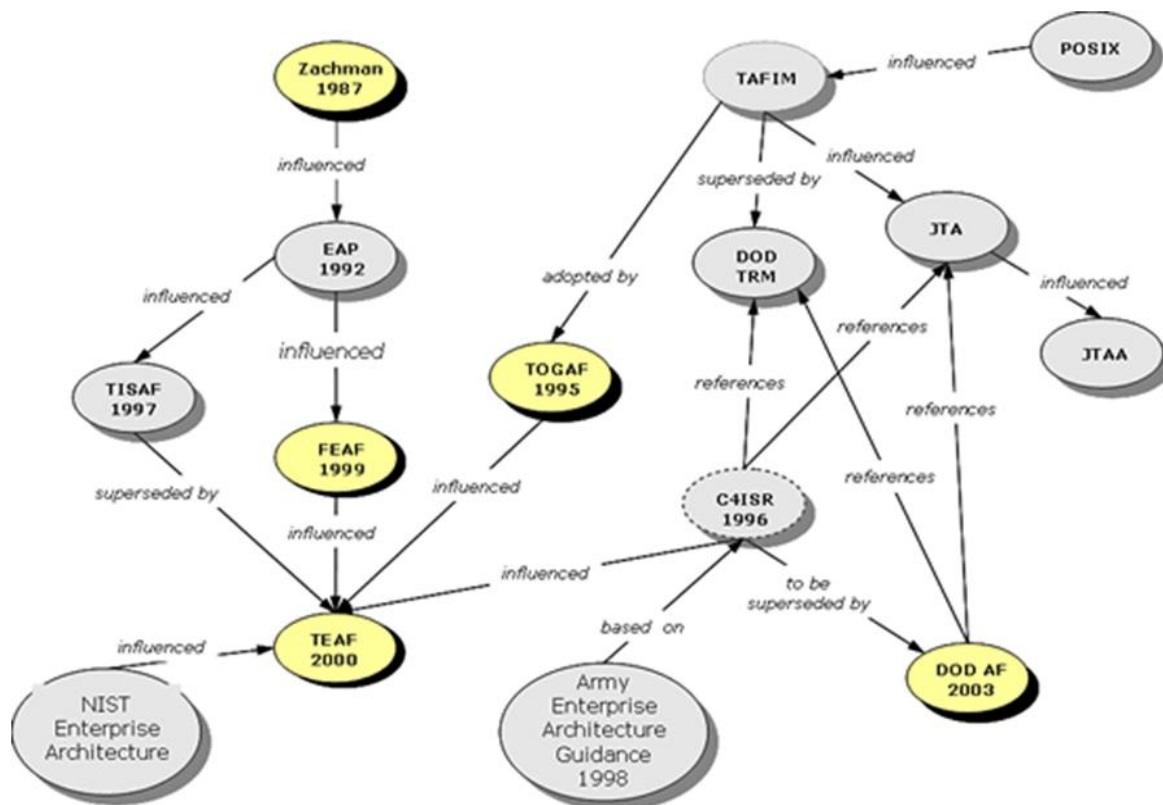
Az Amerikai Egyesült Államokban az 1996-os Clinger-Cohen törvény szerint az egyes szövetségi állami szervezetek informatikai vezetői (Chief Information Officers (CIO) egyszer csak

a **CIO Council**-ban, az **Informatikai vezetők tanácsában** találtak magukat, amelynek fő feladatává tette a törvény a szövetségi állam informatikai erőforrásainak modernizálását, magasabb kihasználtság elérését, a kooperáció fokozását, az informatikai együttműködési képesség, az interoperabilitás, valamint a nyújtott teljesítmény javítását. E feladatkör következményeként az Informatikai vezetők tanácsa 1998 áprilisában elkezdte a FEAF kidolgozását¹³. A munkában a szervezeti (vállalati, üzleti) architektúra vezető szakemberei is részt vettek tanácsadóként, pl. Zachman és Spewak.

Mivel egy architektúra fejlesztése és napra készen tartása a pillanatnyi állapot és a jövőbeli célkitűzések értékelésének egy állandó folyamata ezért a FEAF kifejlesztői abból indultak ki, hogy azt kell megfogalmaznia, hogy ez a folyamat hogyan megy végbe. Ezért nem az architektúra tartalmát alakították ki, hanem tulajdonképpen azt a keretet, „helyőrzőt”, amelyben a tulajdonképpen az architektúra tartalmi leírását el lehet helyezni. A FEAF három leglényegesebb alapelve:

- 1) A FEAF a szervezeti architektúrát négy szintre bontja: **Szervezeti, adat, alkalmazás és műszaki / informatikai**. A három utolsó szintet architektúra tervezési szintnek tekinti, amelyek a *szervezeti architektúra* alapját alkotják.
- 2) A FEAF-ot a *Zachman féle szervezeti architektúra* elvei szerint rendezték el. Ezért az oszlopokban megjelenik a *mit (adat architektúra), hogyan (alkalmazás architektúra), hol (műszaki / informatikai)*, amelyek a az öt sorra vonatkoznak: nevezetesen: **a kiterjedés(célkitűzések), szervezeti modell, információrendszer modell, műszaki/informatikai modell, részletes specifikáció**.
- 3) A FEAF egy architektúra kidolgozásának és napra készen tartását négy lépésben írja le. Nyolc komponensre szorítkozik a leírás során:
 - a. A vezérlő elvek és erők;
 - b. A stratégiai irány;
 - c. A pillanatnyi architektúra;
 - d. A cél architektúra;
 - e. Az áttérés folyamata;
 - f. Az architektúra alkotó elemei (*szervezeti, adat, alkalmazás és műszaki / informatikai*);

- g. Az architektúrát leíró modellek (az architektúra egyes alkotó elemeinek szervezési, szervezeti és tervezési modelljei);
- h. Szabványok beleértve az útmutatókat, a bevált legjobb nemzetközi gyakorlatot.



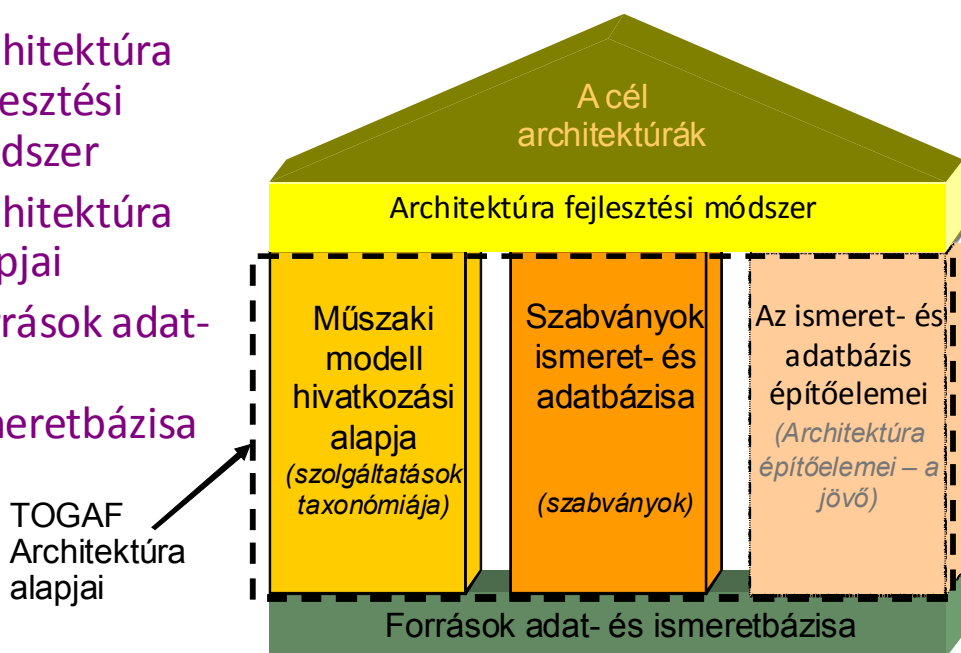
27. ábra Az architektúra keretrendszerek kaotikus fejlődésének rendszer az Amerikai Egyesült Államok-ban¹⁴

5 AZ ARCHITEKTÚRA NÉZETEI ÉS NÉZŐPONTJAI A TOGAF¹⁵ SZERINT

Az *Open Software Foundation* és az *X/Open* mint két ipari szabványosítási testület **Open Group** néven egyesült.

A TOGAF egy architektúra keretrendszer, amit az '*Open Group*' architektúra fórum fejlesztett ki és tart napra készen. 1994-ben hozták létre az első változatát, most tartanak a 9-esnek jelzett változatnál. A keretrendszer része egy architektúra fejlesztési módszer (ADM, Architecture Development Method), amely jelentős szellemi tulajdont foglal magában, az architektúra fejlesztésekben, a fórumban résztvevő cégeknél felgyűlt tapasztalatot. Ez a fejlesztési módszer előírást ad a létrehozandó vállalati architektúra kialakításának módjára és felépítésére. Egy általános megközelítést ad az információ architektúrára vonatkozóan a tervekészítés, tervezés, megvalósítás, és szervezeti irányítás tekintetében. A módszert jellemzi a technológia és eszköz semlegesség, ezáltal minden szervezet eldöntheti, hogy számára melyik informatikai technológia és eszköz az ideális.

- **Architektúra fejlesztési módszer**
- **Architektúra alapjai**
- **Források adat- és ismeretbázisa**



28. ábra TOGAF Szerkezete és alkotórészei

A jó informatikai architektúrával szemben az alábbi követelmények támaszthatók:

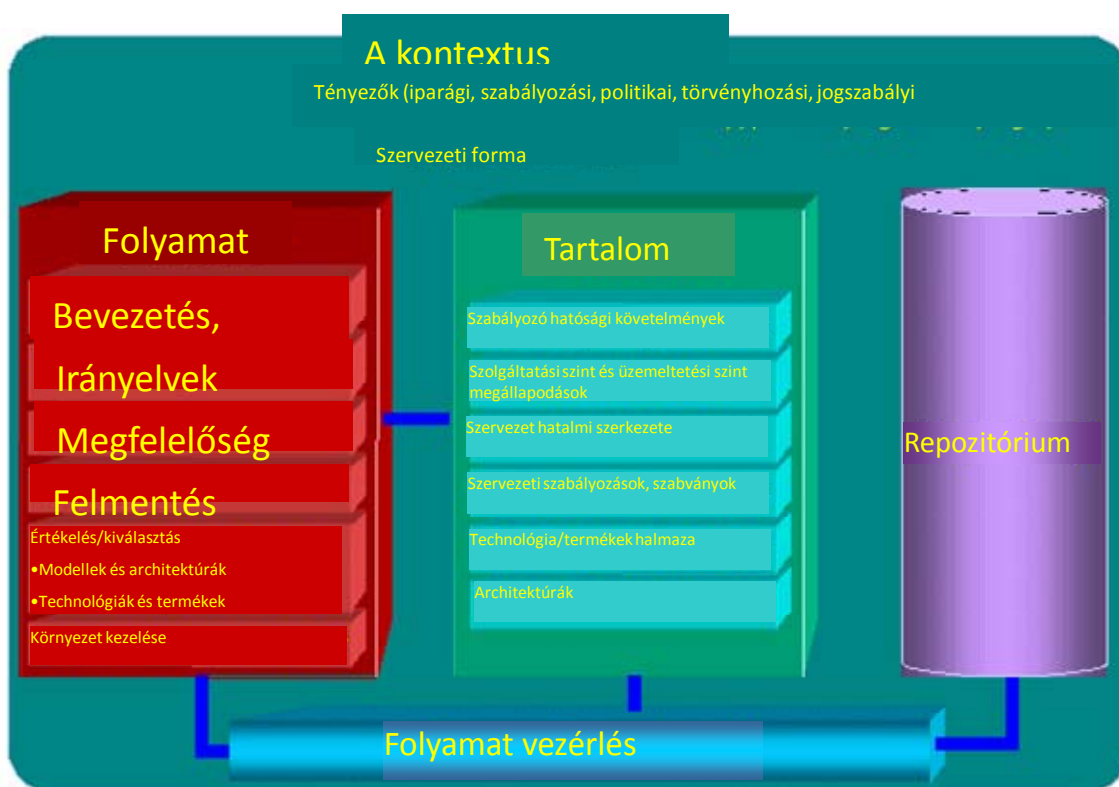
- Gyorsan reagál a külső környezet változásaira;
- A felsővezetők számára is érthető a megfogalmazása; A felső vezetés érti és támogatja;
- Egyértelműen fogalmazza meg a létező rendszerek felépítést;
- Minimális az architektúra leíró, ábrázoló alkotórészek egymáshoz való kapcsolódása;

4.7 E-kormányzati architektúra - FEAF (Federal Enterprise Architecture Framework)

- Könnyen bővíthető, napra készen tartható, aktualizálható;
- Közvetlenül a szervezet szükségleteire irányul;
- A változásokra a külső körülmények által megkövetelt sebességgel reagál (verseny, közigazgatásban jogszabály stb.);
- Világosan fogalmazza meg és ábrázolja a létező rendszerek szerkezetét;
- A leendő beszerzések és fejlesztések számára világos tervet és migrációs stratégiát nyújt;
- Csökkenti az alkotórészek közötti kapcsolatok, kapcsoló felületek számát, megkönnyítve ezzel :
 - Az alkalmazások hordozhatóságát;
 - A komponensek aktualizálását, frissítését;
 - A komponensek cseréjét;
 - A komponensek fejlesztését és karbantartását

A TOGAF kifejlesztése tehát a felhasználói oldal kezdeményezésére indult, formális felhasználói követelményeket 1994-ben fogalmazták meg.

- A főbb célok, amiket el kívántak érni:
 - Az informatikai ágazatra egy egységes architektúra keretrendszert létrehozni;
 - A keretrendszer az egyedi, specifikus szervezeti szükségletek kielégítésére szolgáló architektúrák kifejlesztését támogatja;
 - Nem „konfekció” architektúra, hanem az igényekhez, körülményekhez illeszthető, testre szabható.



29. ábra TOGAF Irányítási keretrendszer

5.1 TOGAF: Architektúra fejlesztési módszer

A TOGAF része egy Architektúra fejlesztési módszer (Architecture Development Method, ADM), amely segítségével lépésről lépésre felépíthető és napra készen tartható a szervezet architektúrája. Az ADM a szervezet főbb területeit fedi le: a szervezetet, az információrendszereket, és a műszaki, informatikai, technológiai környezetet.

5.1.1 Kezdeti szakasz céljai

Ebben a szakaszban kell gondoskodni arról, hogy minden résztvevő - illetve mindenki, akinek haszna származik ebből a megközelítésből - elkötelezett legyen az architektúra kialakítása iránt. Utána meg kell határozni azokat az architektúra elveket, amelyek megfogalmazzák azokat a peremfeltételeket és korlátokat, amelyeknek minden az architektúra kialakításával kapcsolatos munkára érvényesnek kell lenniük. Rögzíteni kell azt is, hogy azok az alkalmazottak, akik architektúrával kapcsolatos bármilyen munkát végeznek, hol találhatóak a szervezetben, és mi a feladat, hatás és felelősségi körük. Ezen kívül meg kell fogalmazni az architektúra tervezésnek, mint projektnek a kiterjedését és az előfeltételeket is.



30. ábra TOGAF ADM módszere

Az adott szervezetre vonatkozóan architektúra keretrendszert és részletes módszertant kell kidolgozni, amelyet a szóban forgó szervezet a szervezeti architektúra kifejlesztésére kíván alkalmazni, ez tipikusan az ADM testre szabását jelenti. Ezek után egy verifikációs és validációs lépést kell végrehajtani, nevezetesen egy architektúra kidolgozási feladat kivitelezését és nyomon követését (általában egy kísérleti („pilot”) projekt indítását), amelynek a célja a létrehozott, testre szabott architektúra keret életképességének bizonyítása. Az előb-

biek kiegészítéseként - ha szükséges -, a szóba jöhető architektúra tervező eszközök kiértékelésére szempontrendszert kell meghatározni¹⁶, adatszótár, tervek-tára / repozitórium ; továbbá elő kell írni az eszközök használatával, kezelésével kapcsolatos folyamatokat, amelyek az architektúra tervek rögzítésével, nyilvánosságra hozatalával és napra készen tartásával foglalkoznak.

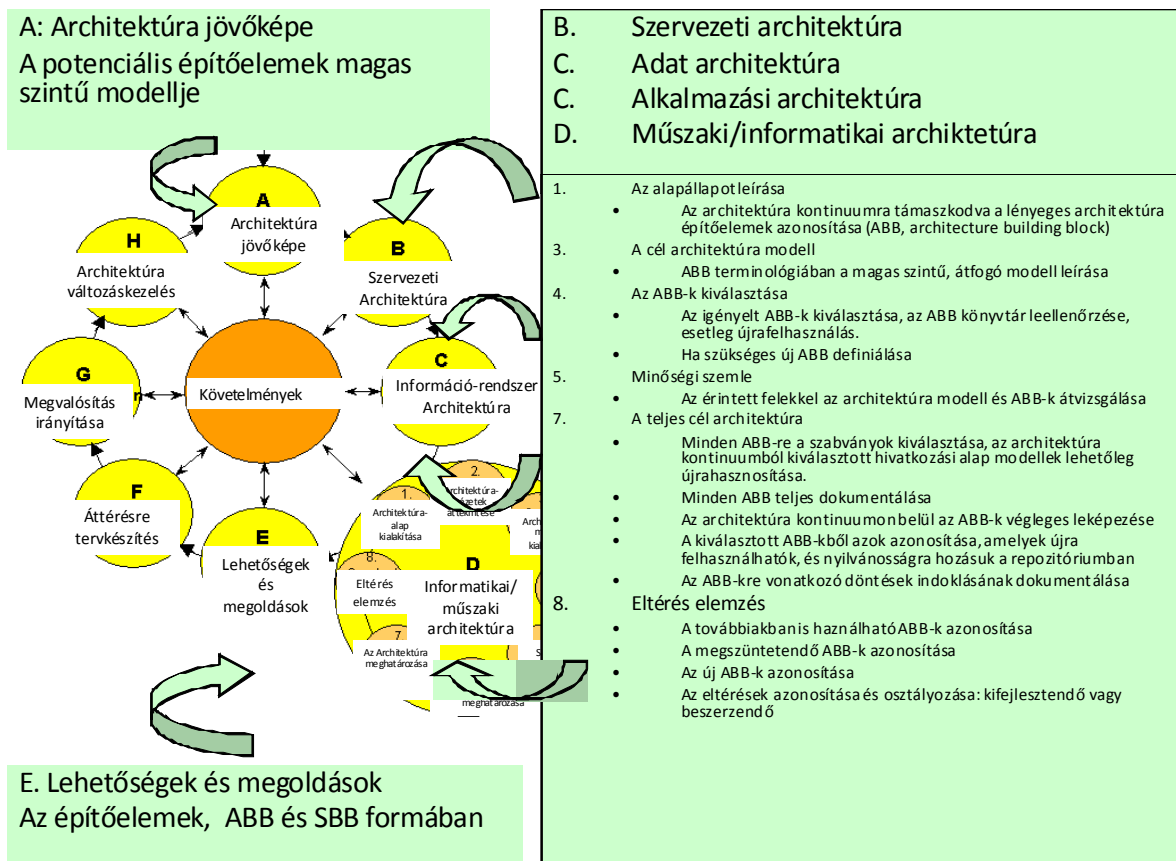
5.1.2 A- Architektúra jövőképe és célkitűzései

Az A szakaszban kell megszerezni a szervezet felső vezetésétől és a középvezetéstől is az architektúra fejlesztés iránti támogatását. Az architektúra fejlesztése, kialakítása ciklikus folyamat, az egyes ciklusokban az architektúra fejlesztési folyamat saját maga is evolúciósan fejlődik, a folyamatjavításhoz is szükség van a vezetés támogatására.

A szervezet elfogadott irányelveit, szervezeti céljait, és a stratégiát meghatározó tényezők megfogalmazásának helyességét, aktualitását ellenőrizni kell. Utána a pillanatnyi architektúra tervezéssel kapcsolatos erőfeszítések kiterjedését, szervezeti határait, a szervezeti architektúra meta-modelljéből kiválasztott komponensek célszerűségét és azok rangsorát meg kell határozni (43. ábra). Az architektúra kialakításban a lényeges érintett feleket, aggályait, célkitűzéseiket meg kell ismerni. Meg kell fogalmazni azokat a kulcsfontosságú szervezeti követelményeket, amelyekkel foglalkozni kell az architektúra tervezés során, továbbá a tervezési folyamat a peremfeltételeit és korlátait.

Az előkészítő munka után egy olyan architektúra jövőképet kell létrehozni, amely az előbbiekben megfogalmazott követelményeket és peremfeltételeket lehetőleg kielégíti, majd ezek után az így kialakított koncepciót hivatalosan jóvá kell hagyatni. Az architektúra tervezésre, mint projektre a többi párhuzamoson folyó szervezeti architektúra fejlesztési projektre, azok aktuális szakaszára, ciklusára gyakorolt valamint az azok által a szóban forgó projekten kiváltott hatásokat figyelembe kell venni.

5.1 TOGAF: Architektúra fejlesztési módszer



31. ábra Az alap architektúra építőelemek (ABB) és szabványos építőelemek (SBB) formájában történő használata

5.1.3 B-Szervezeti Architektúra

A *B szakaszban* a jelenlegi architektúra alapvető jellegzetességeit írják le, amelyre alapozva egy olyan szervezeti architektúra kialakítása a cél, amely leírja:

- Az architektúrával kapcsolatos termék és/vagy szolgáltatási stratégiát;
 - A szervezeti;
 - A funkcionális;
 - A folyamat;
 - Információ tekintetében.
- A szervezet és környezetének földrajzi elhelyezkedéséből, elosztottságából származó sajátosságokat is rögzítenie kell.

A *cél architektúra* megfogalmazásának a jelenlegi szervezeti irányelveken, célokon és a stratégiai tényezőknél kell alapulnia. A jelenlegi architektúra-alapvonalai és a cél szervezeti architektúra eltérésének elemzését el kell végezni, az eltéréseket dokumentálni kell. A lehet-

séges nézőpontok közül az architektúra tervezőnek azokat a lényeges architektúra nézőpontokat kell kiválasztania, amelyek az érintett felek által megfogalmazott aggályokra, problémákra, kérdésekre válaszokat adnak. Ebben a szakaszban kell kijelölni a kiválasztott architektúra nézőpontokhoz szükséges eszközöket és technikákat.

5.1.4 C - Információ-rendszer Architektúra

Ennek a szakasznak az a célja, hogy kialakítson egy olyan cél architektúrát, amely (a projekt kiterjedésétől függően) vagy az adat vagy az alkalmazási rendszerek tartományára vagy mindkettőre vonatkozzon. A szóba jövő, a vizsgálatba bevonandó szervezeti folyamatok azokra korlátozódnak, amelyeket az informatika támogat, és kapcsolatot jelentenek az informatika által támogatott folyamatok és az informatika által nem támogatott folyamatok között.

5.1.4.1 C - Információ-rendszer Architektúra - Adat

Ebben a szakaszban a cél itt az, hogy a szervezet számára szükséges jelentős adattípusokat és adatforrásokat meghatározzák úgy, hogy az érintett felek számára érthető, teljes, ellentmondásmentes és stabil legyen ez a fogalmi szintű adatszerkezet. Fontos szem előtt tartani azt, hogy ez nem adatbázis tervezés. A fogalmilag fontos adat entitások meghatározása a cél, azaz sem nem logikai sem nem fizikai adattárolás tervezése a feladat. A létező adatállományokhoz és adatbázisokhoz a kapcsolatokat esetleg feltárják és rögzítik, a potenciális javítási lehetőségekre rámutathatnak, de nem mennek le adatbázis tervezési szintig, részletekig.

5.1.4.2 C - Információ-rendszer Architektúra - Alkalmazások

Ennek a lépésnek a célja, hogy a szervezet adatfeldolgozási tevékenységének és a szervezetnek magának támogatására szolgáló jelentős alkalmazási rendszereket egységes keretben leírják. De ez a lépés sem tekinthető alkalmazási rendszertervezésnek. Ennek a lépésnek az a célja, hogy a szervezet szempontjából lényeges alkalmazási rendszereket gyűjtse össze, és azt rendszerezze, hogy vajon az alkalmazási rendszernek mit kell az adatkezelés és az adatok megjelenítése területén, a szervezeten belül, az emberek, és egyéb számítógépes szereplő / aktorok számára nyújtania. Az alkalmazásokat nem, mint számítógéprendszereket, hanem mint olyan szolgáltatási, funkcionális képességeket kell felfogni és leírni, amelyek az adat architektúra adat objektumait kezelik és a szervezeti architektúra szervezeti funkcióit támogatják.

5.1 TOGAF: Architektúra fejlesztési módszer

Az alkalmazások és szolgáltatási képességeket az egyes, speciális technológiai megoldásokra történő hivatkozás nélkül írják le.

Az alkalmazási rendszerek időben viszonylag stabilak és nem változnak, míg azonban az a technológia, amelyen megvalósították időben változik annak megfelelően, hogy éppen milyen technológia áll rendelkezésre és hogyan változnak a szervezet szükségletei.

5.1.5 D- Informatikai/ műszaki architektúra – Kimenetek (Outputs)

Az architektúra tervezés munkafadatainak meghatározása (ha szükséges aktualizálják).

A műszaki/informatikai alapállapot leírása

A helyesség szempontjából leellenőrzött műszaki/ informatikai alapelvek, illetve az új műszaki/ informatikai alapelvek (ha itt hozzák létre)

A műszaki/informatikai architektúra jelentés elkészítése, amely összegzi azt, hogy mit csináltak és a kulcsfontosságú feltárt tényeket.

- A célként megfogalmazott műszaki/informatikai architektúra 1. verzióját.
- A műszaki/informatikai architektúra –eltérés jelentés
- Az érintett felek aggályainak kezelésére szolgáló architektúra nézőpontok
- A kiválasztott nézőpontokhoz tartozó nézetek.

5.1.6 E- Lehetőségek és megoldások

Ennek a szakasznak a célja:

A különböző cél architektúrák kialakítása során ki kell értékelni és ki kell választani azokat a megvalósítási lehetőségeket, amelyek szóba jöhetnek

Például, fejlesztés kontra beszerzés kontra újrafelhasználás, és az ezeken belül szóba kerülő további lehetőségek

Az olyan változtatáshoz, a megvalósítandó projektekhez és a legfelső szintű munkafeladat csomagokhoz – amelyek a jelenlegi környezetből a cél környezetbe vezetnek át - kapcsolódó stratégiai tényezőket meg kell határozni.

A különböző projektek közötti függőségeket, a költségeket és hasznokat fel kell tárni és értékelni kell.

Ennek alapján egy átfogó kivitelezési tervet, migrálási stratégiát és egy részletes kivitelezési tervet kell megfogalmazni.

5.1.7 F - Áttérésre tervekészítés (migrálásra)

Ennek a szakasznak a célja az , hogy a különböző megvalósítási projekteket rangsorba rendezze:

A különböző áttérési (migrálási) projektek közötti függőségeket, a költségeket és hasznokat fel kell tárni és értékelni kell.

A rangsorba rendezett projektek listája részletes kivitelezési és migrálási terv alapját fogja alkotni.

A szakasz kulcs lépései:

A projektek rangsorolása

Az erőforrás igények és rendelkezésre állásukra becslés készítése.

A különböző áttérési (migrálási) projektek költség/haszon elemzésének elvégzése

Kockázat értékelés végrehajtása

Kivitelezési tervek előállítása idő dimenzióra vonatkoztatva (roadmap (time-lined))

Az áttérési (migrálási) tev dokumentálása

5.1.8 G- Kivitelezés / megvalósítás irányítása

A szakasz célja:

Az egyes megvalósítási projektekre ajánlások, javaslatok kidolgozása

Egy architektúra megállapodás létrehozása a teljes architektúra kivitelezés, telepítés, üzembe helyezés átfogó irányítására.

Az architektúra kivitelezése telepítése, üzembe helyezése során az irányítási funkciók és feladatok ellátása.

A megfelelőség garantálása az előírt architektúrának az architektúra megvalósítási és egyéb projektek esetében.

5.1.9 H- Architektúra változáskezelés

A szakasz célja egy architektúra változáskezelési folyamat felállítása az új szervezeti architektúra alapállapotára vonatkoztatva, amelyet „Kivitelezés / megvalósítás irányítása” szakasz hoztak létre és fejeztek be.

Ennek a folyamatnak tipikusan gondoskodnia kell az olyan dolgok folyamatos nyomon követéséről mint például az új technológiai fejlemények, a szervezeti / üzleti környezetben bekö-

vetkező változások és annak eldöntéséről, hogy vajon szükség van-e egy hivatalosan kezdeményezett új architektúra továbbfejlesztési ciklusra.

Ebben a szakaszban van lehetőség kezdeti szakaszban kialakított keretrendszer és az alapelvek megváltoztatására is

A szakasz kulcs lépései:

- A technológiai változások folyamatos nyomon követése
- A szervezeti/üzleti változások folyamatos nyomon követése
- A szervezet helyzetében bekövetkező változások és fejlődés értékelése abból a szempontból, hogy vajon kell tenni valamit.
- Az architektúra tervezési testületnek (vagy bármely más irányító testületnek) kell döntést hoznia a változások kezelése ügyében (mind a technológiai mind a szervezeti/üzleti változások tekintetében).

5.1.10 Architektúra fejlesztési módszer (Architecture Development Method (ADM)) – Architektúra követelmény kezelés

Célkitűzés:

Egy olyan folyamat kialakítása, amelynek révén a szervezeti architektúrával szemben szabott követelményeket feltárják, rögzítik és a vonatkozó ADM folyamatba beviszik illetve a folyamat állít elő ilyen követelményeket.

Kimenet:

- Egy strukturált követelmény lista, amely tartalmazza
- A megváltoztatott követelményeket
- A követelmények hatás elemzését

A követelmény repozitórium (adatszótár) tartalmazza a jelenlegi követelményeket a cél architektúrára vonatkozólag. Amikor új követelmény keletkezik, vagy meglévő megváltozik, akkor egy „ követelmény hatás elemzés” jön létre, amely azonosítja azokat az ADM szakaszokat, amelyeket felül kell vizsgálni ahhoz, hogy a változásokat kezelni lehessen.

A követelmény hatás elemzési dokumentum több iteráción esik át addig, amíg a végleges változat el nem készül, amely tartalmazza a követelmények összes következményét (pl., költség, idő, szervezeti, üzleti mérőszámok) az architektúra fejlesztésre.

5.2 Architektúra nézetei

A **szervezeti (vállalati, üzleti) architektúra terület** a végfelhasználók az szervezeti (vállalati, üzleti) tervekészítők és közép-és felsővezetés számára nyújt egy ábrázolást.

Az **adat architektúra terület** az adatbázis tervezők, adatbázis adminisztrátorok és a rendszer mérnökök számára nyújt egy áttekintést.

Az **alkalmazási rendszerek architektúrájának területe** a rendszermérnökök és a szoftvertervezők számára ad egy leírást.

A **műszaki, informatikai architektúra területének** címzettjei a beszerzők, a rendszergazdák, rendszeroperátorok, adminisztrátorok és középszintű vezetők.

Az egyes architektúra területeket részletesebben fel lehet bontani, illetve az igényeknek megfelelő nézetek alakíthatók ki.

1. Az **szervezeti (üzleti) architektúra nézet** a végfelhasználó igényeinek, szempontjainak megfogalmazását célozza.
2. A **biztonsági architektúra nézet** a rendszer biztonsági architekturális kérdéseivel foglalkozik.
3. A **szoftvertervező architektúra nézet** az új szoftver rendszerek fejlesztéséhez szükséges ábrázolást jelenti.
4. A **rendszermérnöki architektúra nézet** a szoftver és hardver komponensek működő rendszerre történő összeépítéséhez szükséges leírást jelenti.
5. A **hálózattervezői és kommunikáció architektúra** nézet a hálózati, kommunikációs, távközlési elemek olyan leírását jelenti, amely leegyszerűsíti a hálózati, kommunikációs elemekre a továbbfejlesztési tervek és a műszaki tervek elkészítését.
6. Az **adatáramlási architektúra nézet** az adatok tárolásával, visszakeresésével, feldolgozásával, archiválásával és biztonsági kérdéseit írja le.
7. A **vállalkozás (szervezet) irányítási architektúra** nézet a működési („operations”), adminisztrációval és a rendszer igazgatásával foglalkozik.
8. A **beszerzési architektúra nézet** a „polcra vehető”, dobozos szoftver és hardver termékek beszerzésével kapcsolatos architekturális kérdéseket ábrázolja.

4. Táblázat Az architektúra nézetek hierarchikus rendje, taxonómiája

5.2 Architektúra nézetei

Előzetes	Architektúra jövőkép		
Katalógusok	Táblázatok, mátrixok	Lényeges diagram	
Architektúra elvek katalógusa	Érintett felek táblázata	Értéklánc diagram	Megoldási koncepciók diagramja
Szervezeti (vállalati, üzleti) architektúra terület	Adat architektúra terület	Alkalmazási architektúra terület	Műszaki, informatikai architektúra terület
Katalógusok	Katalógusok	Katalógusok	Katalógusok
Szervezet, szereplő katalógus	Adat entitás, adatkomponens katalógus	Alkalmazás portfólió katalógus Kapcsoló felületek („interface”) katalógus	Műszaki, informatikai szabványok katalógusa Műszaki, informatikai portfólió katalógusa
Hajtóerők, célok, célkítűzések katalógusa	Táblázatok, mátrixok	Táblázatok, mátrixok	Táblázatok, mátrixok
Szerepkör katalógus	Adat entitás, szervezeti (vállalati, üzleti) funkció mátrix	Rendszer, szervezeti egység mátrix	Rendszer, technológia mátrix
Szolgáltatás, funkció katalógus	Rendszer, adat mátrix	Szerepkör, rendszer mátrix	
Helyszín, telephely katalógus		Rendszer, funkció mátrix	
Folyamat, esemény, ellenőrzési mechanizmus (kontrol), termék katalógus		Alkalmazási rendszerek közötti kapcsolatok mátrixa	
Szerződés, mérőszámok katalógusa			
Táblázatok, mátrixok	Lényeges diagram	Lényeges diagram	Lényeges diagram

Szervezeti (vállalati, üzleti) folyamatok közti kölcsönhatás mátrixa	Osztály diagram	Alkalmazások közötti kommunikáció diagram	A környezet és helyszínek, telephelyek diagramja
Szereplő, szerepkör mátrix	Adat elosztási, szétosztási diagram	Alkalmazások és végfelhasználók elhelyezkedésének diagramja	Platform lebontási diagram
		Rendszer használati eset diagram	
Lényeges diagram	Kiegészítő diagram	Kiegészítő diagram	Kiegészítő diagram
Üzlet (szervezet) architektúra lábnyomának diagramja	Adat biztonsági diagram	Vállalkozás (szervezet) igazgatási, irányítási diagram	Feldolgozási diagram
Szolgáltatás, információ diagram	Osztály (objektum) hierarchia diagram	Szervezeti (vállalati, üzleti) folyamat, rendszer megvalósítási diagram	Hálózatba kötött számítástechnikai egységek, hardver diagram
Funkcionális lebontási diagram	Adat migráció diagram	Szoftver tervezői diagram	Hálózati tervező, kommunikáció tervezési diagram
Termék életrajzi diagram	Adat életrajzi diagram	Alkalmazás migráció diagram	
Kiegészítő diagram		Szoftver terjesztési, telepítési, elosztási diagram	
Cél, célkitűzés, szolgáltatás diagram			
Szervezeti (vállalati, üzleti) használati eset diagram			

Szervezeti felépítés diagramja			
Az szervezeti (vállalati, üzleti) folyamatok diagramja	Követelmények kezelése	Lehetőségek és megoldások	
Esemény diagram	Katalógus	Lényeges diagram	
	Követelményjegyzék	Projekt kontextus diagram	Hasznokat ábrázoló diagram

5. Táblázat Az architektúra meta-modellel, ontológiával és kiegészítéseivel kapcsolatos architektúra nézőpontok

	Infrastruktúra kiegészítés
	Szervezet irányítási, igazgatási kiegészítés
	Motivációs, ösztönzési kiegészítés
	Adatmodellezési kiegészítés
	Folyamat modellezési kiegészítés
	Szolgáltatás modellezési kiegészítés
	Központi elemek

5.3 A TOGAF szerinti architektúra modellezés alapú megközelítés

A szervezeti, üzleti szolgáltatások és az informatikai szervezeti funkció (fejlesztés, beszerzés, üzemeltetés stb.) közötti kapcsolatrendszer egységes kapcsolati és fogalmi keretbe a szervezeti architektúra elméleti és módszertani megközelítésével lehet elrendezni.

5.3.1 Fogalmi struktúra legfontosabb elemei

Az egységes fogalmi keretrendszer és modellezési tartalom (meta-modell) legfontosabb fogalmi elemeit és rövid meghatározásait adjuk meg a módszertani koncepció kialakításához, majd a részletes módszertan kialakítása kiindul pontjaként.

Architektúra modellezési lépés	Módszertani termék	Magyarázat
Szervezeti (üzleti) architektúra	Szervezeti egység/Szereplő katalógus	Szervezeti egység/Szereplő katalógus az összes olyan résztvevő, lényeges érintett, érdekelt fél felsorolását tartalmazza, akik valahogyan kapcsolatban

5. Az architektúra nézetei és nézőpontjai a TOGAF szerint

		lépnek az IT rendszerrel, beleértve a (vég)felhasználókat és az egyes IT rendszerekért felelős személyeket
	Szerepkör katalógus	A szerepkör katalógus célja, hogy a szervezeten belüli összes jogosultsági szintet meghatározza. Gyakran az alkalmazási rendszerek biztonságát és magatartását és viselkedését a jogosultsági szintek helyileg felfogott értelmezése szerint határozzák meg. Ez azonban gyakran vezet nagyon bonyolult és váratlan következményekkel járó eredményekre akkor, amikor ezeknek a jogosultságoknak a kombinációja egy felhasználó esetén a (számítógépes) munkahelyén megtestesül.
	Szervezeti (üzleti) szolgáltatás/ Szervezeti (üzleti) funkció	Ennek a katalógusnak a célja, hogy a szervezet olyan funkcionális felbontását mutassa be, amelyet különböző szempontok alapján szűrni és csoportosítani lehet, jelentéseket és lekérdezéseket lehet készíteni. A szervezet funkcionális felosztását mutató diagram kiegészítése. Szervezeti (üzleti) szolgáltatás/ Szervezeti (üzleti) funkció katalógus felhasználható arra, hogy a szervezet képességeit feltárják és megértsék azokat az irányítási szinteket, amelyek a szervezet egyes funkcióihoz kapcsolódnak.
	Telephelyek, helyszínek, fiókok listája	Azoknak a telephelyeknek a felsorolása, ahol a szervezet valamilyen üzleti tevékenységet folytat, ad otthont olyan információ/ informatikai vagyonnak, amely architektúráisan fontos: pl. adatfeldolgozóközpontok, végfelhasználói számítógépes munkahelyek.
	(Szervezeti egységek közötti kölcsönhatás, kölcsönös kapcsolatok mátrixa)	A szervezeti / üzleti funkciók és a szervezeti egységek közötti kapcsolatok mentén létrejövő kölcsönhatások leírására szolgál az egyes szervezetre tekintettel, átfogóan.

5.3 A TOGAF szerinti architektúra modellezés alapú megközelítés

	Szereplő / Szerepkör mátrix	Ennek a mátrixnak az a célja, hogy megmutassa, hogy mely szereplő, mely szerepköröket tölti be, elsősorban a biztonsági és készség, gyakorlat követelmények követésére szolgál.
	Folyamat/Esemény/Ellenőrzési mechanizmus(kontrol)/termék katalógus	<p>Ez a katalógus azoknak a folyamatoknak, eseményeknek a hierarchiáját írja le, amelyek kezdeményezik a folyamatok végrehajtását, a folyamatok eredményeként állnak elő, és azokat az ellenőrzési mechanizmusokat, amelyeket a folyamat végrehajtása során alkalmaznak.</p> <p>Ez a katalógus bármilyen módszertan által előállított folyamatok folyamat ábrázoló diagram kiegészítésére szolgál, és amelyet azért alakítanak ki, hogy a folyamatok és szervezet egészét átfogóan szűrni lehessen, le lehessen kérdezni és jelentéseket lehessen készíteni azért, hogy a folyamatok terjedelmét, közös használatának mértékét, és hatását ki lehessen értékelni.</p> <p>Ez a katalógus lehetővé teszi egy szervezet számára, hogy a folyamatok és részfolyamataik közötti kapcsolatot átlássa, azért, hogy feltárhassa azt, ha egy magas szintű folyamatot módosítanak, akkor annak mi a hatása a teljes összefüggő folyamat láncon.</p>
	Szerződés /indikátor katalógus	<p>Szerződés /indikátor katalógus az összes elfogadott szolgáltatási szerződést tartalmazza és (opcionálisan) a hozzájuk kapcsolódó indikátorokat, mutatószámokat, mérőszámokat, mérhető paramétereiket.</p> <p>Ez az alap lista, amely kiindulópontja azoknak a szolgáltatási szinteknek, amelyeket az egész szervezetben elfogadtak.</p>
Információrendszerek (Adat architektúra)	Entitás (adat) / Adat komponens katalógus	A katalógus célja, hogy feltárja és napra készen tartsa az adatok listáját a szervezet egészére vonatkozóan, beleértve az adat entitásokat és az adat komponenseket. Egy ilyen elfogadott katalógus támogatja az információkezelés, menedzsment definiálását

5. Az architektúra nézetei és nézőpontjai a TOGAF szerint

		és alkalmazását, továbbá az adatkezelés irányelveit és bátorítja az adatok megosztását és újra felhasználását.
	Entitás (adat) / Szervezeti funkció mátrix	<p>Az adat entitások és a szervezeti funkciók közötti kapcsolatok leírására szolgál. A szervezeti funkciókat szervezeti szolgáltatások támogatják, pontosan meghatározott határokkal és szervezeti folyamatok valósítják meg. A következő feltárását segíti:</p> <p>Az adat entitások felelősét (tulajdonosát) meghatározni;</p> <p>A szervezeti szolgáltatások adat és információcsere igényét;</p> <p>Az eltérés elemzés támogatására, annak meghatározására, hogy vajon adat entitás hiányzik és szükség van-e a létrehozásukra.</p> <p>Az adat entitásokat létrehozó, tároló és rájuk hivatkozó rendszerek meghatározására;</p> <p>Az adatok kezelésének igazgatására vonatkozó szervezeti program kialakítása (adat felelős, adat szabványok kidolgozása, amelyek a szervezeti funkciókra vonatkoznak, stb.)</p>
	Információrendszer / adat mátrix	A rendszerek (alkalmazási komponensek) és az adat entitások közötti kapcsolatok leírására szolgál, mely entitásokat mely rendszerek használnák, olvassák, aktualizálják.
Információrendszerek (Alkalmazási architektúra)	Alkalmazások portfóliója (katalógus)	A szervezet összes alkalmazásának felsorolására szolgál, és a lista napra készen tartására
	Rendszer kapcsolatok, kapcsolófelületek katalógusa, (Interface Catalog)	Az alkalmazások közötti kapcsolófelületek felderítésére szolgál, az alkalmazások közötti összefüggések terjedelmének leírására.
	Információrendszer / Szervezeti egység mátrix	A rendszerek és a szervezeti egységek közötti kapcsolatokat írja le. A szervezeti funkciókat a szervezeti egységek hajtják végre. Néhány szervezeti funkciót

5.3 A TOGAF szerinti architektúra modellezés alapú megközelítés

		és szolgáltatást olyan szervezeti egységek hajtják végre, amelyeket IT rendszerek támogatnak. Az alkalmazási rendszer komponens - Szervezeti egység közötti leképezés fontos leírása a szervezetnek.
	Szerepkör / Információrendszer mátrix	A szervezeti szerepkörök és a rendszerek közötti kapcsolatok leírására szolgál
	Információrendszer / Szervezeti funkció	Az információrendszerek és a szervezeti funkciók közötti kapcsolatok leírására szolgál. A szervezeti funkciókat a szervezeti egységek hajtják végre.
	Alkalmazási rendszerek kommunikációs kapcsolata, kölcsönhatási mátrixa	A rendszerek közötti kommunikációs kapcsolatok leírására szolgál
Műszaki / technológiai architektúra	Műszaki szabványok katalógusa	<p>http://www.opengroup.org/sib.htm</p> <p>Műszaki szabványok katalógusa az adott szervezet elfogadott műszaki, informatikai szabványait írja le, megnevezve a technológiát, annak verzióját, a technológia életciklusának szakaszát, az adott technológia megújításának ciklusidejét.</p> <p>A szervezet műszaki irányelveitől és koncepciójától függően, ez a katalógus tartalmazhatja a helytől, telephelytől és szervezeti, üzleti területtől függő egyedi szabványok leírását.</p> <p>Ha a műszaki szabványok gyűjteménye rendelkezésre áll, akkor ezt a szabvány készletet fel kell használni a „Műszaki technológiák portfóliójának katalógusa” kialakításához azért, hogy egy áttekintést kapjanak a műszaki, informatikai szabványoknak történő megfelelés alap vagy kiinduló helyzetéről.</p>
	Műszaki technológiák portfóliójának katalógusa	Ennek a katalógusnak az a célja, hogy azonosítsa, és napra készen tartsa a szervezetben alkalmazott műszaki, informatikai szabványok listáját., beleértve a hardver, az infrastruktúra szoftver, alkalmazási rendszer szoftvere. A technológiai termékek életciklusának, verzió kezelésének közben tartására szolgál

		<p>az elfogadott műszaki technológiák portfóliója, továbbá a szervezet műszaki technológiai szabványainak meghatározására szolgáló definíciós alap kialakítását segíti.</p> <p>Műszaki technológiák portfóliójának katalógusa nyújtja azt az alapot, amelyre a további (a műszaki, informatikai és infrastruktúra elemek leírására szolgáló) mátrixok és diagramok támaszkodhatnak.</p> <p>Az alkalmazott technológiák osztályozásakor célszerű figyelembe venni TOGAF Műszaki modell hivatkozási alapot (TOGAF Technical Reference Model (TRM) 34. ábra Oldalnézet: Műszaki/technológiai hivatkozási modell — a technológiai szintű szolgáltatások), szükség esetén alkalmasan kiterjesztve a modellt azért, hogy a használatban lévő termékekhez illeszkedjen.</p> <p>A műszaki technológiák portfóliója következőket tartalmazza,</p> <ul style="list-style-type: none"> – A platform szolgáltatások; –A logikai technológiai/ műszaki komponensek; –A fizikai technológiai/ műszaki komponensek.
	<p>Információrendszer - technológia mátrix</p>	<p>A szervezeti / üzleti alkalmazási rendszerek (információrendszerek) és technológiai platformok egymáshoz rendelését, leképezését mutatja.</p>
	<p>Telephely katalógus - Környezet és (földrajzi) elhelyezkedés</p>	<p>A telephelyek katalógusa az összes olyan helyszín felsorolását tartalmazza, ahol a szervezet valamilyen tevékenységet kifejt, vagy architektúra szempontjából lényeges (informatikai) vagyon elemeknek ott-hont ad, mint például adat-feldogozó központok és végfelhasználói berendezések.</p> <p>A telephelyek listájának napra készen tartása lehetővé teszi azt, hogy a változtatás kezdeményezések esetében gyorsan meghatározható az érintett</p>

5.3 A TOGAF szerinti architektúra modellezés alapú megközelítés

		<p>helyszínek száma és vizsgálható a tervezet terjedelmének teljessége. Pl. asztali munkaállomások szolgáltatásainak aktualizálása, frissítése a feladat, akkor a munkaállomások telepítési helyeinek pontos nyilvántartására van szükség.</p> <p>Hasonlóan, amikor egy új rendszer megvalósítása folyik, a telephelyek diagramja fontos szerepet játszik abban, hogy a rendszer telepítési tervét úgy alakítsák ki, hogy az tartalmazza mind a végfelhasználók mind az alkalmazások helyét, és azonosítsa a helyszínekhez kapcsolódó ügyeket, mint például nemzetközisítés, szoftver rendszer lokalizációja, időzónák hatása a rendszer rendelkezésre állására, a földrajzi távolság hatása késleltetésre, a hálózat hatása sávszélességre és a rendszer elérhetőségre.</p>
--	--	--

Architektúra modellezési lépés	Diagram jellegű módszertani termék	Magyarázat
Műszaki / technológiai architektúra	Platform hierarchikus lebontási diagram	<p>Azokat a technológiai platformokat írja le, amelyek az információrendszer architektúrát támogatják. Az infrastruktúrát alkotó platformokat írja le a jelentősnek, fontosnak tartott szempontokból. A technológiai platformokat leképezi a kapcsolódó alkalmazási komponensekre, egy adott (szervezeti) funkció vagy szervezeti folyamat területen belülre. Olyan részleteket is lehet rögzíteni, mint pl. a termékek verziója, a CPU-k száma stb.</p> <p>"Hardver</p> <ul style="list-style-type: none"> - Logikai technológiai komponens" - Fizikai technológiai komponens" <p>"Szoftver</p> <ul style="list-style-type: none"> - Logikai technológiai komponens - Fizikai technológiai komponens"
Információrendszerek (Alkalmazási architektúra)	Alkalmazási rendszerek kommunikáció diagramja	Az alkalmazási rendszerek kommunikáció diagramjának célja az, hogy leírja az alkalmazási rendszerek közötti

5. Az architektúra nézetei és nézőpontjai a TOGAF szerint

		<p>kommunikáció minden olyan formáját, amelyek leképezhetők a TOGAF architektúra modelljének (meta-model) entitásaira.</p> <p>Megmutatja az összes alkalmazási rendszer komponensét és közöttük fennálló kapcsoló felületeket.</p> <p>A kapcsolófelületeket össze lehet rendelni <i>adatentitásokkal</i>, ahol ez értelmezhető.</p> <p>Az alkalmazásokat össze lehet kapcsolni az <i>szervezeti (vállalati, üzleti) szolgáltatásokkal</i>, ahol ez értelmezhető</p> <p>Az alkalmazások közötti kommunikációt logikai szinten kell ábrázolni és csak abban az esetben kell érzékeltetni az adatközpontú technológiát, ha az a technológia az architektúra szempontjából lényeges.</p>
<p>Információrendszerek (Alkalmazási architektúra)</p>	<p>Az alkalmazási rendszerek és a végfelhasználók elhelyezkedésének diagramja</p>	<p>Az alkalmazási rendszerek és a végfelhasználók elhelyezkedésének diagramja megmutatja az alkalmazások földrajzi elosztottságát. Arra is fel lehet használni, hogy leírják azt, hogy a végfelhasználók hol és mely alkalmazásokat használják; a kiszolgáló oldali alkalmazási rendszerek elosztottságát; hol érhetők el ezek az alkalmazások vékony kliensen keresztül; az alkalmazási rendszerek fejlesztésének, bevizsgálásának, tesztelésének, kibocsátásának stb. elosztottságát.</p> <p>Ennek a diagramnak az a célja, hogy világosan megjelenítse azokat a telephelyeket, helyszíneket ahonnan a szervezet alkalmazottai, végfelhasználói az alkalmazási rendszerekkel dolgoznak, továbbá az alkalmazási rendszerek infrastruktúrájának, a kiszolgáló oldali részének helyet adó helyszíneket.</p> <p>A diagram lehetővé teszi:</p> <ul style="list-style-type: none"> – A szoftvercsomag példányok számának meghatározására, amely ahhoz szükséges, hogy a földrajzilag elosztott felhasználókat kielégítő mértékben támogassa; – A szoftvercsomagokhoz és egyéb szoftverekhez szükséges licencek típusának és számának meghatározása. – Az informatikai ügyfélszolgálat által nyújtandó támoga-

5.3 A TOGAF szerinti architektúra modellezés alapú megközelítés

		<p>tás szintjének meghatározása és az ügyfélszolgálati pontok helyének kijelölésére;</p> <ul style="list-style-type: none"> – A rendszer felügyeleti, rendszergazdai eszközök kiválasztását, valamint azt az irányítási rendszert, amely szükséges felhasználók, ügyfelek, partnerek kezeléséhez helyben és távolról is. – A szervezet technológiai, műszaki komponenseire a megfelelő tervekészítést, a kiszolgáló (szerver) gépek méretezését és a hálózati sávszélességet, stb. – A rendszerek teljesítményével illetve a teljesítmény és kapacitásméretezéssel kapcsolatos megfontolásokat, és a műszaki, informatikai architektúra megoldások kialakítását.
Információrendszerek (Alkalmazási architektúra)	Információrendszer használati diagram	<p>Információrendszer használati diagram jeleníti meg az információrendszer szolgáltatások fogyasztóit és a szolgáltatásokat nyújtókat. Az alkalmazási információrendszerek szolgáltatásainak a fogyasztói a szervezet szereplői, vagy más alkalmazási rendszerek. Az alkalmazási információrendszer diagram azzal gazdagítja az architektúra leírást, hogy az alkalmazási információrendszerek funkcionális szolgáltatásait is érzékelteti, azt hogyan és mikor használják az adott funkcionális szolgáltatást.</p> <p>Információrendszer használati diagram a szereplők és az alkalmazási információrendszerek közötti információcsere leírására és a leírás helyességének ellenőrzésére szolgál.</p> <p>Ha szükséges a használat mélyebb, részletesebb kibontása, akkor a leírás a funkcionális szolgáltatásoktól a technikai, műszaki részletek felé mozdulhat el.</p>
Információrendszerek (Adat architektúra)	Osztály diagram	A szervezet szempontjából kritikus fontosságú entitások, objektum osztályok és köztük fennálló kapcsolatok ábrázolására szolgál.
Információrendszerek (Adat architektúra)	Adat elosztási, terjesztési diagram	Az adat entitások, szervezeti szolgáltatások és az alkalmazások komponensek közötti kapcsolatot mutatja. Azt mutatja meg, hogy az alkalmazási komponensek részéről fizikailag hogyan valósítják meg a logikai entitásokat.
Információrendszerek (Adat architektúra)	Adat biztonság diagramja	Az adatok a szervezet vagyonát alkotják, az adatbiztonság egyszerűen azt jelenti, hogy a szervezet adatai nem károsodnak és az adathozzáféréseket megfelelően ellenőrzik.

5. Az architektúra nézetei és nézőpontjai a TOGAF szerint

		<p>A diagram azt ábrázolja, hogy mely szereplő (személy, szervezet vagy rendszer) férhet hozzá a szervezet adataihoz. Ez kapcsolat mátrix formájában is ábrázolható.</p> <p>A diagram a törvényi, szabályozási, hatósági előírásoknak történő megfelelés demonstrálására szolgálhat.</p>
Szervezeti (üzleti) architektúra	Szervezeti (üzleti) szolgáltatás/ Információ diagram	A diagram azokat az információkat mutatja, amelyek egy vagy több szervezeti / üzleti szolgáltatás támogatásához szükségesek. A diagram azt mutatja, hogy mely adatokat használja fel a szervezeti szolgáltatás, és melyeket állítja elő, valamint esetleg az információforrásokat is megmutatja.
Szervezeti (üzleti) architektúra	Szervezeti funkciók hierarchikus lebontási diagramja (Functional Decomposition Diagram)	A diagram egy lapon ábrázolja a szervezet olyan képességeit, amelyek lényegesek az architektúra szempontjából. A szervezet képességeinek vizsgálata egy funkcionális nézőpontból lehetővé teszi, hogy elkerüljenek olyan vitákat, amelyek arról szólnak, hogy a szervezet hogyan végzi a tevékenységét.
Szervezeti (üzleti) architektúra	Szervezeti (üzleti) termékek életciklus diagramja (Product Lifecycle Diagram)	A szervezet kulcsfontosságú „entitásainak” életciklusának megértésére szolgál. A termék életciklus megértése egyre fontosabb, a környezetvédelmi, törvényhozási és egyéb szabályozási változásokra tekintettel, ahol a terméket létrejöttétől a megszűnéséig nyomon kell követni. Ugyancsak azok a szervezetek, amelyek olyan termékekkel foglalkoznak, melyek személyi adatokat, bizalmas (kényes) adatokat tartalmaznak, pontosan érteniük kell a termék életciklusát azért, hogy az ellenőrzési mechanizmusok, folyamatok, és eljárások során a tervezésben az vonatkozó előírásokat szigorúan érvényesítsék.

5.4 Nézetek és nézőpontok

Példa egy nézőpont jellemzésére. Az üzleti területek nézőpont.

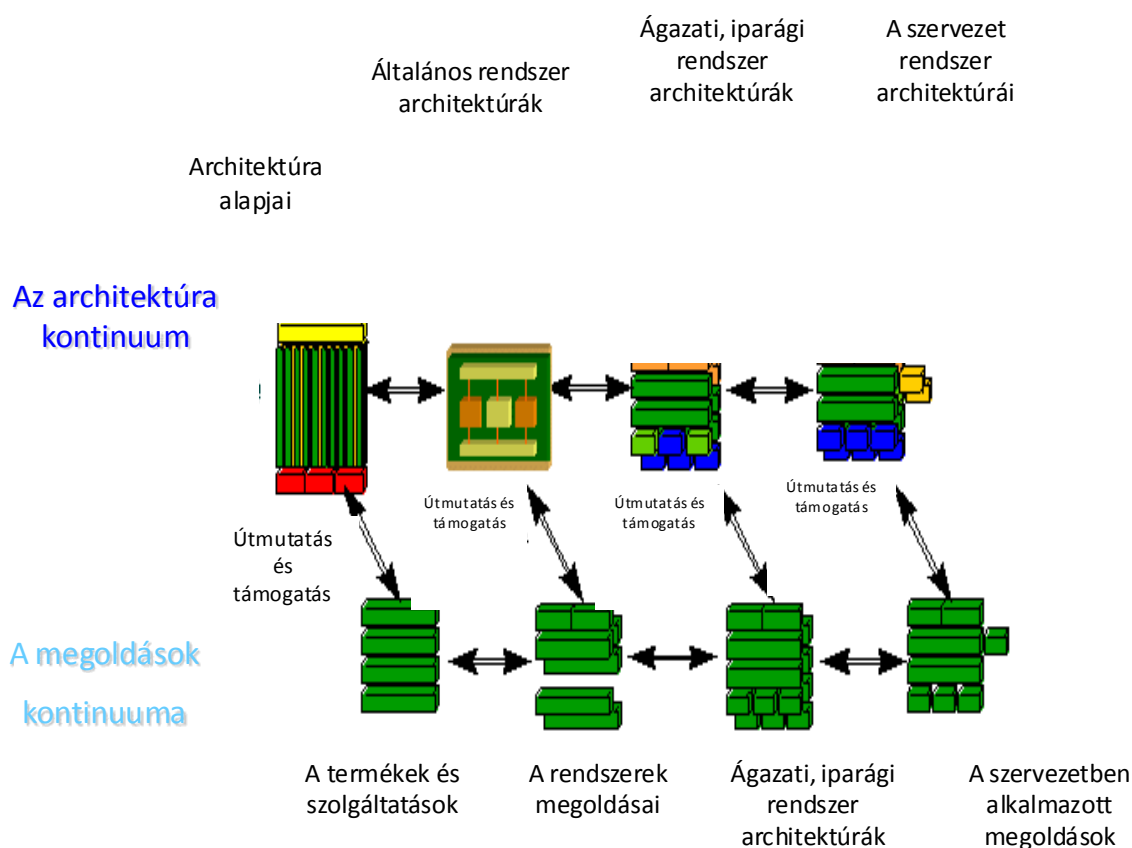
Nézőpont alkotórésze	Leírás
Érintett felek	Igazgatótanács, Ügyvezető igazgató
Érdeklődési, érdekeltégi terület	Megmutatja a magas szintű kapcsolatokat a földrajzi helyszínek és az szervezeti (vállalati, üzleti) funkciók között.

5.4 Nézetek és nézőpontok

Modellezési technika, eljárás	Egymásba ágyazott téglalapokban diagramok. Külső téglalap = helyszínek, belső téglalapok = szervezeti (vállalati, üzleti) funkciók. Az egymásba ágyazás szemantikai tartalma = szervezeti (vállalati, üzleti) funkciót a jelölt helyszínen hajtják végre.

5.4.1 A legfontosabb alapfogalmak meghatározása

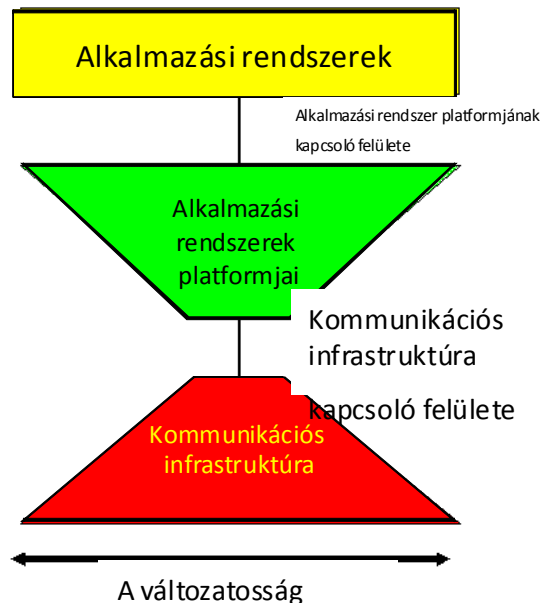
- **Szereplő:** Személy, kapcsolóf felület (interface), szervezeti egység, vagy (információ)rendszer, amely ugyan kívül esik az architektúra modellezésen, de kapcsolatba lép vele..
- **Alkalmazási rendszer komponens :** Az alkalmazási rendszer funkcionalitása egy részének összefogása egy egységbe, amely összhangban áll megvalósított rendszer szerkezetével.
- **Szervezeti (üzleti) szolgáltatás:** A szervezeti (üzleti) képességeket támogatja egy pontosan meghatározott kapcsoló felületen („interface”) keresztül és kifejezetten egy adott szervezet (szervezeti egység) által irányított.



32. ábra A szervezeti architektúra kontinuum

- **Entitás:** Az adatok egy olyan egységbe foglalása, amely a szervezet szakterületének szakértője számára egyértelműen elkülöníthető fogalomként jelenik meg. Az adat entitásokat alkalmazási rendszerekhez, repozitóriumokhoz, és szolgáltatásokhoz lehet kapcsolni, és a megvalósítási igények szerint lehet strukturálni.
- **Szervezeti funkció:** Olyan szervezeti (üzleti) képességeket nyújt, amelyek szorosan kötődnek egy szervezethez, szervezeti egységhez, azonban ezt a szervezeti funkciót kifejezetten nem ez a szervezet, szervezeti egység irányítja.
- **Szervezet (Szervezeti egység):** Önálló, erőforrásait magában foglaló egység, hierarchikus vezetői felelősséggel, hatás- és feladatkörrel, célokkal, célkitűzésekkel és intézkedésekkel.
- **Platform szolgáltatás :** Műszaki képesség, amely szükséges ahhoz, hogy a szervezeti működés feltételeit megteremtő infrastruktúráról gondoskodjanak, amely az alkalmazási rendszerek által nyújtott szolgáltatásokat támogatja.
- **Szerepkör:** Egy szereplő (aktor) egy adott feladat végrehajtása során egy adott szerepkört ölt magára.

- **Műszaki, informatikai komponens:** A műszaki, informatikai infrastruktúra egyes elemeinek olyan egységbe foglalása, amely a műszaki termékek egy adott osztályának megtestesülését jelentik, vagy egy különleges, speciális műszaki terméket.
- **Folyamat (Process) [ebben a módszertani környezetben] általában valamilyen információáramlást, valaminek a lefolyását írja le**



33. ábra Műszaki modell hivatkozási alapja (Technical Reference Model (TRM))

A *folyamat* szervezeti funkciók, szervezeti szolgáltatások közötti információáramlás, csere, és ennek lefolyása, amelyet nem lehet fizikailag valahová telepíteni. Mindegyik folyamat egy szervezeti funkció végrehajtásának lefolyását írja le, ezért a folyamat létrehozása csak a szervezeti funkción keresztül történhet, amelynek végrehajtását támogatja, azaz egy alkalmazási rendszer valósít meg egy szervezeti funkciót, amelynek van egy folyamata, és nem az alkalmazási rendszer valósítja meg a folyamatot.

- **Szervezeti funkció egy szervezeti egység képességeit írja le minden elképzelhető részletezettségi szinten.**

A „szervezeti funkció” kifejezés egy szervezeti egység képességeit írja le minden elképzelhető részletezettségi szinten, önmagában foglalva olyan fogalmakat, mint például érték lánc, szervezeti folyamatok területe, képesség, üzleti funkció stb. Minden olyan korlátozott számú üzleti funkció összességét egy szervezeti funkcióként lehet leírni.

- **Szervezeti (üzleti) szolgáltatások szervezeti (üzleti) célkitűzéseket támogatnak és olyan részletezettségi szinten definiálják, amely a szervezeti irányítás által megkövetelt részletezettségi szintnek felel meg.**

Szervezeti szolgáltatás egy vagy több szervezeti funkciót ölel fel. A szervezeti szolgáltatás felbontásának finomsága, részletezettsége függ a szervezet által támasztott hangsúlyoktól és fókuszpontoktól (ami általában a szervezeti tényezők, hajtóerők, célok és célkitűzések értelmében fogalmazódik meg.).

A SzOA (*Service Oriented Architecture (SOA)*) terminológiában egy szolgáltatás (azaz egy üzembe helyezhető, telepíthető alkalmazási rendszer funkcionalitás) sokkal közelebb áll egy alkalmazási rendszer szolgáltatáshoz, alkalmazási rendszer komponenshez, vagy műszaki / technológiai komponenshez, amely megvalósít vagy támogat egy szervezeti szolgáltatást.

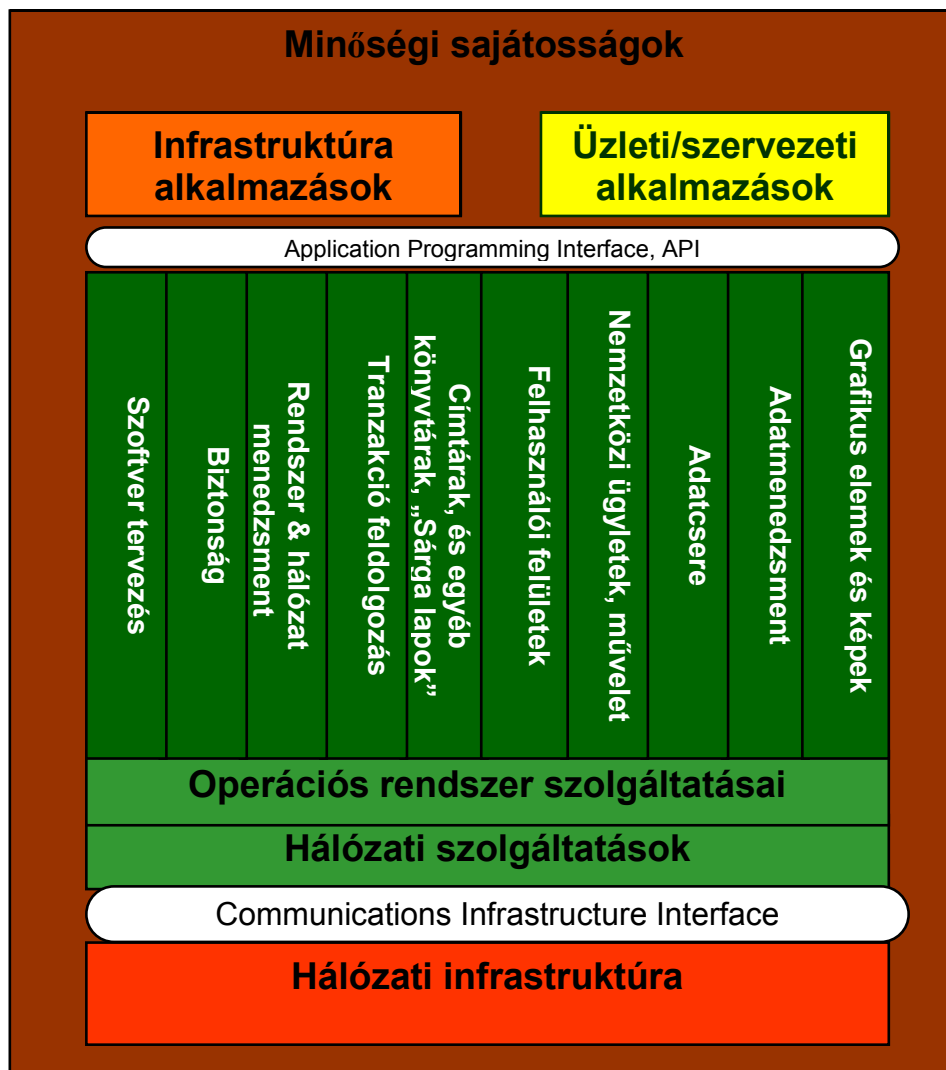
- **Szervezeti szolgáltatásokat az egyes alkalmazási rendszer komponensekhez rendelik hozzá.**

Szervezeti szolgáltatásokat olyan szervezeti tevékenységek valósíthatják meg, amelyeknek semmi közük sincs az informatikához, sem nem kapcsolódnak, sem nem támogatottak az informatika részéről. Az olyan szervezeti szolgáltatásokat, amelyeket az informatika támogat, azokat az alkalmazási rendszer komponensekhez rendelik. Az alkalmazási rendszer komponenseket hierarchikusan lebontják, és egy alkalmazási rendszer komponens egy vagy több szervezeti szolgáltatást támogathat. Egy szervezeti szolgáltatást több alkalmazási rendszer komponens támogathat, ez irányítási, vezetési szempontból problematikus lehet. Ez a jelenleg azonban rámutathat egy részletezettségi, felbontási finomsággal összefüggő problémára, nevezetesen lehet, hogy a szervezeti szolgáltatás nagyon durva fel-

bontású („túl nagy”), vagy az alkalmazási rendszer komponensek felbontása túlzottan részletese, nagy finomságú.

- **Az alkalmazási rendszer komponenseket a műszaki / technológiai komponensekre telepítik, rendelik hozzá.**

Egy alkalmazási rendszer komponens megvalósítása a műszaki / informatikai / technológiai komponensek egy halmaza révén történhet meg. Egy alkalmazási rendszer mint például „Emberi erőforrás rendszer ("HR System") több műszaki komponensen valósul meg, ilyen komponensek: hardver elemek, alkalmazás kiszolgáló (server) szoftverek, és alkalmazási rendszer szolgáltatások.



34. ábra Oldalnézet: Műszaki/technológiai hivatkozási modell — a technológiai szintű szolgáltatások

modell — a technológiai szintű szolgáltatások

Infrastruktúra alkalmazások (34. ábra):

- Elektronikus fizetések, pénzáttalások
- E-mail kliens
- Elektronikus nyilvánosságra hozatal, előfizetés (Publish and subscribe)
- Intelligens ágensek
- Naptár, ütemezése
- Csoportmunka (Groupware services)
- Munkafolyamat (Workflow services)
- Táblázatkezelő (Spreadsheets)
- Bemutató készítő (Presentation software)
- Dokumentum szerkesztés, bemutatás (Document editing and presentation)
- Rendszeradminisztráció hálózati adminisztráció, felügyelet, monitoring stb.
- Szoftvertervező, fejlesztő eszközök, telepítést támogatás

Műszaki/technológiai hivatkozási modell a szolgáltatások részletes taxonómiáját tartalmazza, Mindegyik szolgáltatás kategória kiterjedését, terjedelmét leírja.

- Azonosítja a rendszer képességeit („minőségi sajátosságokat”), pl. :
 - Nemzetköziesség, határnélküliség;
 - Biztonság;
 - Rendszergazdálkodás, menedzsment.

A nyílt iparági szabványok adatbázisa tartalmazza:

- Az Open Group által támogatott összes szabványt;
- A tartalmat az Open Group konszenzus teremtő eljárása határozza meg;
- A strukturálása „Műszaki modell hivatkozási alapja” (TOGAF Technical Reference Model) szerint történik.
 - Rendszeresen aktualizálják;

5.4 Nézetek és nézőpontok

- A Világhálón elérhető¹⁷.

Források adat- és ismeretbázisában található meg az architektúra fejlesztési módszer (TOGAF Architecture Development Method) alkalmazási előírásainak forrás anyagai itt található meg, pl.

- Architektúra leíró nyelv, ADML (Architecture Development Method Language);
 - Architektúra megfelelési minőségi szemlék;
 - Architektúra elvek;
 - Architektúra nézetek;
 - Szervezeti / üzleti forogatókönyvek;
 - Esettanulmányok.

Az informatika irányításának stratégiájára vonatkozó elveket is tartalmazza.

6. Táblázat Egy alkalmazási rendszer architekturális leírásának példa sémája. Információrendszer architektúra szempontjából történő jellemzése.

5. Az architektúra nézetei és nézőpontjai a TOGAF szerint

Információ rendszer: alkalmazási információrendszer			
Alkalmazási rendszer típusa: Szervezeti szintű alkalmazás			
Platform szolgáltatás kategória	Alkategória	Szükséges-e	Megvalósító/támogató technológia
Adatcsere	Általános dokumentum adattípusok és konvertálási szolgáltatások (szöveg [text], kép, numerikus, speciális betű / karakter típusok, logikai és megjelenítési szerkezete a dokumentumnak)		
	Grafikus adatcsere szolgáltatások (vektoros, raszteres vagy bitkép)		
	Specializált elektronikus adatcsere szolgáltatás (orvosi, könyvtári, fogorvosi, biztosítási, olajipari stb.- vertikális piacok sajátos adatcsere formátumai.)		
	Elektronikus adatcsere szolgáltatások (papír-mentes e-kereskedelmi, e-igazgatási környezet). Pl:beszállító keresés, kiválasztás, szerződés kötés, termék adatok, kiszállítás, továbbítás, érkeztetés; vám; fizetési információk, készletgazdálkodás, karbantartás; adózással összefüggő adatok, biztosításokkal összefüggő adatok.		
	Fax szolgáltatások (fax készítése, átolvasása, továbbítása, fax képek fogadása)		
	Alapvető grafikus adatformátumok (TIFF,JPEG, GIF, CGM)		
	Szövegszerkesztésre szolgáló funkciók (létrehozás, szerkesztés, összevonás és formázás)		
	Dokumentum szerkesztésre szolgáló funkciók (létrehozás, szerkesztés, összevonás és formázás, beleértve a grafikus képek, hangok, stilizált szövegek kezelését is, stb.)		
	Nyomdai külalaknak megfelelő dokumentumok előállítás, publikálási célokra (magas grafikus, szín megjelenítése, fejlett formázási szolgáltatások, stb.)		
	Videó kezelés (videó rögzítése, szerkesztése, összeállítása, tömörítése, és kibontása, pl. MPEG stb.)		
	Hangfelvétel kezelés (Audio) (rögzítése, szerkesztése, összeállítása, tömörítése, és kibontása,)		
	Multimédia szolgáltatások, funkciók (ebbe a következők is beleértendők : szkennelés, szkennelt dokumentumok tárolása, optikai adathordozókon, tömörítés stb.)		
	Média szinkronizáció (videó, audio stream)		
	Információ megjelenítés és szétosztás, terítés (kötegelt és interaktív alkalmazások. Információ maszkolás, tárolás, archiválás, rangsorolás, korlátozás, újraállítás. A szervezeti, üzleti területek által előállított olyan információ kezelése, amelyeket a későbbi feldolgozásban kerülnek további vezérlési és ellenőrzési mechanizmusok hatáskörébe.		
	Hipertext, hipervivatkozás (keresés, böngészés, hipertext kapcsolatok/linkek, multimédia információk megjelenítése)		
Adatkezelés	Repozitórium/Adatszótár szolgáltatások (adat adminisztráció, adatbázis-tervezők, meta-adatok kezelése. Belső és külső adatábrázolás, integritási és biztonsági szabályok.)		

5.4 Nézetek és nézőpontok

Információ rendszer: alkalmazási információrendszer			
Alkalmazási rendszer típusa: Szervezeti szintű alkalmazás			
Platform szolgáltatás kategória	Alkategória	Szükséges-e	Megvalósító/támogató technológia
	Adatbázis kezelő rendszer		
	Objektum-orientált adatbázis kezelő rendszer (OODBMS)		
	Adatállomány („File”) kezelő rendszerek(pl. index szekvenciális (ISAM) vagy véletlenszerű elérésű adatállomány szervezés (hashed random access). Egyszerű, alap adatállomány szerkezet, adatállomány könyvtár szerkezet.		
	Lekérdező szolgáltatások (Query processing) (4-dik generációs nyelvek)		
	Képernyő generáló funkciók (adat visszakeresés, megjelenítés, és aktualizálás).[Screen Generation]		
	Jelentés előállító, generáló funkciók (Report Generation)		
	Hálózati, konkurens, párhuzamos adatbázis elérést lehetővé funkciók		
	Adattárház, adatpiac funkciók (OLAP, online analytical processing)		
Grafikus és digitális képszolgáltatások	Grafikus objektumkezelés		
	Rajzolás (képek manipulálására szolgáló eszközök, GKS, PEX, PHIGS, OpenGL.) Rajzolás (képek manipulálására szolgáló eszközök, GKS, PEX, PHIGS, OpenGL.)		
Nemzetközi működést lehetővé tevő szolgáltatások	Betű, karakter készletek és adatábrázolási szolgáltatások (Kódrendszerek)		
	Helyi konvenciók (dátum, idő formátum, valuta, számbábrázolás, tizedes tört)		
	A helyi nyelv támogatása (Üzenetek, menük, űrlapok, on-line dokumentumok. Billentyűzet)		
Címtár és szolgáltatás lokalizáció funkció	Címtárszolgáltatások („kliensek”[ember, számítógép] számára erőforrások nyilvántartása(elektronikus posta cím, nevek, tanúsítványok, nyomtatók, web lapok stb.),		
	Speciális célú név szolgáltatások. Objektumok, amelyeket a névtartományokba szervezhetnek: File (adatállomány) rendszerek; Biztonsági adatbázisok(Security databases) Kiszolgálási sorok (Process queues)		
	Szolgáltatás feltalálási helyének megkeresésére szolgáló funkciók (Sárga /Arany Lapok, „Yellow Pages”)		
	Regisztrációs szolgáltatás (szolgáltatások azonosítójának, leírásának, nyújtott erőforrások, és az elérhetőségének módjára vonatkozó információk feljegyzése.)		
	Információs-zűrés szolgáltatások (előre meghatározott szempontok szerint a hasznos információ kiválasztása)		

5. Az architektúra nézetei és nézőpontjai a TOGAF szerint

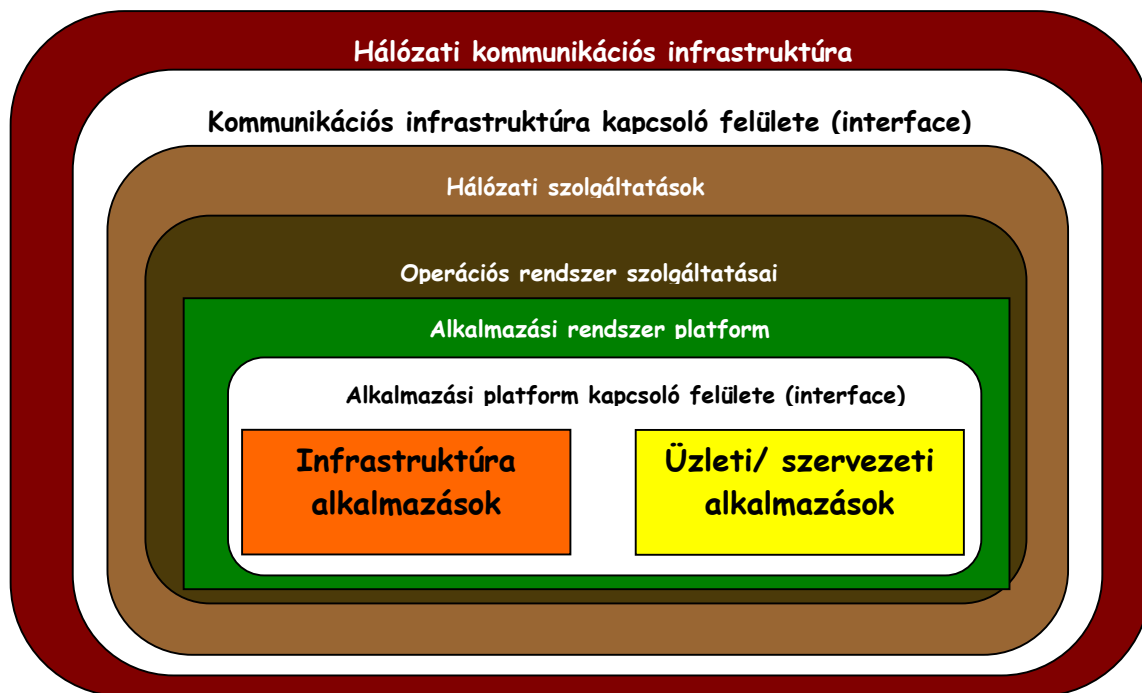
Információ rendszer: alkalmazási információrendszer			
Alkalmazási rendszer típusa: Szervezeti szintű alkalmazás			
Platform szolgáltatás kategória	Alkategória	Szükséges-e	Megvalósító/támogató technológia
	<i>Számlázási szolgáltatások (Számla/bejegyzés létrehozása, aktualizálása, egyenleg számítás, tételek kezelése, lezárása, engedmények, egyeztetés, forgalom, időalapú, erőforrás használat alapú, vagy egyéb specifikus elszámolások)</i>		
Hálózati szolgáltatások	<i>Adat kommunikáció (magas szintű: file transfer, remote login, remote process execution; alacsony szintű: sockets API)</i>		
	<i>Elektronikus levél</i>		
	<i>Elosztott adatszolgáltatások (transzparens címzés, adat replikáció, adatállomány reteszelés (locking), naplózás))</i>		
	<i>Elosztott névszolgáltatás (X.500, hálózati navigáció, stb.)</i>		
	<i>Elosztott időszolgáltatás (szinkronizált időszolgáltatás, különböző időzónák között)</i>		
	<i>Távoli eljárások (remote procedure call (RPC))</i>		
	<i>Távyomtatás (Remote print spooling and output distribution services)</i>		
	<i>Telefon szolgáltatások</i>		
	<i>Osztott képernyő (audio teelkonferencia, közös munkaállomás ablakkal)</i>		
	<i>Videó konferencia</i>		
	<i>Műsorszórási funkció</i>		
	<i>Résztevők listájának funkciója (audio, video, telekonferencián részt vétel lehetővé tétele, ki- és belépés stb.)</i>		
Operációs rendszer	<i>Kernel szolgáltatások</i>		
	<i>Parancs értelmező és segédprogramok (Command interpreter, utility services)</i>		
	<i>Kötegelt adatfeldolgozás</i>		
	<i>Adatállomány és könyvtár szinkronizálási szolgáltatások (File and directory synchronization services)</i>		
Szoftvertervezési szolgáltatások	<i>Programozási nyelvek</i>		
	<i>Számítógéppel támogatott szoftver tervezési környezet és eszközök (CASE).</i>		
	<i>Felhasználói felületkészítési szolgáltatások (Graphical user interface (GUI) building services)</i>		
	<i>Szkript nyelvek (Scripting language services)</i>		
	<i>Programozási nyelv és az alkalmazási platform közötti kapcsolatot létrehozó szolgáltatások</i>		
	<i>Futtató környezet</i>		
	<i>Alkalmazási rendszer, platform bináris kapcsolófelülete (interface)</i>		
Tranzakció feldolgozás	<i>Tranzakció kezelése (tranzakció indítása, koordináció, „comit/rollback”, nyomon követés,felügyelet, stb.)</i>		
Felhasználói felü-	<i>Grafikusfelületű ügyfél/kiszolgáló (kliens/szerver) architektúra szolgáltatá-</i>		

5.4 Nézetek és nézőpontok

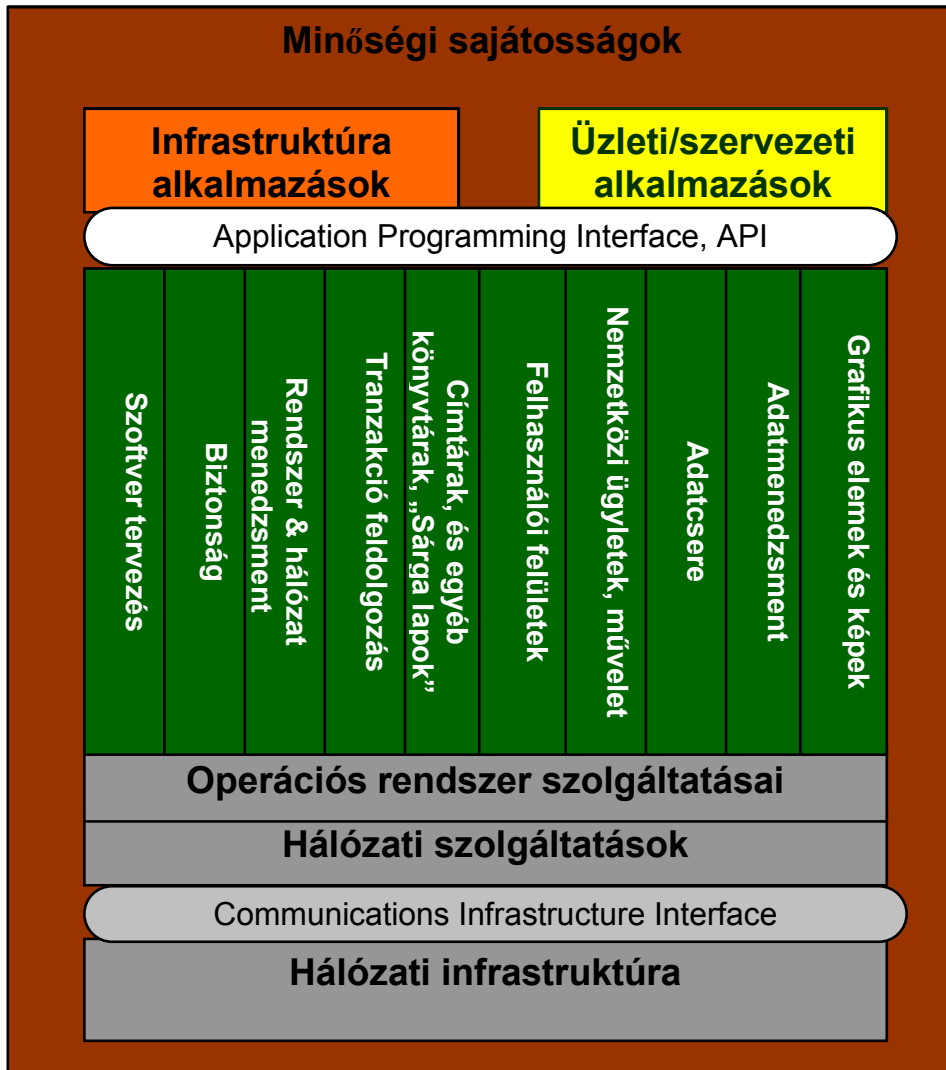
Információ rendszer: alkalmazási információrendszer			
Alkalmazási rendszer típusa: Szervezeti szintű alkalmazás			
Platform szolgáltatás kategória	Alkategória	Szükséges-e	Megvalósító/támogató technológia
letek	sok		
	Megjelenítő (display) felület objektumainak kezelésére szolgáltatások		
	Ablakos kezelő felület szolgáltatásai		
	Ember-gép párbeszéd (dialógus) szolgáltatásai		
	Nyomatási szolgáltatások		
	Számítógép alapú oktatási, tanulás, tájékoztató és segítség nyújtó szolgáltatások.		
	Karakter alapú felhasználói felületek		
Biztonsági szolgáltatások	Személyazonosítási és hitelesítési szolgáltatások		
	Rendszer belépés, beléptetési szolgáltatások és ellenőrzési mechanizmusok		
	Auditálási lehetőségek biztosítása (auditálhatósági, nyomon követhetőségi napló védelme)		
	Hozzáférés ellenőrzés, nyomon követés, kézben tartás		
	Letagadhatatlansági szolgáltatások (Non-repudiation)		
	Biztonságirányítás szolgáltatások (biztonsági rendszer felállítása, beállítása, a biztonsági irányelvekből származó paraméterek ellenőrzése, felügyelete, a felhasználók regisztrálásának, a rendszer erőforrásoknak a kezelése, a rendszergazdai funkciókra vonatkozó korlátozások.		
	Megbízható visszaállítási szolgáltatás (Trusted Recovery services)		
	Kriptográfiai szolgáltatások		
	Bizalmas adatok továbbításának, kommunikációs szolgáltatása (kriptográfiai, algoritmikus védelem, zagyválási funkció (hash), kulcskezelés....)		
Rendszer és hálózat-felügyelet	Felhasználó kezelése (jogosultságok, preferenciák,)		
	Konfigurációkezelés		
	Hálózat felügyelet		
	Teljesítménykezelés		
	Meghibásodások és rendelkezésre állás kezelése		
	Számlázási szolgáltatások (a szolgáltatások kiterhelése, illetve visszatérítések)		
	Biztonsággal kapcsolatos szolgáltatások (a biztonsági irányelvekkel összhangban történő működés ellenőrzése)		
	Nyomatási szolgáltatások (helyi és távoli)		
	Mentés, visszaállítás		
	Merevlemez, háttértár kezelési szolgáltatások		
	Liczenkezelés és felügyeleti szolgáltatások		
	Kapacitáskezelési szolgáltatások		
	Szoftver telepítési, üzembe helyezési szolgáltatások („Software installation“)		
	Hibajegy, hibajelentési szolgáltatások		

5. Az architektúra nézetei és nézőpontjai a TOGAF szerint

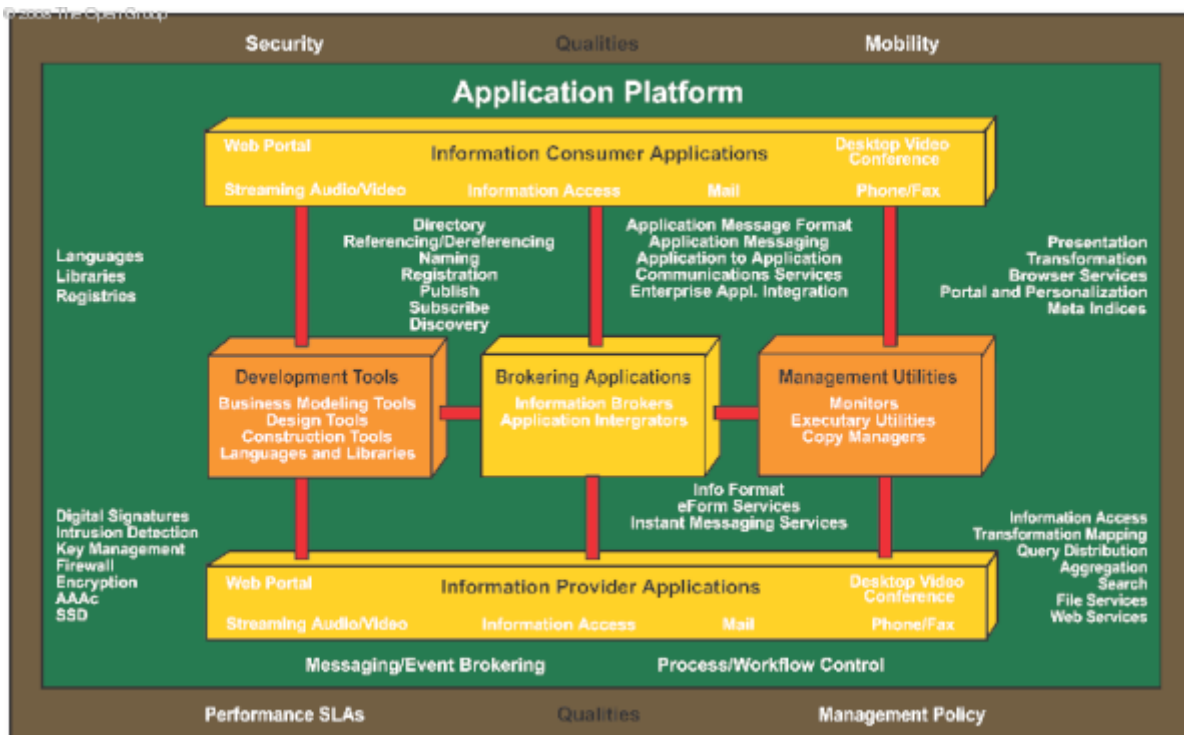
Információ rendszer: alkalmazási információrendszer				
Alkalmazási rendszer típusa: Szervezeti szintű alkalmazás				
Platform szolgáltatás kategória	Alkategória	Szükséges-e	Megvalósító/támogató technológia	
Objektum-orientált rendszer szolgáltatások	Object Request Broker (ORB) Services:			
	Common Object Services:			
Szolgáltatási minőség	Rendelkezésre állás			
	Kezelhetőség, kézben tarthatóság (Manageability,)			
	Szervizelhetőség (Serviceability,)			
	Teljesítmény (Performance,)			
	Megbízhatóság, hibatűrőképesség (Reliability,)			
	Visszaállíthatóság (Recoverability,)			
	Megtalálhatóság, felelhetőség (Locatability,)			
	Értékelhetőség, biztosíték nyújtás (Assurance,)garantálás, szavatolás			
	Biztonság (Security,)			
	Sértetlenség, épség (Integrity,)			
	Szavahihetőség (Credibility,)			
	Használhatóság	Nemzetközi működés (több nyelvűség, multi-kulturálitás)		
Adaptálhatóság	Együttműködési képesség (Interoperability,)			
	Skálázhatóság (Scalability,)			
	Hordozhatóság (Portability,)			
	Bővíthetőség (Extensibility,)			
	Új paradigmákban megjelenő szolgáltatásokhoz (pl. objektum-orientált, komponens-alapú, SOA, stb.) hozzáférés, elérés megvalósításának lehetősége			



35. ábra Felül nézet: Műszaki/technológiai hivatkozási modell — a technológiai szintű szolgáltatások



36. ábra III-RM Integrált infrastruktúra hivatkozási modell szempontjából lényegtelen elemek kiszűrítve



37. ábra Az integrált információ infrastruktúra modell részletes ábrázolása

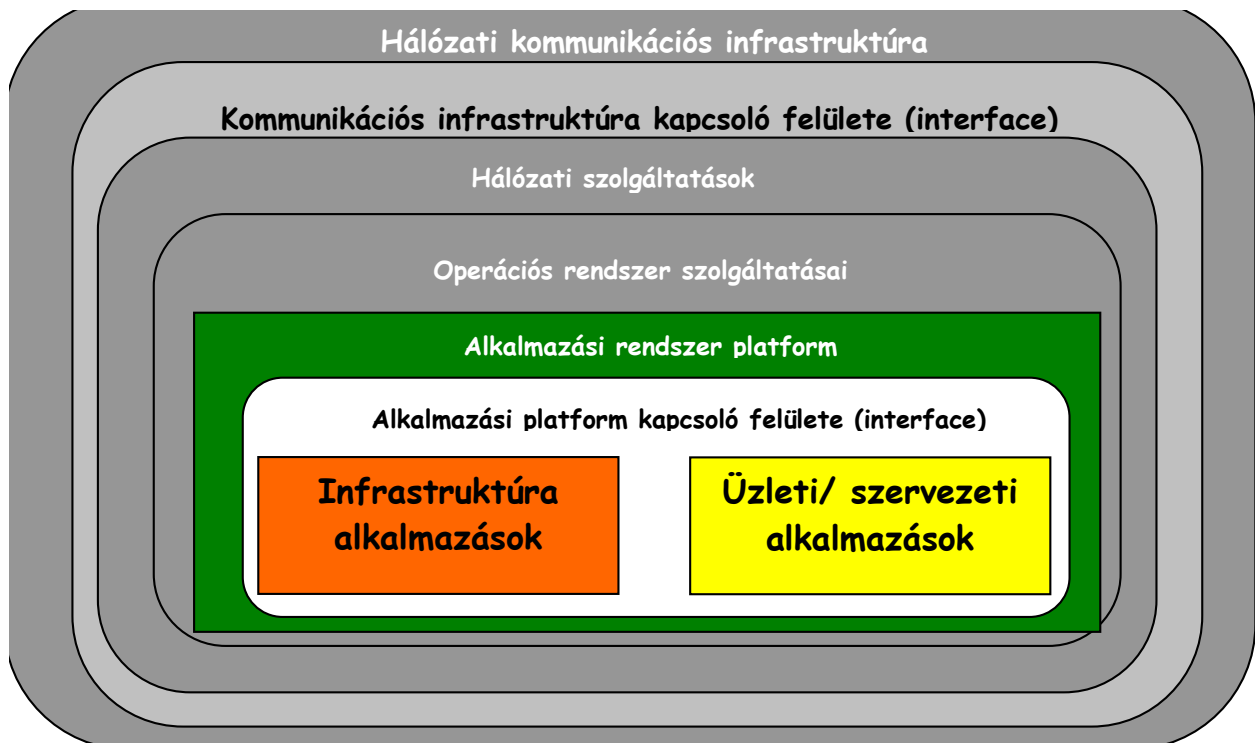
Forrás:TOGAF

- Szervezeti (vállalati, üzleti) alkalmazások, amelyet a sárga téglalap jelöl a magas szintű modellben.(34. ábra Oldalnézet: Műszaki/technológiai hivatkozási modell — a technológiai szintű szolgáltatások). Az szervezeti (vállalati, üzleti) alkalmazásoknak három típusát különbözteti meg a modell:
 - Alkusz / bróker alkalmazások (Brokering Applications), amelyek tetszőleges számú ügyfél („kliens”) alkalmazásoktól származó kérést kezelnek le, és továbbítanak tetszőleges számú „Információ-szolgáltató alkalmazás felé” (Information Provider Applications);
 - Információ-szolgáltató alkalmazások (Information Provider Applications), amelyek az ügyfél („kliens”) alkalmazásoktól származó kérésekre reagálva válaszokat adnak, egyes kiszolgáló centrumok / gépek (server) felügyelete alá tartozó adatok lekérdezése / lekérése révén ;
 - Információ-felhasználó alkalmazások (Information Consumer Applications), amelyek tartalmat szolgáltatnak a végfelhasználók felé, és olyan szolgáltatásokról gondoskodnak, amelyek a magas szintű integrált információ infrastruktúra

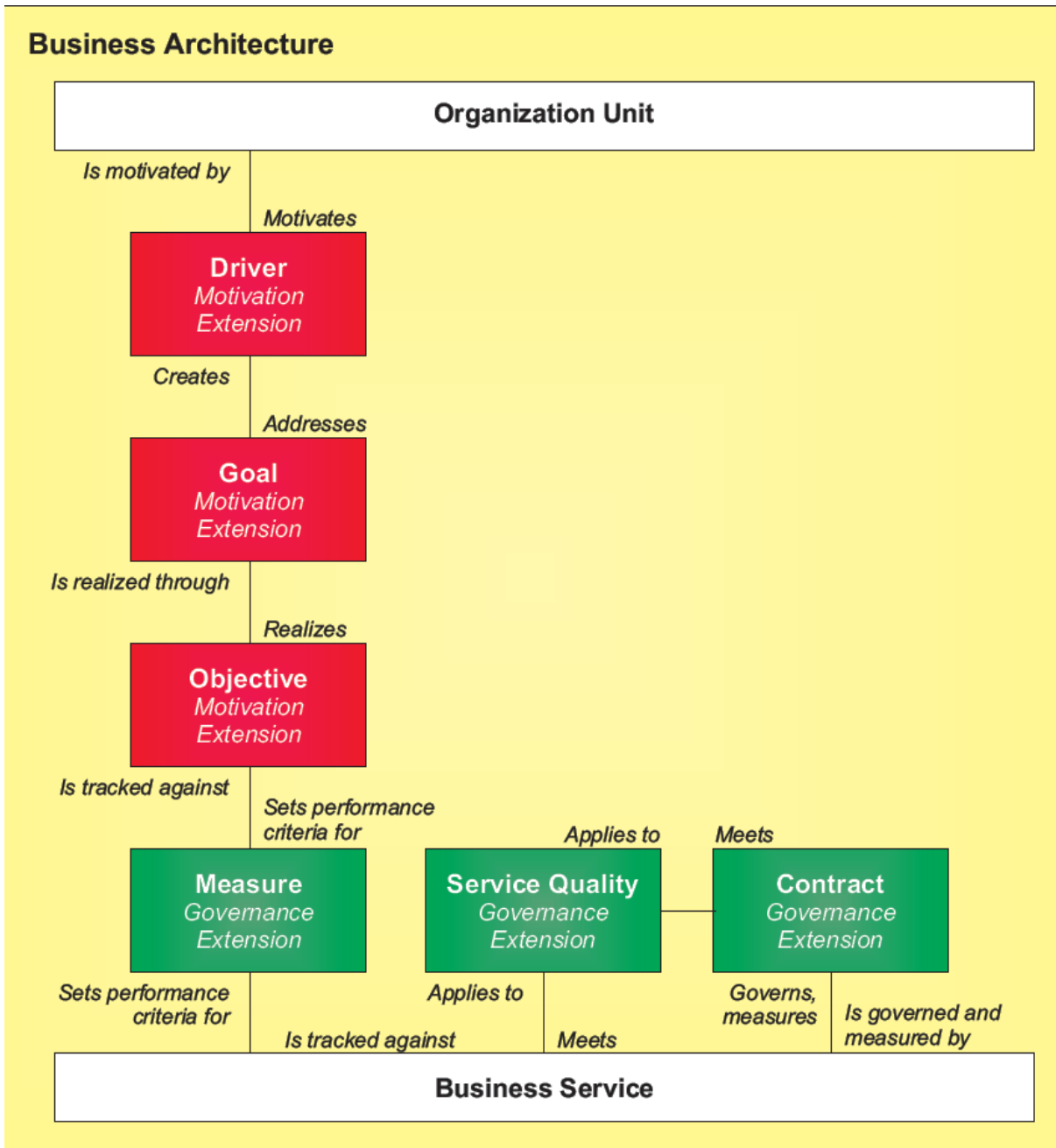
túra hivatkozási alapmodellben található információkra irányuló kéréseket kiszolgálják. (37. ábra Az integrált információ infrastruktúra modell részletes ábrázolása)

- Infrastruktúra alkalmazások, amelyet narancssárga téglalap jelöl a magas szintű modellben. (34. ábra Oldalnézet: Műszaki/technológiai hivatkozási modell — a technológiai szintű szolgáltatások). Két típusa van az infrastruktúra alkalmazásoknak a modellben:
 - Fejlesztő eszközök, amelyek az olyan modellezési, tervezési és kivitelezési képességekről gondoskodnak, amelyek szükségesek az olyan alkalmazások kifejlesztéséhez és telepítéséhez, amelyeknek hozzá kell férniük, el kell érniük, az integrált információ infrastruktúrát olyan módon, amely összhangban áll, konzisztens a körülvevő szervezeti környezetben fennálló szabályokkal és szabványokkal;
 - Rendszermenedzsment, - felügyelet, - adminisztráció eszközei, amelyek gondoskodnak az összes olyan szükséges eszközről, szoftverről, segédprogramról, szolgáltatásról, amelyek segítik megérteni, működtetni, üzemeltetni, hangolni, és kezelni a futó rendszereket, azért, hogy kielégítsék az állandóan változó szervezeti (vállalati, üzleti) környezetből származó igényeket, olyan módon, amely összhangban áll, konzisztens a körülvevő szervezeti környezetben fennálló szabályokkal és szabványokkal;
- Az alkalmazási platform gondoskodik a fent felsorolt, összes alkalmazás számára támogató szolgáltatásokról
 - Például a következő területeken, a szolgáltatás helyének megtalálása, adatállomány könyvtár, munkafolyamatok kezelése, adatmenedzsment, adatcsere stb.,
 - Ezáltal gondoskodik arról a képességről, hogy az adott informatikai környezetben belül megtalálja, elérje és mozgassa az információkat. Ezek a szolgáltatások a műszaki-technológiai hivatkozási modell összes szolgáltatásának egy részhalmazát alkotják, amelyek sötét zöld színnel érzékeltetve jelennek meg a magas szintű modellben (és az alkalmazási platformnak felelnek meg a műszaki-technológiai hivatkozási modellben).

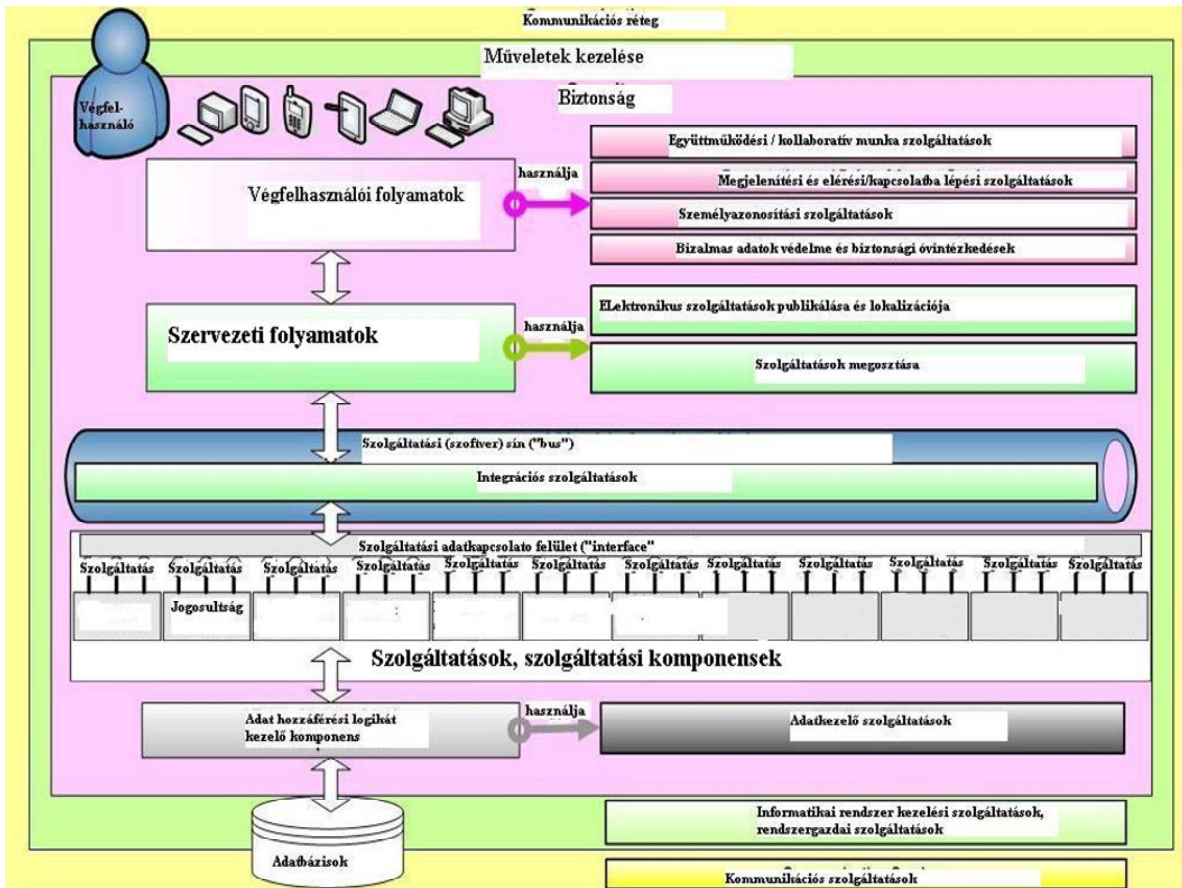
- A kapcsolófelületeket („Interfaces”) a komponensek közötti kommunikációra használják. A kapcsolófelületek, csatolók közé számítjuk az adatformázó, formattáló szolgáltatásokat, a protokollokat, az alkalmazás programozási kapcsolófelületeket (application programming interfaces), a hálózati kapcsolókat (switches), adatérték átalakítókat stb. Az alkalmazási szinten elhelyezkedő komponensek közötti kapcsolófelületeket piros jelzi. A különböző alkalmazási szintű komponensek között és az ezeket támogató alkalmazási platform szolgáltatások közötti kapcsolófelületeket fehérrel érzékelteti (Az API téglalap „34. ábra Oldalnézet: Műszaki/technológiai hivatkozási modell — a technológiai szintű szolgáltatások”-ban)
- Az ábra hátlapjaként megjelenő minőségi tulajdonságokat barna színnel jelzik a magas szintű modellben. (Sztintén barna háttér a „34. ábra Oldalnézet: Műszaki/technológiai hivatkozási modell — a technológiai szintű szolgáltatások”-ban). Az alkalmazási szoftver és az alkalmazási platformnak meg kell felelnie, összhangban kell lennie a minőségi sajátosságok által megfogalmazott irányelvekkel és követelményekkel.



38. ábra III-RM Integrált infrastruktúra hivatkozási modell szempontjából lényegtelen elemek kiszűrítve



39. ábra A szervezeti (vállalati, üzleti) architektúra szervezeti (vállalati, üzleti) szolgáltatáshoz kapcsolódó komponensei



43. ábra Szolgáltatás orientált architektúra szervezeti architektúra megközelítésben

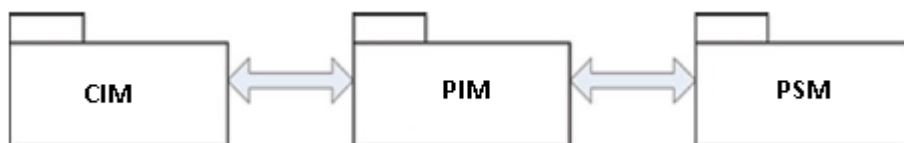
6 MDA

Egy újra felbukkanó téma a szoftverfejlesztés evolúciójában a formális nyelvek minél absztraktabb használata modellezési megoldásokhoz. Leggyakrabban az alapvető szoftverfejlesztésben, az absztrakt leírásokat (például Javában, C#-ban) futtatható formába transzformálják. Az ilyen absztrakt fejlesztések növelik a hatékonyságot, csökkentik a hibákat, mivel a transzformáció automatizált.

A modell-vezérelt architektúra (MDA – Modell Driven Architecture) egy nem régi technológia, amely tulajdonképpen az eszközök egy jóval absztraktabb specifikációjához és fejlesztéséhez vezet.

- Az **MDA**-t az OMG (Object Management Group) definiálta, mint „*egy szemlélet az informatikai rendszer specifikációjában arra, hogy elkülönítsük a funkcionalitás specifikációját a specifikáció implementálásától.*”

Ahogy az a névből is kikövetkeztethető, egy „alkalmazási modell” az, amely mint vezérfonal húzódik meg erő MDA mögött. Az MDA-ban egy modell *funkcionalitásnak, struktúrának és/vagy az alkalmazás rendszer viselkedésének* formális specifikációja. Az MDA szemléletben egy informatikai rendszert először elemeznek/analizálnak és úgy specifikálnak, mint egy „**Számítástechnikától független modell**”-t (**CIM – Computation Independent Model**), amit a szakterületi (*domain*) modellként is ismernek. A *CIM* a rendszerkörnyezetre és a rendszerkövetelményekre fókuszál. A számítástechnikai, informatikai és az megvalósítás részletei ezen a szinten rejtve vannak (vagy még meg sincsenek határozva).



44. ábra Modell transzformáció MDA-ban

Ahogy az ábra (44. ábra) is mutatja a *CIM*-et átalakítják platform független modellé (**PIM – Platform Independent Model**), amely számítástechnikai információkat tartalmazza az alkalmazásra vonatkozóan, de semmilyen információ nincs az alatta levő platform technológiájáról, ahol a *PIM*-et implementálni fogják. Végül a *PIM*-et **PSM**-mé (**Platform Specifikus Modell**) alakítják, ami tartalmazza a szükséges platform specifikus információkat, műszaki informatikai, információtechnológiai részleteket.

6.1 Miért használjunk MDA-t?

Egy „**platform**” az MDA-ban bármilyen alrendszerek és technológiák halmazaként definiálható, ami egy koherens funkcionalitáshalmazt biztosít interfészeken és specifikált használati mintákon keresztül (mint például CORBA, vagy JEE).

Az MDA-t az OMG standardok tömkelege támogatja, mint például UML, MOF (Meta-Object Facility), XMI (XML Metadata Interchange), és CWM (Common Warehouse Model). Az MDA-n belül a modelleket egy modellező nyelvben le kell írni. Ez nyelv lehet bármelyik általános modellező nyelv, ami a követelményeknek eleget tesz (pl. **UML**). A MOF lehetőséget nyújt arra, hogy specifikáljunk bármilyen modellező nyelvet a MOF modellezési rendszerét használva.

6.1 Miért használjunk MDA-t?

A modellek központi szerepet játszanak az MDA-ban. De miért is van szükségünk modellekre. Íme a válasz.

A modellek a rendszer absztrakcióját biztosítják. A modelleket sokféleképpen lehet használni, például a rendszer minőségének jóslására (pl. teljesítmény), tervek validálására, és arra, hogy információt adjon a rendszer sajátosságairól az üzleti és rendszerelemzőknek, tervezőknek és szoftverfejlesztőknek. Az MDA világában, ezek a modellek a rendszer megvalósításának, implementációjának tervrajzaként is szolgálnak.

Három célja van az MDA-nak: hordozhatóság, átjárhatóság és újrahaználhatóság, amit az elemek architektúráis szétválasztottságának magas fokával érnek el.

6.1.1 Hordozhatóság

A hordozhatóságot leginkább a modellek szétválasztásával és transzformációjával érhetjük el. Magas szintű modellek nem tartalmazzak alacsony szintű platform és technikai részleteket. A rendszer alatti platformok cserélődnek, vagy fejlődnek, ezek a modellek transzformálhatóak a platformra közvetlenül, újramodellezés nélkül.

6.1.2 Átjárhatóság

Nagyon ritka az olyan alkalmazás, amely nem kommunikál más alkalmazásokkal. Vállalati szintű alkalmazásoknál szükséges, hogy cégen belül és cégen kívül is kapcsolatot teremtsenek a rendszerek egymással. A legtöbb alkalommal ezekre a külső rendszerekre minimális ráhatásunk lehet.

Az MDA-t használva, az átjárhatóságot a horizontális modell leképezésével (mapping) érhetjük el.

Az átjárhatósági probléma egy horizontális modell leképezési és integrálási problémájaként fogható fel. Az egyszerűség kedvéért tegyük fel, hogy két CIM/PIM/PSM rendszerünk van. Az információcsere és a kölcsönhatás a két magasabb rétegű CIM és PSM között modellezhető és elemzhető. A modelleken áthidaló leképezések és információcsere és a kölcsönhatás aztán leképezzük egy részletes kommunikációs protokollra, vagy egy közös adatbázisba. Mivel explicit vertikális transzformációk léteznek mindkét rendszerben az egyes rendszerbeli a modellek között, az az egyes rendszerek elemei is bekerülnek ebbe a leképezési folyamatba, és könnyen nyomon követhető vagy automatikusan lefordíthatóak alsóbb szintű elemekké.

6.1.3 Újrafelhasználhatóság

Az újrafelhasználhatóság a kulcs a minőség és a termelékenység javítására. Az MDA támogatja a modellek újrahasznosítását és tervezési alkalmazások rutinjait, kifejezetten azért, hogy egy alkalmazás családot lehessen létrehozni.

6.1.4 Gyakorlat és eszközök

Habár lehetséges, hogy az MDA részeit eszközök támogatása nélkül használjuk, ez csak a nagyon bátraknak és a nagyon elkötelezetteknek lehet ajánlani. Mivel néhány MDA szabvány csak gépi értelmezésre alkalmas.

Mivel az MDA szabványok, – főként az útmutatók csak ajánlások – és nem kötelező előírásokat tartalmaznak, eszközök tömkelege állítja magáról, hogy támogatja az MDA-t. Természetesen mind különböző funkcionalitással és képességgel.

6.1.5 MDA és szoftver architektúra

A legtöbb MDA modell legfőképpen a szoftver architektúra reprezentációja. A szoftver architektúra modellek egy absztrakt reprezentációi tulajdonképpen a szakterületi (domain) modellek és a rendszer modellek. A generált kód modellek az architektúra sajátosságait magukban hordozzák, implementációs részletekkel.

Egy architektúrát az **ADL (Architecture Description Language)** architektúra leíró nyelvel írhatunk le. Sok ADL keletkezett az elmúlt idők alatt, mindegyiknek megvan a maguk előnyei és hátrányai; más-más területekre fókuszáltak. Sok hasznos ADL funkcionalitást építettek az UML (könnyű és nehézsúlyú) változataiba.

6.1 Miért használjunk MDA-t?

6.1.6 MDA és nem-funkcionális követelmények

A **nem-funkcionális követelmények** (*Non-functional Requirements – NFR*) alapvető építőkövei a szoftver architektúrának. Az *NFR*-ek tartalmazzák a minőségi tényezőkre vonatkozó követelményeket, mint például a *teljesítmény, módosíthatóság, újrahasználatosság, átjárhatóság és biztonság*. Habár az MDA nem céloz meg minden egyes minőségi tulajdonságot (legalábbis nem direkt módon), de segít elérni ezeket a törekvéseket:

- 1) Egy bizonyos mértékű átjárhatóságot, újrahasználatosságot és hordozhatóságot minden modellbe beépítenek, azok belső elválasztása miatt.
- 2) A MOF és az UML profil mechanizmusok lehetővé teszik, hogy az UML-t kibővítsük olyan modellezési szolgáltatásokkal, és tervezési elemekkel, amelyek az *NFR*-eket célozzák meg.
- 3) Az *NFR* modellezési követelmény és tervezés bővítményekkel, explicit modellezési szabványokkal támogatja a modell transzformációk során a minőségi tulajdonságok elérését.

6.1.7 Model Transzformációk és Szoftver architektúra

Az MDA-ban minden modellező nyelvnek megvan a maga szintaxis és szemantika által jól definiált meta-modellje. Az egyik modellből (pl. követelmények) a másikba (pl. terv) való áttranszformálásának folyamata *szisztematikus* folyamat, pontosan meghatározott *transzformációs szabályokat* követ. Ez a részletes előírás, és meghatározottság nagyban javíthat az architektúra modell minőségén és hatékonyságán.

A model transzformációs szabványt, ami az OMG-ből ered, „**Query, View and Transformation**” (QVT) néven ismerik. A **QVT** egy alapértelmezett módszert biztosít, hogy a forrásmodelleket célmodellekké transzformálja.

6.1.8 A SzOA és az MDA

Az MDA és a SzOA is ugyanazt a problémát próbálják megoldani két különböző szemszögből és különböző szintű absztrakcióval. A **SzOA heterogén** rendszereket hidal át kommunikációs protokollokon keresztül, az Interneten mindenütt fellelhető szolgáltatásokon, és hozzákapcsolódó szolgáltatás-alapú, szolgáltatás-központú, szolgáltatás-orientált architektúra filozófián és megközelítésen keresztül. Az MDA a rendszerek közötti hézagmentes, akadálymentes, magas szintű szemantikus integrációval foglalkozik és a rendszer-modelleket alacsonyabb architektúra szintű SzOA alapú megoldásokká transzformálja.

6.1.9 Az analitikus modellek is modellek

Az analitikus modellek használatával meg tudjuk vizsgálni a rendszer olyan sajátosságait, amelyeket gyakran figyelmen kívül hagynak (még az MDA útmutatóban is). Ezeknek a modelleknek a használata (csakúgy, mint a QVT transzformációs modelleké) kompatibilis az MDA-val és alkalmazásuk nagyon hasznos.

Annak érdekében, hogy UML modellünkhöz analitikus modelleket építsünk, a modellezést manuálisan vagy alacsony szintű transzformációval kell felépítenünk, az UML XML-ben való reprezentációjától függően.

Az analitikus modellt szükségszerűen ugyanabból a tervezési modellből kell származtatni. De mivel az analitikus modell nem kompatibilis az MDA szabványokkal, nehezebb kereszthivatkozást létrehozni a származtatott modellek között.

Az MDA a modell-vezérelt szoftverfejlesztés átfogó szabványosítása, amely bebizonyította, hogy sikeres és folyamatosan fejlődik. Az MDA-nak kihatása van a szoftver architektúra tervezés gyakorlatára, és szükséges ahhoz, hogy a tervező csapat formális modelleket alkosson az alkalmazásról, egy egyszerű modellező nyelv használatával.

Az elért sikerektől függetlenül, még mindig sok támadás érte az MDA-t a korlátai miatt, néhányan ellenkezőjét nehéz bebizonyítani alaposabb vizsgálatok nélkül.

7 SZÁMÍTÁSI FELHŐ (CLOUD COMPUTING)

A számítási felhő az utóbbi években (2010~) a szakirodalom egyik legtöbbet tárgyalt témájává vált és az informatikai alkalmazásokban is egyre nagyobb teret foglal el. A vállalkozások nagy száma helyezi át adatfeldolgozó rendszereit a számítási felhőbe. A hagyományos környezetben használt üzleti alkalmazási szoftvernek és az adatoknak a számítási felhőbe történő átvitelének módszertani kérdései is felmerülnek, kutatási és informatikai tanácsadói feladatként egyaránt; továbbá a *vállalatirányítási rendszerek* (ERP) alkalmazásával, igénybevételeivel, kiválasztási kritériumaival kapcsolatban újabb szempontrendszerek alakulnak ki a számítási felhő kontextusának figyelembe vétele következtében.

A számítási felhő, mint minden időszakban rendelkezésre álló, rugalmasan telepíthető és könnyen továbbfejleszthető számítástechnikai szolgáltatás ma már egyre inkább elterjedt informatikai szolgáltatási megoldásnak tekinthető. Technikai alapja a fejlett telekommunikációs és információtechnológiai hálózatok műszaki képessége és kiszolgáló gépek / szerver virtualizáció . A virtualizáció:

„Valaminek a nem feltétlenül valós hardver alapon nyugvó szoftveres leképezése, mely funkcionalitásban egyezik a valóságos hardver funkcionalitásával. Mit értünk ez alatt? Egy virtuális szerver is végső soron (egy vagy több) valódi szerveren fut, csak szoftver által emulált környezetben, a benne futó (guest, vagy vendég operációs rendszer) szemszögéből viszont csak az emulált hardver látszik.”[1]

Így válik lehetővé az adatok és a feldolgozások gyors és dinamikus átvitele a mindenkori, lokális számítóközpont szerverei és a világszerte telepített globális számítóközpontok szerverei között. A technológia magas szinten skálázható azért, hogy szűk keresztmetszetek elkerülhetőek legyenek. Az internet lehetővé teszi azt, hogy a felhasználók a számítóközpont helyétől függetlenül az adataikhoz hozzáférjenek.

A **számítási felhő** fogalmára még nem alakult ki egy egységesen elfogadott definíció. Az alábbi két idézet azonban tartalmazza azokat a jegyeket, melyeket ez az új számítástechnikai technológia magában foglal, vagyis a technikai lehetőségeket és az előnyös költség szintet:

„A számítási felhő egy olyan modell, mely szükség esetén lehetővé teszi azt, hogy hozzá lehessen férni bármely időpontban és bárholonnan kényelmesen, konfigurálható számítógépes erőforrások osztott halmazához (pl. hálózatok, szerverek, tároló rendszerek, alkalmazások, szolgáltatások) egy

hálózaton keresztül. Ezek az informatikai erőforrások gyorsan és minimális üzemeltetési ráfordítással, vagy kevés szolgáltatói beavatkozással állnak rendelkezésre.” [16]. 14. old.

„A számítási felhő olyan szolgáltatások, alkalmazások és erőforrások sokasága, amelyeket a felhasználónak flexibilisen, skálázhatóan, és testre szabhatóan, az interneten keresztül kínálnak fel anélkül, hogy hosszú távú tőke lekötésre és informatikai, információtechnológia specifikus ismeretekre volna szüksége. A felhasználó – a vertikális integrációs mélység-, függvényében vagy egy komplett szoftveralkalmazást, vagy esetleg csak a szükséges információtechnológiai infrastruktúrát veheti igénybe.” [17]

Joe Weinman definíciója a számítási felhőre ([13]):

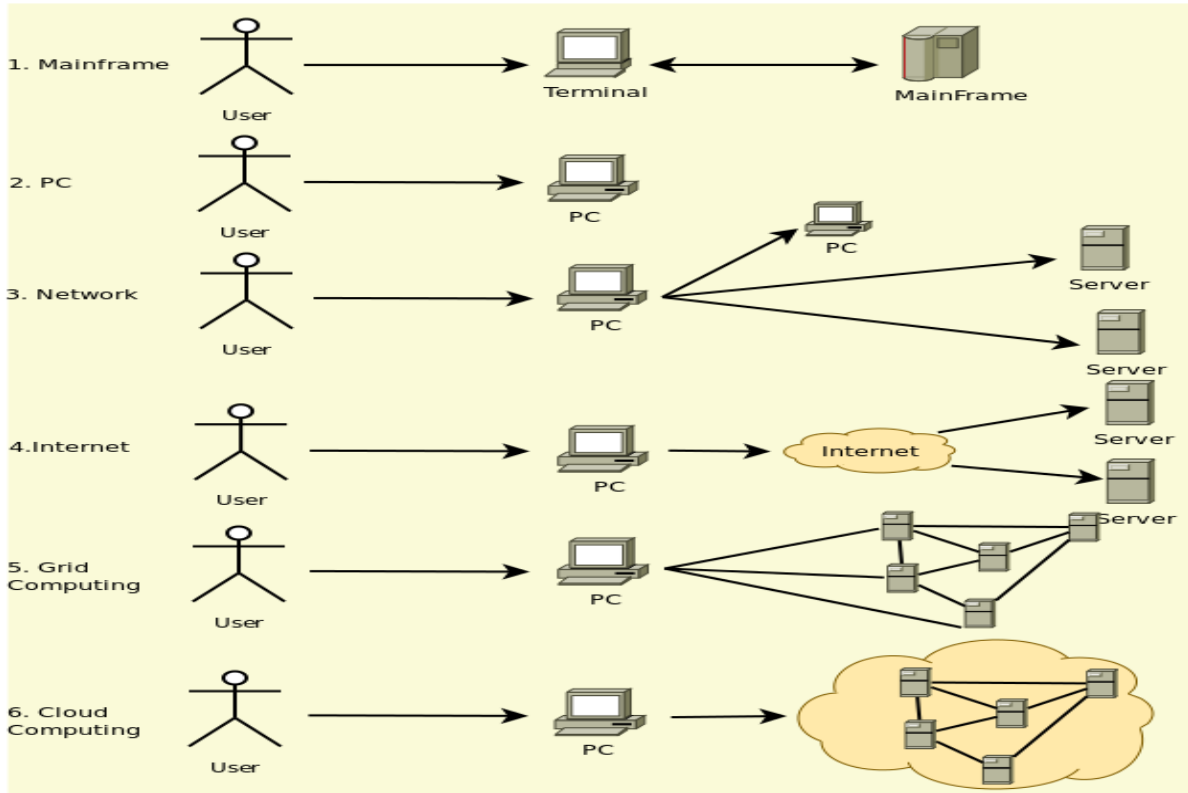
Common Infrastructure	Közös (megosztott) infrastruktúra
Location Independence	Hely függetlenség
Online accessibility	Online (Interneten) keresztüli hozzáférhetőség
Utility pricing	Közmű jellegű árszerkezet
On-Demand resources	Igény szerinti erőforrás rendelkezésre állás nyújtása

7.1 A hálózati és szervezeti architektúra fejlődése

A számítási felhő fejlődési ívét, a különböző adatfeldolgozási formák fejlődési lépéseinek ábrázolásával- az ábra (45. ábra) mutatja.

A történeti fejlődés első szakaszában sok felhasználó osztozott egy nagy teljesítményű számítóközpont kategóriájú gépen, általában egyszerű szolgáltatást nyújtó megjelenítőkön keresztül („buta terminál, dummy terminal”) voltak elérhetők a számítóközpont funkcionalitása. A második szakaszban a személyi számítógépek fejlődése történt meg, amelyek teljesítménye elérte azt a színvonalat, hogy a felhasználók döntő többségének igényét ki lehetett velük elégíteni. A harmadik szakaszban a hordozható és személyi számítógépeket helyi hálózatokban kötötték össze a teljesítmény növelés és az informatikai erőforrások megosztása végett. A negyedik szakaszban a helyi hálózatokat más helyi hálózatokkal kötötték össze, amely egy globális hálózatot kialakulásához vezetett, nevezetesen az Internethez, amely le-

hetővé tette azt, hogy távoli elérésű alkalmazásokat és erőforrásokat lehessen kihasználni. Az ötödik szakaszban a **grid** számítástechnika nyújtott olyan szolgáltatásokat, amelyekkel meg lehetett osztani számítástechnikai adattárolási, kapacitásokat egy elosztott hálózaton keresztül. A hatodik szakaszban a számítási felhő további megosztott erőforrásokat nyújt az Interneten keresztül, skálázható és viszonylag egyszerű módon.



45. ábra Adatfeldolgozási módok fejlődési típusai ([18] Fig.1 alapján)

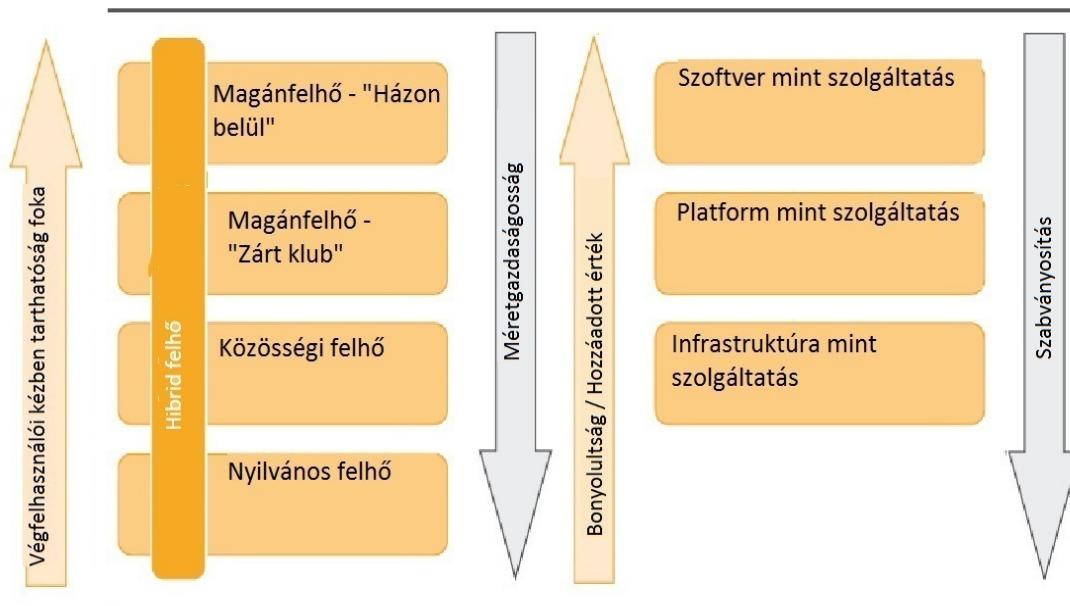
Ezt a hat számítástechnikai – valójában hálózati, műszaki technológiai – architektúrát, áttekintve úgy tűnik mintha a legelső, központosított számítóközpont megközelítéshez térnénk vissza. Azonban a két architektúra paradigma között jelentős különbségek észlelhetők.

A hagyományos számítóközpont megoldás véges számítástechnikai kapacitásról tudott gondoskodni, míg a számítási felhő –legalábbis gyakorlati értelemben – végtelen számítástechnikai kapacitást tud nyújtani. Ezen kívül a végfelhasználói felületet megtestesítő, egyszerű terminálokkal szemben a személyi számítógépek mint végpontoknak saját, helyi számítástechnikai kapacitásuk és gyors elérésű adattárolójuk van.

Számítási felhő

- **Számítási felhő** : információtechnológiai (IT) erőforrások és szolgáltatások, amelyeket mögöttük meghúzódó infrastruktúrától független és absztrakt módon nyújtanak,

több szereplő, több vállalkozás, több igénybevevő számára szolgáltatásokat, kívánásra és igény szerint, és alkalmas mennyiségben. (Cisco Systems által adott definíció)



46. ábra NIST számítási felhő modellje

- A **számítási felhő** egy olyan informatikai szolgáltatási modell, amely lehetővé teszi azt, hogy bárhol, kényelmesen, igény szerint nyújtson hálózaton keresztül hozzáférést számítástechnikai erőforrások megosztott halmazához (pl. hálózatok, kiszolgáló gépek, alkalmazások és szolgáltatások), amelyeket nagyon gyorsan rendelkezésre lehet bocsátani úgy, hogy nagyon kevés rendszer-adminisztratív tevékenységre illetve a szolgáltatóval csak minimális mértékben szükséges kapcsolatfelvételre és/vagy a szolgáltató csak valamilyen csekély beavatkozására van esetleg szükség. Ez a *számítási felhő* modell elősegíti a szolgáltatások rendelkezésre állását és öt lényeges jellemzője van: Magas **elaszticitás**, mérhető szolgáltatási mennyiség, igény szerinti **önkiszolgálás**, bárhol elérhető hálózaton keresztüli **hozzáférhetőség**, helyszíntől független **erőforrás készlet**, valamint alapvetően négy szolgáltatás nyújtási modell: nyilvános felhő, **magán felhő**, közösségi felhő és hibrid felhő. (A National Institute of Standards and Technology (NIST, USA szabványügy testülete) munkahipotézis definíciója).

7.2 Szolgáltatóval szemben szabott követelmények

A számítási terhelés csökkentése érdekében a szolgáltatónak olyan árazási modellt kell alkalmaznia, amely lehetővé tesz különbségtételt és olyan díj struktúrát, amelyet felhasználás tipikus mintázatai, a sávszélesség és a szolgáltatási szint alapján állapítanak meg.

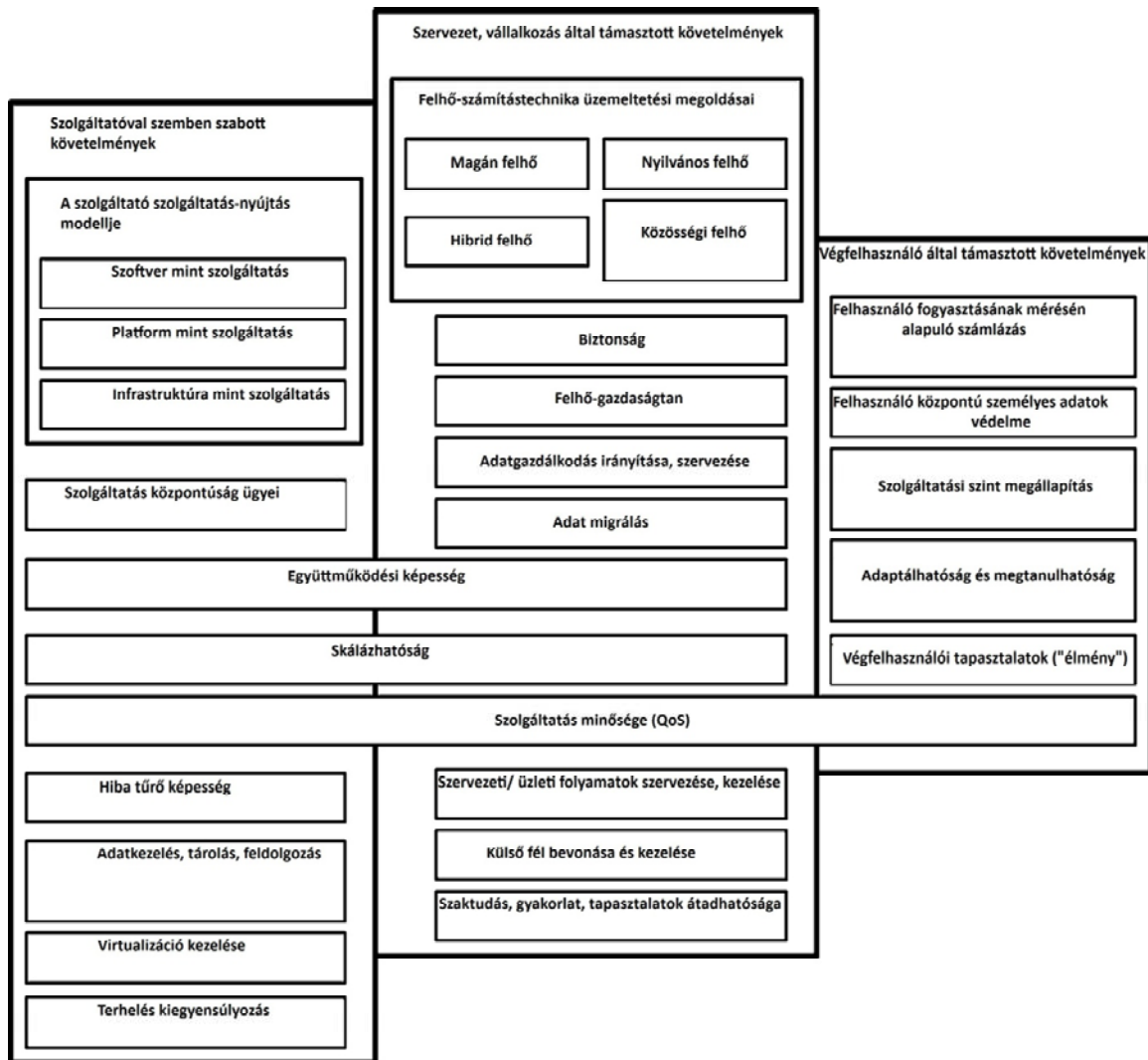
7.3 A szolgáltató szolgáltatás-nyújtás modellje

A szolgáltató szolgáltatás-nyújtás modellje annak alapján sorolható osztályba, hogy milyen szolgáltatásokat tud felajánlani. Alapvetően (és klasszikusan) három szolgáltatási modellről szoktak beszélni :

- Szoftver mint szolgáltatás;
- Platform mint szolgáltatás;
- Infrastruktúra mint szolgáltatás.

Ezeket a szolgáltatásokat általában olyan ipari szabvány felületeken keresztül lehet elérni, mint pl. a **SzOA**, a **REST** (*REpresentational State Transfer*), Web szolgáltatások és egyéb gyártó függő szolgáltatásokon keresztül.

Szoftver mint szolgáltatás(SaaS): Szoftver mint szolgáltatás (vagy alkalmazás mint szolgáltatás) egy olyan szolgáltatási **platform**, amely több igénybevevő vállalkozást, partnert szolgál ki. Közös erőforrásokat használ, a futtatható kódtól kezdve a háttérben működő adatbázisig, egyszerre, szimultán módon szolgálva ki több ügyfelet. Szoftver mint szolgáltatást gyakran alkalmazás-szolgáltatói modellnek is nevezték (*Application Service Provider* (ASP)). Jelentős szereplők: Salesforce Customer Relationships Management (CRM) system, NetSuite, and Google Office Productivity alkalmazások. Az SaaS alkalmazásának fő kérdése az integrációs képességgel szemben támasztott követelmények, együttműködés és összhang más alkalmazásokkal. Egy másik kritikus fontosságú alkotórésze a szolgáltatásnak a különböző technológiák kombinált alkalmazása mint pl. J2EE, .Net, Hibernate, Spring, Scalable Infrastructure and Services. Alkalmazási szinten pedig az architektúra tervező számára a következők a fontos szempontok: *skálázhatóság, teljesítmény, több igénybevevő kiszolgálása, konfigurálhatóság és hibatűrő-képesség.*



47. ábra A számítási felhő architektúrájának 3 szintű modellje

Platform mint szolgáltatás (PaaS): A PaaS mint szolgáltatás mögött az a gondolat húzódik meg, hogy a fejlesztők számára nyújtsanak egy olyan platform *szolgáltatást*, amely a fejlesztés teljes életciklusát átfogja, tartalmazza mindazon rendszereket és fejlesztési környezeteket, amelyek szükségesek a teszteléshez, telepítéshez, üzembe-helyezéshez és olyan kicsiszított web alkalmazások működtetéséhez, amelyeket a számítási felhő alapú platform mint szolgáltatás tud nyújtani. Példák:

- Integráció-központú platform, azaz olyan platform, amelyik e-üzleti, e-kereskedelmi alkalmazásokat nyújt, a különböző platformok közötti kölcsönös kapcsolatokat, kölcsönhatást széles körűen támogatja, a használatban levő nyelveket, és az olyan végfelhasználókat, akik Facebook F8, Salesforce App Exchange.

7.4 Szolgáltatás-központúság ügyei

- Fejlesztés-központú platform, olyan platform, amely rendszerfejlesztőknek nyújt olyan fejlesztési környezetet, amelyben tesztelni, telepíteni és üzembe helyezni tudják alkalmazásaikat, mint pl. Google App Engine, Bunzee connect, és SF force.com.
- Infra-orientált platform, olyan platform, amelyik a fejlesztőknek nyújt skálázható infrastruktúrát és adattároló helyet, mint pl. Amazon EC2, SimpleStorage, Simple DB.

PaaS teljes mértékben a szolgáltató rendelkezésre állási képességére támaszkodik. Nagy valószínűsége van annak, hogy olyan erős kötődés jön létre a szolgáltatóhoz, ami megnehezíti a szolgáltató váltást, abban az esetben, ha PaaS szolgáltató gyártó függő szolgáltatásokat és fejlesztő nyelveket nyújt. Ennek a hatásnak és következményeinek mérséklésére szolgál az „**Open Platform as a Service (OPaaS or Open PaaS)**” vagyis a nyílt platform mint szolgáltatás megközelítése. OPaaS széles körű szolgáltatásokat nyújt a fejlesztőknek mint pl. tetszőleges programozási nyelvek és környezeteik, fejlesztő eszközök, kiszolgáló (szerverek), adatbázisok stb.

Infrastruktúra mint szolgáltatást (IaaS, Infrastructure-as-a-service) gyakran hardver mint szolgáltatásnak (HaaS, Hardver-as-a-service) is nevezik. Ez a modell különösen előnyös a vállalatoknál, mivel nem kell befektetniük informatika rendszerek hardver beszerzésébe és fenntartásába. A nagyobb rugalmasságtól eltekintve, az egyik előnye az, hogy használat alapján lehet a díjat fizetni. Ez lehetővé teszi, hogy a partner a növekedésével arányosan fizessen. Másik előnye az, hogy a legkorszerűbb technológiához jutnak hozzá a szolgáltatáson keresztül. Az IaaS szmben támasztott legfontosabb követelmények: *kívánság szerinti igénybevétel, ön-fenntartó, ön-javító, több partner kiszolgálása, ügyfelek elválasztása*. Ajelentősebb szereplők: GoGrid, MSP On Demand, masterIT, Mosso/Rackspace, NewServers Inc.

7.4 Szolgáltatás-központúság ügyei

A szervezet vagy vállalat vezetésének, informatikai részleg vezetésének igényeinek kielégítése végett a felhő-architektúrának egyetemes szolgáltatás-központú megközelítéssel kell lépnie. A számítási felhő szolgáltatásaira a következőknek kell jellemzőknek lennie:

- **Autonóm:** A felhőbeli rendszereket / alkalmazásokat úgy kell megtervezni, hogy a változó környezethez dinamikusan tudjanak alkalmazkodni viszonylag csekély mértékű humán segítséggel. A szolgáltatások autonóm jellegű működése mind a szol-

gálatások minőségét (QoS) mind biztonságát és hiba-tűrőképességét tudja erősíteni.

- **Ön-leíró képesség:** Az ön-leíró szolgáltatás felületet definiálja a szolgáltatás által tartalmazott információkat és funkcionalitást mint újra felhasználható környezet-független elemeket. A szolgáltatás tényleges megvalósítási módja eközben változhat anélkül, hogy a szolgáltatási megállapodást aktualizálni kellene. Az ön-leíró szolgáltatások azért előnyösek, mert a kliens oldali alkalmazással tudják közölni, hogy milyen módon hívhatók meg és milyen adatokat tudnak átadni.
- **Alacsony költségű** elosztott alkalmazás építés, konstruálhatóság, amelynek több résztvevős környezetben kell olyan infrastruktúra háttérrel nyújtania, amely lehetővé teszi a partnerek közötti együttműködést és kölcsönös kapcsolatokat.

7.5 Együttműködési képesség (Interoperabilitás)

Az együttműködési képesség egy olyan keretrendszert, ontológiát illetve nyílt adatformátumot, protokollokat, API-t jelent, amelyek lehetővé teszik az alkalmazások és adatok migrálását és integrálását különböző felhő szolgáltatók között valamint a biztonságos és védett adatkommunikáció támogatása a különböző felhő platformok között.

7.6 Szolgáltatás minősége (QoS)

A szolgáltatás minősége (QoS) a teljesítmény és a rendelkezésre állás garantálására szolgál, továbbá olyan minőségi jellemzők szavatolására, mint a biztonság, *megbízhatóság* stb. A szolgáltatási szint megállapodások (SLA, Service Level Agreement) a szolgáltatás nyújtó és a végfelhasználó közötti egyezség visszatükrözése.

A szolgáltatás minősége (QoS) fogalma alá tartozik az átlátható rendszeradminisztráció és menedzsment, az erőforrások, adattárolók, hálózatok, virtualizált környezetek, szolgáltatások migrálása és a hibatűrő képesség. A felhő szolgáltatást nyújtók számára szolgáltatás minősége (QoS) elsősorban a virtualizált környezetek és nyomon követő, monitoring eszközök teljesítményét jelenti.

Az alapkérdés természetesen az, hogy végfelhasználó részéről milyen teljesítmény követelményekkel tervezik használni az alkalmazásokat és a szolgáltatásokat. Magas teljesítmény követelményeket tartalmazó szolgáltatási szint megállapodás esetében a felhő-szolgáltató egyáltalán nem fogja tudni teljesíteni a követelményeket az Internetből származó hálózati

7.7 Hiba-tűrő képesség

késleltetés miatt. Mivel a végfelhasználó elvárásai magasak maradnak a szolgáltatás minőségére tekintettel, fontos az, hogy bizonyos tolerancia szintet állapítsanak meg az üzleti, szervezeti folyamatok értelmében,

7.7 Hiba-tűrő képesség

Egy rendszer hiba tűrőképességét az üzemszünetek értelmében lehet mérni. A felhő-szolgáltatás üzemszünetei összefüggenek a platforméval. Néhány üzemszünet meglehetősen hosszú volt a beszámolók szerint, pl. a Microsoft Azure üzemszünete 22 órás volt (2008.03.13 - 14). Cégek, vállalatok, sőt közigazgatási szervezetek számára, ha kritikus fontosságú ügyviteli folyamataikat egy ilyen felhő-szolgáltatásba telepítették volna, akkor súlyos problémával, a piaci szereplőknél komoly anyagi veszteségekkel nézhetek volna szembe. A felhő-szolgáltatás megbízhatósága ezért súlyos problémát jelenthet a felhő-szolgáltatás igénybevevője számára, ha a rendszerleállások és üzemszünetek fölött elveszíti a felügyeleti és irányítási lehetőséget.

Az 7. Táblázat mutatja a jelentősebb felhő-szolgáltatók szolgáltatás kieséseit és ez a táblázat illusztrálja a felhő-szolgáltató rendelkezésre állását és megbízhatóságát.

7. Táblázat Jelentős felhő-szolgáltatások üzemszünetei

	Szolgáltatás és a kiesés oka	Időtartam	Dátum
1	Microsoft Azure: Windows Azure hibás működés	22 h	March 13–14, 2008
2	Gmail , Google Apps Engine	2.5 h	Feb 24, 2009
3	Google search üzemszünet: programozási hiba	40 min	Jan 31, 2009
4	Gmail: üzemszünet a kapcsolódó rendszerekben o	1.5 h	Aug 11, 2008
5	Google AppEngine részleges kiesés, programozási hiba	5 h	June 17, 2008
6	Google AppEngine: részleges illetve teljes üzemszünet	5 h 50 min	July 2, 2009
7	S3: autentikációs kiszolgáló gép túlterhelése, teljes üzemszünet	2 h	Feb 15, 2008
8	S3: egy bit az egyik protokoll teljes összeomlásához vezetett		July 20, 2008
9	FlexiScale: alap hálózati hiba	18 h	Oct 31, 2008

A felhő-szolgáltatónak érzékelnie kell a hibás jelenségeket a számítási felhő rendszerében és alkalmazásaiban egy arra alkalmas eszközkészlet és ellenőrzési mechanizmuson keresztül, ilyen lehet például egy ön-javító (self-healing) és ön-diagnosztikai (self-diagnostic) eszközrendszer és mechanizmus. Továbbá olyan következtető rendszerek, amelyek a hibák osztályozását és csoportosítását (klaszterezését) segítik, nagymértékben tudják segíteni nemcsak a hibák detektálását, hanem a lehetséges okok és azok gyökerének meghatározását.

7.8 Adatkezelés, tárolás és adatfeldolgozás

A helyi kiszolgáló és asztali, személyi számítógépekről az adatfeldolgozási és adattárolási feladatok fokozatos átcsúsztatása a számítási felhő szolgáltatásaiba, az Internetre egyszerre nyújt új lehetőségeket és új korlátokat. Az adatok másolatai ('replikált' példányai) hatalmas földrajzi távolságokban helyezkedhetnek el, rendelkezésre állásuk és tartós elérhetőségük a felhő-szolgáltató kiemelt feladata. Azonban ha az adatok tárolása megbízhatatlan adatközpontokban, számítástechnikai centrumokban történik, akkor ez a személyes adatok, a magántitok védelme szempontjából iszonyatosan magas kockázatot jelent.

A felhő-szolgáltatás egy másik fontos jellemzője az, hogy az adatfeldolgozási kapacitás elasztikus, viszonylag könnyen illeszthető a változó körülményekhez, extra számítástechnikai kapacitás bővítés az igények növekedésével párhuzamosan menetközben is könnyedén megoldható.

A számítási felhő hangsúlya a nagyteljesítményű adatfeldolgozáson van, végtelen nagyságú, igény szerint rendelkezésre álló számítástechnikai kapacitás, kezdeti beruházási költségek nélkül. A szolgáltatásnak ez a jellege sokkal könnyebben értelmezhető az adatfeldolgozási képesség értelmében, sem mint az adattároló kapacitás tekintetében

Az adattárolásra vonatkozóan a felhő-szolgáltatónak olyan adattárolási infrastruktúrát kell nyújtania, amelynek van egy gazdag adat lekérdező nyelve, és viszonylag egyszerű adatszerkezeteken alapul, amely lehetővé teszi a skálázhatóságot mind fel és mind lefelé. Továbbá, a felhő-szolgáltatónak olyan teljesítmény garanciákat kell szavatolni, amely lehetővé teszi a programozó számára, hogy az adattárolási eljárásokat valahogy kézben tudja tartani.

A jelenleg rendelkezésre álló megoldások (Microsoft Azure, Google's AppEngine, Amazon etc.) nem használják ki azokat a lehetőségeket, amelyeket a félvezető diszkek (SSD, Solid State Disk) potenciálisan nyújtanak. Továbbá egyelőre nem foglalkoznak azokkal a skálázhatósági igényekkel, amelyek a No-SQL, vagy nem relációs, oszlop-orientált adatbázis-kezelés

nyers adatokon történő adatfeldolgozásával függenek össze, illetve ugyanakkor a hagyományos adatbázis kezelés deklaratív jellegéből származnak. Az energiafelhasználás szempontjából az SSD-k kevesebbet fogyasztanak mint a hagyományos merevlemezek. Az SSD-k terjedésének nagy akadálya egyelőre az ár, az adattároló kapacitás és az SSD technológiára kihegyezett, kifinomult adatlekérdező megoldások. Ez azonban várhatóan jelentősen változni fog a közeljövőben, a másodpercenként feldolgozott bemeneti, kimeneti adatok mennyiségében jelentős előnye van az SSD technológiának. Továbbá az adattároló kapacitás is növekszik illetve új SSD központú algoritmusok és adatszerkezetek kifejlesztése egyre nagyobb lépésekben történik meg.

8. Táblázat A számítási felhő egy olyan szolgáltatás, amely az információ-technológiai infrastruktúra használatát jelenti

Felhő számítástechnika infrastruktúrája	Web szolgáltatások: Flickr API, Google Maps API, Storage (Adattárolás)	Szolgáltatások	Hardver és szoftver megoldások halmaza
	Web alapú alkalmazások. Google apps, salesforce.com, Flickr, adóbevallás	Alkalmazások	
	Virtualizált kiszolgáló, web alkalmazás, tárhely szolgáltatás. Előre konfigurált eszközök (appliance) vagy egyéb szoftverek halmaza (stack): AMP, GlassFish, etc.	Köztes réteg (MIDDLEWARE)	
	Előre konfigurált operációs rendszer bérlése. Saját alkalmazásokkal bővítés. Pl. DNS kiszolgáló	Operációs rendszer	
	Virtualizált kiszolgáló bérlése. Saját virtulás gép telepítése (VM) illetve a saját szoftver eszközök halmazának üzembe helyezése.	Virtualizált kiszolgálók	

	Számítástechnikai grid (teherelosztó hálózat). Pl.: HPC applications	Fizikai kiszolgálók	
--	--	---------------------	--

7.9 Virtualizáció kezelése

A virtualizáció egy olyan absztrakciót jelent, amelyben a fizikai számítástechnikai erőforrásokat logikai erőforrásokká absztrahálják azért, hogy javítsák az agilitást, rugalmasságot, csökkentsék a költségeket és ezen keresztül a szervezet számára többlet értéket nyújtsanak. A virtualizáció alapfeladata az, hogy operációs rendszerek fölött több virtuális gépet kezeljenek, nyomon kövessék, értékeljék, teszteljék a kiszolgáló gépeket és a célkörnyezetre hajtsák végre a telepítéseket. A virtuális gép rendszer adminisztrációját, menedzsmentjét a **hypervisor** nevű szolgáltatással látják el. A hypervisor felügyelete alá tartozik a CPU, a memória kezelés, az I/O (bemenet/kimeneti adatok kezelése), amelynek évén magasabb teljesítmény, megbízhatóság és kompatibilitás érhető el.

A felhő-számítástechnikában a virtualizáció a következőket tartalmazza a következőket: kiszolgáló ('szerver') virtualizáció; ügyfél ('kliens'), asztali gép, alkalmazás virtualizációt, adattároló virtualizációt (SAN, Storage Area Network), hálózat virtualizációt, szolgáltatás és alkalmazási infrastruktúra virtualizációt.

- kiszolgáló ('szerver') virtualizáció: egy fizikai erőforrás leképezése több logikailag elkülönített kiszolgáló gép megjelenítésre illetve partícióra.
- ügyfél ('kliens'), asztali gép: Vékony kliens technológia, az egyik legolcsóbb megoldás felhő-szolgáltatási környezetben; a biztonsági kérdések jelentős problémákat vetnek fel.
- adattároló virtualizáció: a fizikai adattárolás absztrakciója, amelynek eredményeként logikai adattárolót valósít meg az absztrakción keresztül.
- Hálózati virtualizáció: egy olyan környezetet teremt meg, amely több hálózati szolgáltatást üzemeltet és testre szabható egyedi, különleges igényekre egy alap hálózati réteg fölött. Ez a virtualizációs technika hardver és szoftver erőforrások kombinációjaként jelenik meg.
- szolgáltatás és alkalmazási infrastruktúra virtualizáció: Az alkalmazás virtualizálása történik meg, a hozzáférést egy központosított Web kiszolgálón keresztül valósít-

7.10 Skálázhatóság

ják meg. Az alkalmazás licenc kezelését és a szolgáltatás nyújtást a végfelhasználó felé könnyen lehet kezelni, csökkenti az alkalmazás telepítési költségeit, és lehetővé teszi az alkalmazás szolgáltatásként történő felhasználását.

- infrastruktúra virtualizáció: Az alkalmazási logikai és az infrastruktúra logika szétválasztását jelenti, az alkalmazás fejlesztőjének az alkalmazás szükséges kód kialakítására kell koncentrálnia az infrastruktúra függő kód megtervezése helyett. Az infrastruktúra virtualizáció a fizikai infrastruktúra fölött történhet. Virtuális erőforrások készlete virtuális hálózati erőforrásokkal kiegészítve segíti összekapcsolni a virtuális infrastruktúrát az erőforrások virtualizációjával.
- Erőforrás virtualizáció: Az erőforrás virtualizáció mögött meghúzódó gondolat az, hogy a felhő-szolgáltatásban az erőforrás rendelkezésre bocsátást és az adatközpont környezetet testre lehessen szabni azért, hogy ki lehessen elégíteni a munkaterhelési követelményeket valamint a virtuális erőforrások kihasználtságát felügyelni lehessen. Különböző erőforrásokat lehet a fizikai erőforrások fölött virtualizálni. Az erőforrás virtualizáció központi kérdése egy alkalmas stratégia kialakítása a szolgáltatás-központú és az erőforrás központú irányelvekre.

A virtualizáció tehát a dinamikus felhő infrastruktúrához nagyon jól illeszkedik, mivel jelentős előnyöket nyújt felhő szolgáltatásainak megosztására, az adatok és alkalmazások elszigetelésére. Továbbá lehetővé teszi azt is, hogy a felhő-szolgáltatás számítóközpontjai nem kényszerülnek rá kizárólag egy operációs rendszer használatára a virtualizációs technológiáknak köszönhetően.

7.10 Skálázhatóság

A skálázhatóság azt a képességet jelenti, hogy kezelni tudja azt a növekvő bonyolultságú helyzetet, amely további erőforrásokkal történő bővítéssel jár. A nagy mennyiségű adatfeldolgozással járó műveletek esetében a skálázhatóság alapkövetelmény a számítási felhő tekintetében.

A horizontális skálázhatóság azt a szolgáltatást jelenti, amelyről a felhő-szolgáltatás a terhelés kiegyenlítés és alkalmazási szolgáltatások nyújtása érdekében tud gondoskodni. Az elosztott zagyva táblák (Distributed hash table (DHT)), oszlop-orientáció és horizontális particionálás a horizontális skálázhatóság példái.

A **vertikális skálázhatóság** az erőforrások fel- és kihasználását jelenti, a hagyományos nagy számítóközpontok ('mainframe') megközelítéséhez hasonlóan. Ha egy alkalmazás vertikálisan nem skálázható jól, akkor annak a felhőben történő működtetése jelentős költséget fog jelenteni.

Az alkalmazások skálázhatósága egy számítási felhő környezetben alapvetően az alkalmazás jellegétől és várható felhasználási volumentől függ. Az architektúra tervezőnek gondosan vizsgálnia kell azt, hogy vajon milyen dimenziókban várható a rendszer növekedése, hol van szükség redundanciára, és rendszer heterogenitását, hogyan lehet kezelni. Az architektúra tervezőnek ismernie kell azt, hogy mely eszközök és milyen feltételek mellett használhatók, és mik a buktatók.

7.11 Terhelés kiegyensúlyozás

A terhelés kiegyensúlyozás a számítási felhő integráns része, az elasztikus skálázhatóságot a szoftver, a hardver és virtuális környezet nyújtja. Ennek az egyik mechanizmusa a munkaterhelés önszabályozása a felhő-szolgáltatás alkotórészei között (egy vagy több kiszolgáló gép, merevlemezek, hálózat, és egyéb IT erőforrások). A felhő-szolgáltatás infrastruktúrája és az adatközpontok hatalmas számítástechnikai kapacitást igényelnek, amelyek ki vannak téve a túlterhelés veszélyének, ha az igények határtalanul növekednek. Ennek elkerülése érdekében szükség van feladatok átcsoportosítására, általában a terhelés kiegyensúlyozás révén. A túlterhelés a korlátos erőforrásokból, hardver meghibásodásokból, elektromos betáplálási hibákból és hálózati szolgáltatások megszakadásból adódhatnak, amelyek szükségessé tehetik a feladatok és terhelés átcsoportosítását. A felhő-szolgáltatás egyes komponensit folyamatosan nyomon követik, monitorozzák, és amikor valamelyik alkotórész nem reagál, akkor a terhelés kiegyensúlyozót értesítik és az nem irányít további forgalmat az erőforrás felé. A terhelés elosztó segítségével lehet az egyes alkalmazások kiszolgálását illetve a kiszolgálás megszüntetését szabályozni.

7.12 A magánfelhő és a szervezeti architektúrák kapcsolata

A vállalatok, közigazgatási és kapcsolódó szervezetek között fennálló bonyolult, informatikai viszonyrendszer miatt a számítási felhő, a magán felhő, előnyeinek, hátrányainak feltárása, kiértékelése csak valamilyen szabatos, szisztematikus módszertani megközelítés révén lehetséges. Az egyik adekvát módszer az ún. **szervezeti architektúra** („*Vállalati architektúra*”,

„Enterprise architecture”). Ennek gyakorlatban alkalmazható és fejlett ipari államokban nemzetközileg bevált megközelítése a TOGAF módszer¹⁸. A felhőszámítástechnika informatikai, műszaki architektúra elemei és a közigazgatási folyamatok, szolgáltatások valamint az információrendszerek funkcionális szolgáltatásai is egységes keretben kezelhetők.

9. Táblázat Magánfelhő magas szintű architektúra építő elemei (ld. még 48. ábra)

7.1	Szerver (kiszolgáló) gépek modul
7.2	Háttértároló modul (Storage)
7.3	A hálózati kapcsolatok szövedéke modul (Fabric)
7.4	Wan modul
7.5	1. típusú végfelhasználó - hivatali helyiségben, irodában ügyintéző
7.6	2. típusú végfelhasználó – mobil, külső helyszíneken dolgozó ügyintéző

7.13 Szervezet/ vállalat által támasztott követelmények

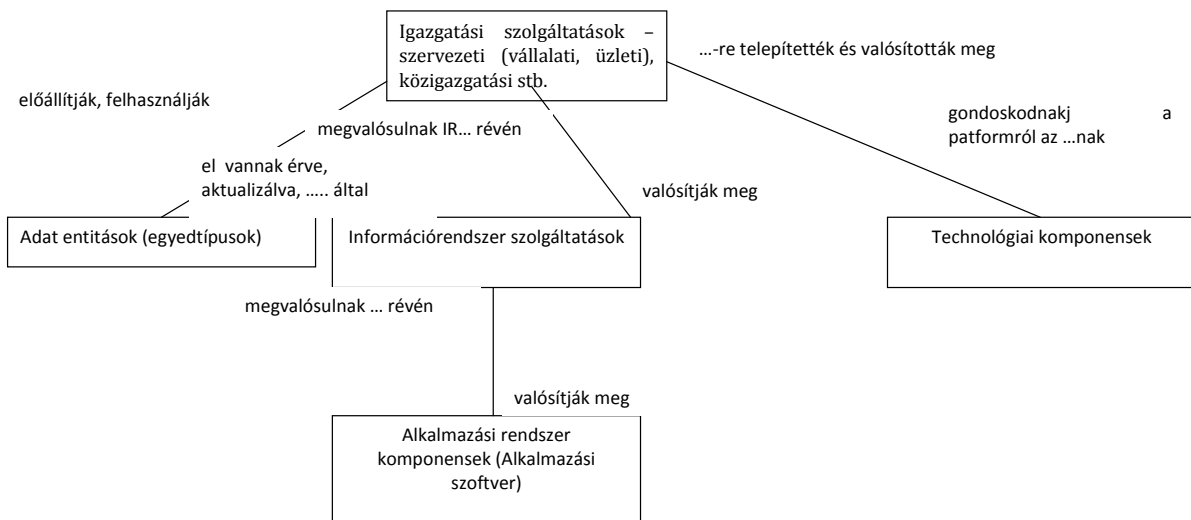
A vállalatoknak, szervezeteknek, közigazgatásnak tisztában kell lennie azzal, hogy mely szolgáltatásokért kívánnak fizetni, és különös figyelmet kell szentelni az olyan kérdéseknek mint például a *szolgáltatási szint, személyes adatok és magántitok védelme, szabályszerűség, megfelelés, adatfelelősség és adatok mozgathatósága*.

7.13.1 Felhő-számítástechnika üzemeltetési megoldásai

A felhő-szolgáltatás bárhol elérhető négy üzemeltetési vagy telepítési modell formájában.

- **A nyilvános felhő:** Alapgondolata a szolgáltatások és az infrastruktúra megosztása, a szolgáltatást egy a szervezet/vállalaton kívüli külső fél nyújtja, külső telephelyen, *több partner, „bérlő”* felé. Ez az értelmezés a felhő-szolgáltatás széles körben elfogadott definíciójának felel meg, ahol az erőforrásokat dinamikusan tudják rendelkezésre bocsátani, egy nagy finomságú teljesítmény skálán, önkiszolgáló alapokon, az Interneten keresztül, web alkalmazások / web szolgáltatások révén. A vállalatok általában a tevékenységük lényegét alkotó üzleti folyamataikat a biztonsági, irányítási és felügyeleti potenciális problémák miatt nem helyezik át a nyilvános felhőbe.
- **A magánfelhő:** A magánfelhő alapgondolata az, hogy egy partner vagy „bérlő” felé nyújtson szolgáltatást és infrastruktúrát, amelyet egy szervezet biztosít. Nem olyan

költségtakarékos megoldás mint a nyilvános felhő, de sokkal olcsóbb mint egy adatközpont beszerzése és fenntartása. A cégeknek lehetőségük van kézben tartani a felhő erőforrások feletti felügyeletet. A nyilvános felhőből a magánfelhőben tartott adatok egy adatszolgáltatási felületen keresztül elérhetők.



48. ábra A magánfelhő szervezeti architektúráis komponensei

- **A közösségi felhő:** A közösségi felhőn több szervezet osztozik és egy olyan közösség támogatja, amelynek vannak közös ügyei (stratégiai célok, biztonsági követelmények, irányelvek és szabályszerűségi és megfelelési követelmények).
- **Hibrid felhő:** Több belső és külső szolgáltatóból játszik szerepet. Költségtakarékos, skálázható és igény szerint rendelkezésre álló infrastruktúrát és biztonsági szolgáltatásokat jelent. A hibrid felhő kombinálása a hagyományos infrastruktúrával életképes megoldás lehet a legtöbb szervezet vállalat számára. A hibrid felhő azonban behoz egy bonyolultsági tényezőt, nevezetesen hogyan lehet megosztani az alkalmazásokat a nyilvános és a magánfelhő között. Ha az adatmennyiség kicsi, vagy az alkalmazás állapot nélküli akkor a hibrid felhő sokkal sikeresebb lehet, mintha nagy mennyiségű adatot kellene a nyilvános felhőbe átvinni és ott csekély mennyiségű adatfeldolgozást végezni.

A vállalatoknak, közigazgatási szervezeteknek ki kell alakítaniuk egy megfelelő stratégiát, amely mind a négy lehetőséget figyelembe veszi. A virtuális magán hálózatok segítenek az in-

formatikai erőforrások elszigetelésében, és a szervezetek informatikai erőforrásainak kiterjesztésében. Ezt a megoldást nevezik *virtuális magán felhőnek*, amely segíti a szervezet ügyviteli folyamatainak optimalizált működtetését.

7.13.2 Biztonság

A vállalatok, vállalkozások és egyéb szervezetek számára a biztonság kiemelt fontosságú terület adat, infrastruktúra és virtualizáció tekintetében. Általános elfogadott nézet az, hogy a számítási felhő új típusú kockázatokat vezet be,

A felhő-szolgáltatásban olyan adatok tárolására kerül sor, amelyeket hagyományosan a végfelhasználó számítógépén tárolnak. Ez a tény a személyes adatok és magántitok védelmével kapcsolatos aggodalmak növekedéséhez vezet, mivel a végfelhasználó elveszti a felügyeletét és ellenőrzési lehetőségeit az adatok fölött. A központosított szolgáltatások felé történő mozdulás a végfelhasználók adatforgalmának adatbiztonságát és védelmét csökkenti. A biztonsági fenyegetések az erőforrás rendelkezésre bocsátás és a végfelhasználói alkalmazások elosztott végrehajtása tekintetében jelennek meg.

Új fenyegetések megjelenése várható, pl. a számítógépes betörők ('hacker') a virtualizált infrastruktúrát fel tudják használni mint indító padot a támadások kezdeményezésére.

A személyazonosításnak (autorizáció) egy vállalati környezethez képest nagyobb finomságnak kell lennie – pl. szerepkör alapú jogosultság kezelés -, amely végig fennáll az adatok teljes életciklusa. A jogosultságok megadásának több tényezőss személyazonosításon kell alapulni., pl. egy alkalomra szóló jelszó, szervezeten belül és kívül biztonsági föderáció, kockázat alapú személyazonosítás, amely figyelembe veszi viselkedési előzményeket, a pillanatnyi környezetet stb. Föderatív egyszeri bejelentkezési protokoll a SAML (SecurityAssertion Markup Language) és OpenID , amelyeket a felhő-szolgáltatásokba történő bejelentkezés kezelését meg tudja oldani.

10. Táblázat A felhő-szolgáltatás üzemeltetési módjai és követelmények

Jellemző	Költség	Migrálás	Szervezet mérete	Biztonság	A felhőszolgáltatás fölött felügyeleti jog	Jogi kérdések
Üzemeltetési mód						
Magán felhő	Drága	Szabványos API-re van szükség a magán, nyilvános és hibrid felhő között	Nagy vállalatok, szervezetek	Magas	Magas	
Nyilvános felhő	Kevésbé drága mint a magánfelhő	Szabványos API-re van szükség az adatáramlás akadálymentessé tételéhez	Nagy és KKV-k	Alacsony, rosszindulatú tevékenységek nagy esélye, pl. túlterheléses támadás. Megbízható virtuális adatközpont szükséges	Alacsony	Nemzeti korlátozások az adattárolásra
Közösségi felhő	Viszonylag olcsóbb mint a többi (három) modell	Szabványos API-re van szükség az adatáramlás lehetővé tételéhez a közösségen belül	Kis KKV-k	Alacsony	Alacsony	
Hibrid felhő	Költségtakarékos	Szabványos API-re van szükség az adatáramlás akadálymentessé tételéhez	Több belső és külső szolgáltatás nyújtás	Alkalmazás kompatibilitási ügyek	Magas	Nemzeti korlátozások az adattárolásra

7.13.3 Biztonsági szempontok

A számítási felhő bérlője/előfizetője (*tenant*) mérhető anyagi előnye abban jelentkezik, hogy miután a kiválasztott szoftvert egy szolgáltatónál lévő számítási felhő számítóközpontban telepítik, a felhasználó ilyen jellegű beruházástól és a rendszer üzemeltetésétől mentesül. Feldolgozása viszont sok más vállalkozás rendszerével együtt kerül át egy külső számítóközpontba. A kihelyezés, a géprendszer másokkal való közös használata például az adatállományok, futtatási eredmények kezelésében adatvédelmi, biztonsági kérdéseket vet fel. „Az adatok, illetve az információ általában a legértékesebb javakhoz sorolhatóak, amelyekkel egy vállalkozás rendelkezik” ([1]) vagyis ezek védelme, a kezelésükkel kapcsolatos kockázati tényezők felmérése kiemelt fontosságú. A szolgáltatás felhő jellegéből fakadó üzleti és műszaki kockázatok széles skálájának felsorolása és definiálása, illetve ezen kockázatok kezelésének lehetőségei Racskó munkájában található. Megállapítja, hogy „Az üzleti kockázatok alapvető oka a számítási felhő-piac éretlensége, és ebből következően alacsony szintű szabályozása. Jelenleg még nem léteznek sem számítási felhőre vonatkozóan széles körben elfogadott

szabványok, sem erre vonatkozó jogi szabályozás.” ([12]). A nagy szabványosító szervezetek (pl. USA NIST) referencia architektúrák kialakításával foglalkoznak. Ezek hiányában a már meglévő adatkezelési törvényeket alkalmazzák a felhő-szolgáltatások esetében. Az EU tagországai az adatvédelemi iránymutatásnak (95/94/EG) megfelelő adatkezelési törvényeket fogadtak el az adatvédelem biztosítására. Németországban például a személyi adatok harmadik félnek történő átadására szigorú előírások vonatkoznak (§4 Abs.1 BDSG) és a feldolgozás is csak a §11 BDSG törvény betartásával engedélyezett.

EU-n kívüli adatforgalom esetén meg kell győződni arról, hogy a számítási felhő szolgáltató csatlakozott-e a **Safe Harbour-Egyezményhez** és annak előírásait biztosítja-e.

Nemzetközi, illetve nagyvállalkozások a meglévő gépparkjukon is létrehozhatnak ún. belső számítási felhő központot, mellyel a cég különböző egységein, telephelyein belüli felhő szolgáltatást biztosítják. A számítási felhő szolgáltatásra kialakított megoldások az alábbiak szerint csoportosíthatók, mely differenciálás az említett adatvédelmi, szerződéskötési, és jogi szempontból is fontos:

- **belső számítási felhő:** megvalósítás a saját számítóközpontban.
- **külső számítási felhő:** megvalósítás a szolgáltatónál, a szolgáltatási hely az EU-ban van.
- **globális számítási felhő:** a szolgáltatás EU-n kívülről, vagy ismeretlen helyről történik.
- **hibrid számítási felhő:** a leírt számítási felhő típusok kombinációja. ([1])

7.13.4 Felhő-számítástechnika üzemgazdaságtana

A számítási felhő és a felhő-szolgáltatás üzemgazdaságtanát **Cloudonomics** nevezik angolul ([1]).

A vállalatoknak választási lehetősége van a felhő-szolgáltatók árazási és számlázás megoldásai között, amelyek általában használat, fogyasztás alapúak. A felhő-számítástechnikánál hiányzik a költségek átláthatósága. Általában nagyon nehéz költség haszonelemzést végezni hagyományos infrastruktúra és a távoli felhő-szolgáltató nyújtók tekintetében (Amazon EC2, GoGrid stb.).

1. **A közmű szolgáltatások kevesebbe kerülnek még akkor is ha többbe kerülnek.**
Igény szerinti szolgáltatók tipikusan extra, közmű felárat számítanak fel - nagyobb egységárat egy erőforrásért mintha azt birtokolnák, megvásárolnák vagy lízingelnék.

Noha a közmű szolgáltatások többre kerülnek akkor, amikor használják, azonban nem kerülnek semmibe, amikor nem használják. Következésképpen az ügyfelek, fogyasztók pénzt takarítanak meg ha a saját infrastruktúrárt helyettesítik felhő-szolgáltatással abban az esetben, amikor a munkaterhelés változó, jelentős csúcsokkal. Ez különösen igaz, ha a csúcs-átlag viszony nagyobb mint a közmű szolgáltatási felár.

2. **Az igények minden előrejelzést meghazudtolnak.** A gyors szolgáltatás rendelkezésre bocsátás azt jelenti, hogy bármilyen váratlan igény kiszolgálhatóvá válik, és vele kapcsolatos árbevétele pedig realizálható. A szolgáltatás igénybevételének gyors megszüntetése azt jelenti, hogy nem termelő eszközök után nem kell fizetni. Előrejelzések mindig rosszak, ezért a gyors reakcióképesség nagyobb árbevételt és alacsonyabb költségeket jelent.
3. **Az összes igény csúcspontja soha sem magasabb mint a csúcsigények összege.** A vállalatok és szervezetek olyan kapacitásokat telepítenek, amelyekkel le tudják kezelni a csúcs igényeket. Pl. az adóigazgatás aggódik a hónap 20-ért, a kiskereskedő a hosszú ünnepek előtti bevásárlási rohamért stb. A teljes kapacitás kiépítése az egyedi csúcsok lekezelésére szolgál. Mivel a felhő sok szervezet számára nyújt szolgáltatást, ezért a különböző idejű csúcsterhelések eloszlása miatt a felhő-szolgáltatásnak ténylegesen kevesebb kapacitás fenntartására van szüksége.
4. **Az aggregált igények összessége kisimítja az egyedi igények változékonyságát.** Több ügyféltől származó az igények a változó csúcsok kisimulásához vezetnek.

7.13.5 Adat-migráció

Komoly kihívást jelent az Internet felhasználóknak szóló információk költségtakarékos és hatékony szétosztása, különösen a korszerű, különféle alkalmazások oldaláról felmerülő követelmények miatt, mint például IP telefon, IP hangtovábbítás, egyidejű adatfolyam (voice-over-IP, streaming media).

A számítási felhő környezetben speciális igények lépnek fel, amelyeknek együttesen kellene biztosítania következő célok elérését:

1. **Adatvesztés** nem lehetséges: A rendszernek garantálnia kell azt, hogy nagy valószínűséggel adatvesztés, tartósan nem következhet be.

2. **Magas rendelkezésre állás:** Az adatoknak akkor és ott kell magas valószínűséggel rendelkezésre állnia, amikor és ahol azt a felhasználó akarja, noha természetesen valamennyi, ideiglenes üzemszünet elfogadható.
3. **Magas teljesítmény:** A rendszernek nem szabad rosszabb teljesítményt nyújtani mint a szokásos file kezelő rendszereknek (pl. NFS, Network File System).
4. **Skálázhatóság:** A rendszer skálázhatóságának lehetővé kell tennie, hogy nagyszámú felhasználót, nagyszámú adattároló helyet stb. tudjon kezelni.
5. **Költség takarékos legyen:** Mivel ma már nagy kapacitású, megbízható adattároló rendszerek könnyen beszerezhetők ezért a számítási felhő szolgáltatásnak olcsónak kell lennie a hardver, szoftver és karbantartás, napra készen tartás tekintetében.
6. **Biztonság:** A rendszernek meg kell felelnie a bizalmas adatkezelésre, az adatok épségére, sértetlenségére (integritására), a személy azonosításra és hitelesítésre vonatkozó a felhasználók által elvárt szabványoknak. Mivel a számítási felhő szolgáltató az adatokat valamelyik távoli gépén tárolja, azért az adatok távoli tárolása miatt különösen komoly kihívást jelentő probléma az **adatbiztonság** és az **adatvédelem**.

A 4. és 5. tétel a számítási felhő definíció szerinti sajátossága, mivel a számítási felhő felhasználóinak „per-definitionem” nem kell foglalkozniuk hardver környezettel és a skálázhatósággal. Az első három tétel azért különösen fontos mert, a jelenlegi adatközpontok szolgáltatásai több hiányosságban szenvednek, nevezetesen üzemszünetektől, leállásoktól – amelyeket túlmelegedés, áramszünet, rack szekrény hibája, hálózati hibák, merev lemezek hibái, hálózat újrakábelezése és karbantartási munkák okoznak , továbbá természeti csapások (árvíz, vihar stb.), illetve rosszindulatú humán támadások mint például az elosztott túlterheléses támadás szolgáltatás leállítása végett, illetve a kiber-terrorizmus okozhatnak.

A három első célt tekintve az adattükrözés tűnik egy adekvát és kényelmes megoldásnak. Azonban az adattükrözés néhány alapvetően különböző sajátosságot mutat a számítási felhő kontextusában. Nevezetesen teljesen elosztottnak kell az adattükrözés megoldásnak lennie, dinamikus alkalmazkodnia kell a tükrözött példányoknak a lekérdezési terhelésekhez és a kiszolgáló gépek kapacitásához. az adatok összhangjának, ellentmondás-mentességének különböző szintjeit kell alkalmazni – nevezetesen erős, gyenge stb. - , költség takarékosnak

kell lennie, a lehető legmagasabb rendelkezésre állást kell biztosítania mind az adatfelhasználók mind az adat tulajdonosok számára.

7.13.6 Szervezeti, üzleti folyamatok szervezése, kezelése (Folyamatmenedzsment)

A szervezeti (vállalati, üzleti) folyamatok szervezésének rendszerei (BPM, *Business Process Management*) meghatároznak egy szervezeti, üzleti felépítést, a szervezeti folyamatokat, a felhasználókat, magát a szervezetet, annak szervezését és földrajzi elhelyezkedését átfogó biztonsági és ellentmondás-mentességi szabályokat. Ezt a klasszikus felfogást a számítási felhő alapú folyamatmenedzsment rendszerének kontextusában kiterjesztették. A számítási felhő egy „üzleti működtetési platformot” nyújt (**Business Operating Platform**) - elsősorban az üzleti vállalkozások, de általában különböző szervezetek számára is - , amelyben az SaaS és folyamatmenedzsment (BPM) témaköréhez tartozó alkalmazásokat kombinálnak össze. A következő vállalatirányítási rendszerekben bejáratott területek mint számítási felhő szolgáltatások jelennek meg: ügyfélkapcsolat, (customer relationship management (CRM)), munkatársak teljesítmény kezelése, (workforce performance management (WPM)), vállalatirányítási rendszerek, (enterprise resource planning (ERP)), elektronikus kereskedelmi portálok, (e-commerce portals) és így tovább. Ez a platform megoldás segít abban, hogy az összetett vállalati alkalmazásokat rendelkezésre állhassanak, rugalmasan, könnyen telepíthetően és üzembe állíthatóan, valamint ár tekintetében is elfogadhatók legyenek a szóba jöhető díjtételek.

A számítási felhő mint informatikai üzemeltetési megoldás elfogadásának gazdasági, pénzügyi értékelése szempontjából a beruházás, befektetés megtérülés elemzése (ROI, Return of Investment, NPV, Net Present Value) fontos módszer. A megtérülés értékelésének módszeréről a szakirodalomban jelentős viták folynak. A „The Open Group Cloud Computing Work Group” egy projektje a „Cloud Business Artifacts (CBA)” (A számítási felhő szervezeti tárgyi-elemei) ([1]) a következő területeket jelölte, meg mint mérendő értékelendő szempontokat:

- A számítási felhő árazása és díjtételei;
- Finanszírozási megoldások;
- Befektetés megtérülés (ROI);
- kapacitás és kihasználtság mint kulcsfontosságú teljesítmény indikátorok;
- A rendszer birtoklás teljes költsége (Key Performance Indicator (KPI));

7.13 Szervezet/ vállalat által támasztott követelmények

- Kockázat kezelés;
- A számítási felhő szolgáltatások értékelésével kapcsolatos döntés és kiválasztási folyamatok.

Ez a „CBA” módszertan ezen kívül leírja azokat a megközelítéseket, amelyek a befektetés megtérülés (ROI) mérésére alkalmazhatók, abszolút értelemben illetve az informatikai beruházások hagyományos értékelési megközelítéseivel összehasonlítva. A szervezeti/ üzleti folyamatok újrafelhasználhatósága segítheti a vállalkozásokat abban, hogy a nyereségüket maximalizálják, továbbá azok az új intelligens és innovatív üzleti folyamatok is eredményeket hozhatnak a vállalkozásoknak, amelyek például a helyzethez dinamikusan alkalmazkodó szervezeti folyamatok szervezési elveit követik.

7.13.7 Külső fél bevonása és kezelése

Külső fél megjelenése a vállalkozás gazdasági hálózatában egy masszív és megbízható kommunikációs távközlési kapcsolatrendszer kialakítását jelenti, amelyhez társulnak olyan területek mint például a számítási felhő szolgáltatás folyamatos fenntartása, jogi következmények (szolgáltatás szint megállapodás, SLA, szellemi tulajdon védelme (szerzői é szomszédos jogok, védjegyek, szabadalmak), számítási felhő auditja, és beszámoló, jelentések készítésére szolgáltatások, képességek.

A cég telephelyén létező informatika rendszerek felhő-számítástechnikába történő mozgatójának fontos előfeltétele egy megfelelő üzleti/ szervezeti modell létezése. A vállalkozásnak, vállalkozásoknak fel kell vállalniuk azt a terhet, hogy a rendszer beszállítóikat rávegyék arra, hogy alakítsanak ki egy számítási felhő licenccelési modellt, a beszállítói láncot kezelő rendszerek (*supply chain management* (SCM), vállalatirányítási rendszerek (ERP), és vállalati tartalomkezelő rendszerek (*enterprise content management* (ECM))).

A nagy, jelentős számítási felhő szolgáltatókkal az a probléma, hogy ha esetleg megsértik a szolgáltatás szint megállapodást, akkor a szolgáltatás igénybe vevő kap valamilyen kötbért, díj visszatérítést, de a *szolgáltatás folyamatos működése*, rendelkezésre állása nem garantálható. Ezért a vállalkozások é szervezetek inkább a kisebb, specializált félig-privát számítási felhő szolgáltatók irányába fognak elmozdulni, noha ezek esetében pedig a különböző kibertámadások veszélye és emiatti sebezhetőségük jelentősebb kockázatokat hordoz.

7.13.8 Szaktudás, gyakorlati tapasztalatok átvihetősége számítási felhő környezetbe

Az átvihető szaktudás, gyakorlati tapasztalatok, képességek a műszaki, technológiai, informatikai ismeretek terjesztését, megosztását, a műszaki támogatást, segítségnyújtást, a tanácsadó testületekkel, cégekkel azok szakértői csoportjaival folytatott konzultációt, vagy szolgáltatás kihelyezést jelent. Ezek az ismeretek elősegítik a rendszerek, alkalmazások adaptálását és stabilitását. A számítási felhő magával hozza a saját információ-menedzsment feladatait, amelyeket a vállalkozás, szervezet saját informatikai személyzetének kell végrehajtania, például a pillanatnyilag rendelkezésre álló, számítástechnikai kapacitást és az igények szerint növelni, illetve csökkenteni kell, illetve a fejlesztőknek új funkciókat kell megvalósítani az üzleti, szervezeti folyamatok változása miatt. Mielőtt kiválasztanának egy számítási felhő szolgáltatót, a szervezetnek át kell tekintenie a munkatársak képzettségét és fel kell tárni azokat a szakismereteket, amelyek átvihetők az új környezetbe azért, hogy az átmenetet minél simábban lehessen végrehajtani, hiszen a számítási felhő vállalati, szervezeti szoftverei és szolgáltatásai érettsége tekintetében nagy változatossággal lehet találkozni.

7.14 Végfelhasználó által támasztott követelmények

A felhasználók követelményei az egyik fontos tényezőt alkotják annak mérlegeléséhez, hogy vajon a számítási felhő szolgáltatások elfogadhatók-e a szervezet szempontjából. A számítási felhő szolgáltatásoknak megbízhatóknak kell lenniük a bizalom értelmében azért, hogy a kritikus fontosságú felhasználó adatokat a szervezet migrálja, átvigye a felhő környezetbe. A felhasználóknak biztosítékok kellene ahhoz, hogy a bizalmas adataikat, információikat megvédik, az adatvesztéstől, sérüléstől, nyilvánosságra kerüléstől. és adataik bármikor és bárhol a világon rendelkezésre állnak.

A felhasználók számára a bizalom kulcsfontosságú kérdés abban a tekintetben, hogy vajon a számítási felhő szolgáltató az adatok védelmét el tudja-e látni kielégítő színvonalon. A szervezeti, bizalmi alapon működő számítási felhő szolgáltatás a szervezeti (vállalati, üzleti) számítási felhő szolgáltatás kulcseleme. A stabilitás és adatbiztonság hasonlóan lényeges kérdés a bizalom növelése érdekében.

7.14.1 Felhasználó fogyasztásának mérésén alapuló számlázás

Az egyedi számítási felhő végfelhasználók fogyasztás mérése és fogyasztás alapú számlázása elveiben teljesen megegyezik a tipikus közművek számlázási módszerével, azaz annak megfe-

lelően, ahogy a vízművek, gázművek, elektromos művek számláznak valamilyen fogyasztási egység alapon. Az árak és a költségek kézbentartása a jövőbeli tervezés és ellenőrzés szempontjából kritikus fontosságú tényező. Segít abban, hogy a felhasznált erőforrásokat és az érték felszámított díjtételeket nyomon lehessen követni és ellenőrizni. A költségek lebontása és a költség tételek elemzése, a felhasznált szolgáltatások nyomon követése és az adaptív költségkezelés szintén lényeges eleme a költségek kézbentartásának. A végfelhasználók elvárják az átláthatóságot a szolgáltatások felhasználás és kiszámlázott tételek tekintetében. Mi az adott szolgáltatás használati *gyakorisága* ? Milyen *gyakori* a szolgáltatás használata? Nagyon gyakori használatú szolgáltatás esetében nincs sok értelme annak, hogy a felhőbe helyezték el a szolgáltatást *használat alapon történő fizetésnek* megfelelő üzleti modell keretében.

Az architektúra tervező figyelembe kell venni ezeket, az elsősorban nem-funkcionális követelmény szempontokat. A tevékenység alapú költség elszámolás (Activity-Based Costing (ABC)), egy olyan vezetői számviteli, kontrolling megközelítés, amely lehetővé teszi azt, hogy a végfelhasználók számára nyilvánvalóvá váljon az, hogy a szolgáltatások felhőszámítástechnikába helyezése, a szolgáltatások igénybevétele ily módon mennyibe kerül. Ez a megközelítés, és ennek megfelelő kimutatás elősegíti az átláthatóságot a végfelhasználó számára.

7.14.2 Felhasználó-központú személyes adatok védelme

A számítási felhő használata tekintetében a végfelhasználók egyik legfontosabb mérlegelendő szempontja a személyes adatok, üzleti adatok és titkok tárolásának lehetséges módjai. A számítási felhő magával hozza azt a sajátosságot, hogy a legtöbb végfelhasználó által előállított adat, szellemi termék, információ - amelyet a végfelhasználó saját szellemi tulajdonának tekint – valahol a világban, valamilyen szuper adatközpontokban kerülnek tárolásra. Egy ilyen környezetben a személyes adatok és magán/ üzleti titkok védelme jelentős problémaként jelentkezik. Ezért érthető, hogy a számítási felhő szolgáltatók mindent megtesznek annak érdekében, hogy a végfelhasználók bizalmát elnyerjék; azonban általában a nagyvállalatok nem nagyon hajlandók a bizalmas adataikat, üzleti titkaikat a „felhőben” tartani. Ez egy megoldhatatlannak tűnő feladat, de vannak azonban arra jelek, hogy mégis legyűrhető a fennálló akadályok.

Egyrészt, a legtöbb végfelhasználó már „*érintett*”, valamilyen számítási felhő szolgáltatást már használ, elsősorban az elektronikus levelezést (pl. gmail) vagy valamilyen Web 2.0 alkalmazást, közösségi portált (Facebook, MySpace stb.) anélkül, hogy különösebben aggódnának a személyes adataik és magántitkaik védelméért. Ennek következtében a végfelhasználók jó része, a manapság megszokottá vált adatbiztonsági és adatvédelmi környezethez már hozzászokott. Másrészt, ma már közönségesen rendelkezésre állnak azok a technológiák amelyek az adatok épségét, sértetlenségét, bizalmas adatkezelést és adatbiztonságot garantálni tudják egy számítási felhő környezetben is. E technológiák közé tartoznak:

- a tároló hely szintjén működő adattömörítési és kriptográfiai védelmi eljárások, amelynek következtében a számítási felhő szolgáltató nem tudja felhasználni a nála tárolt bizalmas adatokat;
- virtuális helyi hálózatok (virtual LAN), amelyek biztonságos távoli kommunikációt tesznek lehetővé;
- hálózati „köztes rendszerek” (mint például a tűzfalak, csomag szűrők, behatolás érzékelők stb.), amelyek tovább erősítik a biztonsági szempontból is hibátűrő kommunikációt.

Ezek a technológiák már érettnek tekinthetők, és semmilyen műszaki problémát nem jelent önmagukban történő alkalmazásuk egy számítási felhő környezetben. Nyilván ezeknek a technológiáknak az illesztése a számítási felhő iszonyatos nagy méreteihez bizonyos fejlesztési erőfeszítéseket követel meg, de ezek a feladatok végrehajthatók.

7.14.3 Szolgáltatás szint megállapodás (Service Level Agreements (SLAs))

Általában a szolgáltatók és a végfelhasználók közötti szerződést szolgáltatási szint megállapodásnak nevezik, amelyben rögzítik azokat a szolgáltatásokat, amelyeket az előre definiált feltételek mellett kell nyújtani. Természetesen jelenleg is a számítási felhő szolgáltatók felkínálnak szolgáltatási szint megállapodásokat, noha végfelhasználók szempontjából meglehetősen gyenge feltételekkel az üzemszünetekre és kötbérezési lehetőségekre vonatkozólag. A fontosabb architektúra kérdések:

- Ki és hogyan fogja mérni a szolgáltatás nyújtás paramétereit?
- Hogyan lehet kialakítani egy kölcsönösen elfogadott módszert a nyújtott szolgáltatás teljesítmény paramétereinek nyomon követésére?

7.14 Végfelhasználó által támasztott követelmények

- Mi fog történnie abban az esetben, ha a szolgáltató nem tudja teljesíteni a szerződésben előírt feltételeket?
- Milyen szolgáltatási szint szerződés módosítási mechanizmust lehet lefektetni az idők folyamán felmerülő, folyamatosan változó helyzetek kezelésére?
- Milyen kötbérezési, kártérítési eljárások legyenek, ha a szolgáltató a szolgáltatási szint szerződésben előírt elemek bármelyikét nem tartja be.

A végfelhasználók természetesen mindig stabil és megbízható szolgáltatást igényelnek. A felhő-számítástechnikáról az elfogadott kép az, hogy magas rendelkezésre állású, a hét minden napján a nap huszonnégy órájában elérhető szolgáltatás. A számítási felhő szolgáltatók komoly erőfeszítéseket tettek megbízható szolgáltatások kialakítására a nagyméretű rendszer szolgáltatásainak rendelkezésre bocsátása mellett. Ennek ellenére a legtöbb szolgáltató nem nyújt garanciákat a magas rendelkezésre állás biztosítása tekintetében. Ez különösen érdekes a **tartalomegyesítésre** vonatkozólag (**Mashup**), amikor is a tartalomegyesítés elemeit alkotó, Különböző Web szolgáltatások más-más számítási felhő szolgáltatóknál működnek, bármelyik Web szolgáltatás akármelyik számítási felhő szolgáltatónál tetszőleges időpontban, véletlenszerűen leállhat. Ha egy szolgáltatás lehet, akkor vajon egy végfelhasználó mit tehet? Hogyan érhetik el a felhőben tárolt dokumentumaikat a végfelhasználók? (Ld.: 7. Táblázat). Ilyen esetekben - ha nem is különösen vigasztaló a végfelhasználó számára, mert neki a szolgáltatásra van szüksége – a szolgáltató kötbért fizethet.

11. Táblázat A funkcionális és nem-funkcionális követelmények összehasonlítása számítási felhő esetében

7. Számítási felhő (Cloud Computing)

Nem-funkcionális követelmények	Biztonság	Interoperabilitás (együttműködési képesség)	Adatkezelés, adatgazdálkodás, tárolás dolgozás	Szolgáltatási szint megállapodás (SLA)	Mérés és számítás	Virtualizáció kezelése	Felhasználó pontú adatkezelése	köz-bizalmas hordozás, adatmigráció, konvertálás

Funkcionális követelmények

Szolgáltató	A szolgáltató igénybevevőkkel a bizalom felépítése	Nyílt adatformátumok, Nyílt API-k	Sok lehetőséggel rendelkező lekérdező nyelv, amely a skálázhatóságot mind lefelé mind felfelé lehetővé teszi. Rugalmasan és elasztikusan változtatható tárolókapacitás, ami a tartós tárolást lehetővé teszi.	99,99%-os rendelkezésre állás az SLA keretében, megbízhatóságot jelent.	A bizalom kialakításának egyik eszköze a megbízható, átlátható, hatékony fogyasztás-mérés és számítás.	A nyereséget maximalizálni a szolgáltatások, alkalmazások, hálózatok, adatbázisok, infrastruktúrák, virtualizációjával lehet.	A kölcsönös bizalom gyors felépíthetősége	Regionális adatközpontokra van szükség, ha az adatok mozgásának, migrálásának az országok határa korlátot jelent.
Nagyvállalat	A szervezeti, üzleti titkokat, valamint az üzletvitel, ügyvitel szempontjából kritikus fontosságú adatokat csak akkor lehet a felhőbe áthelyezni ha a biztonsági feltételek garantáltak	A szolgáltatások hordozhatóságának megteremtése előfeltétele annak, hogy a ügyvitel szempontjából kritikus fontosságú szolgáltatásokat a felhőből lehessen igénybe venni.	A háttértároló kapacitás elasztikus, nem kell aggódnni esetleges kapacitás hiány miatt.	Üzemszünetek estére kötbér, vagy esetleg kártérítés is követelhető.	A korrekt eljárás létezése a felek közötti kölcsönös bizalom gyors kiépüléséhez vezet.	A nyereség, profit maximalizálható	A biztonsági feltételek megteremtése a felek közötti kölcsönös bizalom gyors kiépüléséhez vezet	A kapcsolódó politikai és jogi problémák kezelésének korrekt, szabályszerű megoldása.
Végfelhasználó	A kölcsönös bizalom környezete a számítási felhő könnyű igénybevehetőségét és felhasználhatóságát teremti meg.	A végfelhasználók tetszőlegesen, bármelyik szolgáltatótól igénybe vehetik a szolgáltatást, ha az adatok hordozhatósága nem okoz gondot.	Az igény szerinti háttér tároló kapacitásból előnyök nyerhetnek.	Jogilag kikényszeríthető megállapodási forma létezése nagyon lényeges azért, hogy üzemszünet esetén díjvisszatérítés, kötbér stb. érvényesíthető legyen	A fogyasztás-mérés és számítás átláthatósága jelentősen befolyásolja a végfelhasználók hajlandóságát a felkínált szolgáltatások igénybe vételére.	A virtualizáció a számítási felhő szolgáltatásokat olcsóbbá teszi, ezért a végfelhasználók számára ez előnyös.	A végfelhasználóknak a személyes adatok és magántitkok védelmére vonatkozó szemlélete határozza meg döntően a bizalmi viszony kialakulásának lehetőségét	A regionális adatközpontokban kívánják tartani az adataikat a jogi környezettel és a szabályozással összhangban.

7.14.4 Adaptálhatóság és megtanulhatóság

A számítási felhő infrastruktúrájának sok erőforrást, adatot, szolgáltatást és végfelhasználót kell kezelnie; ezeknek az elemeknek az összessége a számítási felhő alapú nagyvállalati alkalmazások, rendszerek kézben tartása sokkal bonyolultabb feladat, a szolgáltatások és erőforrások közötti összhang fenntartása végett.

A végfelhasználók számára nagy kihívást jelent a nagyvállalatok által biztosított alkalmazási rendszerek megismerése miközben a számítási felhő környezetet is kezelni kell. A végfelhasználóknak a rendszerek kezelésével kapcsolatban olyan felhatalmazással kell rendelkezniük, hogy a saját személyes adataik, magántitkaik, bizalmas adataik fölött eredményes felügyeletet gyakorolhassanak.

A számítási felhő megoldásokat a szervezet üzleti, gazdasági vezetése szokta elfogadni. A számítási felhő felhasználása leginkább olyan kényes és bizalmas szervezeti/üzleti folyamatokra vonatkozik mint például termékek, szolgáltatások on-line vásárlása. Ha a felhasználók nincsenek teljesen tudatában a számítási felhő szolgáltatások felhasználásának módjával, akkor lehet, hogy nem származik számukra semmilyen előny a számítási felhő használatából, sőt különböző jellegű biztonsági támadásoknak tehetik ki magukat. Néhány útmutatás, amelyek segítik a számítási felhő megismerését és a hozzá történő alkalmazkodást:

- keressék meg azokat a megközelítéseket, módszereket, amelyek a felhasználói tevékenység, az adatok rögzítése módjának megfigyelésén alapulnak és azokon a felhasználói igényeken, amelyekre támaszkodva az oktatási, képzési és rendszer adaptálási környezetet a legjobb módon illeszthetik a felhasználói kívánságokhoz.
- az alkalmazásokat úgy kell megtervezni, hogy újrakonfigurálhatók, személyre és testre szabhatók legyenek.

A számítási felhő szolgáltatások intelligens és interaktív formájú bemutatása segíteni tudná a végfelhasználókat a szolgáltatások sajátosságainak megismerésében. A számítógépes, mesterséges intelligencia módszerei komoly támogatást tudnak nyújtani e területen.

12. Táblázat Nem funkcionális követelmények és Funkcionális követelmények leképezése

	Nem funkcionális követelmények	Nem funkcionális követelmények/ Funkcionális követel-	Funkcionális követelmények
--	--------------------------------	--	----------------------------

		mények	
Egyetlen üzleti folyamat	K5, K6, K8, K11, K12, K13, K16, K17, K19, K20, K21	K7, K18	K1, K4, K10, K14
Munkafolyamat vezénylése	K9, K15, K22	K2	K3

7.14.5 Végfelhasználói tapasztalatok („élmény” User Experience (UX))

A végfelhasználói tapasztalat fogalma ahhoz kötődik, hogy hogyan lehet bele látni a végfelhasználók szükségleteibe és magatartási, viselkedési módjukba, vajon hogyan lehet a használhatóságot, az igényeknek történő megfelelést és az alkalmazások termelékenységét maximalizálni. A végfelhasználó igények alapján vezérelt tervezés és üzembe helyezés a számítási felhő fejlődésének következő lépése. A felhasználók számára a képernyők és felhasználói felületek tervének egyszerűnek, nem túlszűfoltnak kell lennie, és a felhasználó által követendő munkafolyamatnak illetve szervezeti/üzleti folyamatnak kell megfelelnie a tervnek.

Szoftver-mint-szolgáltatás (SaaS) esetében az AJAX mint felhasználói felület, illetve a „Smart Client” alkalmazása érdekes tapasztalatokat eredményez a végfelhasználók számára a szoftver termékek és szolgáltatások terjedelmének és üzleti, szervezeti hasznossága, értéke szempontjából. A mobil eszközök és a számítási felhő szolgáltatások fejlődése teljesen új lehetőségeket kínálnak fel a végfelhasználói szolgáltatások interaktivitása és az aktív résztvétel tekintetében. A mobil eszközökön a kutyüket (gadgets) „húzd és dobd le” stílusban lehet kezelni, amelyek az információkat szórakoztató stílusban jelenítik meg, hírek, képek játékok formájában. Az ember-gép kapcsolat (Human-Computer Interaction, HCI), ergonómia irányelvei, a használhatóságra tervezés (usability engineering) szabályai kulcs fontosságú követelményként jelennek meg a számítási felhő szolgáltatások „végfelhasználói élménye” megtervezésének szempontjából. A jelentősebb vizsgálandó szempontok:

- A szervezeti/üzleti és végfelhasználók szempontjából potenciálisan használható szolgáltatások feltárása;
- a szolgáltatások lehetőségeinek, a problémák kontextusának, és a „végfelhasználói élmény” tervezési mintázatai alkalmazhatósága mögött meghúzódó racionális indokok;

7.14 Végfelhasználó által támasztott követelmények

- a „végfelhasználói élmény” figyelembe vételének garantálása a szolgáltatás, a termék életciklusának kezdetétől;

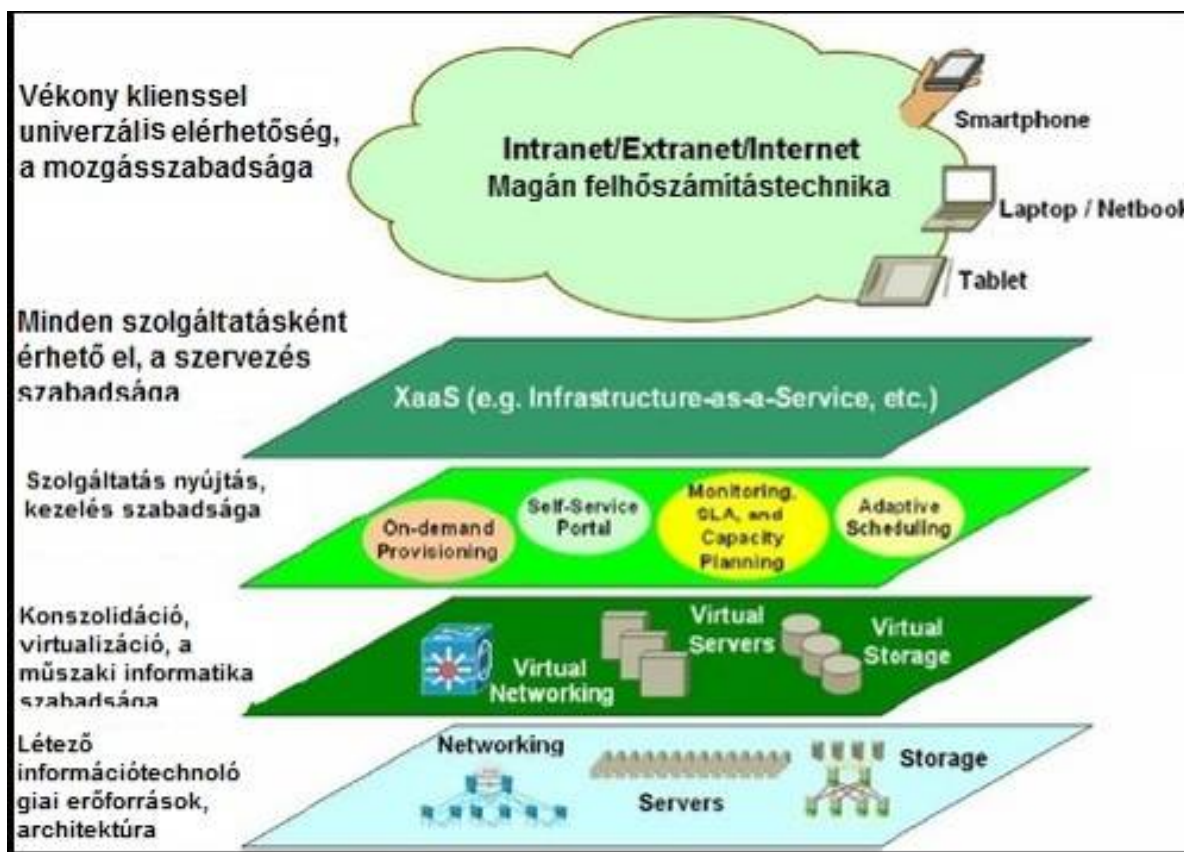
13. Táblázat A követelmények összefoglaló táblázata

1	A szolgáltató szolgáltatás-nyújtás modellje (SaaS,PaaS, IaaS)
2	Szolgáltatás-központúság
3	Adatkezelés, tárolás és adatfeldolgozás
4	Virtualizáció kezelése
5	Felhő-számítástechnika üzemeltetési megoldásai
6	Hiba-tűrő képesség
7	Biztonság
8	Szolgáltatás minősége (QoS)
9	Felhő-számítástechnika üzemgazdaságtana
10	Terhelés kiegyensúlyozás
11	Együttműködési képesség (interoperabilitás)
12	Skálázhatóság
13	Adatgazdálkodás irányítása, szervezése
14	Adat-migráció
15	Szervezeti, üzleti folyamatok szervezése, kezelése (folyamatmenedzsment)
16	Külső fél bevonása és kezelése
17	Szaktudás, gyakorlati tapasztalatok átvihetősége számítási felhő környezetbe
18	Felhasználó fogyasztásának mérésén alapuló számlázás
19	Felhasználó-központú személyes adatok védelme
20	Szolgáltatás szint megállapodás (service level agreements (SLA))
21	Adaptálhatóság és megtanulhatóság
22	Végfelhasználói tapasztalatok ("élmény" user experience (UX))

A számítási felhő alapú szolgáltatásoknak könnyen használhatóknak, megbízhatóknak, nagy sebességű és gyorsan rendelkezésre álló szolgáltatásokra kell képesnek lenniük, könnyen skálázhatóknak, testre szabhatóknak kell lenniük azért, hogy mind a globalizációs mind a lokalizációs követelményeket ki tudják elégíteni. Az internet alkalmazásoknál kilakult megoldások alkalmazhatóságát is támogatni kell, ilyen például „rich Internet applications (RIAs)”¹⁹, EyeOS. Az EyeOS. egy olyan fejlesztői környezet, amely lehetővé teszi a gyors és könnyű RIA alkalmazásfejlesztést, amely alkalmazások eredményességet, használhatóságot, rugalmasságot, sok oldalúságot, igényeknek történő megfelelést és kényelmes környezetet nyújtanak a végfelhasználók számára.

7.15 A funkcionális és a nem funkcionális követelmények elemzése

A 11. Táblázat egyrészt összefoglalóan összehasonlítja a tárgyalt követelményeket, amelyeket döntően a nem-funkcionális követelményekhez sorolunk és az architektúra tervezés egyik vezérlő szempontrendszerét testesítik meg. Másrészt megjeleníti a szervezeti architektúra egyes nézeteit és nézőpontjait.



49. ábra Számítási felhő szolgáltatás-orientált infrastruktúrája (SOI)

A szoftver architektúra tervezésnek különböző **stílusai**, megközelítései **alap-mintázatai** vannak: **ügyfél-kiszolgáló** (kliens-szerver), **csővezeték és szűrő** (pipe-and-filter), **elosztott objektumok**, **esemény-vezérelt integráció**, **virtuális gép** stb. A 12. Táblázat összefoglalja az architektúra követelmények leképezését az általános architektúra komponensekre.

Három architektúra perspektíva emelhető ki, amelyek az általános szoftver architektúra stílusokhoz köthetők – *munkafolyamat/ szervezeti folyamat vezénylése*, *egyetlen üzleti folyamat* és az architektúra *követelmények típusai*.

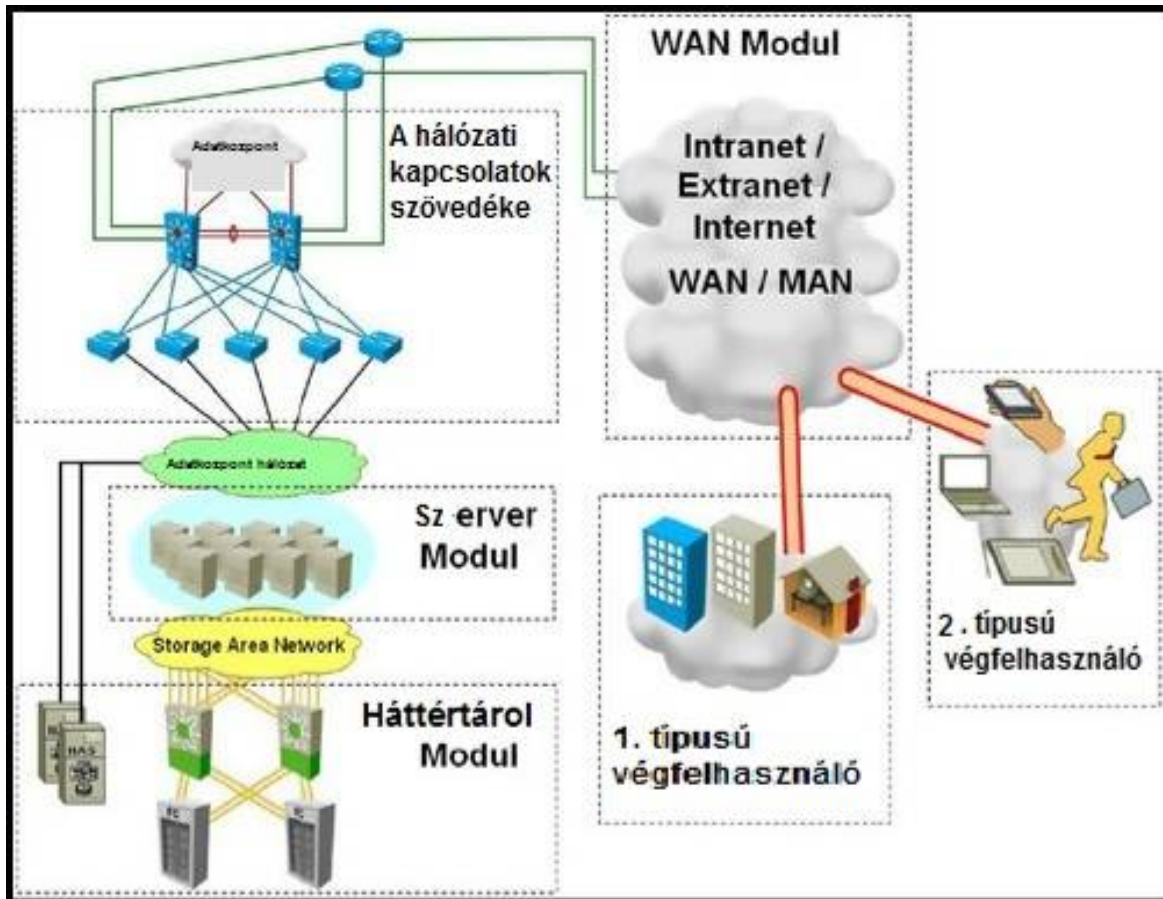
7.16 A számítási felhő és az informatikai tudomány jelenlegi állása

A munkafolyamat/ szervezeti folyamat vezénylése a csővezeték és szűrő (pipe-and-filter) architektúra mintázatnak felel meg: a szervezeti, üzleti, szolgáltatási és tudományos adatok kezelésével, gazdálkodásával, menedzsmentjével összhangban.

A **funkcionális követelmények** azoknak a rendszer által végrehajtandó műveleteknek, funkcióknak felel meg, amelyeket az alkalmazási rendszer fel fog kínálni. A **nem funkcionális követelmények** pedig azokra a minőségi szempontokra vonatkoznak, amelyek meghatározzák azt a környezetet és feltétel rendszert, amelyben a leendő funkciókat végre kell hajtani. A rendszerelemzési módszertanok előírásai szerint a *nem funkcionális követelményeket* már az elemzés korai szakaszában el kell kezdeni gyűjteni, azonban a *nem funkcionális követelmények* általában kapcsolódnak a funkcionális követelmények megfogalmazásához. A követelmények felosztása funkcionális és nem funkcionális követelményekre segít megérteni azt, hogy a felhőszámítástechnikában elhelyezett alkalmazási rendszerek felhasználása milyen forgatókönyv szerint történhet. Ez segítheti az architektúra tervezőt abban, hogy a leendő megoldással, az alkalmazási rendszerrel szemponokat és kritériumokat állítson fel az üzemeltetés, a szolgáltatási szintek tekintetében már a követelmény elemzés és a követelmény specifikáció fázisában.

7.16 A számítási felhő és az informatikai tudomány jelenlegi állása

Bármilyen szervezet számára – nagyvállalat, közigazgatási szervezet stb. – a követelmények megértése és azok viszonya a *lehetséges architektúra* megoldásokhoz kritikus fontosságú, a szervezet, a szervezeti kultúra átalakítása szempontjából egy *kooperatív szervezeti kultúra* irányába, amelyben a *szervezeti/üzleti folyamatokat* megosztják, újrafelhasználhatóvá teszik a számítási felhő szolgáltatásokon keresztül. A számítási felhő alkalmazhatóságával szemben a legnagyobb kihívás az, hogy nem létezik jelenleg (2010~) *de facto* szabvány, illetve egyetlen, egyedüli architektúra megoldás, amely megfelelne a nagyvállalati számítási felhő követelményeinek.



50. ábra A számítási felhő architektúráis építőelemei

7.17 Szinergiák kimutatása és kiaknázása

A magán felhő és az érintett vállalatok, szervezetek, közigazgatás szervezeti, üzleti szolgáltatásai, az érintett információrendszerek szolgáltatásai és az infrastruktúra főbb alkotóelemei, az alkalmazás platformok közötti összefüggés rendszert egy szervezeti architektúra leíró, modellező megközelítéssel, a TOGAF módszerrel lehet megoldani.

7.18 Számítási felhő architektúráis építőelemei

A jelenleg fennálló helyzetben is olyan sok nyitott kérdés és probléma van, hogy ezeknek a megoldására célszerű azt a bevált műszaki megközelítést követni, hogy a probléma területet annyi rétegre bontjuk, amennyi egy tervező számára átlátható, és az egyes rétegek önma-gukban is kezelhetők.

Egy ilyen architektúra réteg modellt láthatunk az ábrán. (49. ábra). Az információtechno-lógiai, architektúra elemek fölött közvetlenül láthatók azok az elemek, amelyek érintettek a virtualizációban — ez a megközelítési mód megszünteti, vagy legalábbis csökkenti azokat a

korlátokat, amelyek egy specifikus, gyártó függő számítógép modell, keménylemezes háttértároló, hálózati csatoló specifikációjának, mint modellnek a felhasználásával járna.

A szolgáltatás nyújtási réteg kijánlja az információtechnológiai, architektúra elemek részéről nyújtott automatizált szolgáltatásokat; lehetővé téve azt, hogy ezeket az erőforrásokat egyaránt jól kihasználhassák a szoftver mint szolgáltatás és az architektúráisan alacsonyabb szintű infrastruktúra szolgáltatás számára.

7.19 A számítási felhő moduljai, funkcionális blokkok

- Szerver modul (Kiszolgálók)
 - Alkalmazási szoftverek/rendszerek rétege (Application software layer)
 - Virtuális gépek szintje (Virtual machine level)
 - Virtuális hozzáférés szintje (Virtual access layer)
 - Számítási kapacitás rétege (Compute layer)
- Háttértároló modul (Storage module):
 - Háttértároló rendszerek (Storage array)
 - SAN réteg (layer)
 - SAN kiterjesztés, kiegészítés (extension)
- Hálózati elérések szövedéke (Fabric module):
 - Hálózati elérés rétege (Access layer)
 - Szolgáltatások aggregációs rétege (Aggregation layer and services aggregation layer)
 - Core layer
 - WAN (Wide Area Network) modul:
 - o Hálózati partner figyelő réteg (Peering layer)
 - o Újabb generációs WAN technológia rétege (Next-generation WAN layer).

7.19.1 Szerver (kiszolgáló) gépek modul

A számítási felhőn belül a számítógép CPU-jának (központi adatfeldolgozó egységének) analógója **szerver (kiszolgáló) gépek modulja**. A kiszolgáló gépek, a szerver farm tulajdonképpen a számítási felhő processzora. Ez a modul az, amelyet az adatközpont hálózata és a háttértárolók hálózata szendvicsként közrefog.

A virtuális gépek (VN, virtual machines), több logika kiszolgálót testesítenek meg ugyanazon a fizikai kiszolgálón. A **virtuális gépek felügyelő eszköze**, (*virtual machine monitor (VMM)*), **hypervisor** teszi ezt lehetővé-

7.19.2 Háttértároló modul (Storage module)

A háttértároló modul gondoskodik a számítási felhő adattárolási képességéről. Általában része egy SAN egy háttértároló alrendszer, amelyben található:

- Merevlemezek sorozata (disk arrays),
- Kötegbe fogott merevlemezek (bunch of disks(JBOD));
- RAID.

7.19.3 SAN kiterjesztés (extension)

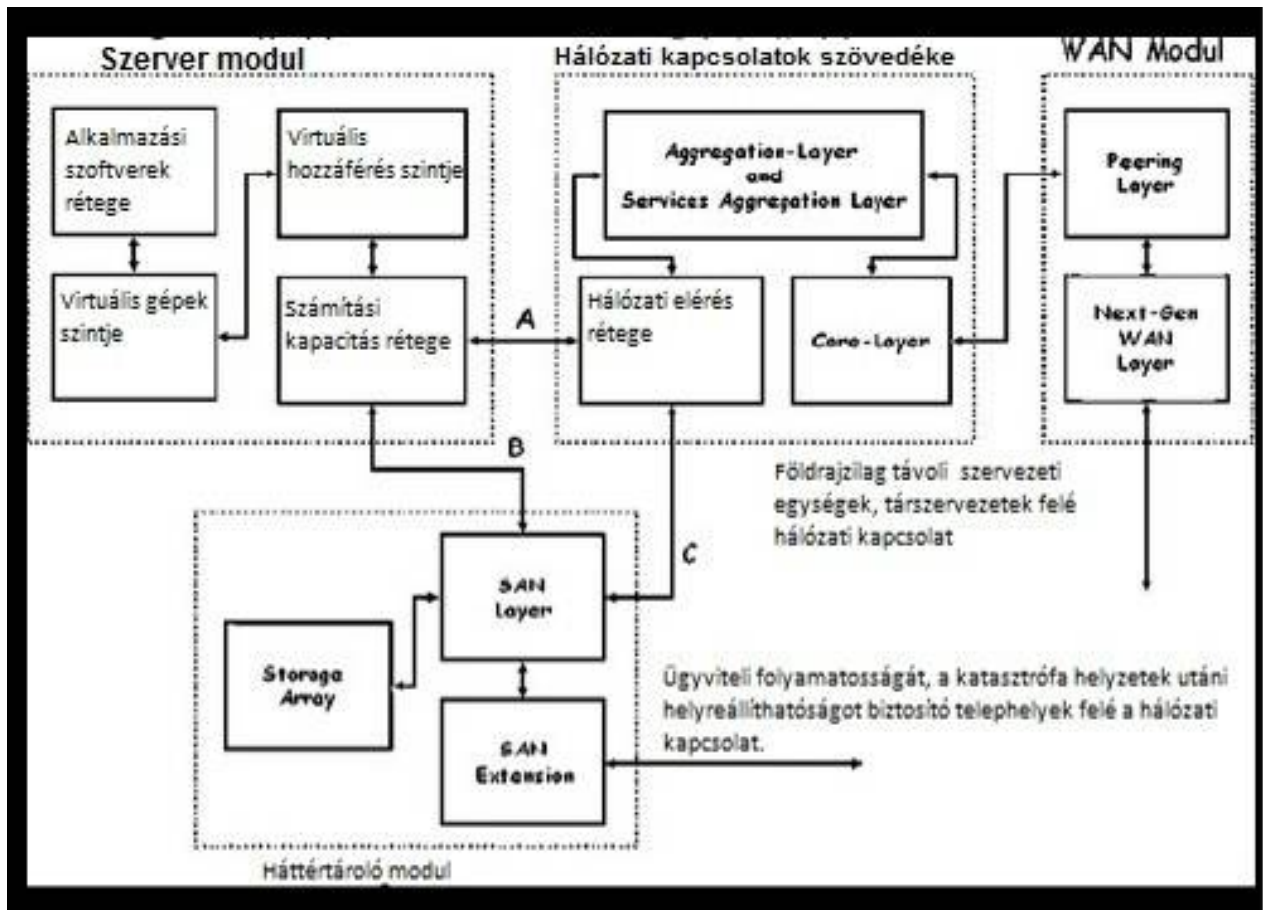
SAN kiterjesztésre akkor van szükség, ha egy vagy több háttértároló modul a **magánfelhőn** keresztül, a WAN modul segítségével kell összekapcsolni, mentés, adatmigráció adattükörzés stb. céljából.

7.19.4 A hálózati kapcsolatok szövődéke (Fabric modul)

A hálózati kapcsolatok szövődéke tulajdonképpen a számítási felhő adatsín rendszere, amely az adatokat továbbítja a számítási felhő egyéb moduljai között. Az ábrán (50.). A *szerver modulban* a szerver farm architektúra logikailag a *hálózati kapcsolatok szövődéke*, az adatközpont hálózat (tipikusan Ethernet) és egy *háttértároló modul*, egy optikai kábelhálózat SAN rendszere között helyezkedik el.

7.19.5 WAN modul

A WAN modul a „szervezet” hálózatait tartalmazza, az intranetet (belső), extranetet (külső), Intranetet (nyilvános elérés) egy WAN és MAN (Metropolitan Area Network) fölött. Ez a WAN modul az, amelyik a magán *felhő* Világhálón keresztüli elérhetőségéről gondoskodik.



51. ábra A magánfelhő architektúráis építő elemei

7.19.6 Hálózati virtualizáció

Mivel a végfelhasználók arra törekcsenek, hogy egymástól elszigetelt módon férjenek hozzá a számítástechnikai erőforrásokhoz, ezért az egyik alapvető követelmény – a magán felhő esetében is – az egymástól független és elszigetelt logikai adatforgalmi utak megteremtése a közösen használt fizikai hálózati infrastruktúra fölött, pl. a WAN fölött.

7.19.7 1. típusú végfelhasználó - hivatali helyiségben, irodában ügyintéző

Az 1. típusú végfelhasználó helye általában rögzített, helyben van, vagy távoli szervezeti egységekben, irodákban, esetleg a lakhelyén dolgozik. A hálózati hozzáférés lehet akár vezetékes, akár vezeték nélküli (3G/4G GSM).

7.19.8 2. típusú végfelhasználó – mobil, külső helyszíneken dolgozó ügyintéző

A mobil, nem rögzített helyszínen dolgozó ügyintéző esetében a mobil eszközökkel, vezeték nélküli hálózati kapcsolattal történő kapcsolattartás a bevett szokás.

7.20 Számítási felhő által nyújtható szolgáltatások típusai

A számítási felhő szolgáltatásai potenciálisan és igény szerint a következő kategóriákba sorolhatók:

1. Tesztelés mint szolgáltatás (TaaS, Testing-as-a-service);
2. Szoftver mint szolgáltatás (SaaS, Software as a Service)
3. Platform mint szolgáltatás (PaaS, Platform-as-a-service);
4. Hardver mint szolgáltatás (HaaS, Hardver-as-a-service);
5. Infrastruktúra mint szolgáltatás (IaaS, Infrastructure-as-a-service);
6. Alkalmazás mint szolgáltatás (AaaS, Application-as-a-service);
7. Köztesrendszer mint szolgáltatás (MaaS, Middleware-as a-service);
8. Mashup mint szolgáltatás (MaaS, Mashup-as-a-service);
9. Biztonság mint szolgáltatás (Security-as-a-service);
10. Együttműködés mint szolgáltatás (CaaS, Collaboration-as-a-service);
11. Információ mint szolgáltatás (InfoaaS, Information-as-a-service);
12. Keretrendszer mint szolgáltatás (FaaS, Framework-as-a-service);
13. Modellezés és Meta-modellezés mint szolgáltatás (MMaaS, Modeling & Metamodeling-as-a-service).
14. Adattároló mint szolgáltatás (StaaS, Storage-as-a-service);
15. Személyazonosítás mint szolgáltatás (IPMAaaS, Identity & Policy Management-as-a-Service);
16. Vállalat/Szervezet irányítás mint szolgáltatás (EaaS, Enterprise-as-a-Service);
17. Üzleti tevékenységek mint szolgáltatás (BaaS, Business-as-a-service);
18. Asztali gép mint szolgáltatás (DaaS, Desktop-as-a-service);
19. Adatbázis mint szolgáltatás (DBaaS, Database-as-a-service);
20. Folyamat mint szolgáltatás (PaaS, Process-as-a-service);
21. Integrálás mint szolgáltatás (Integration-as-a-service);
22. Vezetés / irányítás mint szolgáltatás (Management/governance-as-a-service);

Nem a teljes listát kimerítően de a legfontosabbak rövid magyarázatának kifejtését megadjuk.

7.20.1 Adattároló mint szolgáltatás

A „számítási felhő” szolgáltatónál létező fizikai tároló kapacitás logikailag úgy jelenik meg, mint helyi adattároló kapacitás a helyi alkalmazási rendszerek, szoftverek számára.

A szolgáltató gondoskodik a rendelkezésre állásról a hálózati vonalakon keresztül, a mentésekről stb.

7.20.2 Adatbázis mint szolgáltatás

A szolgáltatónál fizikailag távol fenntartott adatbázis, amelyet más felhasználókkal megosztanak, de logikailag a helyi alkalmazási rendszerek mint helyi adatbázisokat tudják használni. Ebben a formában egy olyan erőforráshoz lehet hozzáférni, amelyet sem nem birtokol sem nem tart fenn és működtet a végfelhasználó, így megtakarítható a hardver, szoftver és karbantartási költség.

7.20.3 Információ mint szolgáltatás (Information-as-a-service)

Ez egy olyan szolgáltatást jelent, hogy az igénybe vevő a felhőben tárolt tetszőleges (a szolgáltatás részét alkotó) információt felhasználhatja egy jól meghatározott kapcsolati felületen keresztül. Az üzleti életből vett példakkal illusztrálva: tőzsdei információk, részvény árfolyamok, cím helyesség ellenőrzés, adós/hitelkérelmező hitelképességének ellenőrzése.

7.20.4 Folyamat mint szolgáltatás (Process-as-a-service);

Ez egy olyan távoli elérésű szolgáltatás, amely több informatikai erőforrást összekapcsol. Ilyen erőforrások lehetnek pl. az erőforrások által nyújtott szolgáltatások és /vagy adatok, amelyek esetleg akár ugyan abban a számítási felhő környezetben található fel vagy akár egy másik távoli elérésű szolgáltatónál, de az összekapcsolás révén egy közigazgatási, államigazgatási, **szervezeti** (üzleti, ügyviteli, ügymeneti) **folyamatot** hoznak létre.

Egy *szervezeti folyamatra* ebben a környezetben és ebben az értelemben úgy gondolhatunk, mint egy olyan meta-alkalmazásra, amely több információrendszert fog át, segít kihasználni és rendelkezésre bocsátani azokat a szolgáltatásokat és adatokat, amelyek összessége egy lépéssorozatba kombinálva egy *folyamatot* hoz létre.

Ezeket a folyamatokat általában és tipikusan sokkal könnyebb megváltoztatni mint az alkalmazási (információ-) rendszereket.

Ez a folyamatmenedzsment központú megközelítés segíti azokat, akik kezdeményező képesek („agilisak”) abban a tekintetben, hogy a *szervezeti / üzleti /közigazgatási folyamat* értel-

mező szoftverek („folyamat motor, process engine”) felhasználását kívánság szerinti igénybevétele szempontjából ösztönzik.

7.20.5 Alkalmazás mint szolgáltatás

Alkalmazás mint szolgáltatás (AaaS), vagy szoftver mint szolgáltatás (SaaS) egy olyan alkalmazás, amelyet az interneten, a Világhálón, az informatikai hálón keresztül lehet igénybe venni, tipikusan valamilyen böngésző szoftveren keresztül.

Példák: Google Docs, Gmail, Google Calendar, Google Map. Tehát nemcsak integrált vállalat irányítási rendszerek sorolhatók ebbe a kategóriába.

Általában a jellemzőjük:

- Felhasználói felület;
- Előre definiált alkalmazási szolgáltatások, funkciók;
- Az adatalemek, adattartalom előre definiált;
- Tetszőleges számú ügyfél (kliens) platform kapcsolódhat az alkalmazási rendszerhez.

Az alkalmazás mint szolgáltatás előnye, hogy lehetővé teszi az alkalmazási rendszerek használatát a nélkül, hogy meg kellene vásárolni és/vagy üzembe helyezni egy alkalmazási rendszert az adott közigazgatási szervezetenél .



52. ábra A számítási felhő komponensei

7.20.6 Platform mint szolgáltatás (Platform-as-a-service)

Ez egy teljes **platform** szolgáltatást jelent, azaz

- alkalmazások;
- fejlesztési környezet;
- kapcsolati felület fejlesztés (adat, program stb);
- adatbázis fejlesztés;
- adattároló;
- tesztelési környezet;
- stb.

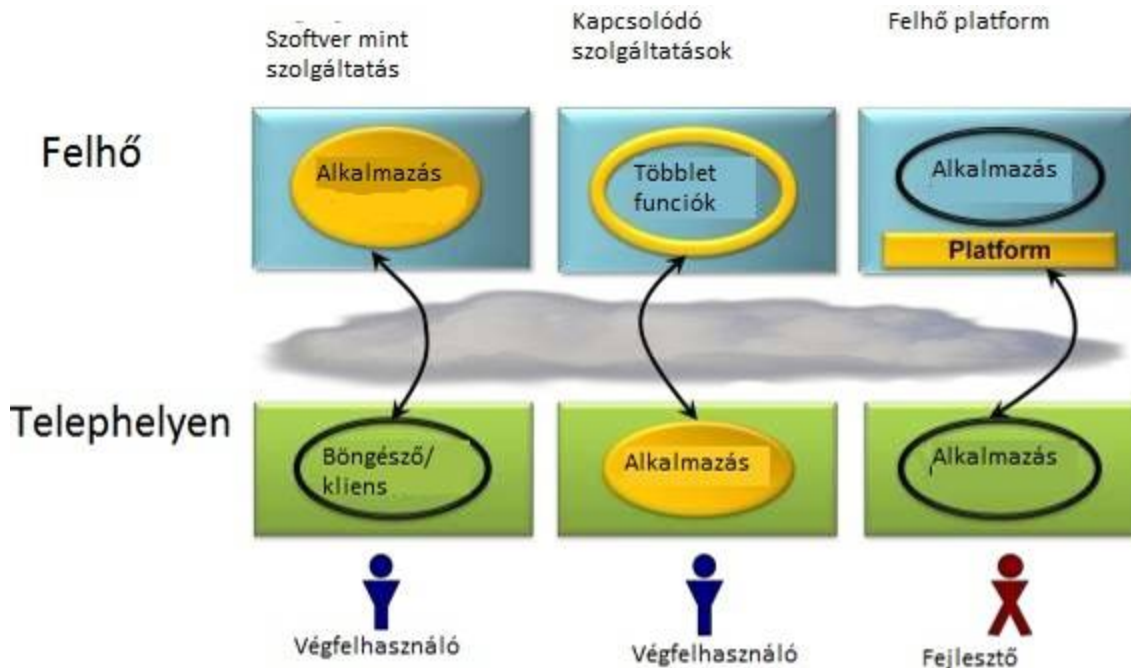
Az „előfizetők” számára nyújtott távoli elérésű szolgáltatás. *Platform mint szolgáltatás* általában a szervezetek egészére vonatkozó és érvényes alkalmazásokat tud szolgáltatni ésszerű költségek mellett.

7.20.7 Integrálás mint szolgáltatás (Integration-as-a-service)

Teljes rendszer integrálási vertikumot nyújtanak a felhő számítástechnikán keresztül, beleértve az alkalmazások összekapcsolását, kapcsolati felületek kialakítását, szemantika mediációt (közvetítést), folyamat és adatfolyam vezérlést, integráció tervezést stb. Lényegében az „Integrálás mint szolgáltatás” magában foglalja azokat a sajátosságokat és szolgáltatásokat, amelyeket egy hagyományos vállalati / szervezeti alkalmazási integráció technológia (EAI) nyújtani tud de ebben az esetben mint szolgáltatást lehet igénybe venni.

7.20.8 Biztonság mint szolgáltatás (Security-as-a-service)

A Világhálón keresztül nyújtott biztonsági, biztonság irányítási szolgáltatások, amelyek a biztonsági óvintézkedések központi vagy leglényegesebb elemeit tartalmazzák. Például a személy azonosítási rendszerek (identity management), autentikáció (azonosítás, hitelesítés), autorizáció(jogosultságok megadása), nyomon követés (AAA, Authentication, Authorization, Accounting) .



53. ábra A számítási felhő három nagyvonalú szolgáltatás kategóriája

7.20.9 Vezetés / irányítás mint szolgáltatás (Management/governance-as-a-service MaaS and GaaS)

Minden olyan kívánságra igénybe vehető szolgáltatás ide tartozik, amelyek egy vagy több számítási felhő menedzselését segítik.

Ezek általában olyan egyszerű feladatok mint például a topológia, erőforrás kihasználtság, virtualizáció, rendelkezésre állás biztosítása. Egyre több irányítást támogató (Governance) rendszer is igénybe vehető a felhő számítástechnikán keresztül mint például az irányelvek, szabályzatok érvényesítése az adatokra és a szolgáltatásokra vonatkozóan.

7.20.10 Tesztelés mint szolgáltatás (Testing-as-a-service, TaaS)

Ez olyan szolgáltatásokat jelent, amelyekkel akár helyi, akár a felhő számítástechnikán keresztül nyújtott rendszer szolgáltatásokat lehet tesztelni, bevizsgálni, különböző távoli elérhető szoftvereket és szolgáltatásaikat. Például más számítási felhő szolgáltatásokat, Web oldalakat, helyeket, belső szervezeti / vállalat irányítási rendszereket, és mindehhez nincs szükség az adott szervezeten belül hardver vagy szoftver beruházásra, üzemeltetésre.

7.20.11 Infrastruktúra mint szolgáltatás (Infrastructure-as-a-service, IaaS)

Ez fogalom tulajdonképpen egy adatközpont vagy számítóközpont szolgáltatást fed, illetve informatikai erőforrások távoli elérési lehetőségét. Lényegében egy fizikai kiszolgáló gép,

7.22 A számítási felhő potenciális előnyei

szerver bérlését, lízingelését jelenti. Az igénybevevő bármit tehet ezen a szerver gépen, és gyakorlatilag a saját számítóközpontja részeként tudja kezelni.

A szokásos számítási felhő megoldástól abban különbözik, hogy az igénybevevő a teljes kiszolgáló géphez és szoftvereihez hozzáférhet, nem egy adott kapcsolati felületen keresztül, amelyben mérik a rendszer használatot is.

Ez a megoldás sokkal kevésbé előre csomagolt, kész megoldás mint több, más számítási felhő megoldás.

7.21 A számítási felhő potenciális előnyei

Nem meglepő módon, a költségek csökkentése az, amelyik elsődleges hajtóerőként jelenik meg a számítási felhő bevezetések kapcsolatban. A költségcsökkentés az informatikai szolgáltatók oldalán is a szolgáltatásértékesítés egyik jelentős marketing tényezőjének számít. A vállalatok egyrészt azt remélik, hogy a rendelkezésre álló infrastruktúrát a számítási felhő elveinek és technológiájának adaptálása révén sokkal hatékonyabban tudják kiaknázni. Másrészt azt várják, hogy az informatikai költségeket, mint *állandó költségeket* át tudják alakítani *változó költségekké*, a külső szolgáltató felhő szolgáltatásának fajlagos használati alapon történő, térítési díj ellenében való igénybevétele révén. Az előbbi üzleti környezetből származó további előnyként jelenik meg az, hogy az informatikai költségeket nagyon finom bontásban lehet kézben tartani és elkülöníteni.

A belső és a külső felhő szolgáltatások integrálásának potenciálisan magas költségeket jelentenek az előnyökkel szemben. Ráadásul a fajlagos használaton alapuló térítési díj fizetés előnyeit is megkérdőjelezhetők. Sok alkalmazás esetében az igénybevételt inkább egy állandó terhelés jellemzi semmint egy időben gyakran változó terhelés (pl. vállalati könyvelés). A költségcsökkentéstől eltekintve, a tényleges előnyök a felhő szolgáltatások skálázhatóságában láthatók, amely a teljesítménnyel kapcsolatban eredményezhet egyes alkalmazásoknál javuló tendenciát. Ezen kívül még az informatikai szolgáltatások rendelkezésre állásának biztosításához szükséges idő és az új üzleti szolgáltatások informatikai támogatással együtt történő piaci megjelenéséhez kapcsolódó idő tekintetében várhatók jelentős előnyök.

7.22 A számítási felhő felhasználói szemszögből

A számítási felhő szoftverek rohamos terjedése egyrészt azzal magyarázható, hogy alkalmazóinak hatékonyabb információs kapcsolatrendszert kialakítását nyújtja, másrészt a szoftver használatban anyagi előnyöket biztosít a hagyományos megoldásokkal szemben.

Néhány idézet a német szakirodalomból a számítási felhőről:

- „A felhő ERP-rendszerek a helyileg installált rendszereknél jobban megfelelnek a mai elvárásoknak, melyek hatékony, problémamentes együttműködést igényelnek, ugyanis az interneten működnek és ez által minden PC-ről, - mely internet kapcsolattal rendelkezik – elérhetőek.“ (1])
- „A számítási felhő olyan szolgáltatások, alkalmazások és erőforrások sokasága, melyeket a felhasználónak flexibilisen és skálázhatóan (testre szabhatóan) az interneten keresztül ajánlanak anélkül, hogy egy hosszú távú tőke lekötés és IT-specifikus know-how lenne szükséges. A felhasználó, a vertikális integrációs mélység függvényében vagy egy komplett szoftveralkalmazást, vagy csak a szükséges informatikai infrastruktúrát tud igénybe venni.“ ([8])
- „A számítási felhő iránti nagy érdeklődés főként azzal magyarázható, hogy a vállalkozások egyre inkább felismerik, hogy a szabványosított informatikai alkalmazások üzemeltetése a cégnek nem ad igazi versenyelőnyt. Számítási felhő alatt skálázható számítástechnikai szolgáltatás értendő, melyet a felhasználó az internet technológia segítségével, igény szerint használhat“. ([9])

A számítási felhő alkalmazásának egyik kézzelfogható előnye, hogy a vállalkozás részéről jelentős mértékben csökkenti az informatikai infrastruktúra kialakításával, fenntartásával kapcsolatos beruházási és működési költségeket. Ez az előny akkor mérhető jelentősen, ha a cég gyártmányai iránt a kereslet nagymértékben ingadozik, egy vállalat terjeszkedik, vagy a vállalkozás mérete miatt egy saját számítóközpont létrehozása gazdaságilag nem lenne kifizetendő.

Nemcsak a szakirodalomban, hanem gazdaságban is napirendre került a számítási felhő költségtervezésének témája. Cégek ajánlják az interneten keresztül például TCO (*Total Cost of Ownership, a birtoklás teljes költsége*) elemzésen alapuló szolgáltatásukat, mellyel egy informatikai beruházás pénzügyi tervezése elvégezhető.

- „Az elemzés a hardver, szoftver és az üzemeltetés direkt költségein túlmenően indirekt költségfaktorokat is figyelembe vesz, mint például a kieső idők okozta termelékenységi veszteséget. Emellett nemcsak a rövidtávú megtakarítási lehetőségeket elemzik, sokkal fontosabb inkább annak vizs-

gálata, hogy egy számítási felhő-modell a szervezet részére gazdaságosabb-e, mint egy saját informatikai környezet kiépítése.” [1]

Az azonban meg kell jegyezni, hogy a gazdaságban szigorúan fennálló ökölszabály: **„Nincsen ingyen ebéd!”**. A beruházásoknak a számítási felhő szolgáltató számára is meg kell térülnie, a humán erőforrás és egyéb állandó és változó költségek fedezetét meg kell keresnie, tehát a szolgáltatást igénybe vevőnek, a bérlőnek ezt a díjakban meg kell fizetnie. A bérlő számára kockázat a saját pénzáramának csökkenése és díjat fedező bevétel megszerzése. A cégek szokásos beruházási költség és tőke (CAPEX, *Capital Expenditure*) a beruházás kivitelezésének a költségei (IMPEX, *Implementation Expenditure*) és a működési költségek (OPEX, *Operation Expenditure*) közötti stratégia és taktikai döntésekkel és kockázat mérlegelésekkel állnak szemben.

7.23 Akadályok és hátrányok a számítási felhő alkalmazásával kapcsolatban

A számítási felhő terjedésének egyik akadály a vállalati szférában különösen a külső fél által nyújtott felhő szolgáltatások esetében, a *biztonsági és szabályszerűségi* kérdésekkel kapcsolatos aggodalmak (a jogszabályi és egyéb szabályozási feltételeknek történő megfelelés, *compliance*). Az Európai Unióban és általában a tagországokban a személyi adatok védelmére vonatkozó jogszabályok nagyon szigorúak. Ezeknek az előírásoknak a betartása megoldható egy olyan magán felhő keretében, amelynek a kereteit a jogszabályokkal összhangban álló szerződésekkel szabályozzák.

A nyilvános számítási felhő szolgáltatások egyelőre nehezen használhatók fel a jogi és a hatósági szabályozással összhangban. Néhány eset jelenthet kivételt mint például egy *virtualizált* környezet használata szoftver tesztelési célokra.

Azonban a vállalatok számára jelentős aggodalomra ad okot az, hogy *kiszolgáltatottá* válhatnak egyes *szolgáltatók*, szállítók irányában, és ezt a veszélyt a számítási felhő szabványosításának jelenlegi hiánya még tovább fokozza.

A számítási felhő központú *informatikai irányítási mechanizmus* („*IT governance*”) kialakítás mind elméleti mind gyakorlati szempontból egyértelműen szüksége. Az egyik konkrét probléma kör, az *erőforrásokért* folyó versenyző igények kezelése egy megosztott, virtualizált környezetben.

7.24 Vállalati irányítási rendszerek (ERP) és a számítási felhő

A gazdasági, piaci környezet és az ezzel összefüggő vásárlói igények változása, a rövidebb termék életciklusok, a széleskörű gyártási/piaci kooperációk kialakulása mind-mind olyan tényezők, melynek követése, kézben tartása a vállalkozásokat is a gyorsabb információs kapcsolatok kialakítására ösztönzi. Az információs hálózat mai, egyik, korszerű kialakítási technológiája a számítási felhő. Sok vállalkozásnál előtérbe kerül a számítási felhő szolgáltatásokra való áttérés, illetve a szolgáltatások különböző formáinak bérlete. A potenciális végfelhasználók, cégek oldaláról a testre szabható, skálázható, olcsó ERP megoldások iránti kereslet az vállalati információrendszer szállítók/gyártók/szolgáltatók oldaláról pedig ennek az igénynek a kielégítése.

Az „ERP a felhőben” koncepció kapcsán ismét fellobbant a nyolcvanas években folytatott vita, amely a rendszerek egyes funkcióinak (pl. *penzügy, beszerzések kezelése, készletvezetés*) a szervezeti (vállalati, üzleti) folyamatok szintjén történő szabványosításával kapcsolatos. Egyes szakemberek úgy vélekednek, hogy a szervezeti (vállalati, üzleti) folyamat szintű szabványosítással a vállalatirányítási rendszerek funkcionális szolgáltatásai olcsóbbá válnak, továbbá a szoftver napra készen tartása, és karbantartása egyszerűbbé válik.

A szervezeti (vállalati, üzleti) folyamat szintű szabványosítás ellenfelei a szolgáltatótól való függőség növekedésével érvelnek. A szabványosítás mind a szervezeti (vállalati, üzleti) folyamatok mind a szoftverek funkciói szintjén bizonyos egyformaságot, egységesítést jelent. A **függőség** a vállalat irányítási rendszer használata, napi működtetése következtében fellépő kérdések megoldásának módjában; az alkalmazottak, végfelhasználók betanításában; a rendszer és szoftver hibák kijavításában jelenik meg. A látszólag szabványosított folyamatok és rendszerek mögött a nyújtott támogatási szolgáltatások jelentős különbségei húzódnak meg, a potenciálisan gyártó függetlenségnek tűnő megközelítés a támogatási szolgáltatások szintjén kialakuló jelentős függőséghez vezet.

Azonban a végfelhasználók számára a legnagyobb problémát azonban az okozhatja, ha a korábban, szabványosított funkciókkal ajánlott, és a telephelyek igényei szerint skálázott rendszert – amelyet a számítási felhőből vesznek igénybe – a vállalat tevékenységi körének bővülése miatt az új vállalati folyamatokhoz kell igazítani. Ez a kiegészítés, szoftver továbbfejlesztése csak akkor lehetséges, ha szabványosított szervezeti (vállalati, üzleti) folyamattal történik meg a bővítés, egyéb esetben a szabványosítás miatt nem lehetséges, hiszen az

adott terméket más cégek is használják. A vállalkozásoknak tehát el kell dönteniük, hogy olcsóbb számítási felhő megoldást választanak-e, vagy a magasabb költségeket okozó, egyedi szoftvereket üzemeltetnek?

A számítási felhőben működő vállalat irányítási rendszerekkel szemben további elvárás, hogy a vállalat irányítási rendszerhez például a cég beszállítói és ügyfél köre is csatlakozhasson. Ezáltal megrendelések egységes keretben adhatók fel, a termékekkel kapcsolatos információk lekérdezhetők, hozzáférhetők, anélkül, hogy vállalat illetékes szervezeti egysége humán erőforrását be kellene vonni. Ez az adott szervezet/vállalat/gazdálkodó egység *ökoszisztémájának* – (beszállító-cég-megrendelő) – a számítási felhőben egységes, központosítottan adatainak elérhetősége, feldolgozhatósága révén valósul meg. A vállalati irányítási rendszerek ilyen irányú tovább fejlesztése a vállalkozás információs hálózatának kibővítését eredményezi és a jogosult felhasználóknak az adatokhoz és információkhoz helytől független elérést biztosít.

A vállalatirányítási rendszerek, amelyek korábban a nagyvállalati szférában voltak honosak, most a számítási felhő technológia révén a KKV területen is erőteljesen terjednek. A növekvő alkalmazási számot kétségkívül a számítási felhő segíti, ugyanis a kisebb anyagi erővel rendelkező cégeknek is elérhetővé válik az új informatikai technológia által korszerű megoldások használata. A vállalatirányítási rendszerek esetében számítási felhőn keresztüli igénybevitel terjedésének okára a következő három tényező adja meg a választ:

- Szoftver architektúra: Komponensekből álló vállalatirányítási rendszer (ERP) szétválaszt és elhatárol architektúra szinteket és rétegeket. Ez az architektúra felépítés lehetővé teszi a későbbiekben módosíthatóságot. Továbbá a komponensekre alapuló architektúra lehetővé teszi a lépésenkénti, moduláris implementálást.
- Skálázhatóság: Szoftver funkcionalitás, rendszer teljesítmény: felhasználószám és kiszolgáló centrumok szintjén.
- Költségcsökkentés: A számítási felhő jelentősen csökkenti a bekerülési és használati költségeket a hagyományos ERP rendszerekkel szemben. ([6])

A vállalat irányítási rendszerek architektúra fejlődése a számítási felhő technológia ösztönzésére:

- a modularitás,
- strukturálás és
- a mobil technológia fokozott alkalmazása irányában történik

Az ügyfélszolgálati rendszerek (CRM, Customer Relationship) megoldások is a számítási felhő megoldások felé tolódnak el. Egy vállalat irányítási rendszer (ERP) sikerét a jövőben, architektúra építőelemeinek, komponenseinek sikere fogja meghatározni, amire a szállítók is a vállalatirányítási rendszer és szoftvere architektúrájának strukturálásán keresztül, fokozatosan felkészülnek.

7.25 Vállalat irányítási rendszerek (ERP) kiválasztási kritériumainak változása a számítási felhő kontextusában

Az *internet/Web/Világháló* széleskörű, mindennapi használata, illetve az erre alapuló *számítási felhő* technológia a *vállalati irányítási rendszerek* (ERP) korábbi kiválasztási, bevezetési módszereire is jelentős hatást gyakorol. A nem számítási felhőben üzemelő (hagyományos) vállalati irányítási rendszerek (ERP) kiválasztási kritériumai a cégek jelentős részében – következőkben foglalhatók össze:

- az vállalati irányítási rendszerek (ERP) szervezeti (vállalati, üzleti) *szakterület-orientáltsága*,
- a szoftverrel kapcsolatos referenciák,
- jelenlegi, illetve jövőbeni üzleti folyamatok kezelése, lefedettsége,
- pénzügyi, gazdasági szempontok,
- a konkurens iparági vállalkozások által alkalmazott vállalat irányítási rendszerek (ERP) alkalmazásának figyelembevétele,
- az vállalati irányítási rendszerek (ERP) integrálhatóságának foka, mértéke, képessége a szervezet meglévő, többi rendszeréhez,
- a vállalati irányítási rendszerek (ERP) továbbfejlesztési, módosítási lehetősége, költsége,
- a megvalósított adatfeldolgozási és szoftver technológia,
- a szállító piaci pozíciója, hosszú távú kilátásai,
- a szállító által nyújtott szakmai támogatás a bevezetés és használat során.

Nemzetközi, multi-nacionális vállalkozások esetén a „*globalizált*” vállalati irányítási rendszerek (ERP)-k kiválasztásánál még további tényezők lépnek fel:

- nyelvi verziók megléte, kezelési megoldása,
- ország-specifikus modulok és megoldások megléte,

- a szoftver bevezetéséhez szükséges támogatás színvonala, partner megléte az adott ország (ok)ban,
- az ország szintű ERP-k és a centralizált (konzernszintű) feldolgozás között kialakítható kommunikációs kapcsolat minősége, modellekhez rendelkezés, ld. ([1]), ([14]).

A számítási felhő használatával további, új kiválasztási kritériumok kerülnek be az elemzés látóterébe, terjedelmébe, amelyek az *adattvédelem, biztonság, szoftverstruktúra és architektúra, skálázhatóság, mobil eszközök integrált alkalmazhatósága* és *használati-fejlesztési költségzinttel* kapcsolatosak.

- Az adattvédelem kérdéseit országoként törvényekkel szabályozzák. Az adatok felügyelete a számítási felhő használatánál kikerül a cég informatikai vezetésének hatásköréből. Az adatok tárolásának, kezelésének védelmét, a feldolgozások biztonságát (mentések gyakorisága, helyreállítások, stb.) az *üzemeltetőnél (számítási felhő szolgáltatónál)* alkalmazott biztonsági rendszer kialakítása határozza meg.
- A szoftver és szolgáltatásai strukturálásán keresztül a vállalati irányítási rendszerekből (ERP) az éppen szükséges funkció választható ki, pl. az *ügyfél kapcsolatok* kezelése, vagy a *pénzügyi* modulok külön-külön is üzembe helyezhetők.
- A rendszer moduljait használó munkaállomások száma, az igényelt szolgáltatási szint tervezhetővé válik, és összhangba hozható a vállalat gazdasági teherbíró képességével – *skálázhatóság* révén.
- Sok alkalmazásnál ma már elvárás a *mobil* eszközök használata és azok integrált alkalmazása a vállalati irányítási rendszerekben (ERP). Szolgáltató cégek, melyek tevékenységüket a megrendelőnél végzik (p. szerviz), a mobil végberendezésről, (iPad, iPhone, MID (mobile Internet device), PDA (personal digital assistant), okos telefon, tábla gép (tablet PC) stb.) is elérhető megoldásokat keresnek.
- A vállalati irányítási rendszerek (ERP), és egyéb informatikai rendszerek fejlesztési és üzemeltetési költségeinek szintje (CAPEX és OPEX) és viszonya a többi vállalati költséghez, a KKV-k területén, a szoftver kiválasztásban meghatározó helyen szerepel. Természetes elvárás, hogy csak a tényleges használat után, tehát pl. a pénzügyi feldolgozásoknál az eseti-napi, illetve a havi, negyedéves, éves futtatásoknak megfelelően kelljen fizetni. További kritérium a kiválasztott szoftver karbantartásának, to-

vábbfejlesztésének, napra készen tartásának (pl. jogszabály követés) kérdése. Kisvállalkozások, melyek saját IT-személyzetet nem foglalkoztatnak, nyilván előnybe részesítik azon termékeket, melyekkel a szükséges szoftverfejlesztést is megajánlják.

Mindezen tényezőket a költségszint elemzésével együtt kell vizsgálni. A korábbi költségtervezési módszereket, mint pl. $ROI = \text{Return on Investment}$ (Gronau, 2010), vagy TCO (Bodri, 2008) most a számítási felhő szolgáltatás keretében kiválasztott funkcióra/szolgáltatásokra és díj fizetési konstrukciókra kell alkalmazni (tranzakció alapú, tartós bérlet stb.).

A számítási felhő szoftver *integrálhatóságának* a vizsgálata, a már meglévő alkalmazási rendszerekhez, vagy az *adatmigráció* lehetősége, mikéntje is a kiválasztási kritérium rendszer és a döntés előkészítési folyamat része. A meglévő rendszerek alkalmazás-integrációra való képessége és számítási felhőből nyújtott szolgáltatások együttműködési képessége (*interoperabilitás*) létfontosságú szervezeti és technológiai architektúra kérdéseket vet fel, és az informatikai tudomány jelenlegi állása színes, változatos és heterogén megoldásokat kínál. (ld. 1.1.4.1)

A tárgyalt, új kritériumok lehetővé teszik, hogy az ERP kiválasztási folyamatban a felhasználó igényeit a számítási felhő által nyújtott lehetőségeknek megfelelően, a korábbinál több szempontból vegyék figyelembe.

7.26 Számítási felhő jövője

A jövőben a számítási felhő széleskörű befogadására lehet számítani. Az a kérdés azonban, hogy a külső felek által nyújtott felhő szolgáltatások vajon helyettesíteni, vagy inkább kiegészíteni fogják-e a házon belüli informatikai szolgáltatásokat.

A szakértők egy része szerint a külső fél által nyújtott felhő szolgáltatások egyre nagyobb szerepet fognak játszani a jövőben, de egy *hibrid* számítási felhő környezetben, amelyben belső és külső információrendszer szolgáltatások egyaránt meg fognak jelenni. Az *informatikai részlegek* szerepe az üzemeltetési feladatokról át fog helyeződni a *szolgáltatóval történő kapcsolattartásra* és a beszállítói / szolgáltatói lánc kézbentartására jövőben. A szakemberek egy része szerint a szolgáltatások *közmű* jelleggel lesznek igénybe vehetők, lényegében megszüntetve hosszú távon a házon belüli informatikai szolgáltatás nyújtást.

Egy több vállalat által közösen üzemeltetett vállalati *közösségi* számítási felhő is életképes elképzelésnek látszik a jövőben. Ezek a modellek lehetővé teszik a méretgazdaságossági elő-

7.26 Számítási felhő jövője

nyök kiaknázását miközben az egyes vállalatok a *biztonsággal* és *szabályszerűséggel* összefüggő célkitűzések kezelését eredményes módon tudják saját hatáskörükben megoldani.

8 INFORMÁCIÓ BIZTONSÁGI ARCHITEKTÚRA (INFORMATION SECURITY ARCHITECTURE)

A hálózati biztonsági követelmények teljesítéséhez jelentős mértékben hozzájárulnak a három A-val jelzett eljárások összetevői, az *autentikáció (authentication)*, magyarul a **hiteleség vizsgálat**, hitelesítés, *autorizáció (authorization)*, magyarul **jogosultságok megadása**, engedélyezése, azaz a felhasználói jog megadása, és *accounting*, a **felhasználói tevékenységek** nyilvántartása.

Az *elektronikus autentikáció*, hitelesítés az a folyamat, amely során egy számítógépet vagy hálózati felhasználót hitelt érdemlően igazolnak, így lényegében különbözik attól, amit jogosultságvizsgálatnak nevezünk, amelyre az egyszerű, jelszavas azonosítást alkalmazzák. A tényleges hitelesítés biztosítja, hogy az a személy valójában az akinek lennie kell. A hitelesítés (autentikáció) különbözik az azonosítástól, amely azt határozza meg, hogy az adott személyt a rendszer nyilvántartja-e, és különbözik az autorizációtól, amely az azonosság alapján feljogosítja a felhasználót bizonyos speciális rendszer erőforrások elérésére.

Az **Autentikációt**, **Autorizációt** és **Accounting**-ot együttesen megvalósító rendszert nevezik „AAA” rendszernek. A biztonsági megoldások, mint az „AAA” szolgáltatás, tűzfalak, titkosítás, behatolás figyelés, az aktív auditálás, a szolgáltatásminőség biztosítása, magasabb prioritást kapnak a rendszerekben. A hozzáférés ellenőrzési lista (ACL), a ponttól-pontig protokoll szerinti jelszavak és az „AAA” szerverek alkalmazása megfelelő megoldást biztosítanak a központi alkalmazások, adatbázisok és a különböző helyszínen azokat elérni kívánó alkalmazások közötti kapcsolattartásra.

A biztonsági megoldások az alábbi három csoportba sorolhatók:

- **Azonosság** Ki az, akinek megengedett arról a helyről valamit csinálni? A biztonságos autentikáció, autorizáció és accounting szerver, és a közigazgatási CA (Certificate Authority) nyújtják ezt.
- **Sérthetlenség**. Az információt és az erőforrásokat a jogosulatlan hozzáféréstől védi tűzfalakkal, hozzáférési ellenőrzési listákkal (ACL). és titkosítással.
- **Aktív audit**. Figyeli a hálózati forgalmat, azonosítja a biztonsági kockázatokat, érvényesíti a biztonsági koncepciót és semlegesíti a jogosulatlan tevékenységet egy valósidejű behatolás érzékelővel és egy proaktív sebezhetőség felderítővel.

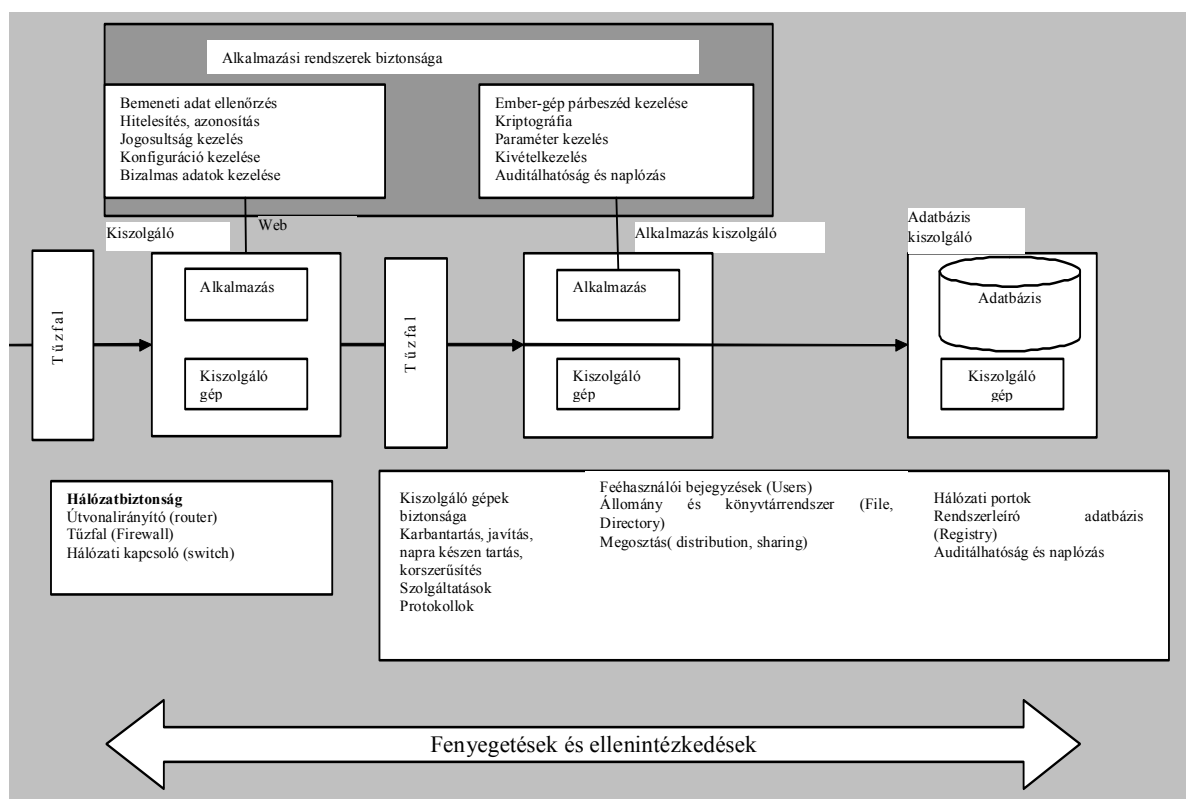
AAA, Authentication, Authorization, Accounting/Access Control

Az **AAA rendszer** egy keret rendszer, mely három egymástól független biztonsági funkciót valósít meg. Ezen funkciók a következők:

- *Authentication* - a felhasználó azonosítása mielőtt hozzáférne bármely erőforráshoz, szolgáltatáshoz a hálózatban.
- *Authorization* - ez a funkció határozza meg, hogy felhasználó milyen erőforrásokhoz, szolgáltatásokhoz férhet hozzá a hálózaton.
- *Accounting* - E funkció segítségével lehet nyomon követni, hogy egy felhasználó milyen hálózati erőforrásokat, szolgáltatásokat vett igénybe.

Az AAA rendszerek a RADIUS és a TACACS+ *protokollokat* használják kommunikációs célokra. Követelmény az AAA rendszerrel szemben:

- **Központosított menedzsment.** Valamennyi felhasználó adatai egy központi adatbázisban legyenek tárolva.
- **Skálázhatóság.** Külső adatbázis szerver használatával akár nagy mennyiségű felhasználó kezelésére is legyen alkalmas.
- Szabványos protokoll az AAA szerver és a „Network Access Server”-k (a hálózat elérhetőségéről gondoskodó kiszolgáló gépek) között.
- **Flexibilitás,** egyszerű kezelhetőség. Felhasználói csoportok definiálásával az adminisztrációs munka jelentősen csökkenthető legyen.



54. ábra Biztonsági architektúra

8.1 PKI (Publikus Kulcsú Infrastruktúra) szolgáltatások és jellemzőik

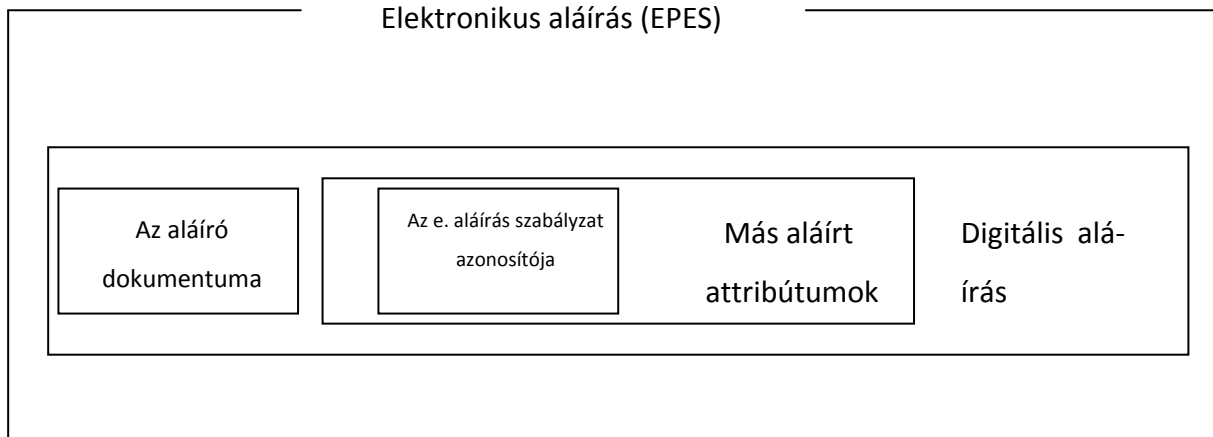
A PKI szolgáltatásai végfelhasználókat **tanúsítványokkal** (certification) látja el, amelyek a kulcspárok nyilvános kulcsát és azonosító adatait tartalmazó digitális tanúsítványok. A magánkulcsok védett tárolására és a kriptográfiai műveletek biztonságos végrehajtására mikro-számítógépet is tartalmazó kriptográfiai eszközt (intelligens kártyát, USB tokent vagy SIM kártyát) alkalmaz az infrastruktúra. A kevésbé védhető személyi számítógépes, illetve szoftveres megoldások is elterjedtek (a számítógépek merevlemez tárolói önmagukban nem biztosítanak kellő védelmet a kulcsok tulajdonos tudtán kívüli ellopása, letöltése, kompromittálása ellen)²⁰. A tároló megnyitásához, illetve a kulcsok, műveletek aktiválásához PIN kód vagy jelszó szükséges.

A felhasználói és alkalmazási követelmények kielégítése, a bizalom megerősítése céljából a PKI az informatikai biztonság fogalmkörében az alábbi szolgáltatásokat jelenti:

Azonosítás

- **Azonosítás** (*autentikáció*, vagy néhol *hitelesítés*) során hitelt érdemlő módon ellenőrizhető, hogy a kérdéses végfelhasználó az-e, akinek a személyazonosítás (identification) során mondta magát. A PKI lehetőséget nyújt a „valamit tud (PIN), valaminek a birtokában van (magán

kulcs), valami egyedi jellemzővel rendelkezik (ujjlenyomat)” együttes alkalmazására, az un. szigorú azonosításra, amikor is az igénybe vett jellemzők száma szerint kettő-, illetve három-tényező azonosításról beszélünk. Az azonosítás különböző jellemzők, attribútumok alapján nemcsak személyek, hanem számítógépek, más hálózati eszközök, elektronikus okmányok, dokumentumok esetében is szükséges és lehetséges a digitális tanúsítványok felhasználásával.



55 ábra A fokozott szintű elektronikus aláírás követelményeinek megfelelő aláírás szerkezete a ETSI szabvány szerint

- Az információ **sértetlenségét**, **épségét**, teljességét (az angol megnevezésből eredően integritásnak is mondják) a PKI digitális aláírás szolgáltatása oly mértékben biztosítja, hogy az eredetől való egyetlen karakter eltérését is jelzi. Bárki, bármikor ellenőrizni képes, hogy egy digitálisan aláírt elektronikus dokumentumot utólag nem módosítottak-e, azonos-e az aláírás kori állapotával. A *digitális* és az *elektronikus aláírás* egymással nem felcserélhető eljárás. Viszonyukat a 55 ábra szemlélteti az ETSI CADES szabvány alapján. A hitelesség megőrzése azonban csak a PKI más szolgáltatásaival együttesen biztosítható. Az elektronikus események (okirat aláírása, tranzakció, üzenet küldése, vétele) **letagadhatatlanságát** megbízható harmadik fél bevonásával lehet biztosítani. A *letagadhatatlanságot* az elektronikus aláírás önmagában csak olyan értelemben biztosítja, hogy a tanúsítványt birtokló és a használatához szükséges PIN kódok vagy jelszavak ismerője lehetett az üzenet küldő (alapfeltételezés az, hogy ez megegyezik azzal a személlyel, akinek a tanúsítványt kiállították). Ennek legelterjedtebb formája az **időbélyeg**-szolgáltató igénybevétele egyik, vagy mindkét fél részéről. Önmagában az időbélyeg csak azt igazolja hiteles módon, hogy az elektronikusan aláírt dokumentum az időbélyegzés pillanatában már létezett.
- Az elektronikus üzenetek, dokumentumok **bizalmasságát** (az eljárást elterjedten titkosításnak, helyesebben rejtjelezésnek, algoritmikus, *kriptográfiai védelemnek*, információ védelemnek

kell nevezni, esetleg titkosírásnak nevezik) - azaz, hogy jogosulatlan ne ismerhesse meg az információt - a nyilvános kulcsú kriptográfia egyszerűen kezelhető módon biztosítja. A megfelelő tanúsítvány ismeretében az üzenet címzettje részére bárki nyílt hálózaton is oly módon küldheti el elektronikus üzenetét, hogy annak tartalmához - például adóbevallása, szolgálati, üzleti titkai, magánélete, személyes adatai védelmében - csak a címzett férhessen hozzá. A PKI technológia hasonló módon alkalmas a végpont-végpont közötti elektronikus kommunikáció védelmére, illetve egyes zárt csoportok, a végfelhasználók adatainak, fájljainak védelmére. (pl. a notebook-okon tárolt adatok esetén.)

A nyilvános kulcsú infrastruktúra elemei



56. ábra A nyilvános kulcsú (PKI) infrastruktúra elemei

Személy azonosításra szolgáló tanúsítvány fokozott biztonságú vagy **minősített** elektronikus személy azonosításra alkalmas tanúsítványt tartalmaz, amely a kártya megszemélyesítésekor kerül a kártyára.

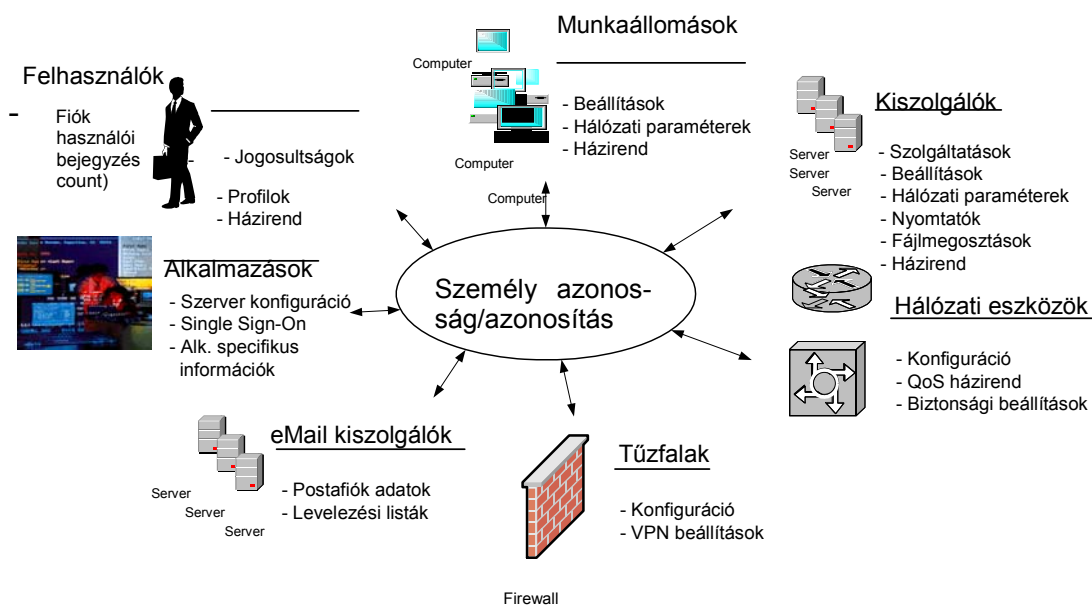
14. Táblázat Tanúsítványokkal és elektronikus aláírással kapcsolatos fogalmak

Tanúsítvány	A hitelesítés-szolgáltató által kibocsátott igazolás, amely az aláírás-ellenőrző adatot egy meghatározott személyhez kapcsolja, és igazolja e személy személyazo-
-------------	--

8.1 PKI (Publikus Kulcsú Infrastruktúra) szolgáltatások és jellemzőik

	nosságát vagy valamely más <u>tény fennállását</u> , ideértve a hatósági (hivatali) jellegget (Magyarországon törvény szabályozza, az „ Elektronikus aláírás törvény ”, <i>Eat.</i> 9. § (3), illetőleg (4) bekezdése szerin) ²¹
Minősített tanúsítvány	Az <i>Eat.</i> 2. számú mellékletében foglalt követelményeknek megfelelő olyan tanúsítvány, melyet minősített szolgáltató bocsátott ki
Fokozott biztonságú elektronikus aláírás	Olyan elektronikus aláírás, amely <ol style="list-style-type: none"> 1. alkalmas az aláíró azonosítására, 2. kizárólag az aláíróhoz köthető, 3. olyan eszközökkel hozták létre, amelyek kizárólag az aláíró befolyása alatt állnak, és 4. a dokumentum tartalmához olyan módon kapcsolódik, hogy minden - az aláírás elhelyezését követően a dokumentumon végzett - módosítás érzékelhető
Minősített elektronikus aláírás	Olyan - fokozott biztonságú - elektronikus aláírás, amelyet az aláíró biztonságos aláírás-létrehozó eszközzel hozott létre, és amelynek hitelesítése céljából minősített tanúsítványt bocsátottak ki
Minősített hitelesítés-szolgáltató	Az <i>Eat.</i> szerint nyilvántartásba vett, minősített tanúsítványt a nyilvánosság számára kibocsátó hitelesítés-szolgáltató
Elektronikus aláírás	Elektronikusan aláírt elektronikus dokumentumhoz azonosítás céljából logikailag hozzárendelt vagy azzal elválaszthatatlanul összekapcsolt elektronikus adat
Elektronikus aláírás hitelesítés-szolgáltató	Az elektronikus aláírást igénylő személyét azonosító, tanúsítványt kibocsátó, nyilvántartó szervezet
Elektronikus aláírást aláíró	Az a természetes személy, aki az aláírás-létrehozó eszközt birtokolja és a saját vagy más személy nevében aláírásra jogosult, valamint az a jogi személy vagy közhiteles nyilvántartásban szereplő jogi személyiség nélküli szervezet, amely az aláírás-létrehozó eszközt birtokolja, és akinek a nevében az őt képviselő természetes személy az elektronikus aláírást az elektronikus dokumentumon elhelyezi, valamint aki meghatározza, hogy a nevében jogszabályban meghatározott feltételeknek megfelelő informatikai eszköz elektronikus aláírást elektronikusan dokumentumon elhelyezzen
Elektronikus aláírás felhasználása	Elektronikus adat elektronikus aláírással történő ellátása, illetve elektronikus aláírás ellenőrzése
Elektronikusan történő	Elektronikus aláírás hozzárendelése, illetve logikailag való hozzákapcsolása az

aláírás	elektronikus adathoz
Elektronikus aláírás érvényesítése	Annak tanúsítása minősített elektronikus aláírás vagy e szolgáltatás tekintetében minősített szolgáltató által kibocsátott időbélyegző elhelyezésével, hogy az elektronikus dokumentumon elhelyezett elektronikus aláírás vagy időbélyegző, illetve az azokhoz kapcsolódó tanúsítvány az időbélyegző elhelyezésének időpontjában érvényes volt



57.ábra Személy azonosítás logikai/technológiai architektúra komponensek

8.2 Tanúsítvány alapú személyi azonosítás

Az azonosság megállapításához a hardver és/vagy szoftver eszköz **azonosításra alkalmas tanúsítványt** (nem elektronikus aláírást és nem titkosításra alkalmas tanúsítványt) tartalmaz.

- *Fokozott biztonságú elektronikus személy azonosításra* alkalmas tanúsítvány a kártya/adathordozó megszemélyesítésekor kerül a kártyára.
- *Minősített elektronikus személy azonosításra* alkalmas tanúsítvány a kártya megszemélyesítésekor kerül a kártyára. A tanúsítvány elektronikus aláírási tranzakcióban történő aktiválásához jelszó/PIN (jel kifejezés) kód szükséges.
- *A minősített elektronikus aláírásra szolgáló tanúsítvány* az, amely a törvény ereje folytán automatikusan egyenértékű a papír alapú dokumentációval, és kézzel aláírt iratokkal, okiratokkal.²²

8.3 Időpecsét (időbélyeg) szolgáltatás

A letagadhatatlanság alátámasztására időpecsét szolgáltatást kell/lehet igénybe venni.

Ezzel a megoldással, az egyes személy-azonosító tanúsítvány (kártya) *tranzakciónál* az eszköz tanúsítványok segítségével, az időpecsét szolgáltató időpontjával és magánkulcsával alkalmas bináris lenyomat készíthető, amely a letagadhatatlanságot egy adott időponthoz kötve tudja igazolni.

8.4 Biztonságos vonali kommunikáció által igényelt megoldások

A bizalmas (kényes) személyi és személyes adatok forgalma igényli a kommunikációs csatornák végpont-végpont algoritmikus, *kriptográfiai* védelmét.

A biztonságos vonali kommunikáció információvédelme érdekében a PKI szolgáltatások közül az eszköz tanúsítványok intenzív használatára mindenképpen szükség van.

A letagadhatatlanság egyik megoldásaként szóba jövő időpecsét szolgáltatáshoz is szükség van eszköz tanúsítványra:

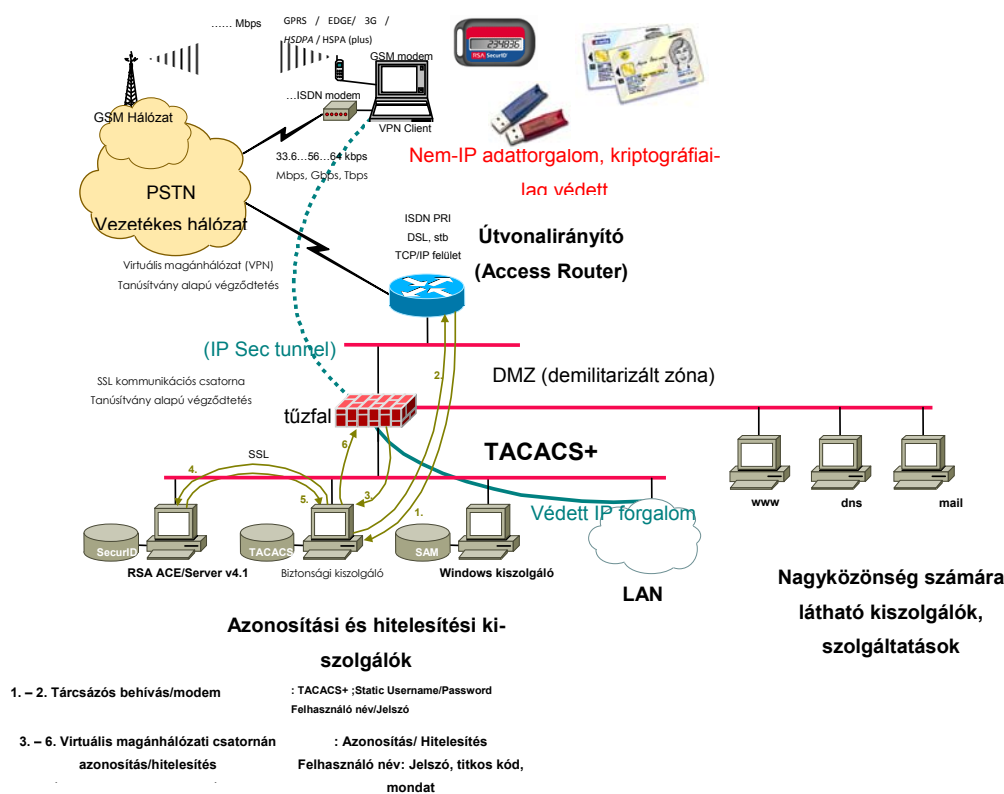
Az ügyfél oldali eszközök (kártya terminál, PC) és a központi rendszerek kommunikációjának védelme a következő eszközökkel támogatható:

- a. SSL csatorna (eszközök PKI tanúsítványára támaszkodva);
- b. VPN alagút, végződtes tűzfalon, vagy egy konkrét alkalmazási rendszeren, kiszolgáló gépen, a szóban forgó végpontok PKI tanúsítványára támaszkodva;
- c. A küldött üzenetek titkosítása, kriptográfiai védelme, sifírozása, rejtjelezése PKI tanúsítvány alapokon, rejtjelezési célokra kibocsátott tanúsítvány segítségével.

8.5 Kommunikációs és hálózatbiztonsági szempontok

Egy információbiztonságot támogató architektúra főbb elemei:

1. Kártyaterminál;
2. munkaállomás (PC);
3. Egyéb számítógép, informatikai eszköz (kiszolgáló gépek);
4. Alkalmazási/ szolgáltató rendszerek;
5. Telefonon (mobil, vezetékes), interneten végfelhasználók. (Itt elkerülhetetlenül alkalmazni kell a dinamikusan generált, korlátozott ideig érvényes titkos kód, a hitelesítés és viszontazonosítás céljára, ha olyan események fordulnak elő, amelyeknél fokozott biztonságú személyazonosításra és az azonosság hitelesítésére van szükség).
6. Világháló (Web/Internet) kapcsolat.



58. ábra Azonosítási és hitelesítési szolgáltatások adatfolyamai

A kommunikáció és hálózati biztonság egy adott szintjének megvalósításához szükségesek az A-val jelzett eljárások és azok elemei, az *autentikáció*, magyarul a **hitelesség vizsgálat**, hitelesítés, *autorizáció*, magyarul **jogosultságok megadása**, engedélyezése, azaz a felhasználói jog

megadása, és *accounting*, a **felhasználói tevékenységek** nyilvántartása. A személy vagy eszköz *hitelesítését* megelőzi az azonosítási (**identification**) eljárás.

A három „A” funkciót ellátó hálózatban a következő elemek találhatóak:

- Távoli hálózati hozzáférést biztosító szerver vagy útvonal-irányító (NAS)
- Biztonsági hozzáférést kontrolláló szerver (ACS) és a hozzáférési lista (ACL)
- Token card szerver (ideiglenesen, rövid ideig érvényes jelszavak, kódok létrehozására) telefonos vagy internetes ügyfélszolgálat számára.
- Ellenőrző/irányító adminisztrátori munkaállomás,
- X.500, X.509, illetve LDAP címszolgálat
- Tanúsítvány kibocsátás és hitelesítési szolgáltatás)

A végponttól végpontig biztonságot nyújtó architektúra az alábbi elemeket tartalmazhatja:

- Intelligens hálózati eszközök, amelyek „alkalmazás biztosak”, azaz alkalmasak a rendszergazdai és rendszer felügyeleti utasításokat értelmezni és biztosítani a biztonsági ellenőrzési követelményeket, minden felhasználó, illetve alkalmazás esetére.
- Szolgáltatásminőség és biztonsági, üzemeltetési irányelvek (*'policy'*) szolgáltatásokhoz kiszolgáló alapú ellenőrzési mechanizmust nyújtó rendszerek, amelyek a rendszeradminisztrátor és a hálózat közötti kapcsolati felületről gondoskodnak. Fordítóprogramok automatizálják a hálózatra kapcsolt eszközök konfigurálását, a központi kezelőpultról és az üzemeltetési irányelveknek megfelelően optimalizálják a hálózatot. Intézik a biztonsági beállítások, a kulcsok hálózati telepítését az egyes elemekre.
- **Regisztrációs** és **címtár** (*directory*) szolgáltatások dinamikus kapcsolatot teremtenek az üzemeltetési irányelvek és a hálózati címek, felhasználói és alkalmazási profilok, valamint biztonsági irányelvek telepítése és érvényesítése szempontjából fontos információk között. Ezeknek a szolgáltatásoknak az alapja tartománynév szerver (DNS)/ dinamikus számítógép konfigurációs protokoll (DHCP) szerver rendszer és az LDAP (Lightweight Directory Access Protocol) alapú címtárak.

8.6 A szoftver architektúra szolgáltatásaival szemben szabott általános követelmények

Az alábbi informatikai szolgáltatások, szoftver architektúra elemek részéről nyújtandó képességek és alkalmazásuk rövid és hosszútávon figyelembe veendőek az informatikai megoldások kialakításánál.

15. Táblázat Szoftver architektúra képességeivel szemben támasztandó igényfelmérési táblázat

Szoftver architektúra képességek (informatikai szolgáltatások)	E projektben/rendszerben érintett	Későbbi fejlesztésben vizsgálandó	
Rendszerfejlesztési módszerek			
Objektum-orientált és komponens alapú fejlesztés.			
Szolgáltatás-orientált architektúra és WEB szolgáltatások technológia			
Személyazonosítás és hozzáférés/jogosultság kezelése			
Címtár szolgáltatások			
Személyazonosítás föderatív (elosztott) mechanizmusa (informatikai megoldása)			
Egy jelszó, egyszeri alkalmalmmal (Single Sign On)			
Föderatív hitelesítés kezelés ('Authentication'). Elosztott hitelesítés szolgáltatás.			
Köztes szoftver rétegek ('Middleware')			
Védett és megbízható üzenet továbbítás (kriptográfiai).			
Web szolgáltatások összekapcsolásához szolgáltatások ('Adapters and Connectors')			
Ember-gép párbeszéd felületek és folyamatok			
Szerep alapú			
Eszközhöz alkalmazkodó			

8.6 A szoftver architektúra szolgáltatásaival szemben szabott általános követelmények

Szoftver architektúra képességek (informatikai szolgáltatások)	E projektben/rendszerben érintett	Későbbi fejlesztésben vizsgálendő	
képes („Eszköz-érzékelő”)			
Az ergonómiai összhang, egységesség megőrzése a különböző eszközökön, platformokon keresztül			
Szervezeti folyamatok vezénylése			
Szinkron és aszinkron műveletek			
Szolgáltatások igénybevétele, „meghívása” ('WEB service invocation')			
Adatmenedzsment			
Fürtözés („Clustering”)			
Hiba esetén feladatok automatikus átvétele ('failover')			
Katasztrófa utáni helyreállítás - Tükrözés, naplózás, mentési eljárások			
Skálázhatóság			
Vertikális és horizontális skálázhatóság ('Scale-up, Scale-out')			
Teljesítmény			
Hangolási szolgáltatások ('Tuning')			
Dialógus optimalizálása			
Biztonsági sajátosságok			
Hitelesítési és jogosultság kezelési szolgáltatások ('Authentication, Authorisation')			
Kriptográfia intenzív al-			

Szoftver architektúra képességek (informatikai szolgáltatások)	E projektben/rendszerben érintett	Későbbi fejlesztésben vizsgálendő	
kalmazása			
Magas biztonságú üzenetovábbítás			

8.7 Biztonsági architektúra követelmények

A megrendelő, felhasználó által támasztható architektúra követelmények felsorolása:

1. Hozzáférés ellenőrzés, nyomon követés
2. Bizalmas (titkos) adatkezelés
3. Üzenetek épsége, sértetlensége (integritás)
4. A szoftver érintetlensége, sértetlensége (integritás)
5. Az adatok épsége, sértetlensége (integritás)
6. Letagadhatatlanság
7. Rendelkezésre állás
8. Költség
9. Teljesítmény
10. Használhatóság

A követelmények kielégítését különböző architektúra elemekkel lehet megoldani:

1. Felhasználói azonosító
2. Jelszó, titkos kifejezés, PIN kód
3. Magas biztonsági fokozatú intelligens kártya (eszköz) (COTS)
4. Biometriaolvasó (COTS)
5. Intelligens kártyaolvasó (COTS)
6. Kártyaolvasó (COTS)
7. Kizárólagosan kriptográfiai eljárásokra szánt hardver kiszolgáló gép (kulcsolás, visszafejtés)
8. Kriptográfiai eljárásokra, algoritmusokra szoftver (kulcsolás, visszafejtés)(COTS)
9. Időpecsét szolgáltatás
10. Kriptográfiailag védett üzenetovábbítás (sifírozás, rejtjelezés, titkosítás)
11. Kriptográfiailag védett adatbázisrekord
12. Behatolás észlelő, védelmet nyújtó szoftver (Intrusion Detection/Protection Software) (COTS)
13. Vírusvédelmi szoftver (COTS)
14. Elektronikus/digitális aláírás

15. Egy jelszó, egyszeri alkalommal (Single Sign-On)
16. Fizikailag védett terem/helyiség a kiszolgáló gépeknek
17. Biztonsági őrk video kamerákkal

8.8 Web szolgáltatások biztonsági kérdései

A Web szolgáltatásokkal kapcsolatos szabványok és protokollok (ld. 3.9.1) a Web szolgáltatásoknak csak a gerincét adják, azon túl még számos kiegészítésre és további részprobléma megoldására van szükség ahhoz, hogy az egész mechanizmus megbízhatóan működő rendszerre álljon össze. Az egyik terület a **biztonság** kérdése, nevezetesen a *hitelesítés* (*authentication*) és a *jogosultságok* megadása (*authorization*) problémája, továbbá a rendszer/szolgáltatás *hozzáférési jogosultságok* (*access control*) kezelése (AAA, ld. 8.1).

Az egyik leggyakoribb probléma, amivel a köztesszoftver protokollok szembenéznek az, hogy nem kifejezetten működnek jól az Interneten, mivel kapcsolatokat akadályozzák a tűzfalak. A legtöbb szervezet nem szeretné, ha az általuk használt protokollok kívülről is hozzáférhetővé válnának, ezért a belső rendszereket tűzfalakkal veszik körbe.

Egy gyakori megoldása ennek a problémának, melyet a Web szolgáltatások is alkalmaznak: a **Web protokollok** alkalmazása, mivel a HTTP-t mint transzport protokollt a legtöbb tűzfal átengedi. A HTTP protokoll ilyen jellegű alkalmazása kényelmes lehet, de ugyanakkor biztonsági fenyegetést is jelenthet, mivel a HTTP kapcsolatot már nem csak arra használjuk, hogy weboldalakat töltsünk le.

A *WS-Security* és a hozzá kapcsolódó szabványok azt tűzték ki célul, hogy ezeket a problémákat erős kriptográfiai módszerekkel küszöböljék ki. Ezt pedig a hívók azonosításával (autentikáció) és az információ védelmével (titkosítás) teszik meg, valamint biztosítják az információ integritását (épségét, sértetlenségét) (pl. digitális aláírással). Ezeket a szabványokat úgy alkották meg, hogy könnyedén kiterjeszthetőek és adaptálhatóak legyenek.

A Web szolgáltatás szabványok támogatják a tranzakciókat és megbízható üzenetküldést. A Web szolgáltatás tranzakciók két típusát támogatja a szabvány. A *WS-AtomicTransactions* a hagyományos elosztott **ACID** (Atomicity (atomicitás), Consistency (konzisztencia), Isolation (izoláció), és Durability (tartósság)) tranzakciókat támogatja és bizonyos szintű biztonságot valamint gyors válaszidőt feltételez. Emiatt ez a módszer csak a belső integrációs feladatoknál alkalmazható, az interneten átívelő alkalmazásoknál nem. A *WS-BusinessActivity* egy

olyan keretrendszer és protokollgyűjtemény, amely a lazán csatolt integrált alkalmazások megszakításának koordinálására szolgál.

A megbízható üzenetküldés támogatása a Web szolgáltatások esetében egyszerűen azt jelenti, hogy biztosítjuk azt, hogy minden elküldött üzenet biztosan célba ér pontosan abban a sorrendben, ahogy elindítottuk őket. A *WS-ReliableMessaging* nem garantálja a biztos célba érést egy fellépő meghibásodás esetén, de a **üzenet kezelős sor** (message queue) megoldások képesek erre egy állandó, perzisztens tár használatával.

8.8.1 WS-Security

A nyilvános kulcsú infrastruktúra (PKI), az SAML és a hasonló szabványosítási erőfeszítések és szolgáltatások megvalósítása tudja segíteni a Web szolgáltatásokkal kapcsolatos biztonsági problémák kezelését.

Web szolgáltatás biztonsága WS-Security **WS-Security (Web Services Security, röviden WSS)** A SOAP üzenetcsere protokoll kiegészítése, amelynek segítségével a Web szolgáltatások biztonsági sajátosságait lehet definiálni.

- **A WS-* (Web szolgáltatás)** leíró szabványokhoz tartozik (OASIS²³)

Ez a Web szolgáltatás leíró szabvány és protokoll az üzenetek **integritására** (*épségét, sértetlenségét*) és **titkosítására** (*rejtjelezésére, sifrírozására*) vonatkozó szabályok leírását teszi lehetővé, és ezen keresztül határozza meg a titkosítás módját és az integritás érdekében hozott óvintézkedéseket. Sokféle biztonsági zsetont (token) támogat; nevezetesen például : X.509, SAML, Kerberos, PKI. A fő hangsúly ennél a protokollnál az *XML Signature* és az *XML Encryption* használatán van.

Az SAML (8.11) protokoll a biztonsággal kapcsolatos három lényeges információ kicserélését támogatja a tények megerősítése végett:

(1) a hitelesítést (*authentication*),

(2) a jogosultságok megadását (*authorization*) – mely szolgáltatások használatára jogosult vagy nem jogosult –;

(3) és a szolgáltatás sajátosságaitól függő egyéb attribútumok értékének ellenőrzését és érvényesítését; a szolgáltatásra előírt szervezeti (vállalati, üzleti) szabályok betartása végett azért, hogy a szolgáltatással végrehajtandó műveletek elvégzésének engedélyezés megtörténhessen.

Ezekre az információcserékre vonatkozó előírásokat XML dokumentumok formájában rögzítik. Az SAML ezen kívül *kérelem-válasz (request-response)* üzenet párok leírását, definiálást teszi lehetővé azért, hogy a **megerősítésekhez (assertions)** szükséges információcserék lefolytathatók legyenek.

A WS-Security az SAML protokoll biztonsági ellenőrzéseihez kapcsolódóan azt írja le, hogy ezek a *megerősítések (assertion)* és egyéb biztonsági *zsetonok (token)* hogyan kapcsolódnak a SOAP üzenetek fejlécéhez. Ez a SOAP üzenet protokoll sajátosságainak kibővítése, amelynek révén ez a protokoll kiterjesztés *védelmi óvintézkedésekről* gondoskodik, nevezetesen az üzenetek *integritásának* (épségének, sértetlenségének), *bizalmosságának* (confidentiality) és az adott üzenet *hitelességének/hitelesítésének* (authentication) a védelméről. E célok megvalósítása érdekében több biztonsági modellt, mechanizmust lehet alkalmazni: *PKI, SSL, X.509, Kerberos*:

- 1) X.509 *tanúsítványok (certificate)*;
- 2) Kerberos „jegyek” (tickets);
- 3) Felhasználó azonosító / jelszó megbízólevél;
- 4) SAML-Assertion (*megerősítés*);
- 5) Felhasználó által definiált zseton (token).

A WS-Security három fő mechanizmust ír le:

- Hogyan azonosítsuk a SOAP üzeneteket, hogyan őrizzük meg az üzenetek integritását. Az üzenetek jelölése "*letagadhatatlan*" (*non-repudiation*) legyen.
- Hogyan titkosítsuk (rejtjelezzük, sifrírozzuk) a SOAP üzeneteket.
- Hogyan csatoljunk az üzenetekhez olyan biztonsági zsetonokat (tokeneket), amellyel meggyőződhetünk a küldő személy azonosságáról.

A specifikáció lehetőséget ad különböző aláírás formátumok, titkosítási algoritmusok és több bizalmi tartományok használatára.

8.8.2 WS-Trust

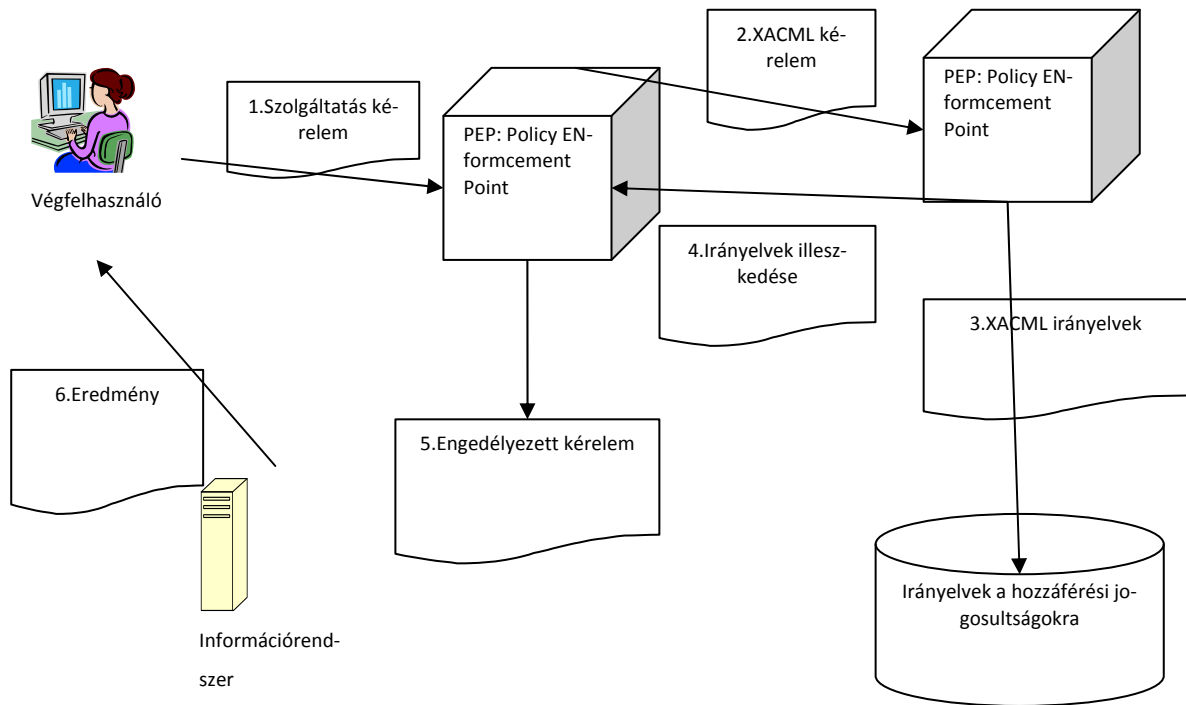
A WS-* szabvány család része, amelyet szintén az OASIS²³ dolgozott ki. A WS-Trust egy olyan keretrendszert nyújt, amelynek segítségével egy bizalmi hierarchiát vagy hálózatot lehet kialakítani. Valójában egy olyan Web szolgáltatás, amelyik biztonsági zsetonok (token) kiadásáért, megújításáért és ellenőrzésért felel.

Két fél közötti biztonságos információcsere megvalósítása érdekében a két fél között olyan információcserének kell lefolynia, amelyben felek azonosságnak igazolására szolgáló igazolványok, **bizonyítványok** (*credentials*) cseréjére sor kerül. A bizonyítványok cseréje közvetlenül vagy közvetve is történhet (egy megbízható harmadik fél közbeiktatásával), a *WS-Security* protokoll segítségével. Azonban mindegyik résztvevő félnek meg kell bizonyosodnia arról, hogy vajon megbízhatnak-e a másik fél bizonyítványában. A **Web Services Trust** (*WS-Trust*) nyelv a *WS-Security* biztonságos üzenetcsere mechanizmusát használja fel arra célra, hogy további elemi nyelvi egységeket (primitíveket) és további kiegészítéseket definiáljon a biztonsági zsetonok (tokenek) kibocsátásár, cseréjére és érvényességük ellenőrzésére. Ily módon a **kölcsönös bizalom** a *biztonsági zsetonok (tokenek) cseréjén, közvetítésén és érvényesítésén keresztül jön létre és reprezentálódik. WS-Trust a különböző bizalmi tartományok között, igazolványok, bizonyítványok kiadására és szétosztására, eljuttatására szintén rendelkezik egy mechanizmussal. Az alkalmazások, Web szolgáltatások felhasználva ezeket a biztonsági mechanizmusokat, biztonságos módon tudnak információt cserélni, miközben természetesen a Web szolgáltatások keretrendszerére támaszkodnak: SOAP+WSDL+UDDI+HTTP.*

Egy Web szolgáltatás először a biztonsági sajátosságait a **WS-Policy** formájában fogalmazza meg, deklarálja (*claims*). Amikor SOAP üzenet formájában egy olyan **szolgáltatási kérelem** (*request*) érkezik meg egy Web szolgáltatáshoz, amely tartalmaz egy biztonsági zsetont (token), akkor a *WS-Trust* megköveteli, hogy az előírt biztonsági igényeknek feleljen meg, azonosítsa magát a nevével, kriptográfiai kulccsal, és (műveleti) engedélyekkel (*deklaráció*). Egyébként a Web szolgáltatás tudomást sem vesz a kérelemről, vagy megtagadja a kérelem kiszolgálását. A *kérelem* valóságának és érvényességének ellenőrzése céljából, a Web szolgáltatásnak rendelkeznie kell egy „*bizalmi motorral*” (*trust engine*), szoftver modullal, amely:

- 1) Leellenőrzi, hogy a kérelmező szolgáltatás deklarációi (*claims*) illeszkednek-e a szolgáltatás által előírt, a biztonsági irányelvekben (*WS-Policy*) szereplő követelményekhez;
- 2) Leellenőrzi, hogy az elektronikus/digitális aláírás bizonyítja-e, hogy a deklarációt tevő állításai megfelelnek-e a valóságnak;
- 3) Leellenőrzi, hogy a biztonsági zsetonok (tokenek) megbízhatóak-e abból szempontból, hogy az állításokat, deklarációkat alátámasszák.

Alternatív lehetőség az, hogyha egy megbízható harmadik fél ellenőrzi és igazolja a kérelmező állításait, deklarációit azzal, hogy alátámasztja a kérelmező azonosságát.



59. ábra. XACML használata

8.8.3 XACML

Az **eXtensible Access Control Markup Language** egy OASIS²³ XML szabvány specifikáció a hozzáférési jogosultságok és azok ellenőrzésének leírására. XML-ben lehet a biztonsági irányelveket megfogalmazni az Interneten keresztül történő információ elérések felügyeletére. Miután elkészítették (megkonfigurálták) ezt a leírást, tartalmazni és publikálni fogja azokat a szabályokat és irányelveket, amelyeket egy hozzáférési jogosultság ellenőrzési mechanizmus meg tud fogalmazni az *objektumok* és azok *attribútumainak* elérhetőségéről. XACML két **architektúra komponens**t tételez fel: egy biztonsági irányelvek kikényszerítésére szolgáló komponens, *Policy Enforcement Point (PEP)*, amelyik arról dönthet, hogy vajon egy kérelem kiszolgálható-e vagy sem. Továbbá egy irányelv értékelési komponens, *Policy Enforcement Point (PEP)*, amely arról hoz döntést, hogy egy irányelv jellegű kérelem engedélyezhető-e. A PEP egyeztet a PDP-vel, a PDP egyeztet az XACML repozitóriumával, adatszótárával. Az ábra (59. ábra)

XACML egyrészt egy **hozzáférési jogok irányelveinek felügyeletét leíró nyelv** (ami lehetővé teszi a fejlesztő számára, hogy deklarálja azt, hogy *ki mit* és *mikor* tehet meg), másrészt a szolgáltatási **kérelmek** és **válaszok** kifejezésére szolgáló nyelv. Ez a nyelv olyan lekérdezések leírására alkalmas, amelyekkel megvizsgálható, hogy vajon egy bizonyos kérelem engedélyezhető-e vagy sem, és a válasz leírását is megadja erre a lekérdezésre. Az XACML specifikációja megadja azokat a definíciókat, amelyeknek segítségével a szabályok lekódolhatók, a szabályokat irányelvekbe lehet összefogni, és azokat az algoritmusokat is meg lehet fogalmazni, amelyeket akkor kell alkalmazni, amikor több szabály is alkalmazható volna. A *hozzáférési ellenőrzési lista* (ACL, Access Control List) négy tömbből áll:

- 1) Az **alany (subject)**: Ez lehet felhasználói azonosító, szerepkör, csoport; pl. „Csak részlegvezetők és magasabb beosztásuk tekinthetik meg ezt a dokumentumot.”
- 2) **Cél tárgy (target object)**: Ez egy XML dokumentum elem, valamilyen hardver/szoftver eszköz, meghajtó, vagy adatállomány lehet.
- 3) **Tevékenység (Action)**: A megengedett művelet : CRUD, LOAT.
- 4) **Szabály szolgáltatás (Provision)**: Ez egy olyan tevékenység, amelyet akkor kell végrehajtani, amikor egy XACML szabály aktiválódott; egy ilyen tevékenység lehet riasztás küldése, további igazolások, bizonyítványok bekérése, vagy egy bejelentkezési eljárás megkezdése.

8.8.4 Logikai következtetések levonása a biztonsági irányelvekkel kapcsolatban

Az XACML a hozzáférési jogosultságok ellenőrzésére vonatkozó irányelveket reprezentálja, leírja, de nem határozza meg, hogy milyen eljárással történjék meg ez az ellenőrzés. A deklaratív megfogalmazásokat egy **szabályalapú** nyelvvel lehet értelmezni, amely kifejezetten a biztonsági irányelvek interpretálását is lehetővé teszi.

Egy ilyen szabályalapú nyelv a következő fogalmakat használhatja:

- **Delegálás (Delegation)**: Létfontosságú sajátosság a Web szolgáltatások számára, mivel a szolgáltatások nem tudják előre megjósolni, hogy ki intéz kérelmeket hozzájuk, és nem tudják előre azokat a követelményeket sem, amelyeket vele szemben támasztanak a hozzá kérésintéző Web szolgáltatások, entitások. Ha egy entitás megkapja egy bizonyos szolgáltatás elérésére vonatkozó hozzáférési jogokat, ez az entitás tovább adhatja, delegálhatja e jogokat megbízható entitások számára anélkül, hogy a

kérelmezett szolgáltatás biztonsági irányelveit és követelményeit explicit módon meg kellene változtatni.

- **Visszavonás (revocation):** Egy olyan információtovábbítási lépés, amely egy entitás létező jogosultságait lenullázza, akár irányelvek révén jutott hozzá, akár delegálás útján.
- **Kérelem (Request):** egy másik entitáshoz intéz kérelmet, hogy megkaphasson egy jogosultságot vagy végrehajthasson egy *cselekményt (action)*.
- **Törlés (Cancel):** Egy korábbi kérelem visszavonása.

Irányelvek közti konfliktusok nyílt rendszerekben teljesen természetesek. Ezért egy szabályalapú rendszernek tartalmaznia kell, konfliktus feloldó mechanizmusokat:

- 1) Modalitási rangsor, preferencia beállítás (negatív a pozitívvval szemben, vagy fordítva);
- 2) Az irányelvek közötti rangsor, preferencia felállítása.

A hozzáférési jogosultságok jelenlegi kezelési mechanizmusának egy hibája az, hogy együttesen kezelik a hitelesítés, jogosultság megadás, és hozzáférési jogok ellenőrzését (authentication, authorization, access control). Ha egy adatfeldolgozási folyamat egy végfelhasználó azonosítása alapján megkap bizonyos jogokat, akkor az összes további adatfeldolgozási folyamat, amely a felhasználó nevében jár el, megkapja ugyanazokat a jogosultságokat és privilégiumokat. Ha az adatfeldolgozási folyamatot ellenséges támadás éri, pl. vírus fertőzés, akkor a felhasználó teljes számítógép környezete veszélybe kerül; adatállományok, elektronikus levelek, adatbázisok, hálózati kapcsolatok.

Ennek a problémának kezelésére a legkisebb felhatalmazás elvét vezették be (principle of least authority (POLA)). Az adatfeldolgozási folyamatoknak és objektumoknak csak annyi jogosultságot és felhatalmazást adnak meg, amennyi ahhoz szükséges, hogy elvégezzék feladataikat és elérjék céljaikat. Ezen kívül a jogosultság kibocsátás három aspektusának (hitelesítés, jogosultság megadás, és hozzáférési jogok ellenőrzése (authentication, authorization, access control)) kezelésének szétválasztása is szükséges azért, hogy egy finomabb felbontású engedély és jogosultság felügyeletet lehessen megvalósítani.

8.9 Rendszerek szabványos információ-architektúra alapjai

16. Táblázat Példa a biztonsági igények megfogalmazására

8. Információ biztonsági architektúra (Information Security Architecture)

<p>Biztonsági szolgáltatások</p> <p>Igényelten, a jövőben rendelkezésre álló szolgáltatási alkategóriák</p>	<p>Személyazonosítási és hitelesítési szolgáltatások</p>	<p>Auditálási lehetőségek biztosítása</p>	<p>Rendszer belépéti szolgáltatások</p>	<p>Hozzáférés ellenőrzés, nyomon kövétel, kézben tartás</p>	<p>Letagadhatatlanság (Non-repudiation)</p>	<p>Biztonságirányítás</p>	<p>Megbízható rendszer-visszaállítás</p>	<p>Algoritmikus, kriptográfiai információvédelem</p>	<p>Védett kommunikáció</p>
<p>Személyazonosítási és hitelesítési szolgáltatások</p>	<p>Új szolgáltatás kell</p>								
<p>Auditálási lehetőségek biztosítása</p>		<p>Új szolgáltatás kell</p>							
<p>Rendszer belépéti szolgáltatások</p>			<p>Fejlesztési kell</p>						
<p>Hozzáférés ellenőrzés, nyomon kövétel, kézben tartás</p>				<p>Új szolgáltatás kell</p>					
<p>Letagadhatatlanság (Non-</p>					<p>Új szolgáltatás kell</p>				

8.9 Rendszerek szabványos információ-architektúra alapjai

repudiation)										
Biztonság- irányítás						Új szol- gáltatás kell				
Megbízható rendszer- visz- zaállítás							Új szol- gáltatás kell			
Algoritmikus, kriptográfiai információ- védelem								Új szol- gáltatás kell		
Védett kom- munikáció									Új szol- gáltatá- s kell	

A leendő architektúra elemei	Architektúrális követelmények	Felhasználói azonosító	++	+	+	+	+	0	-	-	-	-		
		Jelszó, titkos kifejezés, PIN kód	++	+	+	+	+	+	0	-	++	-	-	
		Magas biztonsági fokozatú intelligens kártya (eszköz) (COTS)	++	+	+	+	+	+	0	0	++	-	-	
		Biometria olvasó (COTS)	++	+	+	+	+	+	0	0	0	-	0	
		Intelligens kártya olvasó (COTS)	++	0	0	0	0	0	0	0	0	-	-	
		Kártyaolvasó (COTS)	++	0	0	0	0	0	0	0	0	+	-	
		Kizárólagosan kriptográfiai eljárásokra szánt hardver kiszolgáló gép (kulcsolás, visszatejtés)	0	++	0	0	0	0	0	0	0	-	-	
		Kriptográfiai eljárásokra, algoritmusokra szoftver időpecsét szolgáltatás	0	0	0	0	0	0	0	++	0	-	-	
		Kriptográfiaiag védett üz enettovábbítás (siftrrozás, rejtjelezés, titkosítás)	0	++	+	0	0	0	0	0	0	-	-	
		Kriptográfiaiag védett adatbázisrekord	0	++	+	0	+	+	+	+	-	0	-	
		Behatolás észlelő, védelmet nyújtó szoftver (Intrusion Detection/Protection Software) (COTS)	0	0	+	+	+	+	+	+	+	-	-	
		Virusvédelmi szoftver (COTS)	0	0	++	++	++	++	++	++	+	+	+	
		Elektronikus/digitális aláírás	+	0	+	+	+	+	+	++	0	0	-	
		Egy jelszó, egyszeri alkalommal (Single Sign-On)	++	0	0	0	0	0	0	0	0	-	+	
		Fizikailag védett terem/helyiség a kiszolgáló gépeknek	++	+	+	+	+	+	+	0	0	0	0	
		Biztonsági örk video kamerákkal	++	+	+	+	+	+	+	0	0	0	0	
		Használhatóság												
		Tejjesítmény												
		Költség												
		Rendelkezésre állás												
Letagadhatatlanság														
Az adatok épsége, sértetlensége (integritás)														
A szoftver érintetlensége, sértetlensége (integritás)														
Üzenetek épsége, sértetlensége (integritás)														
Bizalmas (titkos) adatkezelés														
Hozzáférés ellenőrzés, nyomon követés														

60. ábra Példa: Az információ-architektúra biztonsági elemzésére

8.10 Egyszeri bejelentkezés és föderatív személy azonosítási protokoll

Jelmagyarázat:

COTS= kereskedelmi forgalomba kerülő szoftverkomponensek

A mátrix celláiban szereplő jelek érzékeltetik, hogy a leendő architektúra milyen mértékben segíti vagy hátráltatja az architektúrális követelmények formájában megfogalmazott célok megvalósítását. A két plusz jel (++) jelzi, hogy az architektúra eleme erősen támogatja azt, hogy az architektúrális követelmény teljesüljön. Egy plusz jel (+) gyenge támogatást jelent az adott architektúrális követelményre. A zéró (0) azt jelzi, hogy nincs lényeges hatása az adott architektúrális követelményre. Az egy negatív előjel (-) azt jelzi, hogy az architektúrális komponens alkalmazása egy kissé megnehezíti a követelmény elérését. A két negatív előjel (--) pedig azt jelzi, hogy az architektúra elem alkalmazása nagyon megnehezíti, akár lehetetlenné is teszi, azt, hogy a követelményt kielégítsék.

8.10 Egyszeri bejelentkezés és föderatív személy azonosítási protokoll

Security Assertion Markup Language (SAML) az informatikai iparon belül domináns pozícióban lévő protokoll és nyelv a **föderatív személy** (és egyéb) **azonosítás** területén, a telepített és üzemelő rendszerek tekintetében. Elsősorban a mai számítási felhőnek nevezett, Interneten keresztül nyújtott szolgáltatásokkal kapcsolatban több tízezer rendszert helyeztek üzembe; elsősorban nagyvállalatok, kormányzati szervezetek és egyéb Interneten keresztül szolgáltatást nyújtó cégek számára.

8.10.1 Alapműködési modell

2002-ben bocsátották ki **SAML** az 1.0 verziót, amely éveken keresztül fejlődött, 2005-ben jelent meg a **SAML 2.0** verzió. **SAML** szabvány gazdaszervezete **OASIS** Security Services Technical Committee.

SAML 2.0 három föderatív azonosítási szabvány kombinációja: **SAML 1.1**, **ID-FF** (Identity Federation Framework) 1.2, és **Shibboleth**.

8.10.2 Előnyei

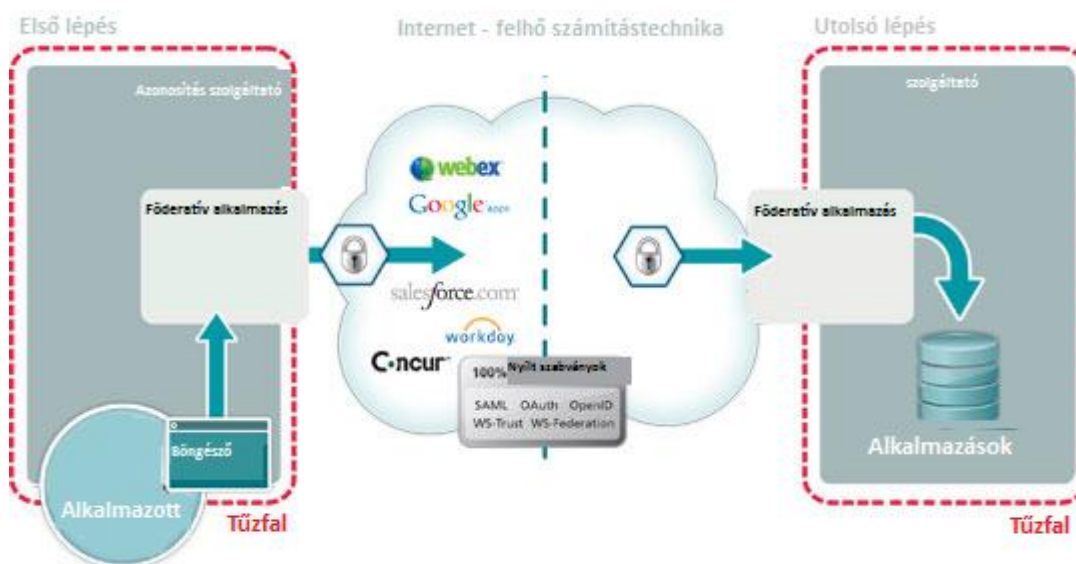
SAML XML alapú, ezért rugalmas, könnyen kiterjeszthető, bővíthető szabvány. A föderációba tartozó két tetszőleges partner kiválaszthatja azt, hogy melyik azonosítást lehetővé tevő attribútumot akarják felhasználni közösen. A **SAML** üzenetek tartalmában ezt tudják használni, feltéve, hogy a kiválasztott attribútum ábrázolható **XML**-ben. Ez a rugalmasság természetesen **SAML** rész szabványokhoz vezet, mint pl. az **SAML** azonosítás **megerősítés (assertion)**, illetve a **SzOA** architektúrához tartozó **WS-Federation** szabványokba történő beillesztése.

Az SAML különös előnye az együttműködési, interoperabilitási képesség több olyan gyártói SSO megoldással szemben, amelyek megkövetelik, hogy mind a szolgáltató (*Service Provider-t (SP)*) mind az azonosítás szolgáltató (*Identity Provider-t (IdP)*) ugyanazt a szoftvert, rendszert telepítse és helyezze üzembe. Ez egy nagyvállalat vagy közigazgatási szervezet számára azt jelenti, hogy minden egyes új kapcsolatrendszer egy új és esetleg egy teljesen különböző szoftver telepítését igényli. Ezzel szemben az SAML egy üzembe helyezett példány támogatni tudja az SSO kapcsolatokat több, különböző föderatív partnerrel.

Ezért az SAML különösen alkalmas felhő-számítástechnika (*Cloud Computing*), alkalmazás szolgáltató (*ASP*) vagy bármilyen más külső, Interneten keresztüli szolgáltatókkal kapcsolatban az egyszeri bejelentkezés megvalósítására (SSO) akkor, amikor a szolgáltatás igénybevétele a **szoftver mint szolgáltatás** minta alapján valósul meg (*Software-as-a-Service (SaaS)*).

8.10.3 SSO az Interneten keresztül és föderatív azonosítás

A föderatív azonosítás azt jelenti, hogy felhasználói, személy azonosságok biztonságosan megoszthatók legyenek eltérő hálózati tartományok és alkalmazások között. Az SAML és aWS-Federation az a két alapvető szabvány és ezek változatai, amelyeket a gyakorlatban használnak (SAML 1.0, SAML 1.1 and SAML 2.0.).



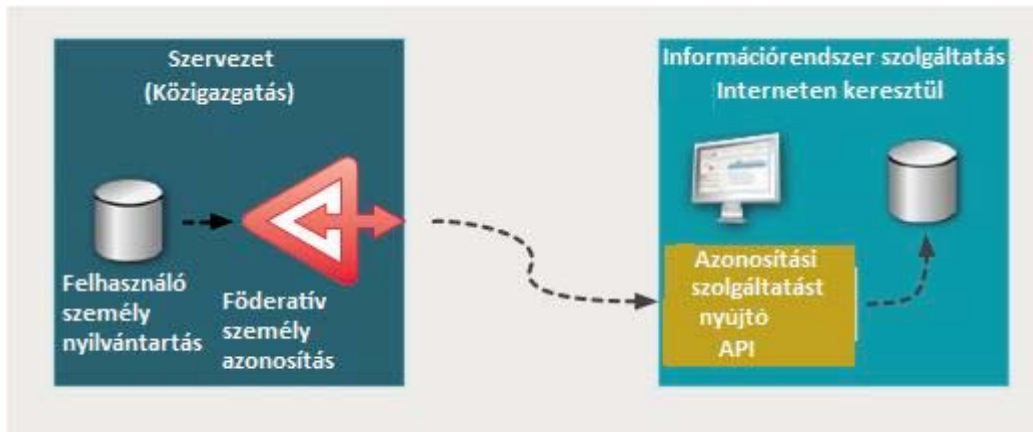
61. ábra Föderatív azonosítás és egyszeri bejelentkezés alap architektúrája

8.10.4 Modellek: Internet szolgáltatások használata egyszeri bejelentkezéssel

Három modell különböztethető meg.

8.10.4.1 1. szint: SSO (egyszeri bejelentkezés): Interneten keresztül nyújtott szolgáltatások használatára stratégia

Ez a stratégia szabvány-alapú internet tartományokon keresztül nyúló azonosítás megoldásáról gondoskodik.



62. ábra A szervezetnél nyilvántartott azonosságokban bekövetkezett változásokat megduplikálja a szolgáltatónál létező nyilvántartásokban.

Az 1. szint bevált szabványokon alapul, amely web alapú párbeszédés kapcsolatokat köt össze alkalmazásokkal és egy központi nyilvántartásban hajtja végre az azonosság hitelesítését, amely a szabályok betartásának, betartatásának és felügyeletének egyetlen ellenőrzési és referencia pontja. Az 1. szintű SSO esetében sem jelszavak küldésére nem kerül sor, sem a felhasználó bejegyzések megduplikálására – a felhasználói azonosságok fölött felügyeletet az azonosítót kibocsátó szervezet informatikai részlege látja el.

Előnyei:

1. Maximális biztonság Interneten keresztüli szolgáltatások igénybevételekor.
2. A legkényelmesebb megoldás az összes érintett félnek: felhasználók, informatikai részleg/ funkció és alkalmazás szolgáltatók.
3. Felhasználói jelszavakat egyszer tárolják el, egy olyan helyen, amelyet szervezet megfelelően tud védeni.
4. A legmagasabb megbízhatóságot nyújtja miközben a böngészők és az alkalmazások folyamatos frissítéseken esnek át.
5. Nincs gyártó függőség a szabványoknak köszönhetően.

6. Általában a legalacsonyabb a teljes bekerülési és üzemeltetési költség (total costs of ownership, TCO).

Hátrányok

1. A modell egyes konkrét megvalósításaiban, az azonosítással kapcsolatos szakértelmnek közvetlenül rendelkezésre kell állnia.
2. A föderációban részt vevő feleknek a megvalósítás érdekében együtt kell működniük.

Az 1. szintű SSO meghatározó jellegzetessége az, hogy az azonosítást és hitelesítést szabványon alapuló zseton, 'token' cserével oldja meg, a felhasználói nyilvántartások azonban központilag kezelt tartományon belül maradnak és nem szinkronizálják más külső erőforrásokkal. Az alkalmazásokba történő bejelentkezésnél a jelszó használatot felszámolják, jelszóra nem lesz szükség kivéve a felhasználói nyilvántartásokra támaszkodó azonosítási és hitelesítési folyamatokat, mint pl. a Microsoft LDAP-ja, az Active Directory-ból (MS AD) történő azonosítás és hitelesítési eljárás lefolytatásához. Minden azonosítással kapcsolatos információ kriptográfiailag védetten közlekedik a hálózaton. Az 1. szintű SSO a nyílt és bevált szabványokon alapszik, nevezetesen az SAML, az OpenID Connect²⁴-en és az OAuth²⁵.

1. szintű SSO megoldásnál, amikor egy kapcsolatot felépítenek egy alkalmazással, a SSO szolgáltatás kikerül az információcsere folyamatából miután a párbeszéd már létrejött az érintett rendszerek között, más modelleknél az információcsere közepén marad egy állandó **proxyként (közvetítő ügynök)**, amelyen keresztül kell az információáramlásnak megtörténnie. 1. szintű SSO megoldást a következő célok indokolják: az ügynök maximalizálja (1) az információcsere áteresztőképességét, (2) csökkentse a késleltetést, (3) számolja fel az egyponthos *hibagócot* (single point of failure), és (4) küszöbölje ki a személyes adatok és magántitkok esetleges megsértésének veszélyét.

Vannak olyan indokolt esetek, amelyekben a *proxyra* szükség van, különös egyes ipari ágazatokban, gazdasági szektorokban, azonban ez a proxy/ügynök modell inkább kiegészítő szolgáltatásnak tekinthető semmint a 1. szintű SSO modell lényegéhez tartozó megoldásnak.

8.10.4.2 2. szintű SSO: Saját, egyetlen hálózati tartományra vonatkozó azonosítás

A **2. szintű SSO** olyan szervezetekben szükséges, ahol sok régi és elavult, korszerűtlen alkalmazás található, és amelyek nem integrálhatók be a modern felhasználói nyilvántartásokba, mint pl. a Microsoft AD-be. A 2. szintű SSO modellt megvalósító gyártói implementációk al-

kalmazkodtak a felhő-számítástechnika, vagy általánosabban az Interneten igénybe vett szoftver szolgáltatásokhoz, a Web Access Management²⁶ megközelítés révén. A 2. szintű SSO modell a szoftver mint szolgáltatás jellegű alkalmazások (SaaS) számára tulajdonképpen egy szub-optimális megoldás, amely lehetővé teszi az olyan régi és elavult alkalmazások életciklusának kiterjesztését, amelyek nem tudnak váltani a régi *azonosítási, hitelesítési és jogosultság kezelési* módszereikről az újra.

Előnyök

1. Az olyan régi elavult alkalmazási rendszerekben is megszabadulhatnak a jelszó használatától, amelyekben a modern felhasználói nyilvántartások nem használhatók.
2. Központosított, hálózati tartomány központú, információcserét nyújt, amelyben keretében a biztonsági irányelvek betarthatók.
3. A 2. szintű SSO modellhez szükséges komponensek esetleg már rendelkezésre állnak, és az SSO megvalósításához szükséges kezdeti lépésekhez elegendő funkcionalitást tudnak nyújtani.

Hátrányok

1. Általában gyártó, vagy tulajdonos függő, jelentősen módosították a testre szabhatóság végett.
2. Magas bekerülési és üzemeltetési költségek.
3. Mind a böngészők mind az alkalmazások kézben tartása szükséges, vagy a rendszer összedől.
4. A felhő-számítástechnikához, vagy általánosabban az Interneten keresztül igénybe vett szoftver szolgáltatásokhoz, nem igazán illeszkedik.
5. A biztonsági sebezhetőség lehetősége több ponton is fennáll.

A 2. szintű SSO modell valójában érett technológia, van vonzereje, azonban egyedi, gyártói technológiát használ általában a felhasználó által kezdeményezett ember-gép párbeszéd nyomon követésére. Web Access Management esetében kriptográfiailag védett sütiket használnak az állapot megőrzésre – azonban ez a megoldás a felhő-számítástechnika esetében nem működik a sütik másféle kezelése miatt. 2. szintű SSO modell zseton, token alapú protokolljai (pl. **Kerberos**) pedig nem tudnak több hálózati tartományon keresztül működni. Néhány 2. szintű SSO modell ezt úgy próbálja áthidalni, hogy 1. szintű SSO modell egyes

funkcionális szolgáltatásaira próbál támaszkodni, ezek lehetnek például az SAML modulok. Más 2. szintű SSO modell megvalósítások pedig a 3. szintű SSO modell olyan funkcionalitását kívánják felhasználni, mint például a képernyő tartalom lelopása vagy a jelszavak megduplikálása.

8.10.4.3 3. szintű SSO modell: azonosítók visszajátzsása, tartományokon keresztüli azonosítás és hitelesítés

3. szintű SSO modellben a felhasználói neveket és jelszavakat az adott szervezet informatikai funkciójának, részlegének hatás körén kívül tárolják, és a szóban forgó alkalmazások számára a felhasználói neveket és jelszavakat az Interneten keresztül visszajátsszák az adott alkalmazásnak. Ez a megoldás gyors probléma kiküszöbölés, amely megszabadít a felhasználói jelszavak kezelésétől, és újra beállításától, de ennek az ára a biztonságirányítási szabályozás és irányelvek optimális megvalósíthatóságának elvesztése. Egyes iparágakban és gazdasági szektorokban nem felel meg a szabályozó hatóságok előírásainak.

Előnyök:

1. Viszonylag alacsony bevezetési költségek (az eszközök általában ingyenesek).
2. A felhasználók a személyes adataikat egy tároló helyen tudják tartani valahol az Interneten.

Hátrányok:

1. Magas biztonsági kockázatok - a legtöbb megoldásnál a felhasználó tudja meghatározni a jelszó kriptográfiai erősségét; nem-SSL Internet tartományok esetében az azonosításra szolgáló adatok kriptográfiai védelem nélkül közlekednek az Interneten.
2. Folyamatos napra készen tartást igényel, mivel a jelszó átadás mechanizmusa elakad minden olyan esetben, amikor az alkalmazás, web hely tulajdonos, internet szolgáltatója megváltoztatja bejelentkezési mechanizmust illetve képernyőt.

3. szintű SSO modellben több architekturális megoldás között lehet választani. Lehet a felhasználói nyilvántartás adatbázisát valahol az Interneten, a felhőben tartani, vagy magánfelhőben, helyi kiszolgálón vagy akár egy alkalmazott asztali számítógépén. 3. szintű SSO modell főbb jellemzői:

8.11 SAML alapú SSO használati esetei

1. Legalábbis részlegesen egy egyedi adattárhelyre kell támaszkodni a felhasználói nevek és jelszavak tárolásánál szemben a zseton, token alapuló információcserével.
2. A személynek saját magának kell gondoskodnia az azonosító adatok bejuttatásáról az adatbázisba. Néhány 3. szintű SSO modell automatikusan felerősíti a jelszót és a felhasználó elől el is fedí.
3. Ez a megoldás támaszkodhat egy központi felhasználó nyilvántartásra, amelyet azonban elsődlegesen arra használnak, hogy feltöltsék az adatbázist.
4. 3. szintű SSO modell részben olyan szolgáltatók azonosítási szolgáltatásaira támaszkodnak mint például a „Facebook” vagy az „MSN Live”.

Ez a 3. szintű SSO modell fokozza a felhasználók kényelem érzetét és csökkenti az informatikai részleg adminisztrációs terheit (kevesebb jelszó beállítás). Azonban ebben a modellben nehéz egy központi személy azonosítási *irányelvet* és *szabályzatot* érvényesíteni. A konkrét megvalósítások általában lehetővé teszik, hogy felhasználó saját maga által előállított felhasználó nevet és jelszót használjon.

8.11 SAML alapú SSO használati esetei

Az SAML-alapú Interneten/Weben/ Világhálón keresztül történő és az azonosítási föderáción keresztül történő egyszeri bejelentkezés használati esetei.

8.11.1 SAML párbeszédnek részt vevői

SAML alapú párbeszédnek legalább két résztvevője, az egyik fél az **SAML megerősítést kibocsátó partner (asserting party)**, a másik **rátámaszkodó partner (relying party)**. Sok esetben egy felhasználó, aki egy web böngészőt használ éppen vagy egy SAML-képes alkalmazást hajtát végre szintén részt vevő lehet, sőt *megerősítést kibocsátó partner* is lehet.

A megerősítést kibocsátó partnert – az, aki az SAML szabvány szerinti megerősítést előállítja –, **SAML szervezetnek** is hívják egyesek (**SAML authority**). A **rátámaszkodó partner** általában a megkapott megerősítést használja fel. Amikor egy *SAML szervezet* vagy *rátámaszkodó partner* közvetlenül intéz kérést egy másik SAML partnerhez akkor ezt a partnert **SAML kérelmezőnek** nevezik (**SAML requester**), a választ adó felet pedig **SAML reagálónak (SAML responder)** nevezik. Egy válaszoló fél készsége, hogy a megerősítést kibocsátó partner információira támaszkodjon, attól függ, hogy vajon kiépült-e a bizalmi kapcsolatrendszer a megerősítést adó partnerrel.

SAML rendszerébe tartozó komponensek több olyan SAML szerepkörben jelenhetnek meg, amelyet az SAML szolgáltatások és a felhasználandó protokoll üzenetek és az előállítandó és feldolgozandó megerősítések típusai határoznak meg. Ilyen lehet a több hálózati tartományra kiterjedő egyszeri bejelentkezés (MDSSO, vagy csak SSO). Az egyik szerepkör az *azonosítás szolgáltató (identity provider (IdP))*, a másik a *szolgáltató, a szolgáltatást nyújtó (service provider (SP))*.

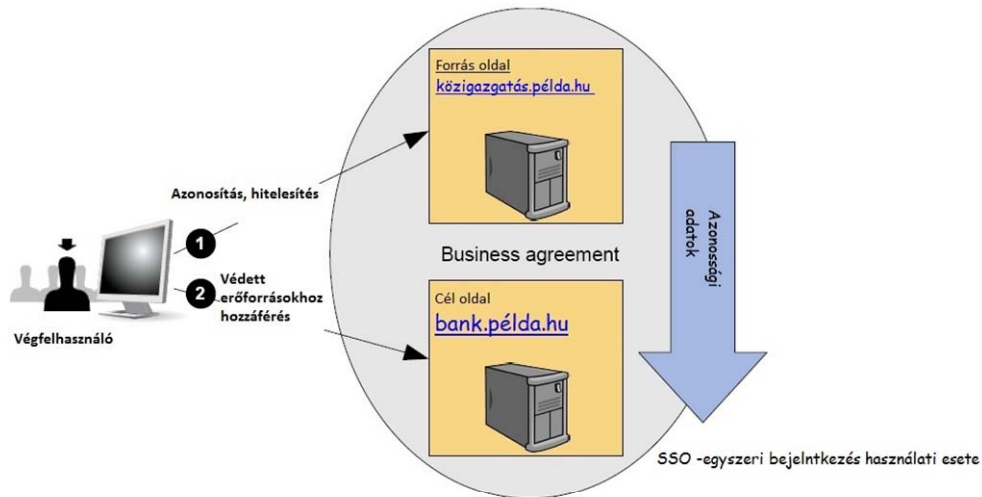
Az azonosítás **alanya** – egy egyed, entitás, akinek az azonosítása és hitelesítése egy adott biztonsági tartományon belül történik meg – az akiről, amiről az azonosítás *megerősítése* szól. Az azonosítás alanya lehet emberi személy, vagy bármely más entitás mint például cég, vagy gép.

Egy megerősítés tipikus forgatókönyve szerint a következő információk átadására kerülhet sor „Ez N.N, akinek az elektronikus posta címe n.n@pelda.hu , és az azonosítása felhasználó név és jelszó segítségével történt meg.” Egy Interneten keresztül szolgáltatást nyújtó dönthet úgy, hogy ennek az információnak a birtokában N.N-nek egyszeri bejelentkezéssel keresztül megadja a jogot a szolgáltatáshoz és a kapcsolódó erőforrásokhoz történő hozzáférésre.

8.11.2 Interneten/ Weben/ Világhálón keresztüli egyszeri bejelentkezés használati esete

A több hálózati tartományon keresztül nyúló egyszeri bejelentkezés vitathatatlanul a legfontosabb használati eset, amelyre az SAML-t alkalmazni lehet. Ebben a használati esetben, a végfelhasználó egy bejelentkezési párbeszédet folytat le – ebben a *biztonsági kontextusban* –, egy világháló helyen (kozigazgatás.pelda.hu). A párbeszéd egy adott pontján – vagy közvetlenül vagy nem észrevehető, transzparens módon – átirányítják egy másik partner világháló helyére, illetve oldalára (bank.pelda.hu). Ebben az esetben fel kell tételeznünk, hogy a végfelhasználó számára a *föderatív azonosítás* rendszerét a két fél - kozigazgatás.pelda.hu és bank.pelda.hu - *kialakította*, két érintett fél között valamilyen megállapodás vagy szerződés jött létre. Az azonosítás szolgáltató világháló hely kozigazgatás.pelda.hu megerősíti a szolgáltatást nyújtó világháló oldal felé bank.pelda.hu azt, hogy a végfelhasználó ismert (hivatkozva a föderatív azonosítójára, azonosítása és hitelesítése megtörtént, és azonosságának egyedi jellemzői (attribútumai) vannak mint pl. cég könyvelő banki aláírási joggal). Mivel bank.pelda.hu megbízik a kozigazgatás.pelda.hu –ban ezért a végfelhasználó azonosságát érvényesnek és korrektül azonosítottak fogadja el, és ennek megfelelően a végfelhasználó

számára létrehozza azt a párbeszédés környezetet a saját tartományában, amely a tranzakciók további folytatásához szükségesek.



63. ábra Az egyszeri bejelentkezés használati esetének általános sémája

A magas szintű leírás azt mutatja be, hogy a végfelhasználó először az *azonosító szolgáltatónál* jelentkezik be azonosítás végett, megtörténik az azonosítás és hitelesítés, majd utána tud a *szolgáltató* védett erőforrásaihoz hozzáférni. Ezt a forgatókönyvet általában az *azonosító szolgáltató* által kezdeményezett Web SSO-nak, Interneten keresztüli egyszeri bejelentkezésnek nevezik általában.

Az *azonosító szolgáltató* által kezdeményezett SSO, egyszeri bejelentkezés, bizonyos esetekben hasznos; azonban egy sokkal gyakoribb forgatókönyv az, hogy a végfelhasználó egy *szolgáltató* oldalát látogatja meg egy böngészőbeli könyvjelzőn keresztül, először olyan erőforrásokat érve el, amelyek nem igényelnek semmilyen különleges azonosítást illetve jogosultságot. Egy SAML-képes Világháló szolgáltatás esetében akkor, amikor a végfelhasználó megkísérel a védett erőforrásokhoz hozzáférni a *szolgáltatónál* akkor a *szolgáltató* egy azonosítási kérést intéz az *azonosító szolgáltatóhoz* annak érdekében, hogy engedélyezhesse a végfelhasználó bejelentkezését a *szolgáltató* rendszerébe. Ezt a forgatókönyvet a *szolgáltató* által kezdeményezett Web SSO-nak, Interneten keresztüli egyszeri bejelentkezésnek nevezik. Ha egyszer már sikeresen bejelentkezett a végfelhasználó, akkor az *azonosító szolgáltató* előállít egy olyan *megerősítést*, amelyet a *szolgáltató* fel tud használni végfelhasználó hozzáférési jogainak ellenőrzésére, amelynek alapján a végfelhasználó a védett erőforrásokat el-

érheti. SAML V2.0 támogatja mindaz *ú azonosító szolgáltató* mind a *szolgáltató* részéről kezdeményezett azonosításhoz kapcsolódó információáramlást.

Az SAML több olyan változatát támogatja a fentebbi két alap információáramlási mechanizmusoknak, amelyek a végfelhasználói azonosítás és hitelesítés különböző típusaival, eltérő erősségű azonosítási módszereivel, a föderatív azonosság alternatív leíró formátumaival, az Internet protokoll üzenetek továbbítási módjának különböző **leképezéseivel (binding)**, az azonosításhoz szükséges attribútumok továbbításával és felhasználásával foglalkoznak.

8.11.3 A föderatív azonosítás használati esete

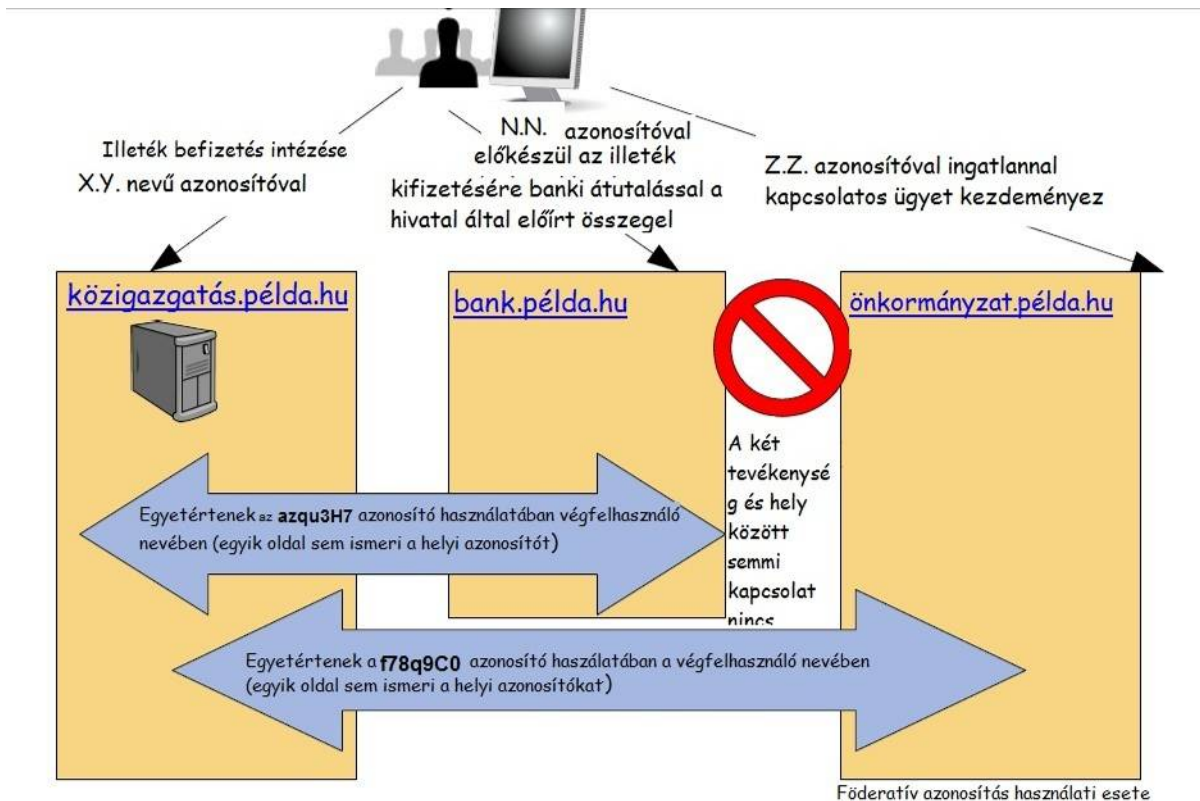
Egy végfelhasználói azonosság akkor tekinthető a föderatív azonosítás rendszer részének, ha létezik egy megállapodás a szolgáltatók között az olyan azonosságok és az azonosító attribútumok tekintetében, amelyeket az Internet helyek a végfelhasználó *megnevezésére* tudnak felhasználni.

A föderatív azonosítási rendszer kialakításának több olyan előfeltétele van, amelyek vizsgálatára szükség van a rendszer kiépítése előtt:

- A végfelhasználóknak az összekapcsolandó internet helyeken vajon létezik-e már olyan helyi azonosítójuk, amelyeket dinamikusan össze lehet kapcsolni egy föderatív azonosítási rendszeren keresztül?
- A szóban forgó internet helyek vajon előre létrehozott föderatív azonosítókat fognak-e használni vagy a föderatív azonosság megállapítását illetve az azonosító megszüntetését dinamikusan fogják kezelni?
- Szükség van-e a végfelhasználó kifejezett és egyértelmű egyetértésére ahhoz, hogy föderatív azonosítót hozzanak létre számára?
- A végfelhasználó azonosítójának egyes attribútumait át kell-e adni?
- A föderatív azonosítás rendszere vajon olyan ideiglenes azonosítókra alapuljon-e, amelyeket a felhasználói párbeszéd végén megsemmisítenek?
- A személyes adatok védelme kiemelt fontosságú kérdés-e, és ennek következtében az áramló adatokat kriptográfiai védelemmel kell-e ellátni?

A legtöbb azonosítást kezelő rendszer napra készen tartja a *helyi azonosítókat* és azonosságot. Ezek a helyi azonosítókat általában a felhasználó helyi felhasználói bejegyzése (*account*) és más egyéb helyileg azonosítható felhasználói profil formájában jelennek meg. Ezeket a helyi azonosítókat kell a föderatív azonosítóhoz hozzákapcsolni azért, hogy a felhasználót ezzel

az azonosítóval nevezhesse meg a szolgáltató a többi partnerrel folytatott üzenet váltás során. A föderatív azonosító és helyi azonosító összerendelését az egyik vagy a többi partnernél, ahol a föderatív azonosító fogják használni, általában a **felhasználói bejegyzések összekapcsolásának** nevezik (**account linking**).



64. ábra Föderatív azonosítás forgatókönyve

A folyamat lépései a következők:

1. Dzsóni a hivatal által kivetett illetéket rendezni kívánja az X.Y (Dzsóni_A) felhasználói bejegyzés alkalmazásán keresztül.
2. Ezután a böngésző könyvjelzőjét használva, vagy valamilyen URL hivatkozásra rákattintva, a bank.példa.hu oldalt látogatja meg az illeték kifizetésének banki átutalással történő intézésének kezdeményezésére. Ez az oldal észleli, hogy a böngésző nem jelentkezett be helyileg, de korábban meglátogatta a föderációhoz tartozó azonosító szolgáltató partner oldalát közigazgatás.példa.hu-t (szükség esetén kiaknázva a SAML 2.0 szabványban megtalálható partner **megtalálási, felfedezési szolgáltatást** (discovery feature)). A bank.példa.hu oldal megkérdezi Dzsónit, hogy

- vajon egyetért-e azzal, hogy a helyi bank.pelda.hu oldalon használt azonosítóját a föderatív azonosítás rendszerében összekapcsolják kozigazgatás.pelda.hu oldalon használt azonosítójával.
3. Dzsóni egyetértését fejezi ki a föderatív azonosítás alkalmazásával és így a böngészőjét visszairányítják a kozigazgatás.pelda.hu oldalra, ahol az oldal előállít Dzsóninak egy **pszeudó nevet** (pseudonym) **azqu3H7**, amelyet fel tud használni bank.pelda.hu oldal meglátogatásánál. A *pszeudó nevet* hozzákapcsolták X.Y (Dzsóni_A) felhasználói bejegyzéséhez. Mindkét szolgáltató egyetért abban, hogy ezt az azonosítót használják Dzsóni megnevezéséhez a következő tranzakciókban.
 4. Dzsónit illetve böngészőjét visszairányítják bank.pelda.hu oldalra egy SAML *megerősítéssel* együtt jelezve, hogy az **azqu3H7** állandó föderatív azonosítóval megnevezett felhasználó bejelentkezett az azonosító szolgáltatójánál. Mivel a bank.pelda.hu oldalon ez az első alkalom, hogy ez a föderatív azonosító megjelent, ezért ez az oldal nem tudja, hogy melyik helyi azonosítóhoz, felhasználói bejegyzéshez kapcsolja.
 5. Ezért Dzsóninak be kell jelentkeznie a bank.pelda.hu oldalon az N.N (A-Dzsóni) azonosítóval. Ezek után a bank.pelda.hu hozzákapcsolja **azqu3H7** azonosítót a helyi azonosítóhoz azért, hogy kozigazgatás.pelda.hu mint azonosító szolgáltatóval történő kapcsolattartás során a jövőben ezt az azonosítót használni lehessen. Ennek révén a *szolgáltatójánál* és az *azonosító szolgáltatójánál* a föderatív azonosítót, **azqu3H7-t** hozzákapcsolták a helyi azonosítókhoz.
 6. A banki fizetés tranzakció elindítása után Dzsóni kiválasztja onkormanyzat.pelda.hu könyvjelzőt, hogy az ingatlanával kapcsolatos ügyintézkést kezdeményezzen.
 7. A föderatív azonosítás lépéseit az azonosító szolgáltatóval, kozigazgatás.pelda.hu-val megismétlik, létrehozva egy új *pszeudó nevet* **f78q9C0** X.Y (Dzsóni_A) felhasználói bejegyzéshez, amelyet az onkormanyzat.pelda.hu oldalon fog tudni használni akkor, amikor azt meglátogatja.
 8. Dzsónit illetve a böngészőjét visszairányítják az onkormanyzat.pelda.hu oldalra, amely a *szolgáltató* szerepét játssza, egy új SAML megerősítéssel. A *szolgáltató* megkívánja, hogy Dzsóni bejelentkezzen a helyi Z.Z. (Dzsóni_Árnika) felhasználói

azonosítóval és a föderatív azonosítót, az *azonosító szolgáltatóval* - [közigazgatás.példa.hu](#) - történő jövőbeli használat végett a pszeudó nevet hozzáköti a helyi azonosítóhoz . A föderatív azonosítón - **f78q9C0** – keresztül az azonosító szolgáltató és a szolgáltató helyi azonosítói összekapcsolódtak.

Ezek után bármikor a jövőben, Dzsoninak valamilyen ügyet kell intéznie, akkor elegendő [közigazgatás.példa.hu](#)-n egyszer bejelentkeznie mielőtt meglátogatná [önkormányzat.példa.hu](#), és a [bank.példa.hu](#) oldalakat. Dzsonit a [közigazgatás.példa.hu](#) oldal mint azonosítás szolgáltató **azqu3H7**-ként fogja a [bank.példa.hu](#) felé azonosítani míg **f78q9C0**-ként fogja az [önkormányzat.példa.hu](#) felé azonosítani. Mindegyik szolgáltató Dzsoni helyi felhasználó bejegyzésének megtalálása révén, amelyet összekötöttek az állandó pszeudó névvel, teszi lehetővé Dzsoninak, hogy a kívánt tevékenységét lefolytathassa miután az egyszeri bejelentkezéshez szükséges információcsere megtörtént.

8.11.4 Föderáció

Ugyanaz a felhasználó több rendszerben különböző identitásokkal rendelkezhet (ezeket a rendszereket ráadásul mások is üzemeltethetik).

- **Cél:** hálózati tartományokon keresztül nyúló egyszeri bejelentkezés megteremtése: (*cross-domain single sign-on.*)
- **Identitás föderáció:** paradigma olyan technológiákat, szabványokat és használati eseteket ír le, melyek lehetővé teszik a személy/felhasználói azonosítás biztonságos együttműködését, interoperabilitását az önálló, egyedi és saját biztonsági rendszerekkel védett vállalati/ szervezeti rendszerek között.

Megkülönböztetik a szolgáltatást nyújtót (**Service Provider-t (SP)**) és személy/felhasználó azonosítást szolgáltatót (**Identity Provider-t (IdP)**), ez utóbbi végzi az autentikációt (azonosítást) , másik nyújtja a szolgáltatást. SP-hez tartozó helyi felhasználói bejegyzés és az IdP által tárolt összekapcsolására több megoldás létezik (attribútum, fix azonosító, pszeudonim azonosító, stb.).

Az SAML az egyik föderációs személy/ felhasználó azonosítási szabvány:

SAML forгатókönyv a föderatív azonosításra:

1. hozzáférés kérése SP-nél
2. SP hitelesítési kérést állít ki és átirányít egy IdP-hez
3. IdP hitelesít

4. *IdP* megerősítést (assertion) ad
5. *IdP* visszairányítja a felhasználót az *SP*-hez
6. *SP* elfogad
7. *SP* lokális munkamenetet indít a megerősítés (assertion) alapján
 - a. **SAML Assertion** (Megerősítés, visszaigazolás) : a hitelesítés tényét igazolja, 3 részből áll:
 - i. Authentication Statement (azonosítás ténye) +
 - ii. Attribute Statement (ki azonosította magát) + Authorization Decision Statement (azonosítás alapján hozott döntés közlése)
 - iii. SAML logout (kijelentkezés).
1. **Kijelentkezési kérelem**, a *LogoutRequest* vizsgálata:
 - a. Munkamenetért felelős entitások (rendszer komponensek) értesítése
 - b. Minden entitásnak küldeni kell egy kijelentkeztetési kérést
 - c. Aktuális párbeszéd (session) megszüntetése
 - d. Ha siker -> Kijelentkezési kezdeményezésre adott válasz. *LogoutResponse* küldése sikeres státusszal
 - e. Ha nem -> státusz: *PartialLogout* (részlegesen sikerült kijelentkezés).

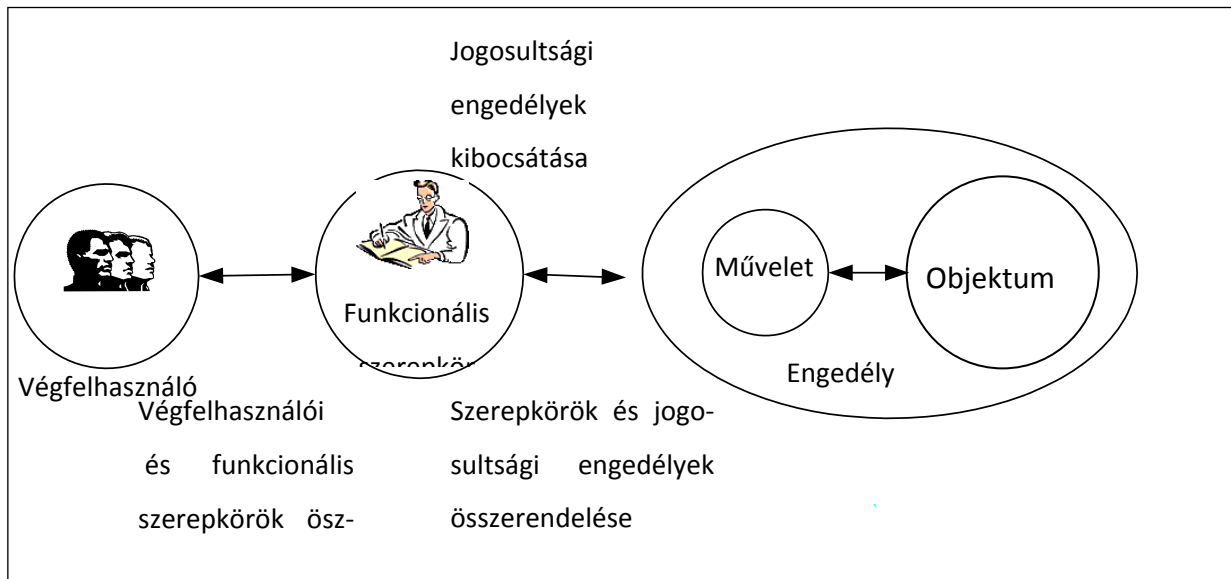
8.12 Jogosultsági viszonyok ábrázolására szolgáló architektúra megoldások

A **hozzáférési jogosultságok leírása** alapvetően két bevált módszer áll rendelkezésre a **szerepkör alapú** és az **attribútum** (sajátosság, tulajdonság) **alapú jogosultsági rendszer**. E két rendszer kombinálható is.

Az információrendszerek alapvető *szereplői* („actor”) az ügyfél (természetes személy), végfelhasználó, ügyintéző, egyéb belső és külső érintett/érdekelt felek. Illetve az Internet, Web, világháló, számítási felhő kontextusában megjelenik a **kibertérben létező szervezetek** elektronikus megszemélyesítése, mint potenciális szereplő, aki számára jogosultságokat kell kibocsátani (pl. virtuális vállalkozások, „Virtual Enterprises”).

Az egyik jogosultsági dimenzió az elektronikus/informatikai hozzáférési jogosultság. A másik az ügyintézés, ügyvitel, üzletvitel viszonylatában a szervezeti (vállalati, üzleti) folyamattal kapcsolatos, feladatkör, hatáskör, felelősségi kör, rendelkezési jog, felhatalmazott, meghatalmazott, saját jogon, önállóan jár el stb.

A rendszerrel érintkezésbe lépő szereplők (ügyfél, meghatalmazott, megbízott, ügyintéző, végfelhasználó stb.), amikor a *kibertérbe* kerülnek és ott tevékenykednek akkor a feladataiknak, jogosultságainak leképezése a kibertérben használható fogalmakra a *hozzáférési és adatkezelési jogosultsági rendszeren* keresztül ragadható meg. E hozzáférés *jogosultsági rendszer* a kibertér alapfogalmaival dolgozik: adat, e-dokumentum vagy objektum, amelyeken *műveletek* bizonyos elektronikus adatfeldolgozó tevékenységek elindítása révén hajthatók végre.



65. ábra A jogosultságok megadásának alapsémája egy elektronikus információrendszer környezetben

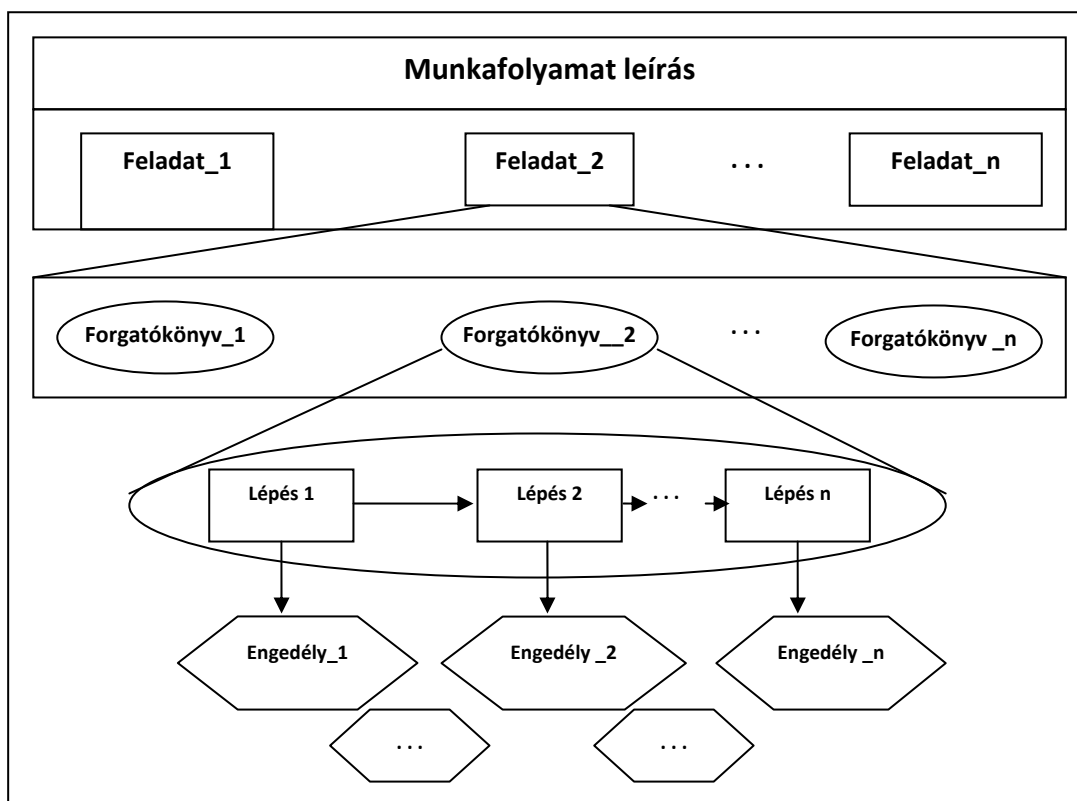
8.12.1 Szerepkör és jogosultság tervezés

Két magas szintű szerepkör modell jelenik meg egy információrendszerben: funkcionális és strukturális.

Funkcionális szerepkör („role”)

- **Funkcionális szerepkör** azoknak az engedélyeknek a halmazából áll, amelyek egy feladat lefolytatáshoz szükségesek. A *funkcionális szerepkör* nevek egy-egy jogosultsági engedély csoporthoz kapcsolhatók azért, hogy a végfelhasználók felé történő engedély kiadást leegyszerűsítsék. A *jogosultsági engedélyek* végső soron ahhoz szükségesek, hogy az adatok, e-dokumentumok (félig-strukturált, strukturálatlan), és funkcionális-, információrendszer-, Web szolgáltatások tekintetében meghatározza azokat a *műveleteket*, amelyeket a jogosultsági engedély birtokosa elvégezhet. Ez általában a klasszikus CRUD (LOAT)²⁷ mátrixban történő e-dokumentum, adatszerkezeti elem (en-

titás, adattábla stb.) szintű műveleti jogosultság megadását jelenti, a szoftverben megjelenő információrendszer szolgáltatások, funkciók tekintetében pedig a végrehajtási („futtatási”) jogosultságokat. A funkcionális szerepköröket meg lehet adni egy *azonosító tanúsítvány végfelhasználói attribútum tanúsítvány* formájában; vagy egy jogosultságok és engedélyeik tárolására szolgáló elosztott *címtárban* lehet megadni (pl. LDAP stb.).



66. ábra A jogosultsági engedélyek és forgatókönyvek lépéseinek összekapcsolása munkafolyamat munkafeladatain belül

Strukturális szerepkör

- **Strukturális szerepkör** a szervezet szereplőit a szervezeti hierarchiában helyezi el, annak megfelelően, hogy milyen alkalmazotti kategóriákhoz tartoznak, amelyek egymástól eltérő hozzáférési jogosultsági szinteket jelentenek. A strukturális szerepkör azt jelenti, hogy lehetővé teszi a szerepkörhöz tartozó végfelhasználók számára, hogy részt vegyenek a szervezet munkafolyamatában (elvégezzenek feladatokat), a munkaköri leírásuk, beosztásuk, rangjuk, címük szerint, de nem specifikálja részleteiben a hozzáférési jogosultságokat egyes információ objektumokra vonatkozóan. A strukturális szerepkör lehetővé teszi azt, hogy egy végfelhasználó kapcsolatba léphessen egy infor-

mációforrással, de nem adja meg a jogosultsági engedélyeket. Strukturális szerepkörre példa lehet: Főosztályvezető, osztályvezető, ügykezelő stb.

Strukturális szerepkörök azt határozzák meg, hogy mely munkafolyamatban vehet részt az adott végfelhasználó, míg a *funkcionális szerepkör* definiálja azokat hozzáférési jogosultsági engedélyeket, amelyek lehetővé teszik a különböző bizalmassági fokozatú információk, e-dokumentumok elérését.

8.12.2 *Munkafolyamat modell forgatókönyv felfogásban*

A **forgatókönyv modell** (ld. 66. ábra) a munkaköri leírások, munkafeladatok, forgatókönyvek és azok lépéseit illusztrálja. A jogosultsági engedélyeket az egyes forgatókönyv lépésekhez köti (ennek mikéntjét a szerepkör szervezés keretében kell megoldani).

A *forgatókönyv* alapú *szerepkör* kialakítás, szervezési eljárás esetében mindegyik tevékenységet és eseményt egy forgatókönyv keretében ábrázolunk. A *forgatókönyv lépése* mindig egy *specifikus hozzáférési művelethez* kapcsolódik. Azokat a forgatókönyveket – amelyeket egy előírt sorrendben kell végrehajtani, azért hogy egy adott munkafeladat célját elérjük – a jogosultsági engedélyek kialakításához, mint információforrások használhatók fel. Egy végfelhasználónak, aki egy bizonyos forgatókönyv szerint folytatja le a tevékenységeket, az összes olyan hozzáférési jogosultsággal rendelkeznie kell, amelyre a forgatókönyv egyes lépéseinek befejezéséhez szüksége van.

Jogosultsági engedély az *információ-architektúra* megfelelő finomságú bontásához illeszkedő szintekhez hozzákapcsolódó „információ építőelemek”-re kell definiálni. Ennek a felbontásnak a finomsága nem lehet sem túl részletes, sem túl nagyvonalú, az információ architektúra hierarchiájában sem túl magas sem túl alacsony. A 17. Táblázat a lehetséges hierarchia szinteket mutatja. Kiemeli az aggregált dokumentum szintet, amely interoperabilitás szempontjából a legalacsonyabb hierarchia szintet kívánja jelölni. Az aggregált dokumentum szint egy olyan fogalmi szintű elektronikus információrendszer objektumot vagy funkciót jelöl (pl. Ügyintézési rendelkezési dosszié) amelynek részlet komponenseire már nem utalunk.

17. Táblázat **Jogosultsági engedélyek**

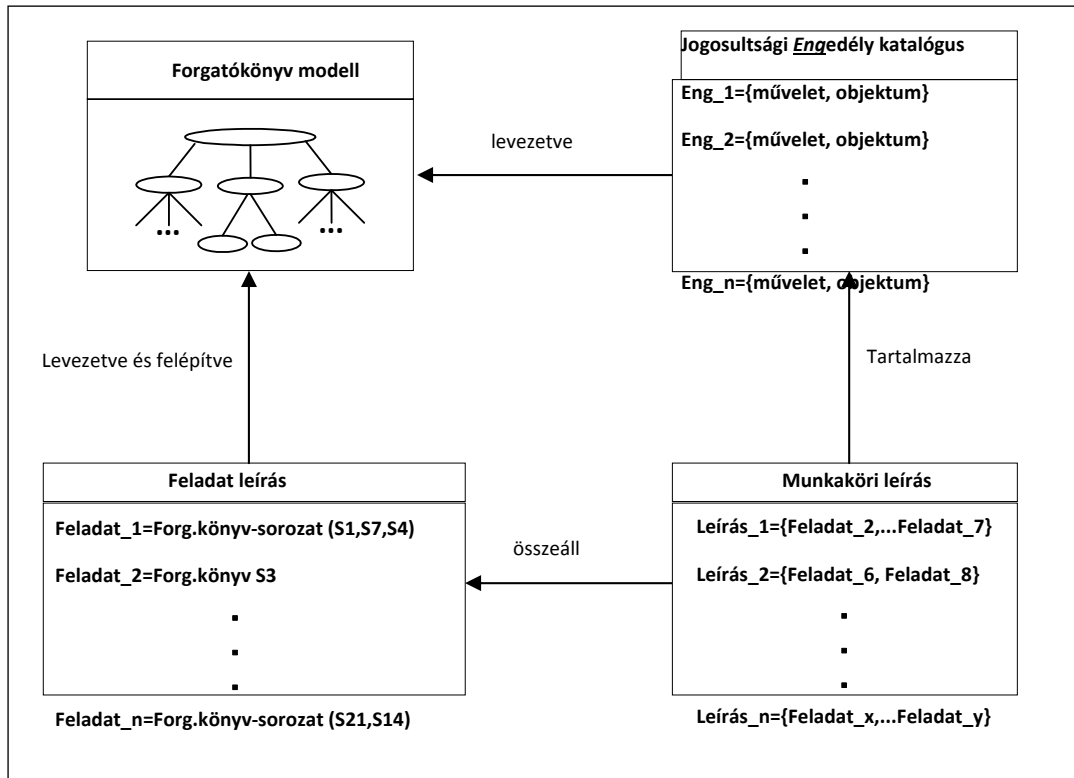
Tevékenység	Szerepkörök, jogosultságok megtervezés, megszervezése	Szerepkör
Részt vesz	Munkaköri leírás	Strukturális szerep-

		kör
Részt vesz	Végfelhasználó leírás	
Végrehajt	Feladat	
Végrehajt	Forgatókönyv	
Lefolytat	Lépés	
Létrehoz, Olvas, Aktualizál, Töröl	Aggregált dokumentum szint	Funkcionális szerepkör
Végrehajt	Funkció	Funkcionális szerepkör
Létrehoz, Olvas, Aktualizál, Töröl	Adat entitás, objektum, tábla, dokumentum	
Létrehoz, Olvas, Aktualizál, Töröl	Adatelem	

A rendszer tulajdonosainak, felügyelőinek, adminisztrátorainak, *szervezeti architektúra tervezőinek* és rendszer gyártóknak, szállítóknak lehetősége van az információ objektumok pontosabb, konkrétabb megragadására, és a saját környezetükben használt információ objektumokra történő leképezésére.

8.12.3 Szerepkör tervezés, szervezés

A *szerepkör tervezés, szervezés* alapvető célja az, hogy szabványosított és újra hasznosítható jogosultsági engedélyeket alakítson ki.

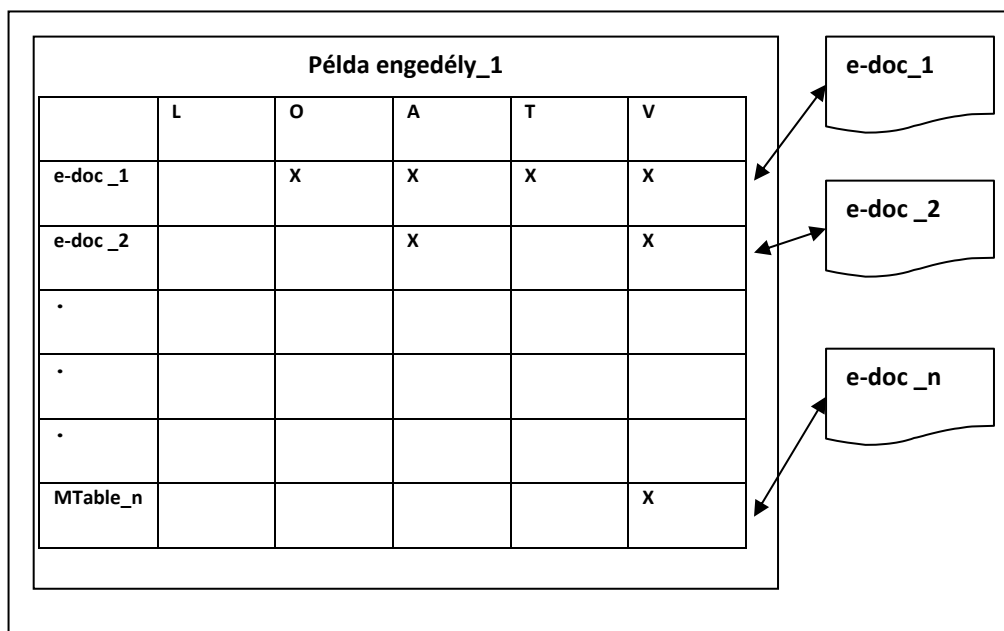


67. ábra A forgatókönyv és az engedély modell közti kapcsolatok

8.12.3.1 Alapfeltevések

Ahhoz, hogy a részletes, jogosultsági engedély rendszer kialakítható legyen a következő előfeltételeknek kell fennállniuk:

- **Feladat:** egy szervezet munkaköri feladatát tükrözi vissza és felhasználható arra, hogy a jogosultsági engedélyek levezethetők legyenek belőle.
- A *strukturális szerepek* meghatározzák, hogy a végfelhasználók milyen jogosultságokkal vesznek részt egy munkafolyamatban és hogyan kapcsolódhatnak egyes védett, bizalmas információforrásokhoz.



68. ábra Példa engedély katalógusra

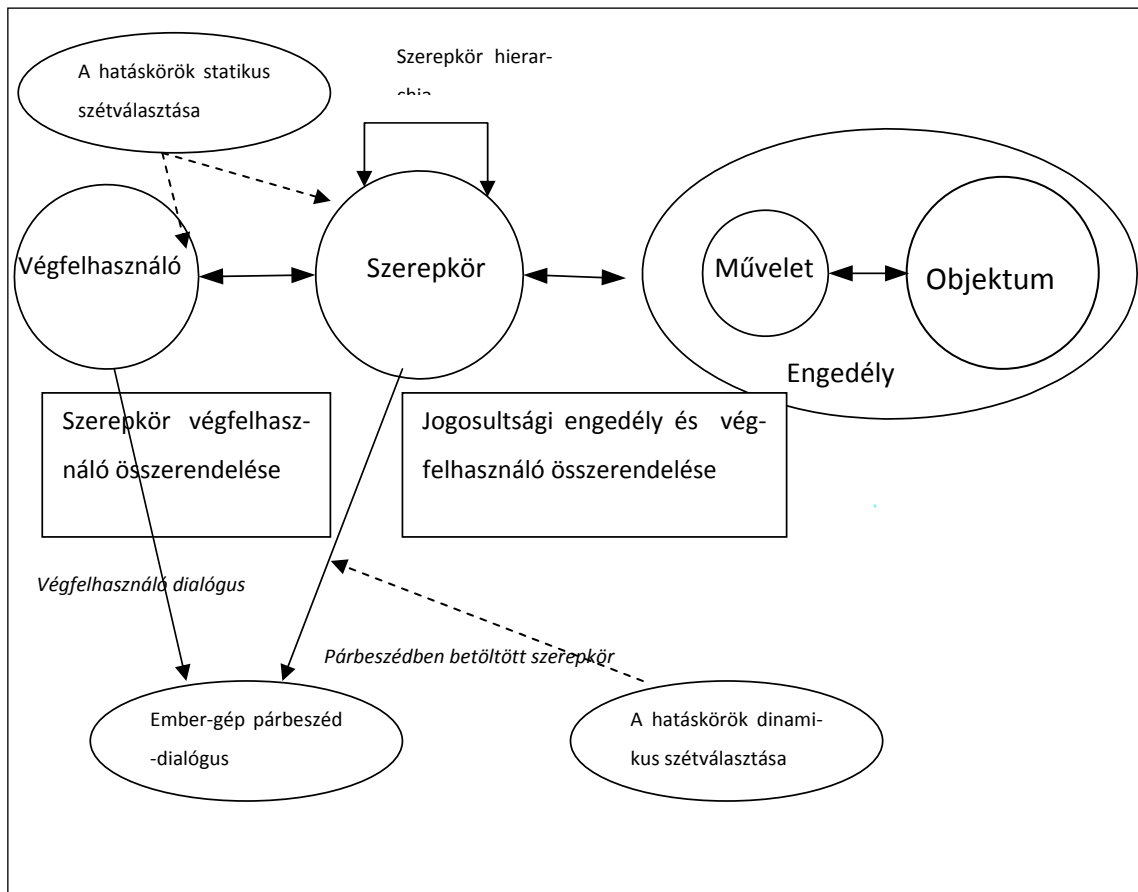
- Az *engedélyek* meghatározzák, hogy mely műveletek vannak engedélyezve egyes védett információforrások esetében.
- Az *engedélyeket* a *funkcionális szerepkörökhöz* kapcsolják hozzá.
- A szabványosított funkcionális szerepkörök a csoportosított és szabványosított engedélyekből állnak össze, amelyeket úgy definiálnak, hogy támogassák a szervezeten belüli információáramlást.

8.12.3.2 Modellek közötti kapcsolat

A használat kialakítása érdekében először a forgatókönyv modellt kell kifejleszteni. Az 67. ábra mutatja azt, hogy hogyan kapcsolódik össze a szerepkör tervezés és forgatókönyvek kialakításának folyamatai.

8.12.4 A hozzáférési és kezelési jogosultsági modell tovább finomítása

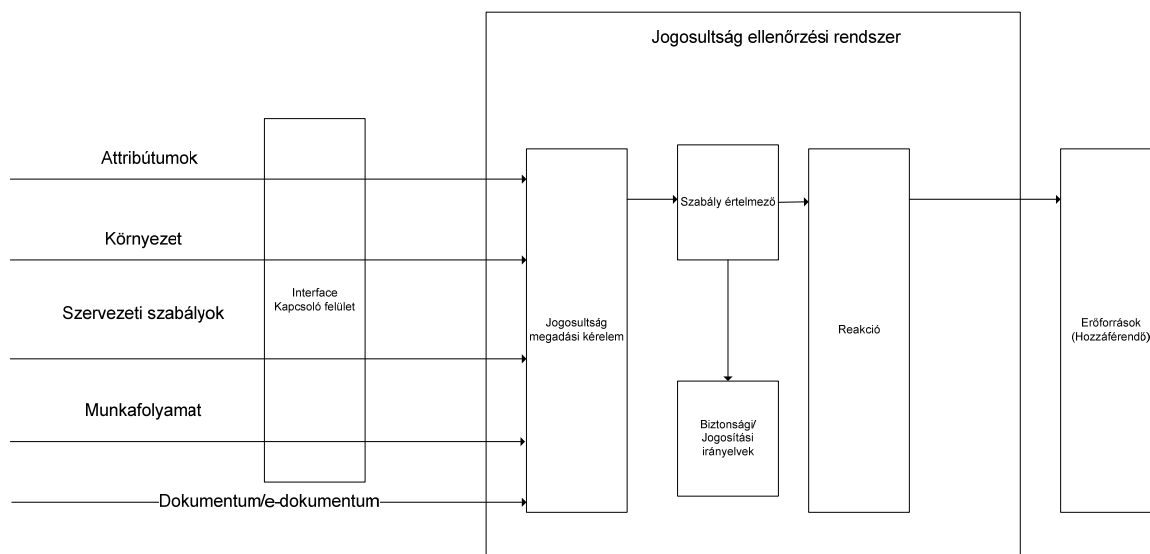
A szerepkör hierarchia egy öröklődési relációt definiál azért, hogy a rendszeradminisztrációs terheket csökkentse. Két fajta szerepkör hierarchia definiálható, egy **általános szerepkör hierarchia** és egy **korlátozott szerepkör hierarchia**. Az öröklődési hierarchia azt jelenti, hogy az engedélyek örökölhettek. Ez egy parciális rendezéshez vezet az engedélyek között.



69. ábra Az ember-gép párbeszéd dinamikusan változó jellegének figyelembe vétele egy elektronikus információrendszer környezetben

A korlátozott szerepkör hierarchia esetében az engedélyek parciális rendezése azt jelenti, hogy egy olyan **gráf elméleti erdővel** állunk szemben, amelyet invertált fák halmaza alkot. Ez lehetővé teszi az erőforrások megosztásának leírásához szükséges engedélyezési rendszer leírását.

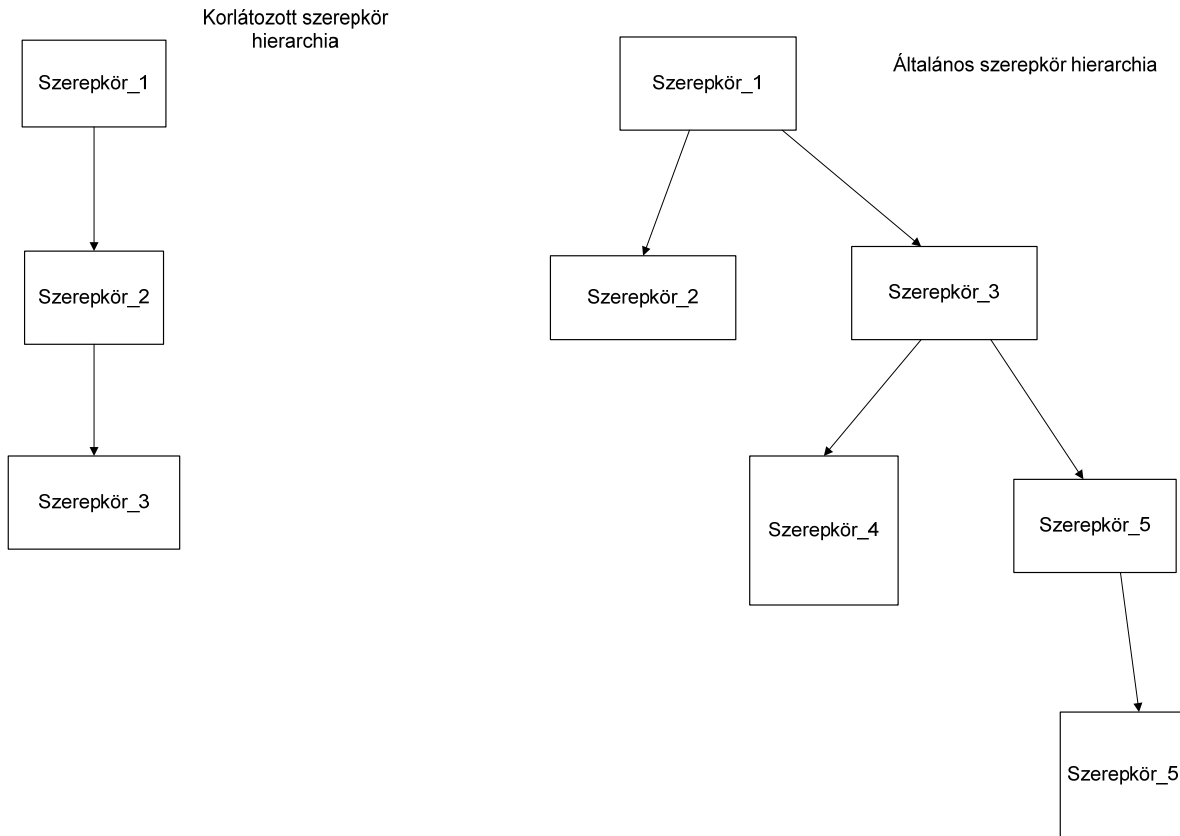
Attribútum alapú hozzáférési jogosultság megadására is lehetőség van. Ez sokkal rugalmasabb, az igényekhez könnyebben illeszthető megoldás, de számítási komplexitása lényegesen nagyobb, a szóba jöhető attribútumok közötti hatványával arányos. A *szerepkör* és *attribútum* alapú megközelítés kombinálható. A tényleges engedélyek kiadása a „**legalacsonyabb privi-*légium*”** elve alapján történik meg.



70. ábra Hozzáférési jogosultság ellenőrzési rendszer

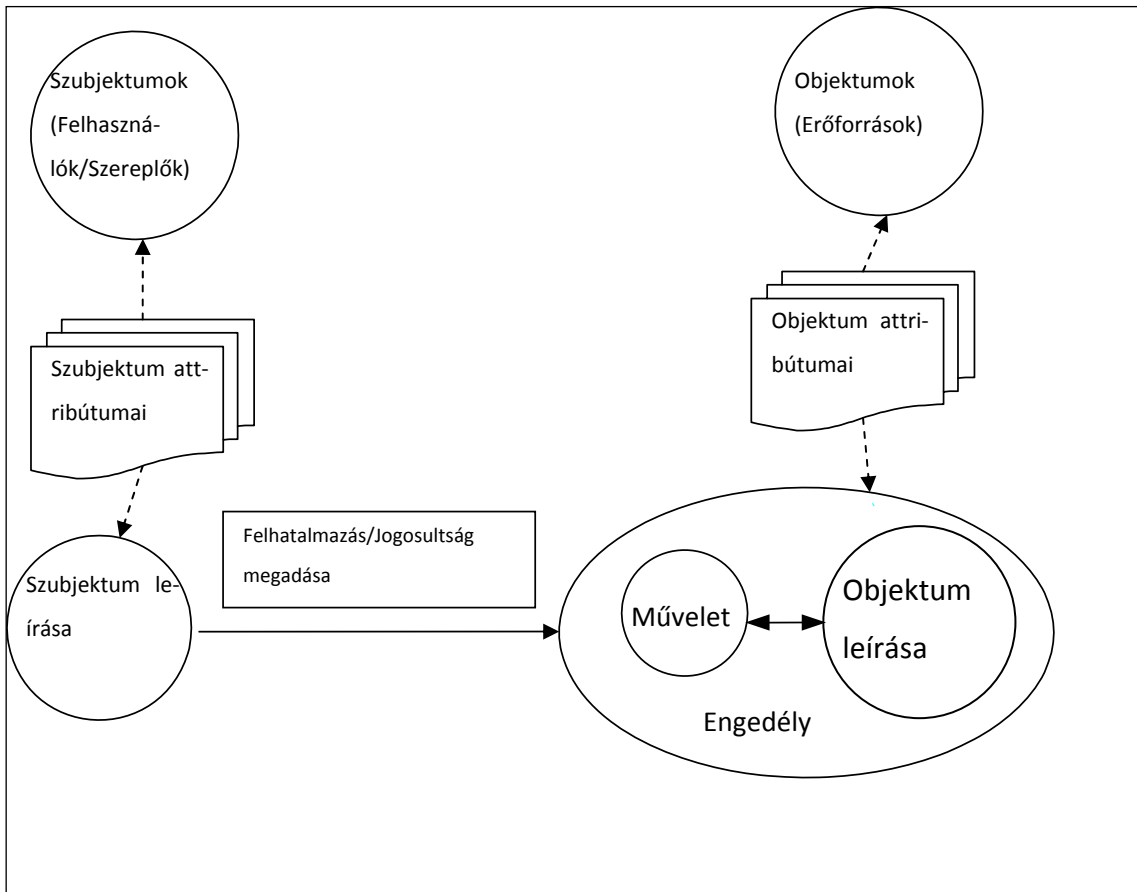
1. **Környezet**- a környezeti tényezőktől függő állapot,
 - 1.1. idő korlátok használatára megengedett időablak;
 - 1.2. biztonsági szint: kibertér védelmi riasztási szint.
2. **Dokumentum** – e-dokumentum
 - 2.1. A konkrét dialógusban dinamikusan megjelenő adattartalom
 - 2.2. Kérdőívre adott válaszok
3. **Szervezeti szabályok** –testre szabható szervezeti előírások
4. **Munkafolyamat** előrehaladásának fázisai, munkafeladatai

Az attribútum alapú hozzáférési jogosultságnak az alapelve az, hogy a felhasználó/szubjektum és az objektum/erőforrás között nem definiál közvetlen kapcsolatot, hanem az aktuális attribútum értékeket használja fel a jogosultság meghatározására. A felhasználó/szubjektumok attribútumai lehetnek statikus jellegűek, pl. munkakör, beosztás a szervezetben. De lehetnek dinamikus jellegű attribútumok is pl. az alany kora, pillanatnyi földrajzi vagy kibertéri helyzete, vagy éppen megszerzett jogosultsága.



71. ábra Szerepkör hierarchiák- Általános és korlátozott

Az objektumok attribútumai a meta-adatok lehetnek, pl. egy dokumentum tárgy, cím attribútuma. Mind a szubjektumok mind az objektumok az attribútumok egy halmazával jellemezhetők. Az engedélyek az objektum leíró adatokból, amelyek az attribútumok és feltételek egy halmazával írhatók le – pl. olyan feltételekből állnak össze mint „kora > 18” vagy „Meghatalmazott=IGEN” –, az adott objektumon megengedett műveletekkel. A jogosultságot a szubjektum leírója és az objektumra vonatkozó engedélyek együtt alkotják. E jogosultsági feltételek kiegészíthetők további feltételekkel, pl. a szubjektum attribútumai és az objektum attribútumainak összehasonlítása révén (feltételek előírása nélkül, csak konstans értékekkel történhet meg összehasonlítás).



72. ábra Attribútum alapú jogosultság kezelés

9 A SZERVEZETI (VÁLLALATI, ÜZLETI) ARCHITEKTÚRA INTEGRÁCIÓJÁNAK KÉRDÉSEI

Szervezeti architektúra megközelítés szempontjából a rendszer integrációnak két nagy területe van az egyik az információ-architektúra, a másik az alkalmazások integrációja.

9.1 Információ architektúra és szervezeti (vállalati, üzleti) integráció

Az információ architektúra kezelését az adatgazdálkodás, adatkezelés, adatmenedzsment fogalmán keresztül lehet megragadni.

Ha az adatgazdálkodás ajánlásainak megfelelően kezelik az adatokat, jelentős eredmények érhetők el:

- csökken az adatok redundanciája, sőt, olykor az adatokat feldolgozó folyamatok redundanciája is, mivel egy nagyvállalat esetében könnyen előfordul, hogy több folyamat is szinte teljesen ugyanazt az adatot állítja elő;
- hatékonyabban és jobb minőségben képezhetők a szükséges információk, vagy akár eddig nem elérhető információk is hozzáférhetővé válnak a hatékony adatmodellezés eredményeképpen;
- az adatfeldolgozás erőforrásigénye és költsége csökken;
- az adatgazdálkodással kapcsolatos folyamatok egyértelműen hozzárendelhetők lesznek a megfelelő csoporthoz.

9.1.1 Az adatgazdálkodás/adatmenedzsment területei

Az adatmenedzsment az adatokkal való gazdálkodás minden területére kiterjed, mondhatni a teljességre való törekvés jellemzi. Rengeteg további részterületre bontható. Egy általánosan elfogadott felosztás szerint az adatmenedzsment a következők szerint tagolható:

9.1.1.1 Adatgazdálkodás irányítás (Data Governance)

A vállalati adatok megfelelő és konzisztens kezelésében résztvevő emberi erőforrások, folyamatok és információs rendszerek irányításával foglalkozik. Néhány kitűzött cél például, hogy optimalizálja, hatékonyabbá tegye az adatmenedzsmentben résztvevők munkáját, vagy, hogy az egyes folyamatok teljesítményével szembeni alapelvárásokat megfogalmazza.

9.1.1.2 Adatarchitektúra- elemzés és tervezés (Data Architecture, Analysis and Design)

Ez a terület az adatok vizsgálatával, transzformálásával, modellezésével és az új vállalati *adatarchitektúrák* kialakításával foglalkozik. E szakterületen történik a létrehozandó *adatarchitektúrákra* vonatkozó követelmények és elképzelések meghatározása, majd az ezek megvalósításához szükséges tervek elkészítése.

9.1.1.3 Adatbázis-kezelés (Database Management)

Az egyik legjobban ismert terület az *adatbázis-kezelés*, *adatbázis-menedzsment*. Három fő feladata az adatok karbantartása, az adatbázisok adminisztrálása, valamint az adatbázis kezelő rendszerek üzemeltetése.

9.1.1.4 Adat-biztonságirányítás (Data Security Management)

Az *adathozzáférés szabályozásával*, a szükségtelen, helytelen, vagy hibás adatok megsemmisítésével, *adatvédelemmel* és *adatbiztonsággal* foglalkozik. Erre a területre minden vállalat esetében kiemelt figyelmet fordítanak.

9.1.1.5 Adatminőség kezelése (Data Quality Management)

Az adatok *minőségellenőrzésével* foglalkozik. Az adatgazdálkodás ezen része szorosan kapcsolódik az előző területhez, mivel ahhoz, hogy bizonyos adatok eltávolításra kerüljenek, szükség van azok minőségének, milyenségének ellenőrzésére az azokat birtokló szervezet szempontjából, hiszen csak a valóban szükségtelen adatok törölhetők.

9.1.1.6 Törzsadatkezelés (Master Data Management)

A törzsadat-kezelés lehetőséget nyújt vállalaton belüli *létfontosságú adatok* összekapcsolására, egymáshoz rendelésére, az összekapcsolás révén ezek az adatszerkezetek egy közös kiindulópontként, tengelypontként („*hub*”) szolgálhatnak a különböző vállalaton belüli részlegek számára. A törzsadat-kezelés segítségével egységes nézet alakítható ki a vállalati adatokról. Ehhez kapcsolódó terület az **adatintegráció**, mely ezt az egységes nézetet képi a különböző forrásból származó adatokból.

9.1.1.7 Adattárház és üzleti intelligencia (Data Warehousing and Business Intelligence Management)

Az üzleti intelligencia és az adattárházak fogalma szorosan kapcsolódik egymáshoz. Az üzleti intelligencia olyan technikákat foglal össze, melyek segítségével eredményesebben hozhatók

meg az üzleti döntések. Ilyek például a különféle jelentés, beszámoló készítési, elemzési, lekérdezési technikák. Segítségükkel a nyers adatok hasznos információkká alakíthatók át.

9.1.1.8 Dokumentum, irat és tartalomkezelés (Document, Record and Content Management)

A dokumentum- és tartalomkezelés koordinálásáért felelős terület. A különféle tartalomkezelő- (CMS, Content Management System), iratkezelő- (RMS, Record Management System) és dokumentumkezelő-rendszerek (DMS, Document Management System) felügyeletére és irányítására vonatkozó módszereket tartalmaz.

9.1.1.9 Meta-adat kezelés (Metadata Management)

Meta-adatokkal különböző típusú és formátumú adatok fő jellemzőit írhatjuk le. A meta-adat kezelés ezeknek az adatoknak a feltárására, majd kialakítására, rendszerezésére és publikálására szolgáltató módszereket és irányelveket.

9.1.1.10 Közönségkapcsolati adatok kezelése (Contact Data Management)

Egy vállalat ügyfelei kapcsolati adatainak kezelésével foglalkozik, pontosabban az ügyféladatok integrálásával, azaz összehangolásával, összeillesztésével. Másrészt az ügyfélkapcsolati rendszerek (CRM) kezelését, sőt, az üzletfolyamatossági tervek (Business Continuity Planning, BCP) kialakításának is az egyik koordinációs eleme.

9.1.1.11 DAMA International

Az előzőekben tárgyalt felosztást a **DAMA International** szervezete határozta meg. Ez egy globális, független, non-profit egyesület, melynek tagjait olyan szakemberek alkotják, akik azon fáradoznak, hogy az információ- és adatmenedzsment módszereit és technikáit tökéletesítsék és továbbfejlesszék. Évenkénti nemzetközi konferencián döntenek az új célkitűzésekről, melyek az egész adatmenedzsment fejlődésének irányát meghatározzák. A DAMA International által meghatározott irányelvek és módszerek gyűjteménye a *Data Management Body of Knowledge* (DAMA-DMBOK Guide) című könyvben is megjelent ([1]).

Amennyiben egy vállalat a korszerű adatkezelés alkalmazása mellett teszi le a voksát, mérlegelnie kell, hogy érdemes-e egyáltalán újraszerveznie belső felépítését (ld. BPM, BPR stb.).

Erre a mérlegelésre, elemzésre feltétlenül szükség van, hiszen a belső átszervezések minden esetben költségekkel járnak. Az egyes folyamatok megvalósításához új rendszerekre van szükség, melyek már önmagukban is elég költségesek, továbbá elengedhetetlen a hozzáértő

szakemberek alkalmazása is, akik felkészítik a vállalat alkalmazottait az új eszközök használatára. Így már érthető lehet, hogy a vállalatok egy része miért nem dönt a szakszerű adatgazdálkodás megvalósítása mellett. Hiánya sok - egyébként kihasználható - lehetőség elszalasztásához vezethet, alkalmazása hosszútávon viszont közvetlen erőforrás megtakarítást eredményezhet.

9.1.2 Törzsadat-kezelés

A következő részben az adatgazdálkodás témakörén belül a törzsadat-kezeléssel foglalkozunk. Az irodalomban gyakran **MDM**-ként is hivatkoznak rá, ami az angol **Master Data Management**-ből ered. A törzsadat-kezelés segítségével magas szintű, egységes nézet alakítható ki a vállalaton belüli adatokról.

9.1.2.1 Miért fontos a törzsadat-kezelés?

Az elmúlt évtizedek folyamán jelentősen eltérő módszerek és technikák alakultak ki az adatgazdálkodásban attól függően, hogy milyen jellegű adatokról van szó. Az üzleti élet egyes területei más és más információkat igényelnek, így az azonos adatokkal szemben is más és más szempontokat részesítenek előnyben. Ebből kifolyólag különböző *alkalmazás architektúrák* alakultak ki, melyek csak az adott terület adatainak nyilvántartására és kezelésére specializálódtak. Általánosan elmondható, hogy bármilyen területről legyen is szó, van néhány fogalom, amely mindig előkerül ezekkel a rendszerekkel kapcsolatban: *adatdefiníció*, *adatstórák*, *táblaszerkezetek*, az *alkalmazás funkcionalitása*. Ezeknek a fogalmaknak persze más és más definícióival találkozhatunk, attól függően, hogy éppen melyik üzleti terület szempontjából vizsgáljuk őket.

A különböző üzleti területeken használt adatok sokszor átfedik egymást, redundánsak, vagy ugyanazt az adatot tartalmazzák csupán valamilyen más szempont szerint csoportosítva. Gyakran előfordul az is, hogy ugyanolyan néven, két teljesen eltérő jellegű és tartalmú adat szerepel. Ezen adatok adminisztrálása egészen addig nem okoz problémát, míg külön-külön kezeljük őket. A mai szervezeti (vállalati, üzleti) trendek és törekvések viszont abba az irányba mutatnak, az összegyűjtött adatokat egységes információ-együttessé alakítsák át. Ez az *információbázis* egy jól felhasználható *tudásbázist* eredményez. Ennek az információbázisnak alapja az volna, hogy az egyes szervezeti (vállalati, üzleti) területekre jellemző létfontosságú adatokat, adatvagyonot egy egységes adatgyűjteménnyé transzformálják át. Mind ehhez az adott szervezetnek egyértelműen meg kell tudni határoznia szervezeti (vállalati, üz-

leti) *fogalmait* és az ezeket a fogalmakat *reprezentáló adatokat*. Ez az igény tette lehetővé olyan módszerek kialakulását, melyek célja a *vállalati információk integrálása, kezelése, menedzselése és megosztása*, ezáltal kialakítva egy ténylegesen *integrált vállalati adatkörnyezetet*. Ez a lényege a *törzsadat-kezelés* törekvéseinek is - megmutatni, hogyan lehet kialakítani a vállalat szempontjából fontos és jó minőségű üzleti információ-objektumokat, irányítani azok használatát, szinkronizálását és úgy optimalizálni használatukat, hogy azok minél inkább megfeleljenek az adott vállalat stratégiáinak.

A kilencvenes évek közepe felé olyan centralizált és integrált rendszerek kezdtek elterjedni, nevezetesen a vállalat irányítási rendszerek (ERP), illetve az ügyfélkapcsolati rendszerek (CRM) stb. Ezeknek köszönhetően jöttek létre az első *adattárházak*, melyek az összegyűjtött, centralizált, aggregált és kumulált adatokat tartalmazzák.

9.1.2.2 A törzsadat fogalma

Törzsadat

- Minden szervezetben vannak olyan általánosan meghatározható fogalmak, amik az üzleti folyamatok fókuszában állnak. A különböző létfontosságú szervezeti (vállalati, üzleti) *objektumok*, üzleti *fogalmak* vagy üzleti entitások, az ezek mögött álló adatok reprezentálják a vállalat **törzsadatait**.

A különböző vállalati alkalmazások által használt *törzsadatok* jellemezhetők *metaadatokkal*, attribútumaikkal, definícióikkal és kapcsolataikkal. A törzsadatok, azok a kulcsfontosságú adatok, amelyeknek a legnagyobb jelentősége van a vállalat szempontjából. Azok a dolgok, amelyeket mérnek, amikről jelentéseket, beszámolókat készítenek, amiket elemznek. Általános példák a törzsadatokra a következők lehetnek:

- vásárlók,
- termékek,
- szállítók,
- kereskedők,
- alkalmazottak,
- pénzügyek,
- előírások,
- partner arculatok adatban („profilok”), stb.

Egy adott törzsadat kategórián belül implicit és explicit hierarchikus kapcsolatok lehetnek. Például az ügyfél adatai – kontextustól függően – , más és más ábrázolásban szerepelhetnek egyes rendszereken belül, vagy egy termék több alkotó termék, alkatrész együtteseként jelenhet meg (ld. például darabjegyzék, Bill of Material, BoM).

9.1.2.3 Törzsadatok meghatározása

Az egyik nehézség a törzsadatok meghatározásával kapcsolatban az, hogy törzsadatok a szervezet több részlegén belül is meghatározhatók, tehát mondjuk ugyanarra az *ügyfélre* vonatkozó adat előfordulhat az értékesítés adatai között vagy a számlázáson belül.

- Ha olyan *entitásokat* tudunk azonosítani az egyes adatbázis rendszerekben, melyek ugyanazt az *üzleti fogalmat* képviselik akkor, ezeket a fogalmakat fogjuk össze egy egységbe. Így egyértelműen azonosíthatjuk azokat a fogalmakat, attól függetlenül, hogy a vállalat melyik részlegén belül használják őket.
- Állapítsuk meg az egyes adatgyűjtemények (halmazelméleti) metszeteit, ha lehet.
- Alakítsuk ki azokat a *folyamatokat*, melyek lehetővé teszik a vállalati alkalmazás számára az újonnan létrehozott szervezeti (vállalati, üzleti) objektumok egységes nézetének elérését, lekérdezését, visszakeresését.

9.1.2.4 Kulcs entitások és létfontosságú adatelemek

Kulcs adatentitások

A **kulcs adatentitások** azok az alapvető információ-objektumok, melyeket az üzleti alkalmazások azért használnak, hogy funkcióikat elláthassák. Ha *kulcs adatentitások* koncepciójáról beszélünk, akkor azok többnyire az üzleti felhasználásuk szempontjából kerülnek szóba és ilyen szempontból vizsgáljuk őket. Technikai értelemben, ha például *multi dimenziós adattárházak* esetében vizsgáljuk őket, a kulcs adatentitások egy-egy dimenzióként jelenhetnek meg. Törzsadat-kezelés esetében azok a törzsadat-objektumok lesznek a kulcs adatentitások, melyek köré a vizsgált rendszert szervezzük. Megfelelő kezelésük természetesen elengedhetetlen, hiszen ezeket az adatokat egyedi entitásokként kell tárolni, elkerülve ebből kifolyólag a nem kívánt redundáns előfordulásukat a különböző alkalmazások közt.

Létfontosságú adatelemek

Létfontosságú adatelemeknek nevezzük azokat az adatelemeket, melyek hiányában a vállalat működéséhez szükséges tevékenységek, műveletek nem megvalósíthatók. Például egy vállalat szempontjából létfontosságú adatelemek lehetnek azok az entitások, melyek vé-

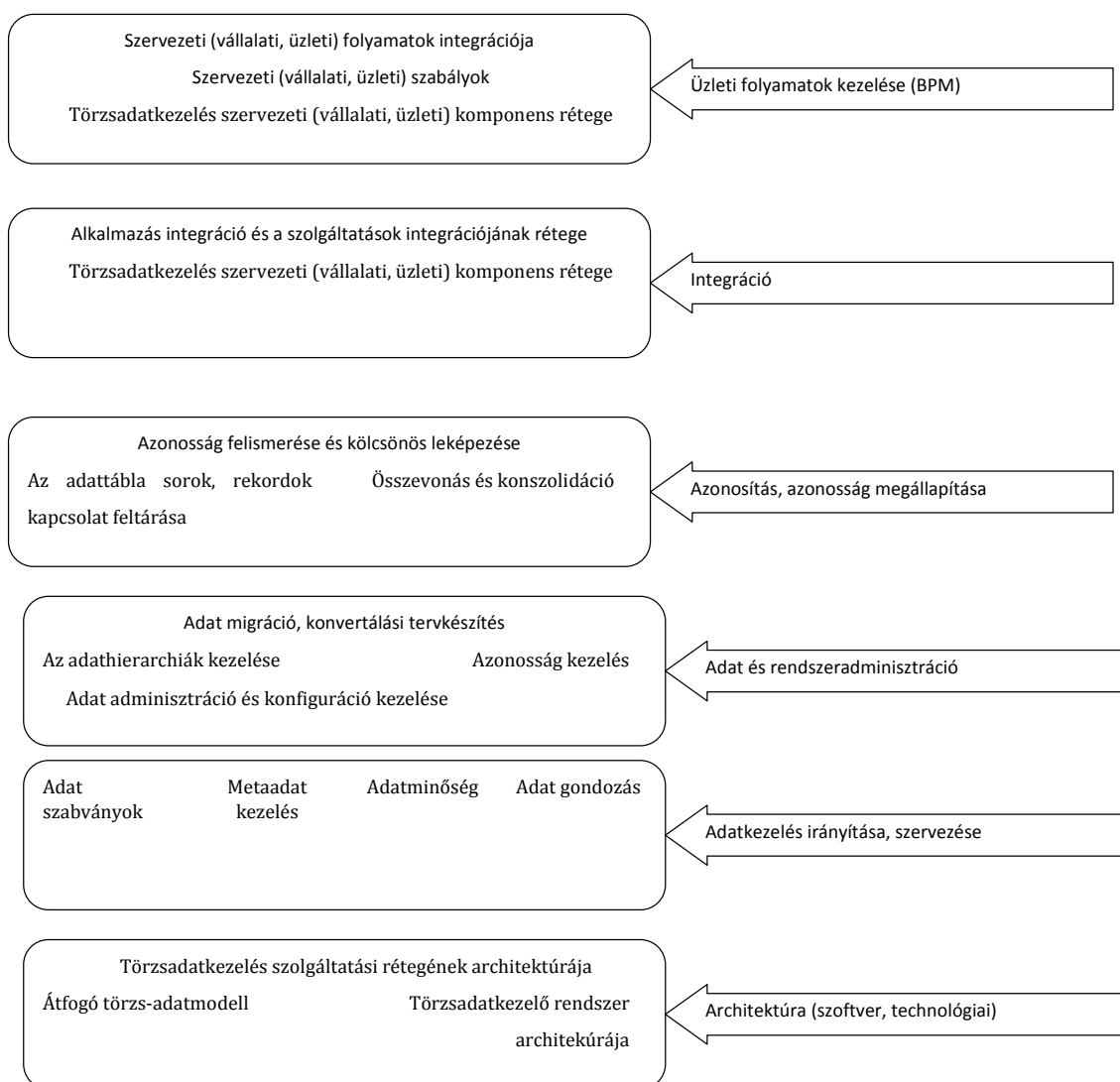
dett személyes információkat tartalmaznak, amiket a különböző pénzügyi jelentésekhez használnak fel. Vagy az olyan adatelemek melyek létfontosságú döntéshozási folyamatoknál, vagy a vállalat eredményességének mérésénél használnak.

Létfontosságú adatelemek meghatározásának nem az a legmegfelelőbb módja, ha az összes vállalati adatelemet tömegesen próbáljuk elemezni valamilyen szempont rendszer szerint. Hanem ehelyett azokra az „információtermékekre” kell összpontosítani, amelyekkel a végfelhasználók közvetlen kapcsolatba kerülnek. Át kell gondolni azt, hogy melyek azok a kimutatások, jelentések, elemzések, műveletek, melyek *kritikus fontosságúak* lehetnek a vállalat számára. Meg kell keresni azokat az adatelemeket, melyek a leggyakrabban vannak jelen ezekben. Miután azonosítottuk az ilyen adatelemeket, fel kell térképezni azok kapcsolatait más adatelemekkel, és meg kell keresni azokat, melyektől a kritikus fontosságúnak talált adatelemek függenek. Vegyünk például egy kimutatást, mely tartalmazza az adott vállalat teljes éves eladásának nagyságát. Ekkor ez az érték egy kritikus fontosságú adatelemnek tekinthető. Ennek az adatelemnek az értéke azonban függ a különböző részlegek eladási adataitól, melyek tovább függhetnek az egyes termékek eladásától. Látható, hogy minél inkább haladunk visszafelé a függőségi láncban, annál több adatelemet azonosíthatunk kritikus fontosságú adatelemként.

Minden adatelem kritikus fontossága arányban van azzal, hogy milyen mértékben járul hozzá a vállalat üzleti életének tovább viteléhez. Egy adatelem minősége attól függ, milyen hatással van az üzletre. Éppen ezért a létfontosságú adatelemek minősége mérhető, és meghatározható olyan küszöbértékek melyekhez viszonyítva minőségük megfelelő.

9.1.2.5 A törzsadat-kezelés komponensmodellje-architektúra építőelemek

Ahogy az ábrán (73. ábra) látható, a törzsadat-kezelés komponens és szolgáltatásmodellje nagyon sok területet érint.



73. ábra A törzsadat-kezelés szolgáltatás- és komponens architektúra

9.1.2.5.1 Adatintegráció és alkalmazásintegráció

Az *MDM* módszer egyik fő célja vállalati adatok egy olyan egységes nézetének és ábrázolásának létrehozása, ami az alapját adja az adat, mint **vállalati vagyon** kezelésének. Ennek révén ez az adatszerkezet, adatrepresentáció magas minőségű *vállalati vagyonná* válhat, mely könnyen megosztható az egész vállalaton belül. Viszont a törzsadat-kezelés célkitűzéseinek eléréséhez nem csupán az *adatok integrációjára* van szükség. Az igazi eredmény akkor érhető el, ha a kialakított törzsadat-entitások reprezentációi integrálva lesznek az üzleti folyamatokba is, így az őket megvalósító *alkalmazások* ténylegesen egy egységes, szinkronizált módon végezhetik műveleteiket az egyes törzsadatokkal. A vállalati alkalmazások esetében valamilyen módon lehetővé kell tenni az egységesített törzsadatokhoz való hozzáférést. A törzsadat-kezelés segítségével biztosítható, hogy a már meglévő alkalmazásinfrastruktúrát a

lehető legkisebb mértékben kényszerüljenek megváltoztatni az új struktúrára való áttérés-kor.

9.1.2.5.2 Törzsadatkezelés komponensszolgáltatási rétege (MDM Component Service Layer)

Minél tovább jutunk a törzsadat-kezelés megvalósításában, minél inkább integráljuk azt, az üzleti alkalmazások egyre inkább támaszkodni tudnak a törzsadat-objektumok koncepciójára és funkcionalitásaira, így egyre inkább támogatva az új szervezeti (vállalati, üzleti) architektúrát. A törzsadat-objektumok reprezentációinak *szabványosításával* az alkalmazásfejlesztőknek egyre kevesebb figyelmet kell fordítani az olyan hagyományos adat-orientált problémákra, mint az adathozzáférés szabályozása, adatkezelő műveletek jogosultsági rendszere, stb. Helyette az új rendszer kínálta funkcionalitást kihasználva elégíthetik ki az üzleti követelményeket azokra az alacsony szintű adatvezérelt szolgáltatásokra támaszkodva, melyek felépítését és tervét az MDM komponensszolgáltatás rétege határozza meg.

9.1.2.6 Törzsadatok és törzs-metaadatok azonosítása

Ahhoz, hogy a törzsadatok megfelelő kezelését megvalósíthassuk, először képesnek kell lennünk a törzs-metaadatok azonosítására és kezelésére. Ehhez azonban megfelelően kell az egyes entitások különböző megjelenési formái alapján kialakítani a meta-adatokat. A meta-adatok meghatározásakor három szempontból kell vizsgálni az adatokat: strukturális, szintaktikai és szemantikai szempontból. A szintaxis szempontjából az adatokat közvetlenül összehasonlíthatjuk *típusuk* és *méretük* szerint, így azonnal láthatók a különbségek és hasonlóságok. *Strukturális* szinten is láthatók az eltérések. Ha jól megvizsgáljuk az egyes reprezentációkat, láthatjuk, hogy két reprezentáció tartalmaz olyan attribútumokat is, melyekből akár egy újabb törzsadat típus is kialakítható. Ilyenek lehetnek például az egyes kapcsolati információk, melyek struktúrákban kék háttérrel vannak jelölve. Végül, *szemantikai* szempontból kell megvizsgálni az entitásokat. A gyakori eset, hogy az ügyfél fogalma szemantikailag más jelent a két szervezeti egység esetében.

9.1.2.7 Adatmenedzsmenten belüli résztvevők, szerepkörök

Az MDM programban résztvevő emberek megfelelő szervezéséhez pontosan azonosítani kell az egyes **szerepköröket**, és pontosan el kell határolni azok a feladat-, hatás – és felelősségi körét egymástól. Az, hogy mikor pontosan milyen feladatot rendelünk az egyes szerepkörök-

höz, mindig az adott környezettől függ. A szerepkörök egy általános felosztása lehet a következő:

Szervezeti (vállalati, üzleti) ügyfél		
Meta-adat elemzés	Alkalmazás felelős (alkalmazási rendszer gazdája)	Adatgazdálkodás, adatvagyon menedzsment irányítása („Governance”)
	Adatminőség felügyelet	
	Rendszerüzemeltetés alkalmazottai	
	Rendszerfejlesztők	
	Információ architekt (architektúra tervező)	
Szervezet (vállalat, vállalkozás) felső vezetői		

6. ábra - A törzsadat-kezelés során felmerülő szerepkörök

Azért, hogy meggyőződhessünk arról, hogy az egyes résztvevők számára a feladataikhoz szükséges körülményeket, feltételeket megteremtették-e, szükséges legalább körvonalában ismerni az egyes szerepeket betöltő csapatok, vagy egyének felelősségi, hatás és feladat körét, a különböző adatgazdálkodási folyamatok megvalósítása során felmerülő feladatokra vonatkozóan. Ez jól szemléltethető az úgynevezett **RACI** mátrix segítségével, mely hatékonyan alkalmazható új folyamatok bevezetésénél. A RACI mátrix négy szempontból vizsgál egy folyamatot:

- ki a felelős érte (R - Responsible),
- ki hozza a döntéseket (A - Accountable),
- kivel kell konzultálni (C - Consulted),
- kit kell informálni (I - Informed).

	Törzsadat kezelés felelős	Szervezeti (vállalati, üzleti) ügyfél	Alkalmazás felelős (alkalmazási rendszer gazdája)	Információ architekt (architektúra tervező)	Adatgazdálkodás, adatvagyon menedzsment irányítása	Meta-adat elemző	Rendszerfejlesztők	Rendszerüzemeltetés alkalmazottai
Adat összehango-	A	C	C	R	R	R		C

lási folyamat kialakítása								
Adatgazdálkodással szemben támasztott követelmények elemzése	I	C	C	A	R	R		
Meta-adatok elemzése	I	C	C	R	R	A	C	C
Törzsadat adatmodell létrehozása	I	I	I	A	C	R	C	I

18. Táblázat - Példa egy folyamat RACI mátrixára

9.1.2.7.1 A felső vezetés (Senior Management)

A felső vezetés támogatása nélkül igazán nehéz lenne szervezni bármilyen vállalati tevékenységet, projektet. Ezen a szinten a vezetők majdnem mindig motiváltak, mivel szeretnék megmutatni, hogy csapataik teljesítménye közvetlen hatással van a vállalat eredményességére. *Törzsadat-kezelés* megvalósítása során dolgozniuk kell azon, hogy a meglévő rendszerek funkcionalitását is fenntartsák, és az új rendszer/projekt kezdeményezéseknek is eleget tegyenek. Ez szintén ösztönzőleg hat.

A felső vezetésnek kell úgy szervezni az egyes folyamatokat, hogy a vállalat azok a részlegei, melyek nem érintettek az éppen folyó migrációban, továbbra is végezhesék feladataikat. Továbbá nekik kell gondoskodni arról is, hogy az új rendszer által megkövetelt alkalmazási rendszer változásokat megismertessék az alkalmazottakkal, hogy azok megfelelően kezdjék alkalmazni az új rendszert, így a vállalat ki tudja aknázni az új lehetőségeket.

9.1.2.7.2 Szervezeti (vállalati, üzleti) ügyfelek (Business Clients)

Ők az egyes vállalati alkalmazások felhasználói. Ezeknek az ügyfeleknek/végfelhasználóknak a sikeressége attól függ, mennyire jó az alkalmazások által használt adatok rendelkezésre állása. Ha a már meglévő alkalmazások által használt adatok megfelelnek a végfelhasználók elvárásainak, akkor az egyes adatmigrálási, adatmigráció folyamatok nem érintik őket. Az ada-

tok beleolvasztása a fő törzsadat tárolóba csak akkor válik lényegessé a végfelhasználók számára, ha ez a folyamat rontaná valamiért az adatok rendelkezésre állását.

Az ügyfelek támogatásának minősége két fontos dologtól függ. Először, a törzsadatkezelést megvalósító csapatnak egyrészt dokumentálni kell a végfelhasználók adatokkal kapcsolatos elvárásait, az alkalmazások szolgáltatásaival kapcsolatos elvárásait és biztosítani kell a végfelhasználókat arról, hogy ezek nyomon követik, felügyeli, monitorozzák és meg fognak valósulni. Másodszor, a megvalósító csapat számára elengedhetetlen megérteni, hogy a meglévő adatobjektumokat jelenleg hogyan használják, hiszen ekkor tudják csak megfelelően megvalósítani a *törzsadat-objektumokat az alkalmazásokban*. Nyilvánvaló, hogy az szervezeti (vállalati, üzleti) folyamatok modellezésében és az adatokkal szembeni követelmények meghatározásában részt kell vennie a végfelhasználóknak is.

9.1.2.7.3 Alkalmazásért felelősök (Application Owners)

Minden alkalmazást, amelyik érintett olyan adatobjektumok használatában, melyek az MDM megvalósítása kapcsán felmerülnek, meg kell változtatni, hozzá kell igazítani az új rendszerhez, úgy hogy tovább már ne a lokális adataikat és a másolatokat használják, hanem a *törzsadatokat*. Ez azt jelenti, hogy a siker érdekében a törzsadatok által képzett vállalati vagyont meg kell osztani az *alkalmazás felelősökkel*. Minden alkalmazás-felelősnek tisztában kell lennie azzal, hogy mi az alkalmazás szolgáltatásaival kapcsolatos igény. Ők az egész MDM-et gyakran veszélyforrásnak tekintik, nem pedig a lehetőséget látják benne, mivel ha jelentős változások megvalósítására van szükség, akkor a már bevált adatvagyon egy még nem használt, bizonytalan adatstruktúrára kell lecserélni.

Az adatokkal szembeni követelmények meghatározása során az alkalmazásfelelősökkel meg kell értetni, hogy mennyire fontos az, hogy határozzák meg és pontosan dokumentálják azokat a követelményeket az *új adatmodellel* szemben, melyeket az *alkalmazások műveletei* megkövetelnek. Ez azért fontos, mert nem az a cél, hogy egy találmányra kialakított modell köré építsük óriási módosítások árán a meglévő alkalmazásokat. Épp ellenkezőleg, egy jó adatmodell kialakításával és az elvárások pontos meghatározásával elérhetjük, hogy valóban csak akkor kelljen módosítani az alkalmazásokon, ha az elengedhetetlen.

9.1.2.7.4 Információ-architektúra tervező (Information Architect)

A törzsadat-objektumok modelljének ki kell elégítenie az alkalmazások már meglévő igényeit, miközben eleget kell tenniük az új változtatásoknak is. Ennek a két követelménynek az információ-architektúra tervező közreműködésével kell eleget tenni.

9.1.2.7.5 Adatgazdálkodás irányítása és adatminőség (Data Governance and Data Quality)

Egy új vállalati rendszer/projekt/folyamat kezdeményezés az adatok létrehozásával, elérésével, használatával, módosításával és megszüntetésével kapcsolatban mindig valamilyen új korlátozásokkal, követelményekkel is jár. Azért, hogy biztosak lehessünk abban, hogy ezek az új megszorítások nem sértik meg az adatkezelési előírásokat, és nem rontják az adatminőséget, a megvalósítást végző csapatnak meg kell ismernie az adatok tulajdonosait, felelőseit és az adatkezelési előírásokat.

9.1.2.7.6 Meta-adat elemző (Metadata Analysts)

Tudjuk, hogy a meta-adatok az MDM egyik kulcs komponensét képezik. Ezek kialakításáért és kezeléséért pedig a meta-adat elemzők a felelősek.

9.1.2.7.7 Rendszerfejlesztők (System Developers)

Az MDM megvalósítása érthető módon hatással van a meglévő rendszer teljesítményére, tárhely igényére is. A rendszerszervezők/rendszerelemzők/rendszerfejlesztők feladata, hogy újraszervezzék a rendszer struktúráját informatikai, műszaki, technikai szempontból úgy, hogy az új törzsadat-kezeléssel integrált rendszernek is megfelelő háttérrel biztosítson a működéshez.

9.1.2.7.8 Üzemeltetés (Operations Staff)

Az egyik rejtett veszély a törzsadat-kezelés kialakítása során, hogy ez a csoport gyakran átlépi a szabványosított és szabályozott adatelérési és módosítási eljárásokat és előírásokat, az adatok elérésének és módosításának – gyakran *ad-hoc* – érdekében. Valójában, néhány vállalatnál bevett eljárás, hogy amennyiben valamilyen módosítást kell végrehajtani az adatokon, direkt módon férnek hozzá azokhoz (pl. SQL utasítások futtatása), ahelyett, hogy végigjárnák a szabályzatok előírásának megfelelő útvonalat (Informatikai Biztonság Irányítási Szabályzat, Informatikai Biztonsági Szabályzat, Adatvédelmi és biztonsági szabályzat). Az ilyen szabálytalan tevékenységek azonban elfogadhatatlanok egy törzsadat-kezelést alkalmazó

környezetben. Az ilyen módszereket alkalmazó csoport tagjainak haladéktalanul változtatnia kell szokásain, és a szabályszerűsége, és a szabályzatoknak megfelelően, velük összhangban kell eljárniuk, hiszen a centralizált adatok sérülése nagy károkhoz vezethet.

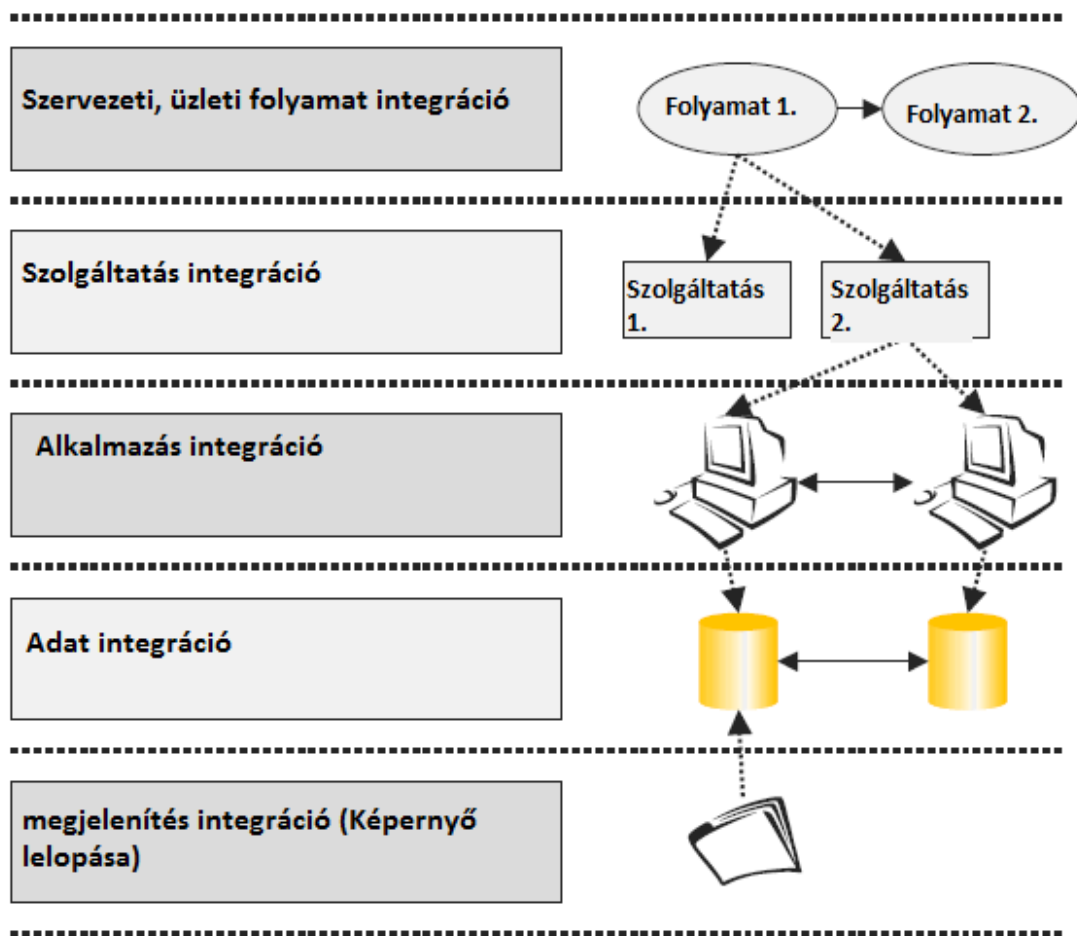
9.2 Alkalmazásintegráció

A **holisztikus** (*rendszer szemléletű*) megközelítés egyik irányító elve az, hogy mint az egész több mint a részek egyszerű (pl. aritmetikai) összege (**szinergia**). Ebben a felfogásban megkülönböztetünk:

- Adatintegrációt, információ-integrációt;
- Funkcionális integrációt;
- Alkalmazásintegrációt.

Integrált rendszer

- Az **integrált rendszer** alatt egy olyan újonnan létrehozott rendszert értünk, amely logikusan összetartozó elemeket, részrendszereket kapcsol össze vagy egyesít
- Az **integráció** alatt azokat az erőfeszítéseket értjük, amelyek az elkülönülő folyamatokat és struktúrákat összekapcsolják illetve e tevékenység révén létrejövő eredményt, amely *összhangot* teremt szervezeti (vállalati, üzleti) tevékenységek, funkciók, folyamatok, adatok, rendszerek között.



74. ábra Integráció szintjei

1. **Megjelenítés szintű integráció:** Az adatok aggregált megjelenítés több forrás információrendszerből. Egy Web portál, amely aggregálja és megjeleníti egy ügyfél részvény portfólióját az egyik informatikai rendszerből, és bank folyószámla egyenlegét egy másik információrendszerből a megjelenítési szinten történő integrációnak tekinthető. Ebben az esetben azonban nincs közvetlen adatcsere, kommunikáció a két informatikai rendszer között.
2. **Adat integráció:** A különböző adatbázisokban tárolt adatok között szinkronizálás révén megvalósított integráció. Ha például két különböző adatbázis tárolja törzsadatként az ügyfél lakcímének adatait, akkor az egyik adatbázisban történt változtatásokat szinkronizálni kell a másik adatbázissal. Ha a szinkronizálás csak egy bizonyos idő elteltével következik be, akkor az adatok aktualitásának értéke jelentősen csökken.
3. **Alkalmazásintegráció:** Akkor beszélünk alkalmazás integrációról, amikor az alkalmazások bizonyos funkciókat kölcsönösen elérhetővé tesznek egymás számára. Elterjedt vállaltirányítási rendszerek (pl. SAP, PeopleSoft, Baan, Siebel, ORACLE , MS Navision stb.) gyakran elérhetővé teszik bizonyos szolgáltatásaikat jól definiált alkalmazásprogramozási kapcsolófelületeken (interface) keresztül (API, Application Programming Interface).

4. **Szolgáltatás integráció:** Újra felhasználható szolgáltatások, amelyek más alkalmazások számára is rendelkezésre állnak. Például egy olyan szolgáltatás, amelynek segítségével egy ügyfél állapota lekérdezhető létrehozható olyan szolgáltatásokból, amelyek már léteznek a szervezet informatikai infrastruktúrájában. Az ilyen szolgáltatásokat általában több egyedi és specifikus alkalmazási funkcióból lehet felépíteni.
5. **Folyamat integráció:** Egy olyan modell megfogalmazását jelenti, amelyben a szervezeti (vállalati, üzleti) folyamatokat, vagy másképpen a munkafolyamatokat (workflow) definiálják oly módon, hogy az újrafelhasználható szolgáltatások e definiált és végrehajtható modellből meghívhatók, végrehajthatók. A folyamatintegráció különösen jelentős szerepet játszik a B2B (Business to Business), azaz vállalkozások közti rendszerekben, ahol az együttműködési képesség (kollaboráció, collaboration) lényeges sajátosság. A szervezeti (vállalati, üzleti) folyamatok vezérlik az üzleti partnerek közti tranzakciókat és kapcsolat tartási lépéseket.

Adatcsatolt és állomány-átviteli technikák során az egyes alkalmazások adataikat egy vagy több közvetítő adaterőforráson (adatbázison, állomány-kiszolgálón, állományokon) keresztül cserélik ki. Előnyeik közé főleg az tartozik, hogy relatívan olcsó, egyszerűen programozható.

Az **EDI** különösen a gépjárműiparban és a kereskedelmi láncoknál elterjedt megoldás. A *B2B*-piac technológiáinak egyik első módszerének tekinthető, vállalatközi területen. Alapvető célja elektronikus bizonylatok, okmányok, okiratok (pl. számla) különböző, jogi személyiségű szervezetek közötti cseréje.

Egyik viszonylag korszerű a **közteszoftver-technológia** (*Middle-ware*). Közteszoftveren azokat a technológiákat magában foglaló informatikai eszközöket értik, amelyek osztott adatfeldolgozási környezetben képesek valós időben integrálni az egymástól eltérő alkalmazásokat, adat-erőforrásokat és szervezeti (vállalati, üzleti) folyamatokat, függetlenül azok operációsrendszer-, hálózatprotokoll-környezetétől és helyétől. A legkorszerűbbnek tekintett modell a Web szolgáltatásokon, üzenetcsere és **szolgáltatási sínen** (*Enterprise Service Bus*) alapuló szolgáltatás-központú integráció megvalósítása.

Az integráltság biztosításához feltétlenül szükséges, hogy legyen egy egységesen definiált vállalati adatmodellre épülő **központi adatbázis**, amelyhez minden alrendszer/modul hozzáférhet. Ennek révén lesz garantált az adatok *konzisztenciája* és *redundancia-mentessége*. Az információ-architektúra kialakítása szempontjából egyik fontos módszere e megközelítésnek a **törzsadat-kezelés**.

9.2.1 Alkalmazási rendszerek integrációja

Az alkalmazások integrációja és integrált rendszerek iránti igény a következő problémák miatt merül fel. A vállalaton belüli, helyi/lokális rendszerek nem támogatják a szervezeti (vállalati, üzleti) folyamatokat sem vállalaton belül, sem azon kívül. Az *ad hoc* alapon, külön-külön

összekapcsolt rendszerek rendszerüzemeltetés, rendszeradminisztrációja nehézkes, nem skálázható és megbízhatatlanok.

A vállalati alkalmazások integrációja (Enterprise Application Integration, röviden EAI)

- **A vállalati alkalmazások integrációja (Enterprise Application Integration, röviden EAI)** a folyamatok, szoftverek, hardverek, szabványok olyan kombinációja, amely két vagy több vállalati rendszer gördülékeny, hézag- és akadálymentes együttműködését teszi lehetővé. Az *alkalmazás integráció* oly módon takarít meg költséget és időt, hogy megteremti az együttműködés lehetőségét a meglévő (elavult, régi, „legacy”) és az új rendszerek között, csökkenti az üzleti reakcióidőt, lehetővé teszi a több rendszert is érintő tranzakciók feldolgozását.

19. Táblázat A szocio-technológiai rendszerszemléletű és az információrendszerek informatikai szemlélet leképezése

Rendszer	Integrált információrendszer
Alrendszer	Modul
Részrendszer	Részmodul

A szoftver-, programcsomagok általában nem tartalmazzák a szervezet működéséhez szükséges valamennyi funkciót. Ennek következtében más - meglévő vagy új - alkalmazások bevonása szükséges. A programcsomagok egyik-másik modulja az adott feltételek között nem szabható testre kellő mértékben a helyi követelmények által támasztott igények kielégítéséhez. Elterjedt nemzetközi gyakorlat, hogy több, egymástól különböző integrált csomagot – vállalati információrendszert – , használnak, amelyekből felváltva használják az egyes modulokat.

A **vállalati architektúra-alapú integráció** (EAI, *Enterprise Architecture Integration*) az 1980-as években jelenik meg a gyakorlatban. Ekkor a fő cél a működő szoftverek átalakítása, tökéletesítő karbantartó, továbbfejlesztése valamilyen fokú integráltság irányába. Mégsem voltak képesek az igényeket kielégíteni, különösen abban a tekintetben nem, hogy azokat az információkat szolgáltatassák, amelyeknek valahol a vállalat adatbázisában meg kellett lenniük. A sok, különböző próbálkozás kudarcba fulladt, mert tévesen azt feltételezték, hogy az adatmodellben lehet a hiba. Valójában a korai alkalmazások még az átalakítások után is képtelenek voltak integráltan működni. Nagyon nehézkes volt az adatok áramoltatása. A 90-es években elméleti és technológia váltás következett be, amely két irányba terelte az *integrá-*

ciót: a *Data Warehouse* (**Adattárház**) és a **vállalatirányítási (ERP)** rendszerek felé. Az adattárház sok tekintetben megoldja az *adat és információintegrálást*, azonban nem segít a folyamatok integrációjában. A törzsadat-kezelés *adatintegráció* oldaláról segíti az *alkalmazás integrációt*, az *információ-architektúra* tekintetében.

Vállalati alkalmazások integrációja"A **vállalati alkalmazások integrációja** (*Enterprise Application Integration, röviden EAI*) a folyamatok, szoftverek, hardverek, szabványok olyan kombinációja, amely kettő vagy több vállalati rendszer gördülékeny együttműködését biztosítja. Az alkalmazás integráció oly módon takarít meg költséget és időt, hogy együttműködésre bírja a meglévő és az új rendszereket, csökkenti az üzleti reakcióidőt, lehetővé teszi a több rendszert is érintő tranzakciók feldolgozását.²⁸¹¹

Információ/adat integráció révén megoldandó szervezeti (vállalati, üzleti) problémák:

1. Naprakész információk nyújtása;
2. Humán eredetű szűk keresztmetszetek csökkentése;
3. Alkalmazottak termelékenységének növelése a feladatukhoz szükséges gyors és naprakész információ elérésének biztosításával;
4. Ügyfél elégedettség növelése;
5. Mobil eszközök bekapcsolása az üzletmenetbe.

Megoldások

1. Elkülönült, heterogén platformú rendszerek integrálása;
2. Intelligens információtovábbítás a heterogén rendszerek között;
3. Releváns információk publikálása;
4. Mobil eszközök biztonságos és gyors használata.

Alkalmazásintegráció révén megoldandó szervezeti (vállalati, üzleti) problémák:

1. Új alkalmazások fejlesztése, különösen a hálózati tranzakciós környezetben
2. Nyitott szabványokra épülő alkalmazás infrastruktúra kialakítása
3. A meglévő erőforrások felhasználása és kiterjesztése az új üzleti funkciókra is
4. Alkalmazás szolgáltatások publikálása a hálózaton az üzleti partnerek és a felhasználók integrálására és az információ könnyű elérésére

Megoldások

9.2 Alkalmazásintegráció

1. Új üzleti logika fejlesztése optimalizálva a létező erőforrásokat és információkat;
2. Új alkalmazások fejlesztése, melyek szabványokra épülnek (J2EE, JMS, XML, WSDL, SOAP – e technológiák mind szóba kerülnek mint a kivitelezés architektúra építőelemei Web szolgáltatás, REST illetve SzOA alapú szoftver architektúrák megvalósításánál.)
3. Alkalmazások fejlesztése üzleti tranzakciók végrehajtására (Web szolgáltatások, REST stb.)

Ügyfél integráció révén megoldandó szervezeti (vállalati, üzleti) problémák:

1. Ügyfél elégedettség növelése;
2. Naprakész információk elérése az ügyfelek, alkalmazottak, partnerek számára;
3. Mobil eszközök használata;
4. A vállalati információk a vállalati portálon történő megjelenítése.

A lehetséges megoldások milyen szolgáltatásokat nyújtanak:

1. Személyre szabott információk biztosítása az ügyfeleknek, partnereknek, alkalmazottaknak
2. Az üzleti tartalom implementálása a portál technológia alkalmazási felületein és szolgáltatásian keresztül valósul meg („portlet”), valamint a tartalomkezelésen keresztül (Content Management);
3. A tartalomhoz való hozzáférés kezelése jogosultságokon keresztül
4. Ügyfél interakciók biztosítása az üzleti rendszerekkel

Folyamatszintű integráció révén megoldandó szervezeti (vállalati, üzleti) problémák:

1. Az üzletmenet hatékonyságának és eredményességének a növelése;
2. A piaci versenyképesség növelése gyors és hatékony informatikai beruházásokkal;
3. Naprakész információkra épülő döntési rendszerek kialakítása;
4. A teljesítmény üzleti szintű mérése.

A lehetséges megoldások milyen szolgáltatásokat nyújtanak:

1. Az üzleti folyamatok mérhetőségét, rugalmasságát és könnyű bevezetést;
2. Eszközöket a gyors fejlesztéshez, telepítéshez, üzemeltetéshez és a rendszer optimalizálásához;

3. Lehetővé teszik, hogy a "MIT" független maradjon a "HOGYAN"-tól; azaz „deklaratív” módon, szervezeti (vállalati, üzleti) szabályok formájában fogalmazhatók meg, hogy mit kell elvégezni, megtenni; „procedurális” végrehajtás, az eljárások lépéseinek kivitelezése, az értelmező rendszer automatizmusaira támaszkodik.

9.2.2 Alkalmazás-integráció és szervezeti ellenállás

A szervezeti architektúra kontextusában informatikai projektekkel szemben támasztott legkomolyabb kihívás az alkalmazottak ellenállása. Az alkalmazottak ellenállásával a vezetés, szervezéstudományi szakirodalomban mint szervezeti ellenállással találkozhatunk. Ugyanis ezek az alkalmazottak fogják a későbbiekben használni az új vagy felújított rendszert, őket kell megtanítani a rendszer használatára és meggyőzni a változtatás szükségességéről.

Bárminemű informatikai újítás változással jár. Az alkalmazottak többsége a 'status quo' fenntartásában érdekelt, mert így látja biztosítottnak jövőjét, megélhetését. Ugyanakkor elképzelhető egy-egy hatékonyságnövelő projekt kapcsán, hogy alkalmazottakat kell elbocsátani a vállalattól.

9.2.3 Alkalmazás-integráció és technológiai architektúra

A szervezeti ellenállás az egyik legkomolyabb kihívás a leendő alkalmazásintegráció projektrel szemben azonban az, informatikai, műszaki és technológiai architektúra szinteken és rétegekben is jelentős akadályok állhatnak fenn:

- A meglévő alkalmazások általában eltérő koncepciók, algoritmusok szerint működnek, gyakran eleve úgy fejlesztették ki azokat, hogy nincsenek fölkészülve az együttműködésre (interoperabilitás) más alkalmazásokkal.
- Eltérő adat-, rekordformátumokat (szintaktikát) és adat illetve információ értelmezési tartalmat (szemantikát) használnak.
- Eltérő, heterogén operációsrendszer-környezetben működnek.
- Esetlegesen különböző hálózati protokollokat használnak.
- A több alkalmazás együttműködése során kialakuló osztott rendszer nem funkcionális tulajdonságai, architektúra sajátosságai komoly feladat előtt állnak egy minőségileg megfelelő szolgáltatási szint nyújtása tekintetében, nevezetesen a rendelkezésre állás, a tranzakciók biztonsága és a rendszerüzemeltetése követelményei értelmében.

9.2 Alkalmazásintegráció

9.2.4 Az alkalmazás-integráció tradicionális módszerei

A hagyományos, *ad-hoc*, alkalmi módszerek közös jellemzője, hogy eredetileg egyik módszert sem kifejezetten alkalmazások integrálására alakították ki, valamennyit más célra fejlesztették. Alkalmazás-integrációra való alkalmazásuk ezért egyedi, *ad-hoc* jellegű. Azonban ha sem általános alkalmazásintegráció technológia sem azok ismerete nem áll rendelkezésre, akkor hagyományos informatikai megközelítések, módszerek használata magától értetődőnek tűnhet.

9.2.4.1 Adatcsatolt technológiák

A gyakorlatban elterjedt, szokásos módszer. Az egyes alkalmazások adataikat egy vagy több közvetítő adaterőforráson (adatbázison, állomány- kiszolgálón, állományokon) keresztül cserélik ki.

Előnyei:

- Egyszerű, kézenfekvő, első pillantásra olcsó megoldásnak tűnik.
- Hagományos termékek használhatók hozzá (például állomány-kiszolgálók, Oracle, DB/2 adatbázis-kezelők stb.).
- Az alkalmazások kapcsolatát lazává, idő függetlenné teheti, ha nem tartunk igényt a tranzakciós biztonságra, azaz a kétfázisú *commit* protokoll használatára.
- Megfelelő adatbázis (elosztott adatbázis képességekkel rendelkező) esetében lehetőséget teremt a tranzakciós biztonság megteremtéséhez a kétfázisú *commit* protokoll révén.

Korlátai:

- Laza, idő független, aszinkron csatolás esetén az üzenetek garantált továbbítását megvalósító protokollokat külön meg kell tervezni és platformként megvalósítani.
- Ha az alkalmazott adatbázis támogatja a kétfázisú *commit* protokollt, a csatolt alkalmazások közti garantált üzenettovábbítás megoldható. Ez viszont az integrálandó rendszerek között igen merev, szoros, szinkron-csatolást eredményez. Ha a célalkalmazás nem elérhető, mert például éppen nem működik, a küldő alkalmazás is blokkolódik/ blokkolódhat.
- Általában hiányoznak az egységes API-felületek az alkalmazások integrálásához (megtervezendők, kifejlesztendők külön-külön minden platformra).

9.2.4.2 Állománycsere

Az egyes alkalmazások valamilyen állomány-átviteli protokoll segítségével cserélnek egymással adatokat együttműködésük érdekében.

Előnyei:

- Szabvány alapú megoldások (például FTP, ODETTE).
- A hálózati hibák kezelése egy ember-gép párbeszeden ("session") belül megoldott.
- A legtöbb rendelkezésre álló hálózati program, operációs rendszer támogat ilyen eljárást.

Korlátai:

- A hálózati ember-gép párbeszéd ("session") valamilyen hiba következtében történő megszakadására nem nyújt támogatást, ezt az alkalmazásoknak kell megoldaniuk.
- A rendszerkiesések kezeléséhez, a tranzakciós biztonsághoz a protokollok hiányoznak (ezeket meg kell tervezni és minden platformra külön-külön kifejleszteni).
- Az integrálandó alkalmazásoknak közös ember-gép párbeszédben ("session") kell részt venniük, az időfüggelenség és a laza csatolás köztük nem valósítható meg.
- Hiányoznak az egységes API-k az alkalmazások integrálásához.
- Menedzselhetőségi, biztonsági problémák állnak fel, amelyek kezelése megint külön erőfeszítéseket igényel, és nehezen szabványosítható.

9.2.4.3 EDI

Egyes ipari ágazatokban, gépjárműiparban és a kiskereskedelmi láncoknál alkalmazott adatcsere megoldás, elsősorban a vállalatok, szervezetek alkalmazásai közti együttműködést támogatja. A B2B-technológiák egyik első adatcsere rendszerének és szabványának tekinthető. Alapvető célja elektronikus okiratok, okmányok, polgárjogilag hiteles dokumentumok cseréje különböző, jogi személyiségű szervezetek között.

Előnyei:

- Szabvány alapú megoldások (UN/ EDIFACT, X12).
- A gyakorlatban bizonyított, létfontosságú vállalati alkalmazásokat kapcsol össze adat és dokumentum szinten.

Korlátai:

- Az EDI szabványok dokumentumformátum-orientáltak, a protokollok a szabványokból hiányoznak (ezeket az **EDI-szolgáltató** adja, általában egyedi megoldással).

9.2 Alkalmazásintegráció

- Függés a közbülső szolgáltatótól (*EDI-szolgáltató*).
- Kötegelt és nem valós idejű működésű, hosszú válaszidőkkel (legalább 15 perc, általában több óra).
- Korszerű megoldás a Web EDI, a szabvány továbbfejlesztése, a PKI-ra, elektronikus aláírásra alapozva.

9.2.5 Alkalmazásintegrációs köztesszoftver-technológiákkal (*Middle-ware*)

Az alkalmazásintegráció problémájának növekvő jelentősége és az addig alkalmazott hagyományosnak tekinthető módszereknek a korlátai hívták életre az alkalmazásintegráció köztesszoftver-technológiákat (**Middle-ware**) mint egy olyan általános műszaki, technológia szintű architektúra megközelítést, amely a szervezeti architektúra integráció és a vállalati alkalmazások integrációjának (EAI) problémájára ad egy megoldást.

Köztesszoftver (*Middle-ware*)

A **köztesszoftver**-technológiák (**Middle-ware**) általános fogalmát az alábbiak szerint határozhatjuk meg:

- **köztesszoftveren** azokat a technológiákat magában foglaló informatikai kategóriát értjük, amelyek osztott adatfeldolgozási-környezetben képesek valós időben **integrálni** (összehangolni, összekapcsolni, összeilleszteni) az egymástól eltérő *alkalmazásokat, adat-erőforrásokat* és szervezeti (vállalati, üzleti) *folyamatokat*, függetlenül azok műszaki, technológiai architektúra megvalósításától, azaz operációs rendszertől, hálózati protokolloktól, egyéb környezeti tényezőktől és helyétől.

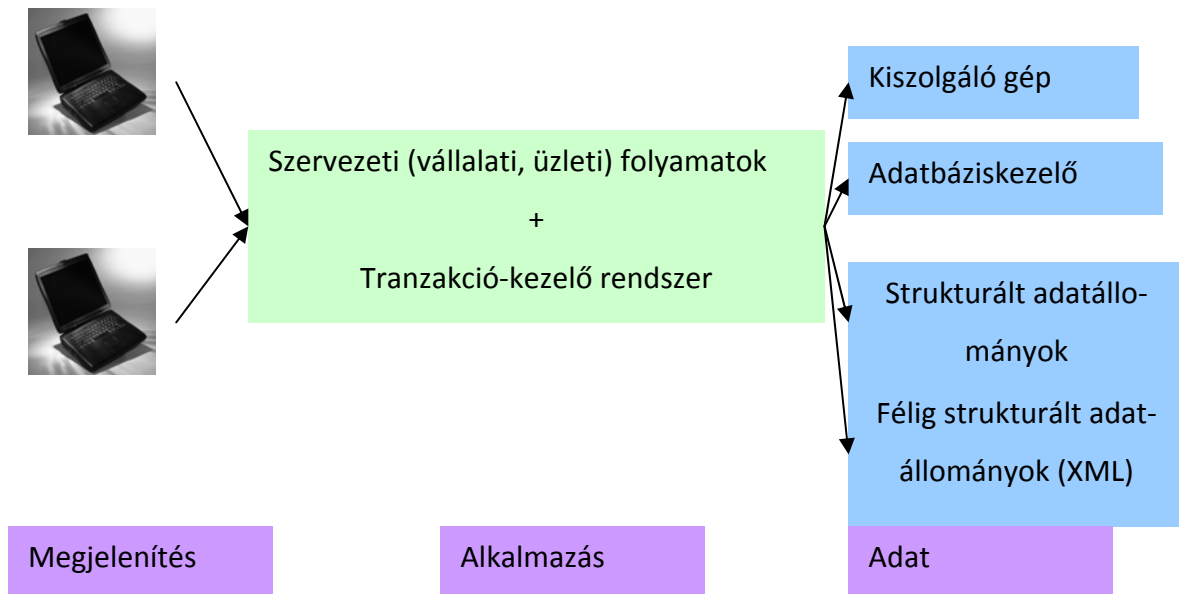
Tekintsük át röviden a *köztesszoftverek* osztályait, az alkalmazásintegráció köztesszoftverek osztályát, majd vizsgáljuk meg az elterjedtebb technológiák előnyeit és korlátjait.

9.2.5.1 Köztesszoftver osztályok

A köztesszoftvereket négy fő osztályba sorolják.

- Az adatvezérlő köztesszoftverek távoli adaterőforrások (adatbázisok, állományok) használatát teszik lehetővé programok számára. Tipikus példái a távoli állománykezelők („*file server*”), például a Network File System (NFS) vagy a távoli adatbázisok elérésére szolgáló köztesszoftverek: Open Database Connectivity (ODBC), Java Database Connectivity (JDBC) stb.

- A *platform* jellegű *köztesszoftverek* a szervezeti (vállalati, üzleti) folyamatok integrálását valósítják meg az osztott feldolgozási környezetben, a *három- vagy többretegű* architektúrák kialakításának támogatása révén. Tipikus példái az olyan elosztott és objektum-orientált tranzakció kezelő rendszer, mint például a CICS, az Encina, a Tuxedo vagy a Top End (76. ábra). Ezek közé tartoznak a webes alkalmazás- kiszolgálók (IBM WebSphere, Oracle Application Server stb.) és a portálkiszolgálók.



75. ábra A platform jellegű közteszftver architektúra

- Az alkalmazásintegráció közteszftverek különböző alkalmazások valós idejű integrálását szolgálják. Főbb típusai az objektum-orientált technológiák (CORBA, MDA, JMS, JCA, COM, COM+, DCOM) és a kommunikációs közteszftverek - az üzenetcsatolt közteszftverek és az RPC (*Remote Procedure Call* - távoli eljáráshívás) alapú szftverek (mint az integráció brókerek és a BPM (*Business Process Management*, - szervezeti (vállalati, üzleti) folyamatmenedzsment)).
- A segéd-közteszftverek az előző osztályok működését támogatják. Tipikus példái az átjátszó közteszftverek (például: CICS/Tuxedo gateway).



76. ábra Köztesszoftver és elosztott rendszerek architektúrája

9.2.5.2 A korszerű köztesszoftver technológiák osztályzása

Köztesszoftver

A **köztesszoftver** egy olyan program amely az alkalmazás és az operációs rendszer között helyezkedik el, az architektúra közepén (middle). A különböző alkalmazási területeken más és más technológiákat definiálnak köztesszoftverként.

Szervezeti (vállalati, üzleti) folyamat vezénylő/vezérlő Business Process Orchestrator	BizTalk, TIBCO, Staffware, Active BPEL
Üzenetközvetítő ügynök Message Broker	Mule, WebSphere Message Broker, Son-icMQ
Alkalmazás kiszolgáló Application Server	JEE, CCM, .NET
Szállítási réteg Transport	Message-Oriented Middleware, Distributed Object Systems, SOAP

77. ábra A köztesszoftverek (Middleware) egyik fajta osztályzása

- A **szállítási réteg (Transport)** réteg felel azért, hogy az egyes elosztott szoftver komponensek között az adatcsere biztonságosan módon történjen.

- Az **alkalmazási kiszolgáló (Application Server)** réteg általában közvetlen a *szállítási réteg (Transport)* rétegre épül rá. Ebben a rétegben találhatóak meg tranzakciós és biztonsági szolgáltatások, illetve az egyes erőforrás hozzáférésekhez kapcsolódó funkciók is. E réteg általában támogatja a több szálú (*thread*) alkalmazások létrehozását a szerver teljesítményének növelése érdekében.
- **Üzenetközvetítő ügynök (Message Broker)** vagy az *alkalmazás kiszolgáló (Application Server)* réteg vagy a *szállítási réteg (Transport)* réteg funkcióit felhasználva nyújt **üzenet feldolgozó szolgáltatást**. A réteg fő funkciói a gyors üzenet transzformáció, illetve a magas szintű programozási eszközökről gondoskodik az üzenetek manipulálására és a különböző szoftver komponensek felé történő továbbítására.
- **Szervezeti (vállalati, üzleti) folyamat vezénylő/vezérlő (Business Process Orchestrator)** réteg segítségével az *üzenetközvetítő ügynök (Message Broker)* számára készíthetünk automatizált munkafolyamatokat (*workflow*). Amennyiben az egyes folyamatok több időt vesznek igénybe és a programnak a további futásához be kell várnia a szükséges információkat, úgy ezekkel az eszközökkel lehet szabályozni, hogy a munkafolyamat rögtön folytatni tudja az adatfeldolgozási folyamatot, amennyiben a szükséges lépésekkel végeztek a megelőző feldolgozó elemek.

9.2.5.3 Alkalmazásintegráció köztesszoftverekkel

Az *alkalmazásintegráció köztesszoftverek* alkalmazásszintű – közvetlen –, **algoritmikus** együttműködést tesznek lehetővé több meglévő vagy újonnan kifejlesztett rendszer között. Ha az algoritmikus szintű kapcsolat kiépítése egy adott meglévő alkalmazás esetében nem lehetséges, mert például a programfelületek nem hozzáférhetők, akkor az alkalmazásintegrációs köztesszoftverek általában támogatják az *adatcsatolt integráció* biztonságos megvalósítását is, mint például az állományok cseréje.

Alkalmazásintegráció funkciók más köztesszoftver-osztályoknál is vannak. A platform jellegű köztesszoftverek is rendelkeznek olyan funkciókkal, amelyek háttérrendszerekkel való integrációt támogatják. Az elosztott tranzakció kezelő rendszerek tipikusan valamilyen alkalmazásintegráció kommunikációs szoftvert használnak. Például Tuxedo esetében a TuxedoQ-t, Encina esetében DCE/RPC-t, J2EE szabványos webes alkalmazás-kiszolgálók pedig a JCA (Java Connector Architecture) szerinti adaptereket. Mindegyiknél igaz azonban az, hogy nem álta-

lános, tetszőleges alkalmazások közti integrációt céloznak, hanem az adott platform jellegű közteszoftver által kezelt szervezeti (vállalati, üzleti) folyamatokat képviselő alkalmazások és a háttérrendszerek közti kapcsolatot. A megvalósításhoz a platform jellegű közteszoftverek valamilyen alkalmazásintegráció közteszoftverre támaszkodnak.

Alkalmazások integrálásához *adatvezérelt közteszoftvereket* is használnak. Ezek azonban nem képesek olyan színvonalú megoldást nyújtani alkalmazások integrálásához, mint az alkalmazásintegráció közteszoftverek. Nem alkalmasak például garantált megbízhatóságú, laza, aszinkron kapcsolatok kiépítésére. Ez persze nem von le semmit az adatvezérelt közteszoftverek értékéből, csupán azt kell tudnunk, hogy nem alkalmazások integrálása a rendeltetésük, hanem adaterőforrások távoli elérése. Ez utóbbi alkalmazásintegráció közteszoftverek segítségével is megoldható, ha közvetlenül nem a távoli adatbázis- vagy állományrendszereket akarjuk elérni, hanem az azokat kezelő alkalmazásokat.

Ha egy elosztott adatfeldolgozó-rendszer egészét tekintjük, akkor az architektúra szempontjából az alkalmazásintegráció közteszoftverek önálló réteget képviselnek az alkalmazások és a hálózati protokollok között (76. ábra). A modell az elosztott adatfeldolgozó-rendszerek ötrétegű architektúráját mutatja.

1. A fizikai réteg az olyan fizikai erőforrásokat képviseli, mint amilyen például a memória, a merevlemez vagy a processzor.
2. A fizikai erőforrásokkal az operációs rendszer gazdálkodik, és lehetővé teszi a fölötte lévő rétegek számára, hogy azok már logikai erőforrásokat lássanak és használjanak, alapesetben egyetlen gépen belül.
3. A hálózati réteg a különböző távoli logikai erőforrások elérhetőségét biztosítja a fölötte levőrétegek számára, kiterjesztve azokat többgépes környezetre.
4. Az alkalmazásintegrációs közteszoftverek rétege egy olyan egységes környezetet alkot a felette lévő alkalmazások számára, amelyben az egyes alkalmazások eltérő tulajdonságai (platform, hálózati, algoritmikus és adatkülönbségek) eltűnnek, így azok képesekké válnak az együttműködésre.
5. A legfelső az alkalmazások rétege, az információrendszer szolgáltatások szintje.

Az alkalmazásintegráció közteszoftverek egységes infrastruktúra-réteget alkotnak az osztott feldolgozó rendszereken belül, és ez a réteg a meglévő alkalmazások integrálásán túl kellő

rugalmasságot nyújt minden további, esetleg előre nem látható módosítás, fejlesztés hatékony kivitelezéséhez is.

9.2.6 Az alkalmazásintegráció köztesszoftverek típusai

Az alábbiakban az egyes alkalmazásintegrációs köztesszoftverek technológiáit tekintjük át.

9.2.6.1 Objektumorientált technológiák

9.2.6.1.1 CORBA

Az Object Management Group (OMG²⁹) által kidolgozott Common Object Request Broker Architecture (CORBA) modell képviselte ez ideig a legáltalánosabb, konzisztens objektumorientált, alkalmazás integráció megközelítést. A CORBA modell az elosztott adatfeldolgozási környezetre terjeszti ki az objektumok világát. Ebben a világban egy hivatkozott objektum az elosztott térben bárhol lehet, helyét a modell elfedi, transzparenssé teszi az alkalmazások számára. A tényleges objektumok és az általuk képviselt alkalmazások egymáshoz rendelését együttműködő brókerek, ORB-k (Object Request Brokerek) valósítják meg.

A szabványos, egységes modell előnyei a konzisztens, általános objektummodell, a helytől való függetlenség megteremtése és a tranzakciós biztonság.

A CORBA modell a gyakorlatban nem terjedt el szélesebb körben. A kidolgozó OMG 2001 márciusában újabb, átfogó modell kidolgozását kezdeményezte helyette: az **MDA-t**, a **modellvezérelt architektúrát** (*Modell Driven Architecture*). Ez az architektúra tervezési és leírási megközelítés egy keretrendszer. A CORBA mellett más elemeket is beemel a modellbe, mint például az *EJB*, *XML* vagy a *SOAP*, amely az XML üzenetek továbbítását rögzíti.

Az elosztott objektumok az egyik alapvető tagja köztesszoftver technológiáknak. A CORBA (OMG) már a korai 1990-es években specifikálta ezt az architektúra megközelítést.

Adott egy szerver számára előírt interfész (program és adatkapcsoló felület valamilyen programozási nyelvben, környezetben). Az ügyfél (kliens) ezen keresztül megszólítja kliens oldali ORB (**Object Request Broker**) komponenst. A kliens oldali ORB kapcsolatba lép szerver ORB-jével, ahol az interfész megvalósítása található. A választ aztán ismét ORB-hez továbbítják, majd az ORB továbbküldik egymás között addig, amíg a kliens a válasz birtokába jut.()

A CORBA kapcsolófelület (interface) kódból **IDL (Interface Description Language)** fordító egy forrás kód vázat (skeleton) hoz létre, amit a programozónak csak ki kell töltenie. Az ORB nem követeli meg, hogy kliens és szerver oldalon futó alkalmazás azonos nyelven kerüljön megvalósításra.

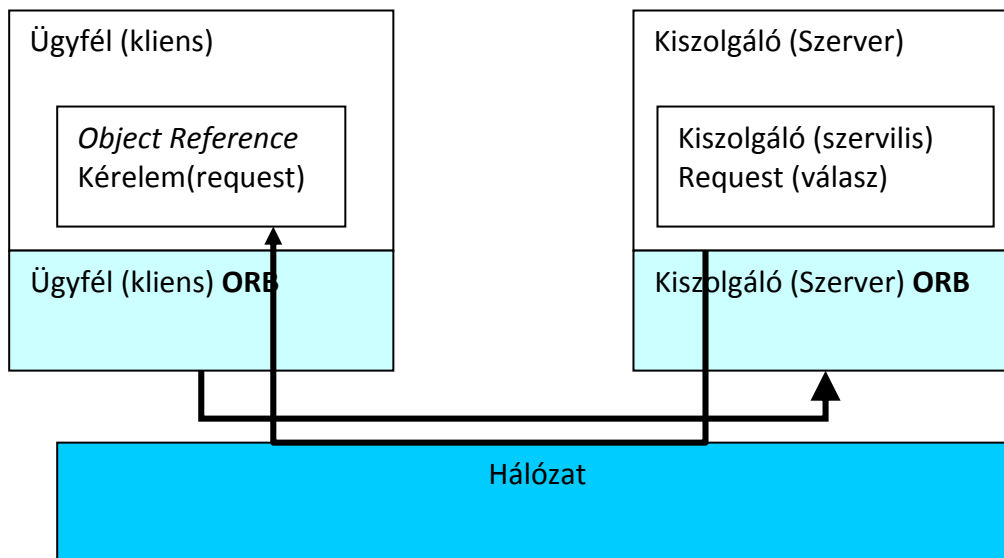
9.2 Alkalmazásintegráció

Milyen hátrányai vannak ennek a megoldásnak.

- A távoli eljárás hívások költségesek, minden esetben átfutnak az ORB-n és a hálózaton is, ami lassúvá teszi őket. Ahhoz, hogy ezt a hátrány kiküszöböljük általában csak nagy komplex hívásokat intézünk szerver felé, így arányosan csökkentve a kommunikációs költségeket.
- Lehetséges, hogy megszűnik a kapcsolat a két gép között, ezért fel kell készítenünk a programunkat, ezeknek az eseteknek kezelésére.
- Amennyiben a kliens meghibásodik, elveszik a kommunikáció pillanatnyi állapota és az összes adat, erre különös gonddal fel kell készülni mivel, a szervezeti (vállalati, üzleti) létfontosságú rendszerek estén ez megengedhetetlen.

9.2.6.1.2 COM, COM+, DCOM

A Microsoft COM, COM+, DCOM megközelítéseinek alapvető motivációja az volt, hogy a CORBA és objektum bróker típusú integráció köztesszoftver és ezzel együtt a hálózati rétegek funkciói az operációs rendszer beépített részévé váljanak. Elsőként a COM modell született meg, amely még a *kétrétegű ügyfél-kiszolgáló* architektúrát követte. A COM+ bővített modell már a *háromrétegű ügyfél-kiszolgáló* környezetet célozza, és a Windows 2000 operációs rendszer kiterjesztésének is tekinthető. Végül a DCOM (Distributed COM - Osztott COM) szintén a COM modell bővítésének tekinthető, az OSF (Open Software Foundation) DCE (Distributed Computing Environment) ajánlását követő MS-RPC szinkron távoli eljárás hívására épülő osztott köztesszoftver-infrastruktúrát célozva.



78. ábra Elosztott objektumok CORBA architektúra alkalmazásával

Előnye, hogy a köztes szoftvereket az operációs rendszer részévé teszi, és a meglévő operációs rendszerhez kapcsolódó eszközökkel készen használható integrációs módszert kínál.

Legnagyobb korlátja, hogy a gyakorlatban csak a Windows-környezetben használható, heterogén rendszerekben nem. Ráadásul az egyes modellek a Windows szintjén sem általánosak. A DCOM modell szoros, szinkron kapcsolatot tételez fel az alkalmazások között.

9.2.6.1.3 J2EE

A **Java2 Enterprise Edition** csomag több olyan szabványt is tartalmaz, amelyek az alkalmazásintegrációt célozzák meg. A JMS (*Java Message Service*) egy API felület szabványa. A JMS az üzenettovábbító (üzenetcsatolt) köztesszoftver-technológia *objektum-orientált* megközelítése. A szabvány azonban kizárólag az API felületre vonatkozik, ezért a JMS nem független a ténylegesen megvalósított üzenetcsatolt köztesszoftver-technológiától.

A JCA (*Java Connector Architecture*) Java programok meglévő rendszerekhez való kapcsolódását támogató *adapterek* szabványa. Tipikus alkalmazási területe a platform jellegű köztesszoftverek és a háttérrendszerek közti kapcsolat kiépítése.

9.2 Alkalmazásintegráció

9.2.6.2 Kommunikációs köztesszoftverek

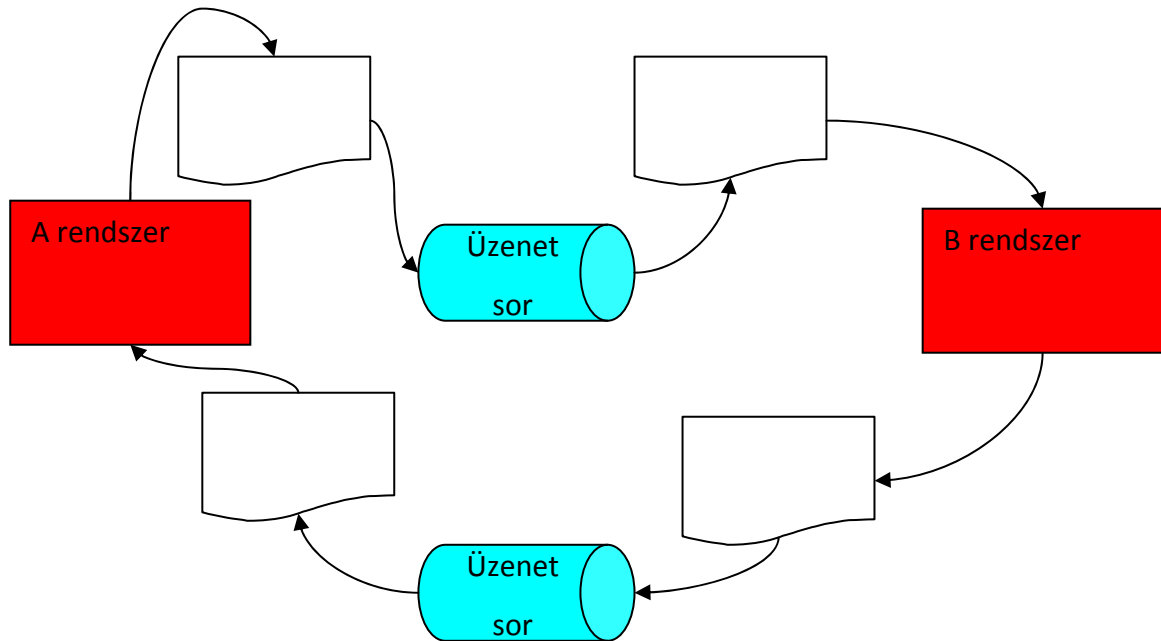
A kommunikációs köztesszoftverek között két jelentős megközelítés alakult ki. Az egyik a *szinkron*, távoli eljáráshívásra alapozott RPC architektúra (*Remote Procedure Call*), a másik pedig az *üzenettovábbító (üzenetcsatolt)* köztesszoftvereké.

9.2.6.2.1 RPC

Az RPC előnye, hogy hűen modellezi az egygépes környezetben használt eljáráshívást, annak egyfajta kiterjesztésének tekinthető. A szoros, szinkroncsatolást megvalósító megoldások alapeleme. Az OSF (Open Software Foundation) DCE (*Distributed Computing Environment*) modellje, a legtöbb platform-köztesszoftver, a CORBA és a DCOM modell használja. Alkalmazások integrálására a gyakorlati jelentősége azonban egyre inkább elhalványul a szoros csatolás hátránya miatt. Ez utóbbi ugyanis az együttműködő rendszerek azonos rendelkezésre állását igényli. Ez esetben számolni kell ugyanis a holtponthelyzetek vagy a csökkent teljesítményű állapotok lehetőségével is. Ezeknek az állapotoknak, kivételes helyzeteknek, rendkívüli eseményeknek a lekezelésére egyedi megoldásokat kell kifejleszteni, leprogramozni, ami azt jelenti, hogy a megoldás hordozhatósága romlik, a szabványos megközelítésektől ezek a megoldások távolodnak a konkrét esetekben.

9.2.6.2.2 Üzenettovábbító köztesszoftverek

Napjaink meghatározó kommunikációs technológiáját a *laza, aszinkron* csatolást megvalósító **üzenettovábbító köztesszoftverek** képviselik. Az üzenettovábbító köztesszoftverek olyan általános alapot jelentenek, amelyre az integráció bróker és a BPM (*Business Process Management*) technológiák is épülnek. Az alapkoncepció az operációs rendszerek gyakorlatában a folyamatok közti kommunikációra (IPC - Inter Process Cooperation) már bevált üzenettovábbítási modell általánosításának tekinthető.



79. ábra Üzenet továbbító rendszer modell

A modell lényege, hogy az egyes alkalmazások egymással üzenetek segítségével kommunikálnak. Egy üzenetet az adott közteszoftver *várakozási sorba* („*queue*”) helyez, és akkor közvetíti ki a célalkalmazásnak, amikor ez utóbbi lekéri azt, illetve kész az adott alkalmazásnak, szolgáltatásnak címzett üzenetek fogadására. Mind a küldő, mind a célalkalmazás szabványos API felületen keresztül kapcsolódik a közteszoftverhez. A módszer előnye, hogy az *együttműködő alkalmazásokat* térben és időben *függetlenné* teszi egymástól. Ez rendszer filozófiai megközelítés a **laza (adat)csatolás** tervezési elvének követését jelenti („*loose coupling*”). Az *aszinkron* működési mód révén a valós idejű kapcsolat az alkalmazási rendszerek között – az informatikai tudomány jelenlegi állása szerint – a lehető legrugalmasabb megoldás, alkalmazkodóképessége, ellenálló („*resilience*”) és hibatűrő képessége magas fokú. Az információ-biztonsági architektúra tervezés szempontjából is kedvező sajátosságai vannak.

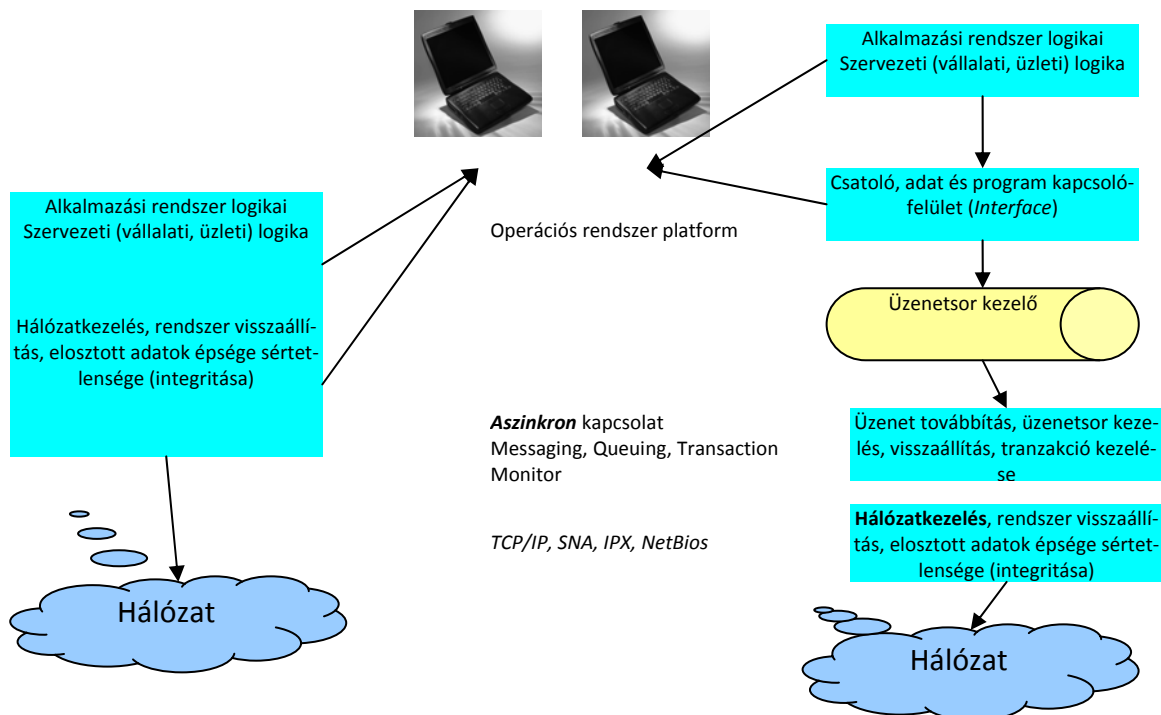
Az üzenettovábbító közteszoftvereknek számtalan előnyük van:

- *alkalmazásszintű integrációt* tesz lehetővé, de lehetőség van az *adatcsatolt* megoldásokra is;
- az együttműködő alkalmazások közti, térben és időben független laza kapcsolat (moduláris tervezés elve: „laza csatolás, erős kohézió!”);

9.2 Alkalmazásintegráció

- garantált egyszeri és csakis egyszeri üzenettovábbítás időszakos rendszerkiesések esetén is;
- platformfüggetlenség - azonos API felület minden platformon;
- függetlenség a hálózati protokolltól;
- a tranzakciós feldolgozás támogatása („*transaction intensive system*”);
- valós idejű csatolás („*real-time*”);
- meglévő, tetszőleges alkalmazások integrálhatóak, ha a csatolófelületeiket (program/adat) szabványossá alakítják, szabványossá fejlesztik tovább.

Az *üzenettovábbító közteszoftverek* (IBM MQSeries, Oracle Advanced Queueing, Progress SonicMQ) az ábrán (80.) látható módon segítik a különböző alkalmazások pont-pont kapcsolódásának megvalósítását.



80. ábra Az üzenettovábbító/üzenetsatolt közteszoftverek szerepe az alkalmazások integrálásában

Az ábra bal oldalán (80.) egy hagyományos operációsrendszer-környezetben működő alkalmazás főbb funkciói láthatók. Üzenetsatolt közteszoftver nélkül, a tényleges szervezeti (vállalati, üzleti) *értéket* jelentő alkalmazási logika mellett a programoknak foglalkozniuk kell a hálózati protokollok vezérlésével, az esetleges hiba utáni visszaállítással és az adatok integritásának biztosításával is. Az ábra jobb oldalán látható, hogy az üzenetsatolt közteszoftver

ezeket a funkciókat, továbbá az üzenetek kezelését is megvalósítja, és ehhez egységes API felületet nyújt az alkalmazások számára.

API-ként több kapcsoló felület is választható. Az alap köztesszoftver API felület mellett gyakori a magasabb absztrakciós szintet jelentő felületek alkalmazása is, így például az OAG által ajánlott AMI (*Application Messaging Interface*), és a J2EE szabvány szerinti JMS egyre szélesebb körű alkalmazása is terjed. A gyakorlati megvalósításban többféle irányzat érvényesül. Az *MQSeries* például a hagyományos *MQI (Message Queue Interface)* és AMI felületei mellett a JMS-t is választhatóvá teszi az adapter fejlesztők részére, míg például a Progress SonicMQ köztesszoftvere eleve célzottan a JMS használatát teszi kötelezővé.

Az üzenetközpontú köztesszoftver (**MOM, Message-oriented Middleware**) a kulcs technológiája annak, hogy nagyméretű üzleti alkalmazásokat tudjunk létre hozni. Ez az csatoló, integráció architektúra elem, amellyel az egyébként független és önállóan működő rendszereket egy egységes integrált programmá lehet alakítani. A részrendszereknek nem szükséges azonos nyelven íródniuk és a futásukhoz használt platformok is különbözhetnek. Ahhoz, hogy ez technológiát használhassuk még az alkalmazások forrás szövegéhez sem kell hozzányúlni, ami különösen előnyös ipari körülmények között. A MOM alapja – szoftver architektúra értelemben – egy üzenettovábbító **cső (pipe)**, amelyet a küldő és fogadó rendszer közé helyeznek el. A szoftver **csővezeték** feladata, hogy az adatokat közvetlenül fogadó rendszerbe juttassa. (79. ábra).

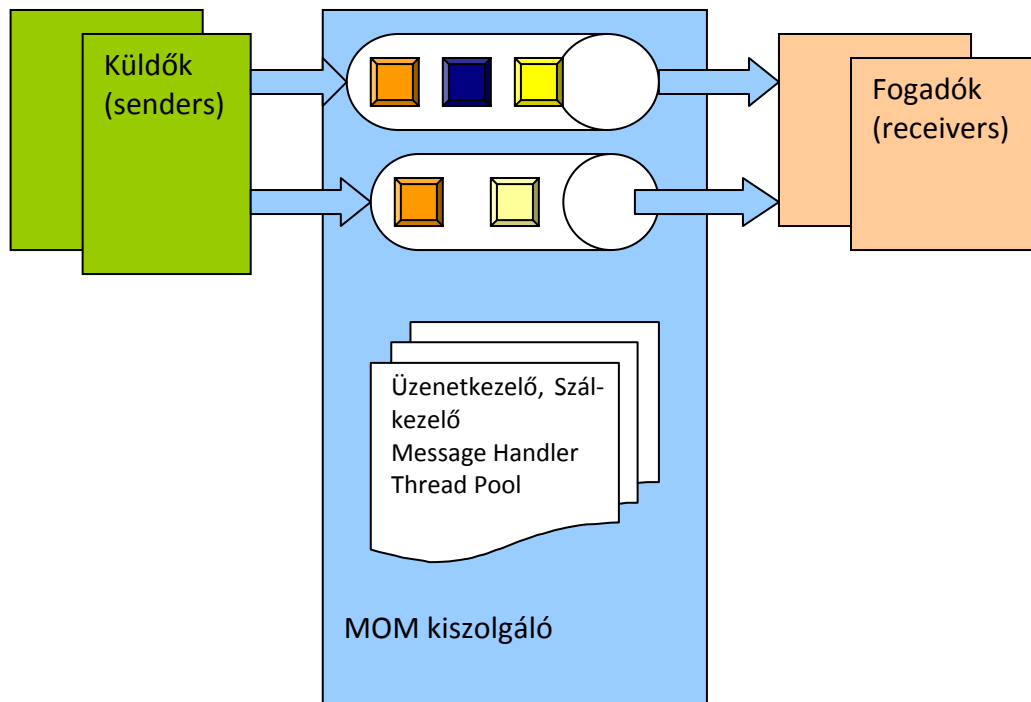
A MOM eredendően egy *lazán kapcsolt aszinkron technológia*. Ez azt jelenti, hogy a küldő és fogadó nincs szorosan összekapcsolva, mint azt a CORBA technológia esetében látható. Az üzenet váltási infrastruktúra szétválasztja a *küldőt* és a *fogadót* egy köztes *üzenetsor* segítségével. A küldő ezért biztos lehet abban, hogy a fogadó előbb vagy utóbb akkor is megkapja az üzenetet, amennyiben éppen aktuálisan elérhetetlen. A küldő csak átadja az üzenetet a MOM számára és folytatja a munkáját. A küldőnek nincs ráhatása arra, hogy végül melyik alkalmazás melyik adatfeldolgozási folyamata dolgozza fel a kérést.

MOM architektúrát általában úgy valósítják meg, hogy egyszerre több konkurens klientsől is tud fogadni üzeneteket. A MOM szerver egyszerre több üzenet kezelését is el tudja látni mivel, egy *szál kezelő szolgáltatás (thread pool)* segítségével szét tudja osztani a fogadó szálaknak az üzeneteket.

A MOM szerver alapvető feladatai a következők. Először az általa fogatott üzenetekről egy

9.2 Alkalmazásintegráció

tértivevényt kell adnia a küldő számára. Ezt követően a megfelelő üzenetküldési sor végére kell illesztenie az új üzenetet. Mivel lehetséges, hogy a küldők gyorsabb ütemben küldik a az üzeneteket, mint ahogy a szerver feldolgozni képes, ezért ezeknek üzeneteknek a tárolására fel kell készíteni a MOM szervert.



81. ábra Az üzenetközpontú köztesszoftver kiszolgáló architektúra- logikai architektúra komponens

Az aszinkron üzenetküldés és laza kapcsolat nagyon hasznossá teszi ezt a szoftver eszközt. Segítségével több általános program tervezési problémát meg lehet oldani. Néhány példa:

- Amennyiben a küldőnek nem szükséges választ kapnia az üzenetre. Ebben az esetben nem kell szükségszerűen várakozni egy fölösleges válaszra. Ez a típus *send-and-forget*.
- Amennyiben a küldőnek *nem szükséges azonnali választ* kapnia az üzenetre. Amennyiben az üzenet feldolgozása több percet vesz igény és a kliensnek van jobb dolga is.
- Amennyiben fogadó vagy hálózati kapcsolat csak bizonyos időközönként nem érhető el. A MOM szerver biztosítja, hogy az üzenetek a probléma megoldását követően kézbesítve legyenek a fogadó számára.

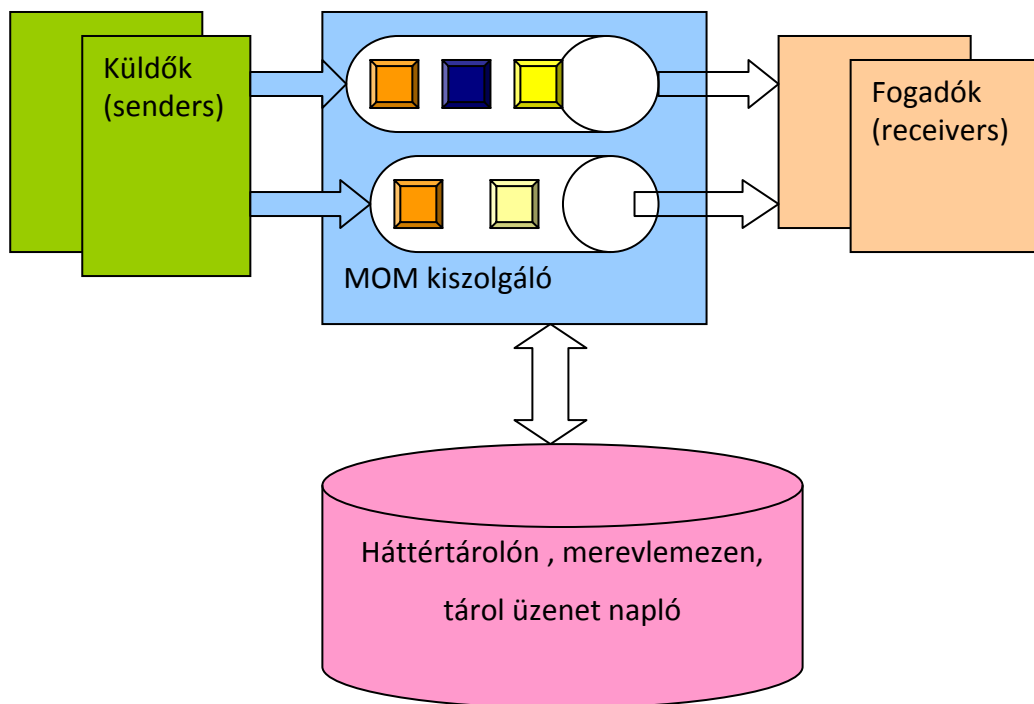
9.2.6.3 A MOM kiaknázása – Speciális funkciók

Az MOM szerver egyszerű funkciói ritkán elegendőek a nagyvállalati rendszerek működte-

téséhez. Létfontosságú vállalati rendszerek esetén szükséges, az üzenetek fogadásának erősebb garanciája, illetve bizonyos teljesítmény problémákon átlépjen a MOM szerver. COTS (Commercial Off-The-Shelf) MOM implementációk sokkal több funkcióval rendelkeznek a *megbízhatóság, felhasználhatóság* és skálázhatóság területén, a korábban leírtakhoz képest.

9.2.6.4 Üzenet továbbítás garanciával

Sok üzleti alkalmazásnál szükséges, hogy az adott üzenet biztosan feldolgozzák. Vegyünk példának egy bankkártyás tranzakciót. A tranzakció adatai bekerülnek az *üzenetküldési sorba* azért, hogy a számlán el tudják végezni a tranzakció könyvelését. Amennyiben a MOM szerver összeomlik, a tranzakció elveszik, kártya tulajdonos örülhet mivel, nem vonják le a pénzt számlájáról, viszont a kártyatársaságot veszteség éri. Mint a példából is látszik az ilyen folyamatok esetén nem fogadható az üzenetek elvesztése, ezért garanciát kell adni arra, hogy ez nem fog megtörténni.



82. ábra Üzenetközpontú köztesszoftver esetében az üzenet továbbítás garanciája

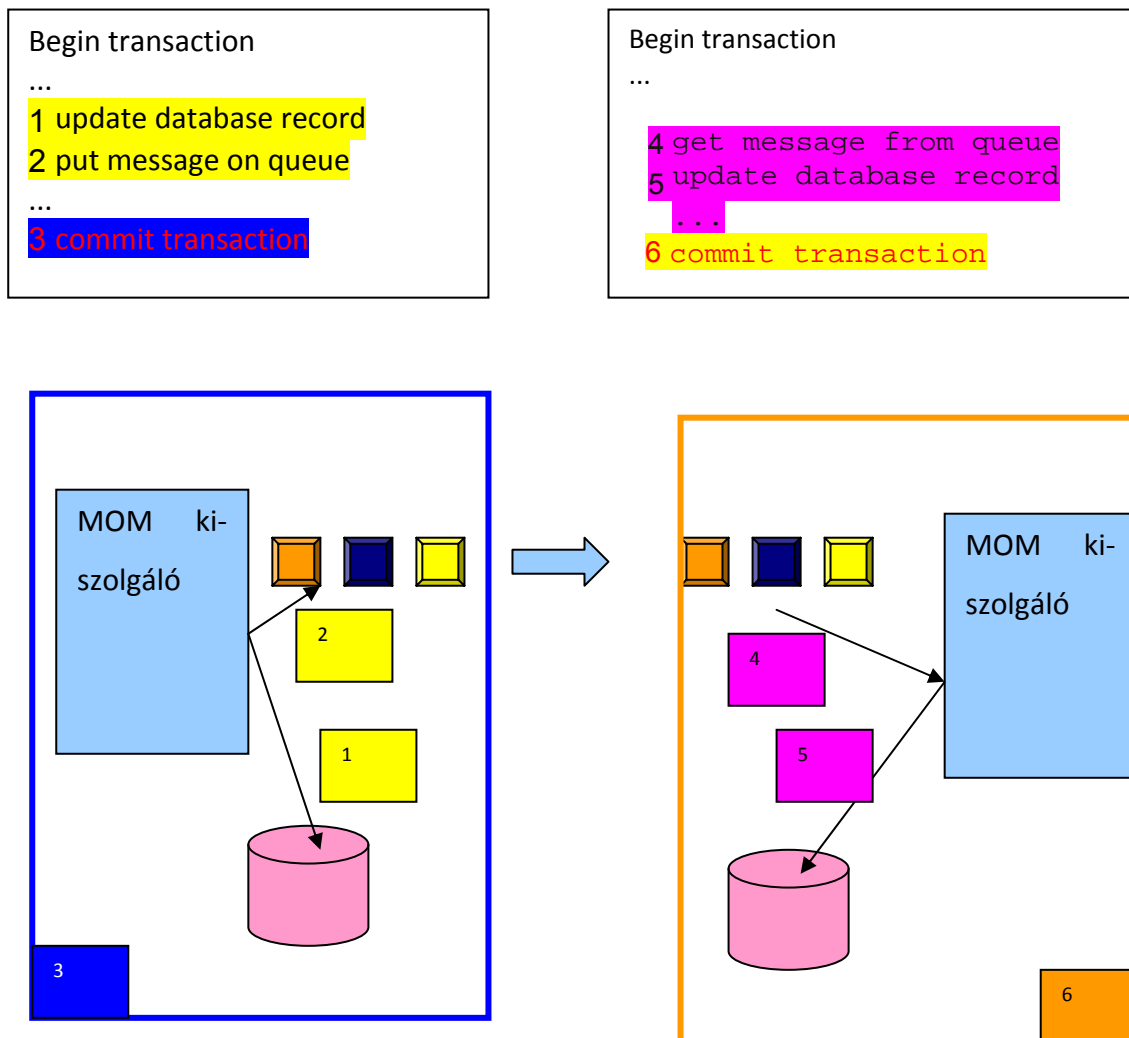
Általában három szolgáltatási minőség van, amit beállíthatnak egy MOM szerver esetében, mindegyik növeli biztonságot, de lassítja a szerver működését.

- **Legjobb akarat (Best effort):** A MOM szerver minden tőle telhetőt megtesz az üzenetek továbbítása érdekében. A továbbításra váró üzenetek a memóriában tá-

9.2 Alkalmazásintegráció

rolja és amennyiben az adott feldolgozó szolgáltatás elérhetetlen egy idő után eldobja az üzenetet.

- **Tartós tárolás (Persistent):** A MOM server garantálja az üzenetek továbbítását, a várakozó üzenetek a tartós háttértárra, merevlemezre kerülnek egy esetleges rendszerhiba esetére felkészülve. Az üzenetek számának csak a háttértároló mérete szab határt. Ld. ábra
- **Tranzakció kezelés (Transactional):** Az üzenetek beburkolhatóak egy „mindent vagy semmit” borítékba. Ezeket az üzeneteket a server egyenként veszi át, de csak egyben továbbítja. Általában a tartós objektum, adattárolás elvei alapján történő továbbfejlesztése az üzenetkezelésnek.



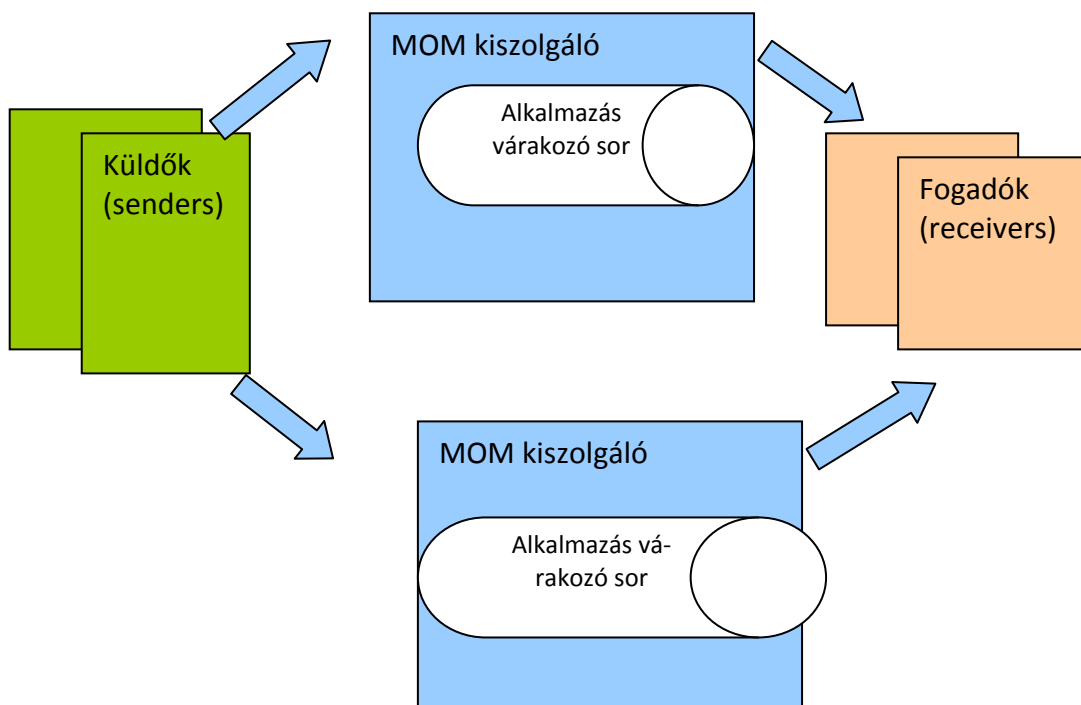
83. ábra Üzenetközpontú közteszofrtver esetében tranzakció jellegű feldolgozás

9.2.6.5 Tranzakció kezelés

Tranzakció kezelő jellegű MOM szerver esetén a szerver az üzenetkötegeket egyben kezeli és csak akkor kezdi el kiosztani, amikor az adott köteg teljes. A folyamatot lásd az ábrán. A kliens új tranzakciót kezdeményez. Ekkor az üzenetek először háttértárolóra és várakozó sorba kerülnek beírásra, ezek az üzenetek nem kerülnek feldolgozásra amíg a *commit* el nem hangzik. Amennyiben a küldő kiadja a *commit* utasítást a fogadó is hasonlóan kötegelve átveszi az üzeneteket. Amennyiben feladó oldalán szakad meg a kapcsolat tranzakció közben, a szerver eldobja az addigi üzeneteket. A fogadó oldalon lévő hiba pedig nem zavarja működést legfeljebb újra próbálkozik. (83. ábra)

9.2.6.6 Klaszterezés (fürtözés)

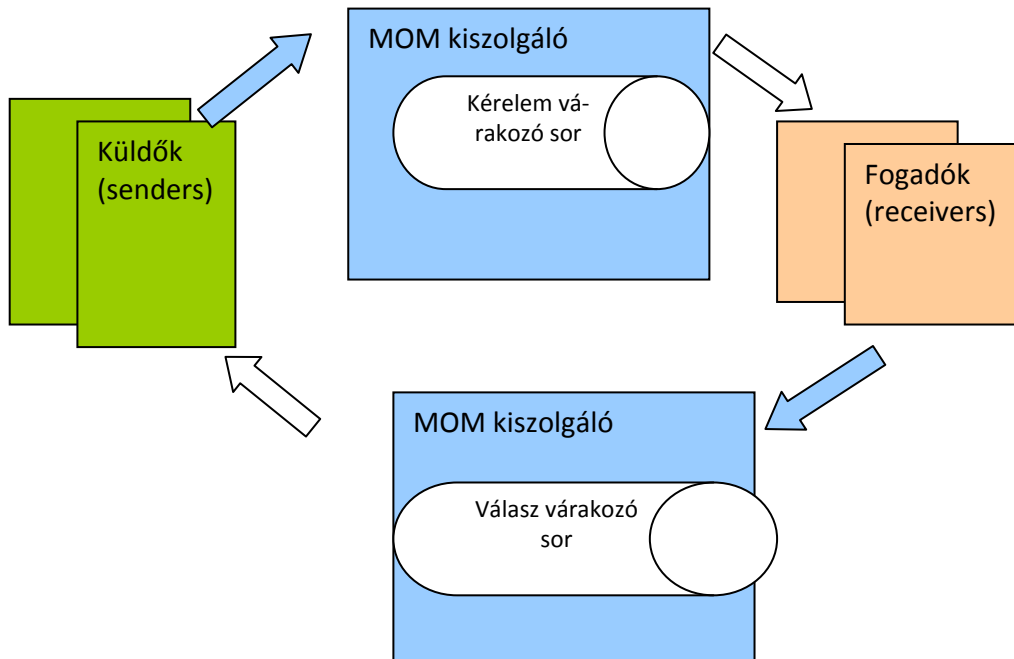
Sok esetben előfordulhat, hogy az adott MOM szerver kiesik, vagy esetleg a hálózati hiba miatt megszakad a kapcsolat vele. Ezt a nem kívánatos eseményt a MOM szerverek tükrözésével szokás kivédeni. Lényegében van egy tartalék MOM szerver egészen addig ameddig az elsődleges elérhető, ha pedig megtörténik a baj a szinkronba tartott szerver könnyen átveheti az elsődleges pozíciót. Ld. 84. ábra ábra.



84. ábra A kiszolgáló gépek klaszterezése (fürtözése) megbízhatóság növelése és a skálázhatóság érdekében

9.2.6.7 Két irányú kommunikáció

Az eddigi esetekben a küldő nem volt szüksége a fogadótól jövő válasz üzenetekre. Amennyiben ez szükséges egy adott probléma megoldásához a megoldás triviálisnak tekinthető. Bevezetünk egy új MOM szervert ami az ellentétes irányba továbbítja az üzeneteket. Ld. 85. ábra.



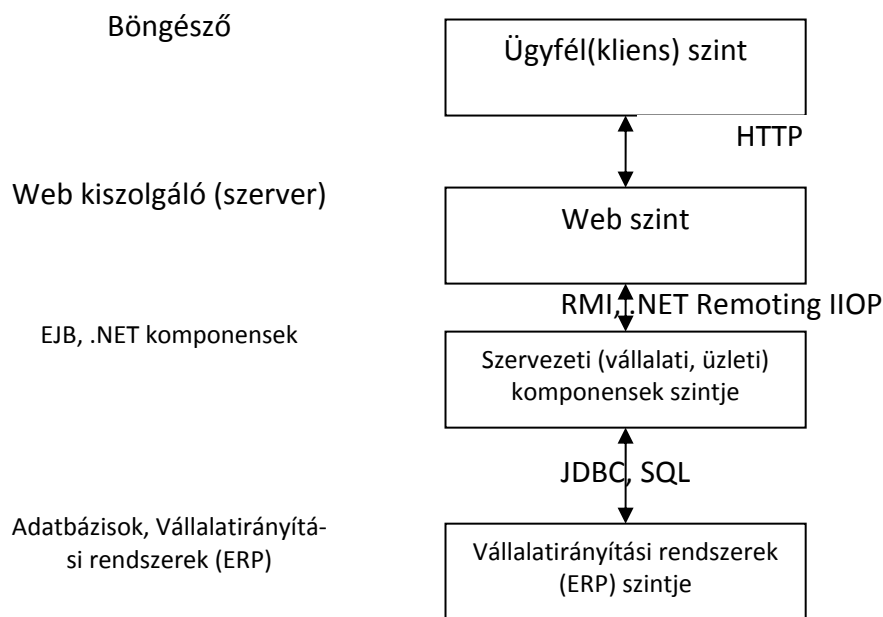
85. ábra Kérelm válasz üzenetváltás

9.2.6.8 Publikálás és előfizetés (Publish-Subscribe)

A MOM szerver alapvetően 1-1 vagy N-1 adatkapcsolatot oldotta meg. Az alkalmazások esetében előfordulhat 1-N kapcsolat is. A MOM szerver megfelelő módosítás után (a sorból nem kerülnek ki az elemek) képes megoldani ezt a feladatot is.

9.2.7 Alkalmazás kiszolgálók (Application Servers)

Az alkalmazás szerverekkel kapcsolatban többféle definíció létezik, de az összes egyetért abban, hogy központi eleme a *köztesszoftver* architektúrájának. Az alkalmazás szervernek nevezet réteg egy komponens alapú szerver architektúra középső rétege, amely elosztott kommunikációt, biztonsági, tranzakciós és perzisztencia eszközöket biztosít. A Java Enterprise Edition fogjuk példaként bemutatni. A N rétegű Web alkalmazás felépítését az ábra mutatja (86. ábra).



86. ábra N-szintű (N-tier) Web alkalmazás architektúra

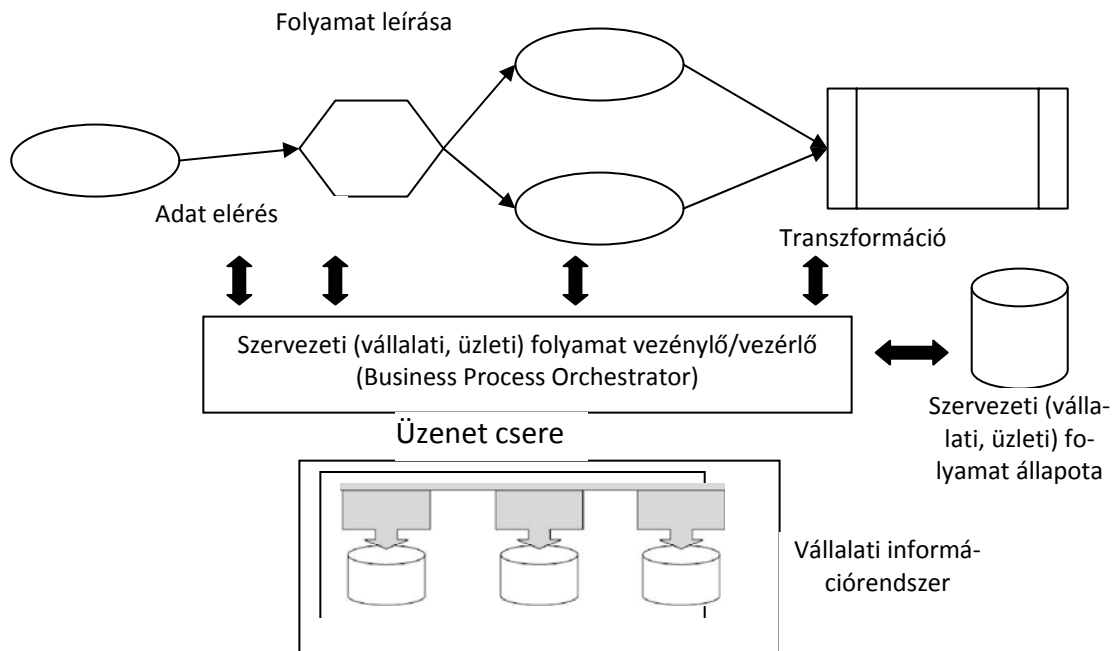
A Web alkalmazás rétegei és funkciói:

- **Ügyfél/kliens szint (Client Tier³⁰):** A Web aláalkalmazásoknál, a megjelenítést általában a böngészők végzik. Az utasítások HTTP kérések segítségével futnak a következő réteghez, a válaszok pedig HTML oldalak.
- **Web szint (Web Tier):** A kientől származó kérések feldolgozása, az üzenetek továbbítása üzleti logika réteg megfelelő komponensének. Amikor az alatta lévő réteg elkészült a feldolgozással, itt készül el a válasz üzenet.
- **Szervezeti (vállalati, üzleti) komponensek szintje (Business Component Tier):** Az *alkalmazás logika* helye. EJB, .NET, CORBA komponensek. Általában azért, hogy ezek a komponensek működhessenek adatbázis kapcsolat kell létrehozni, amelyet a következő szint tud nyújtani. (Rendszer által nyújtott szolgáltatások: component lifecycle management, state management; security, multithreading, resource pooling)
- **Vállalatirányítási rendszer szint (Enterprise Information Systems Tier):** Az adatkapcsolati réteg egységes felületet biztosít adatbázisok és más háttér információrendszer (*backend*) szolgáltatások felé.

9.2.8 Szervezeti (vállalati, üzleti) folyamat vezénylő/vezérlő (Business Process Orchestrator)

Ahogy az ábrán látható, az **üzleti folyamatokat vezénylő** (BPO) platformokat úgy tervezték, hogy viszonylag egyszerűen megvalósíthatóvá váljanak a hosszúideig tartó, hosszú futási idejű, magas fokon integrált szervezeti (vállalati, üzleti) folyamatok automatizálása. BPO platformokat különálló szinként valósítják meg azért, hogy használhatóak legyenek különböző üzenetküldő, üzenet központú architektúráknál, nevezetesen a SzOA vagy az üzenetbróker architektúrák esetében. A következő tulajdonságokkal bővül az üzenet küldő réteg:

1. **Állapotkezelő:** a végrehajtott üzleti folyamatok állapotát folyamatosan tároljuk egy adatbázisban. Ez alkalmazkodó-és ellenálló-képessé teszi a rendszert BPO meghibásodása esetén is. Ugyancsak nagy előny, hogy amint a folyamat állapota rögzítésre kerül az adatbázisban, nem szükségeltetik további erőforrást hozzárendelni a BPO-hoz addig, amíg az adott folyamat nem folytatódik.
2. **Fejlesztő eszközök:** vizuális folyamat definiáló eszközök állhatnak rendelkezésünkre, melyek segítségével kialakíthatók az üzleti folyamatok.
3. **Telepítési eszközök:** ezeknek az eszközöknek a segítségével megkönnyítjük a fejlesztők munkáját, abból a szempontból, hogy könnyebb lesz a logikai szervezeti (vállalati, üzleti) folyamatokat az szervezeti (vállalati, üzleti) információrendszerhez csatlakoztatni, kihasználva a különböző csatlakozási típus támogatásokat (beleértve az üzenet sortípusokat, Web protokollokat, SOAP-t vagy fájl rendszereket).



87. ábra Egy üzleti folyamat levezénylési platform felépítése

9.2.8.1 Az integráció bróker

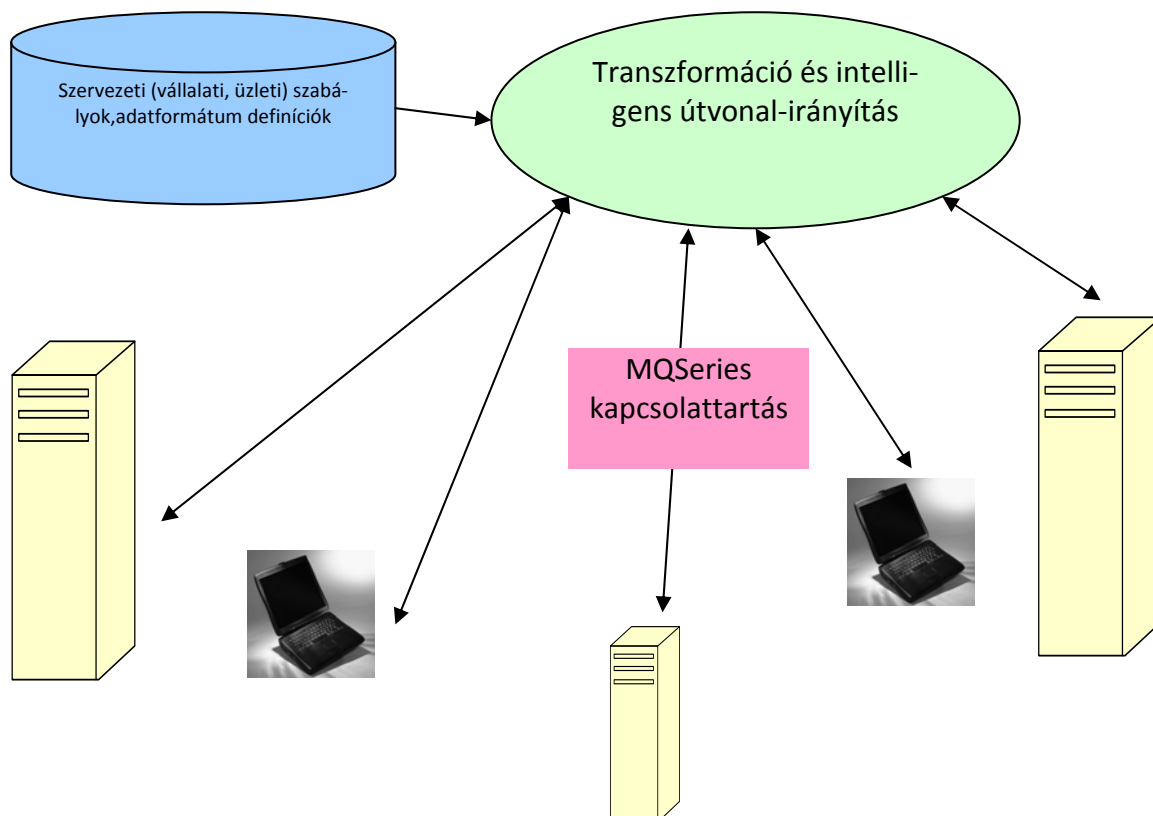
Jóllehet a kommunikációs köztesszoftver általános, egységes, szabványosítható alapot jelenthet valamennyi meglévő és később fejlesztendő alkalmazás integrálásához, néhány fontos probléma megoldásához önmagában nem elegendő. Például nem foglalkozik az alkalmazások algoritmikus különbségeinek áthidalásával, nem segít az alkalmazások által használt eltérő adatformátumok és adat tartalom értelmezések (*szintaktika* és *szemantika*) problémáinak feloldásában. Olyan esetben, amikor az alkalmazások száma négy vagy annál több, az egyes alkalmazás párok közti együttműködések lehetséges száma jelentősen megnőhet, négyzetesen növelve ezzel a rendszer fejlesztési idejét, költségeit és menedzselhetőségének bonyolultságát. A megoldást ilyen esetekben az *integráció bróker technológiája* jelenti. Ilyen eszközök pl. az IBM MQSeries Integrator, vagy a SeeBeyond e*Gate. Az integráció brókerek valamilyen kommunikációs köztesszoftverre, tipikusan üzenetcsatolt köztesszoftverre épülnek.

Alkalmazásuk esetén a kapcsolódó rendszerek egymásról, egymás létezéséről sem tudnak, függetlenek egymástól. Az együttműködési folyamat úgy valósul meg, hogy az algoritmus kooperációt igénylő pontjaiban az alkalmazások a lényeges adatokat üzenetekbe csomagolva

9.2 Alkalmazásintegráció

elküldik az *integráció brókernek*. Az *integráció bróker* az üzleti szintű integrációs szabályok alapján elemzi az adattartalmat, majd meghatározza, hogy mely más alkalmazásoknak és milyen formátumban kell azokat továbbítania. Az integráció bróker tehát szervezeti (vállalati, üzleti), alkalmazási szinten integrálja az egyes alkalmazási rendszereket. Ezen túl, képes az egyes rendszerek által használt adatok szintaktikai és szemantikai különbségeinek a feloldására is.

Az integrációs brókerek általában támogatják az *XML*-kezelést, valamint a csatlakozó alkalmazások menet közbeni, futásidejű módosíthatóságát, a téma és tartalom szerinti dinamikus **publikáció/előfizetés** (*publish/subscribe*) funkciójával. Ez nagymértékben hozzájárulhat a módosítások egyszerű kezelhetőségéhez. A dinamikus *publikáció/előfizetés* funkció témaköröihez rendelt ACL (*Access Control List*) alapú jogosultságkezelés a gyakran alkalmazott biztonsági funkciók közé tartozik. Egyszerű grafikus fejlesztő- és karbantartó környezettel rendelkeznek, és az igényesebb termékek általában több lehetséges platformon is működnek.



88. ábra Az integráció brókerek helye az integráció architektúrában

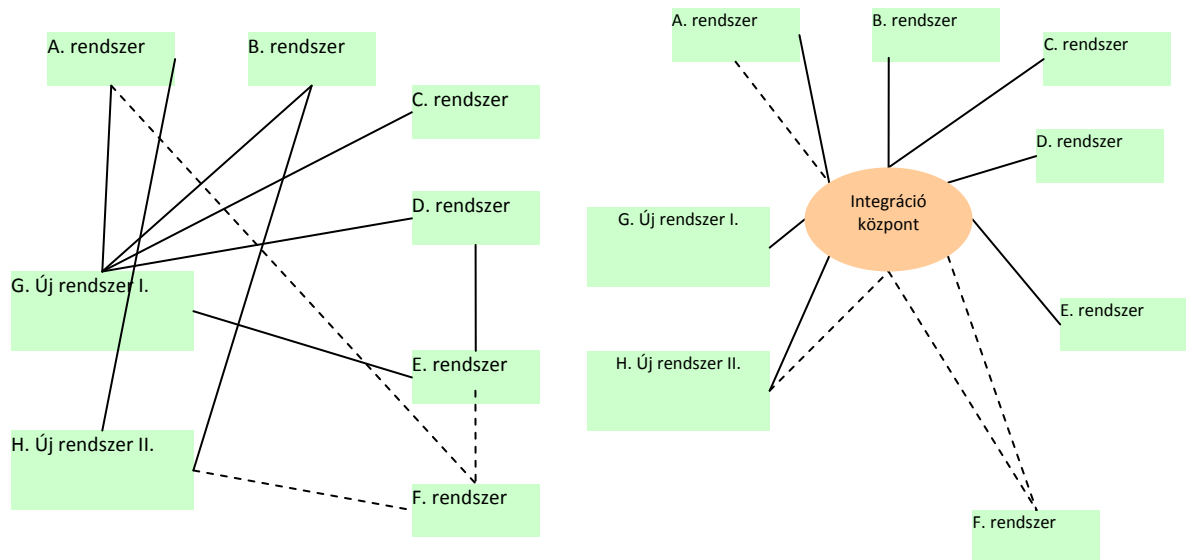
9.2.8.2 BPM (Business Process Management) - üzleti folyamatok szervezése

Az általános **BPM-rendszerek** lehetőséget nyújtanak az alkalmazások szervezeti (vállalati, üzleti) folyamatok által *vezérelt* integrálására is. Ezekbe a folyamatokba ugyanis a személyi közreműködés, feladat- végrehajtás mellett a különböző alkalmazások által végrehajtandó tevékenységek is bevonhatók. A BPM általában *kommunikációs köztesszoftverre* és esetlegesen *integráció brókerre* épül. Tipikus példák az IBM MQSeries Workflow, a CrossWorlds InterChange Server vagy az Extricity Partner Agreement Manager szoftverei.

Az szervezeti (vállalati, üzleti) folyamatok szervezése, irányítása és vezérlése az alkalmazás-integrációnak egy magasabb szintje. Látható ugyanakkor, hogy logikáját tekintve nem választható el a klasszikusnak tekinthető alkalmazásintegráció filozófiától, és eddig kifejlődött megközelítésektől, megoldásoktól.

9.2.8.3 A köztesszoftver technológia és megközelítés összefoglalása

A köztesszoftverek használatának egyik legfőbb előnye az alkalmazások közti kapcsolatok számának csökkentése. Amennyiben minden alkalmazást össze kívánunk kötni minden alkalmazással, a kapcsolatok száma pontosan $n*(n-1)/2$ (általános iskolai feladat, a sokszög átlói, mint lehetséges kapcsolatok száma). Egy új rendszer (vagy alkalmazás) rendszerünkbe történő integrálásakor az első esetben –amikor mindenkit mindenkivel össze akarunk kapcsolni – $\sim n^2$ db új kapcsolatot kell létesítenünk, míg köztesszoftverek használata esetén csak a köztesszoftverhez kell az új rendszert (alkalmazást) csatlakoztatnunk. Mindez nem jelenti azt, hogy m alkalmazásból álló rendszer esetén egy új alkalmazás integrálása $1/m$ annyi munkával jár, hiszen az üzleti logika szintjén értelmeznünk kell az alkalmazások közötti kapcsolatokat, az integráció brókerben rögzítenünk kell az alkalmazások közti kommunikáció szabályait. A két filozófia közti eltérést az alábbi két ábra szemlélteti.



89. ábra Integráció közteszoftver használata nélkül és közteszoftver használatával

A közteszoftver technológiák felépítését összefoglalva a következő megállapításokra juthatunk. A vállalati alkalmazások nem közvetlenül, hanem egy **integráció platformon** keresztül kapcsolódnak egymáshoz, illetve a külvilághoz. Ennek központja legtöbbször egy olyan *alkalmazáserver*, amire ráépülnek az integráció szoftverek és szolgáltatások egymás feletti rétegei. Az *alkalmazáserver*, mint fizikai technológiai architektúra építőelem és platform fogható fel, a rátelepülő integráció szoftverek pedig logikai technológiai komponensként értelmezhetők (ld. TOGAF).

Az architektúra legalsó szintjének feladata az *adatmozgatás*, az *üzenetek továbbítása*; a külvilág felé ez történhet **Web szabványokra** (XML, SOAP, UDDI) alapuló technológiák révén is. Efölött helyezkedik el az integráció bróker, amely már nemcsak továbbítja, hanem intelligensen át is alakítja az üzeneteket a megfelelő formátumra. Ezt követi az szervezeti (vállalati, üzleti) folyamatok megszervezése, integrálása, automatizálása; végül az architektúra felépítmény tetején helyezkedik el az **eseménykezelés** (*Business Event Handling*): ez a logikai komponens lehetővé teszi az üzleti eseményekben megmutakozó trendek, tendenciák csaknem valós idejű elemzését is, más technológiákkal kiegészítve, pl. adattárház stb.

9.2.8.4 Adatszintű integráció

Az alkalmazásintegráció alternatívája az **adatszintű** integráció. A Dataquest és a Gartner Group felméréseiből az derül, hogy egy nagy integrált vállalatirányítási rendszer képtelen lefedni a vállalatok teljes tevékenységi körét.

(A Dataquest elemzőcég felmérése szerint a cégeknek csupán a 20 százaléka használt egyetlen csomagot, 41 százalékuk kettőt, hármat vagy négyet, 39 százalékuk pedig legalább öt különböző programot alkalmazott. A Gartner Group pedig azt állította felméréseiben, hogy egy tipikus vállalat informatikai rendszere átlagosan hét ilyen alkalmazászigetet tartalmaz.)

Erre a kihívásra, ennek a problémának a kezelésére a nagy szoftver gyártók kidolgoztak megoldásokat. Az Oracle az pl. **e-Business Suite**, illetve az **IBM Web Sphere, MQ-Series** és kapcsolódó szoftver eszközei.

A 1990-es évek közepén a vállalatirányítási rendszerek (ERP) három lényeges problémával küszködtek:

- (1) csak egyedi programozási nyelven lehetett őket bővíteni;
- (2) nem fedték le az összes vállalati folyamatot;
- (3) nem voltak igazán integrált rendszerek.

Az *e-Business Suite* kísérletet tesz a fogyatékoságok kiküszöbölésére. Először is, az új fejlesztésekben csak **Javát** használnak, új funkciók írásához tehát csak ezt kell ismerni. Másodsor, az Oracle alkalmazáscsomagja - néhány különleges iparág kivételével - minden vállalatnak teljes körű funkcionalitást, az üzleti folyamatok teljes automatizálását kínálja.

Végül a harmadik kérdés az *integráció*: ezen az azt értik, hogy minden adat egyetlen helyen, egyetlen adatbázisban van. (*Adatszintű integráció.*)

A **törzsadatkezelés** az a módszertani és technológiai válasz arra az esetre, amikor az egyetlen, egységes, mindenképp fölötte álló adatbázis nem valósítható meg, mégis *adatintegrációra* is szükség van a teljes körű *alkalmazás integráció* megvalósításához.

9.2.8.5 Kritika az alkalmazásintegrációval szemben

Nem vált be teljesen az az elképzelés, amely szerint sikeresen össze lehetne kapcsolni a különböző cégektől származó "csúcskategóriás" (*best-of-breed*) alkalmazásokat. Mindnek *külön adatbázis* kell, és olyan alkalmazásokból nem lehet hatékonyan adatokat kinyerni, amelyeket nem terveztek már eleve az együttműködésre.

9.2 Alkalmazásintegráció

A kritika részben jogos, ezért van szükség az **alkalmazások szabványos fejlesztése**. A szabványos fejlesztés alapja nem lehet más, mint egy *szabványokra épülő fejlesztői környezet*. Erre hívják fel figyelmet az IBM-szakemberei, és erre biztosíthat alapot a *Web Services (Web szolgáltatások)* elterjedése (ld. 3.1).

9.2.8.6 Alkalmazások fejlesztése, tekintettel integrálhatóságukra

A vállalatoknak át kell gondolniuk, hogy miként kívánják megvalósítani az *alkalmazási rendszerek fejlesztését* akkor, ha ezeket az alkalmazásokat zökkenő-, akadály- és hézagmentesen akarják integrálni, és ha ennek az integrációnak a természetét – felkészülve a permanens változásokra a rendszer környezetében – menetközben módosítani szeretnék.

Ennek a problémának egyik lehetséges kezelése az ha a nagy, *monolitikus* szervezeti (vállalati, üzleti) információrendszer funkciók helyett az alkalmazásokat *moduláris* rendszerfelfogásban építik fel azért, hogy az üzleti folyamat megváltoztatása nélkül átalakíthatók legyenek, új környezetbe beilleszthetők legyenek.

Ezért az egyik megoldás az, ha egy **független közteszoftver** szinthez integrálják az *alkalmazási rendszereket*, az alkalmazásokat le kell választani infrastruktúráról, és nem *vertikálisan* integrálni az *operációs rendszerhez*. A vállalati információrendszerek integrációjának problémája kezelhető bonyolultságú megközelítések rendelkezésre állásával, kezdésnek tarthatóvá válni azért, mert a *TCP/IP* és a *HTTP* szabvány biztosítja a közös kommunikációs szabványt, az *XML* pedig biztosítja a közös adatcsere formátumot – meglehetősen rugalmasa, a félig strukturált adatszerkezet révén. Az informatikai tudomány jelenlegi állása és a fejlődő technológia ígérete szerint a szervezetek (vállalatok, üzleti szereplők) közti *integráció* a *szabványok* alkalmazásával, az *önleíró* adatbázisokkal és a *címtárak* telepítésével megoldható.

A másik alapfeltétel, a rendszerek **nyitottsága**, hiszen irreális az az elképzelés, hogy egy cégen belül mindenki pontosan ugyanazt a technológiát használja.

A harmadik előfeltétel a **virtualizáció**. A cégek kihasználatlan kapacitásokon ülnek. A nagygépekről a megosztott rendszerekre történő áttéréssel a funkcionalitás a végfelhasználó kezébe került ugyan, de ennek az ára az informatikai erőforrások gyenge kihasználtsága lett. Ez orvosolható az "*igény szerinti kapacitás biztosítással*", amit a teherelosztó *hálózati (grid)* számítástechnika, illetve a *számítási felhő* tesz megoldhatóvá, oly módon, hogy lehetővé teszi a megosztott erőforrásokon való osztozkodást.

9.2.9 Vállalati architektúra integráció SzOA segítségével

20. Táblázat A SzOA befogadásának szintjei

Befogadás szintje	Elnevezés	Leírás
1	Egyedi Web szolgáltatások megvalósítása	Az új vagy régi, létező alkalmazások által tartalmazzott egyes feladatokra szolgáltatások készítése
2	A szervezeti (vállalati, üzleti) funkciók szolgáltatásorientált integrációja	A szervezeten (vállalaton) kívüli és belüli több alkalmazás integrálása úgy, hogy a kialakított szolgáltatások egy szervezeti (vállalati, üzleti) célkitűzés elérését segítsék.
3	A szervezetet (vállalatot) átfogó informatikai átalakítás	Megtervezett architektúra alapján az integráció lehetővé tételére a szervezet egészét átfogóan a szervezeti (vállalati, üzleti) funkciókra vonatkozóan.
4	A szervezeti (vállalati, üzleti) tevékenység igény szerinti alakítása	A meglévő szervezeti (vállalati, üzleti) működési modell átfogó és széleskörű átalakítása, illetve az új működési modellek bevezetése, rendszeresítése.

21. Táblázat SzOA kivitelezésének hat lehetséges megközelítése

Módszer	Leírás (a tipikus projekt felelős által adott jellemzés)	Típus
A szervezeti (vállalati, üzleti) folyamatok által vezérelt	A szervezeti (vállalati, üzleti) folyamatoknak a rendelkezésre álló erőforrásokból kell táplálkoznia, mindegyik szervezeti (vállalati, üzleti) tevékenység informatikai funkcionális szolgáltatásokat igényel, ezeknek a funkcionális szolgáltatásoknak, rugalmasoknak, helyettesíthetőeknek kell lenniük.	Felülnézetből (felülről-lefelé)
Modell vezérelt architektúra fejlesztés eszköz alapokon (Modell	A modellt (a szervezeti (vállalati, üzleti) működési modellt) egy arra alkalmas eszközben definiálják, majd az eszköz előállítja a modell részleteit.	Felülnézetből (felülről-lefelé)

Driven Architecture, MDA)		
A régi rendszer funkciók becsomagolása (Wrap legacy)	Régi rendszerek vannak, amelyeknek a rugalmas alkalmazkodó képessége alacsony. Új funkciókkal való gyors bővítésre volna szükség, de a rendszer nagyon particionált. Az egyes alkalmazási rendszerek tulajdonképpen önmagukban álló silók, amelyek magukba zárják az egyes funkciókat.	Alulnézetből (Alulról-felfelé)
A régi rendszerek komponensekre bontása (Componentize legacy)	A monolitikus régi rendszerek modulokra bontása, fordító programokra támaszkodó eszközök segítségével.	Alulnézetből (Alulról-felfelé)
Adat vezérelt	Az információkat szolgáltatásokon keresztül teszi elérhetővé anélkül, hogy a szolgáltatást nyújtó oldallal az adat sémákat, szerkezeteket vagy a megvalósítás során hozott tervezési döntéseket közölnék, tudomására hoznák.	Adat-központú
Üzenet vezérelt	A meglévő rendszereket szabványos, nem egyedi gyártói protokollokon keresztül, kell integrálni, a köztük a kommunikációt megvalósítani.	Az alkalmazások és rendszerek szolgáltatás orientált integrációja.

22. Táblázat SzOA kivitelezésének lehetséges mintázatai és hasznai

Értelmezési környezet	Mintázat	Alkalmazhatóság
Siló, koncentrált funkcionalitás	Bedrótozott programkód (nem igazán mintázat, inkább egy pillanatnyi állapot az	Egy adott időpillanatban; alacsony kockázat; nehezen változtatható, magas teljesítőképességű rendszerek.

	időben) .	
Elosztott, több pontról történő elérés.	Pont-pont kapcsolat	A létező funkciókat gyorsan elérhetővé lehet tenni, gyorsan értéket hoz létre, a beágyazott funkciók rendelkezésre állnak.
A régi információrendszerek funkcióinak a becsomagolása és hívhatóvá tétele Web szolgáltatásokon keresztül.	(Web) szolgáltatás adapter.	A Web szolgáltatás felhasználójának hozzá kell férnie a nem szolgáltatásként létrehozott funkcióhoz (a régi rendszert egy (Web) szolgáltatás meghívásán keresztül lehet elérni.)
Egy szolgáltatást a proxy kiszolgálóján keresztül lehet elérni, ha nincs közvetlen elérés a szolgáltatás nyújtó szolgáltatás leírása felé és nem tudja közvetlenül meghívni a szolgáltatást.	Szolgáltatás proxy	A végfelhasználó felé egy SzOA kapcsolófelületet nyújt.
A szolgáltatás nyújtó rugalmas kiválasztásának lehetőségéről gondoskodik.	Távoli szolgáltatás stratégia	A szolgáltató nyújtó váltás rugalmasságáról gondoskodik, ami akár a szolgáltatás minősége vagy a funkcionális szolgáltatás miatt bekövetkezhet. Ez távoli szolgáltatás stratégiai megközelítés lehetőséget ad a szervezet összevonások, egyesülések, felvásárlások felgyorsítására illetve a szolgáltató rugalmas megváltoztatására akkor, amikor az alkalmazás portfólió konszolidációja történik meg.
A redundáns funkciók felszámolása, refaktoring és konszolidálás, illetve egyes esetekben a létező rendszerek helyettesítése.	Egy ponton keresztül történő elérés (Single point of access)	Egy ponton keresztüli elérésről gondoskodik potenciálisan jelentős változottságú funkció irányában. A szolgáltatás stratégia általában megkívánja az

9.2 Alkalmazásintegráció

		egy pontos elérést.
Egy projekt vagy egy olyan vertikális szervezeti funkció, tevékenység, amely hierarchikus szervezeti egységeket fog át, de mégis néhány olyan funkcióra támaszkodik, amelyet még nem alakítottak át szolgáltatássá.	Virtuális szolgáltató (Virtual provider)	Nincs szolgáltatást nyújtó; a szolgáltatások bővítésének el kell érnie a kritikus tömeget.
Egy ponton keresztül történő elérés (Single point of access)	Szolgáltatás integrátor (Service integrator)	Útvonal irányítás, transzformáció.
Általános szervezeti integráció megközelítés.	Szolgáltatási sín (Enterprise service bus)	Közvetítés (mediation); útvonal irányítás; átalakítás; irányelvek; szabályok; események; a szervezeten belül, vagy a partnerek között a szervezet ökoszisztémáján illetve értékteremtő hálózatán belül.
A SzOA nirvánája; A szolgáltatások dinamikus újra konfigurálása a környezetet ismerő szolgáltatások segítségével, amelyek a szervezeti terület specifikus képességeire támaszkodnak.	Integrál szolgáltatási ökoszisztéma. (Integrated service ecosystem)	Bizonyos, szemantikailag összefüggő szervezeti partnerek részére olyan dinamikus konfiguráció képességekről gondoskodik, amelyek az ökoszisztéma képességeit kiaknázzák, és újra kombinálják azért, hogy saját maguknak és az ökoszisztéma teljességének magasabb értéket nyújtsanak.

9.2.9.1 Szolgáltatás integráció érettségi modellje

23. Táblázat A szervezet szolgáltatás integrációra való érettségének felmérésére szolgáló modell táblázata

	Sziló	Integrált	Komponens alapú	Szolgáltatások	Összetett szolgáltatások	Virtualizált szolgáltatások	Dinamikusan újra konfigurálható szolgáltatások
Szervezet	Szervezeti funkció alapján megszervezett	Szervezeti funkció alapján megszervezett	Szervezeti funkció alapján megszervezett	Szolgáltatás központúan szervezett	Szolgáltatás központúan szervezett	Szolgáltatás központúan szervezett	Igény szerint
Szervezet irányítása	Az informatikai funkció irányítása <i>ad hoc</i>	Az informatikai funkció irányítása <i>ad hoc</i>	Az informatikai funkció irányítása <i>ad hoc</i>	SOA jellegű irányítás kezdetei	SOA és az informatikai funkció irányítás összeillesztése	SOA és az informatikai funkció irányítás összeillesztése	SOA és az informatikai funkció irányítás összeillesztése
Módszerek	Strukturált elemzés és tervezés	Objektum-orientált modellezés	Komponens alapú fejlesztés	Szolgáltatás központú modellezés	Szolgáltatás központú modellezés	Szolgáltatás központú modellezés	Használati-eset nyelvtan központú modellezés
Alkalmazások	Modulok	Objektumok	Komponensek	Szolgáltatások	Folyamatos integráció szolgáltatások révén	Folyamatos integráció szolgáltatások révén	Dinamikus alkalmazás összeépítés
Architektúra	Monolitikus	Réteges architektúra	Komponens alapú architektúra	Kezdeti SzOA	SzOA	Grid alapú SzOA	Dinamikusan újra konfigurálható architektúra
Információ	Alkalmazás specifikus	Szervezeti egység, területfüggő	Kanonikus modell	Kanonikus modell	Szervezeti szintű adatszótár	Virtualizált adat szolgáltatás	Szemantikus adat szótár
Infrastuktúra	Platform függő	Platform függő	Platform függő	Platform függő	Platform függő	Platform független	Dinamikus érdeklés és reagálás
	1. szint	2. szint	3. szint	4. szint	5. szint	6. szint	7. szint

9.2.9.2 Enterprise Architecture Assessment

9.2 Alkalmazásintegráció

Szervezeti Architektúra Mutatószámrendszer (Enterprise Architecture Score Card)									
Érthető/világos = jól definiált és dokumentált									
Részben érthető/világos = részben jól definiált és dokumentált									
Nem érthető/világos = NEM felismert és NEM kezelt, NEM definiált és NEM dokumentált									
SzAM (ASC)	Állapot meghatározás: Érthető/világos = 2 Részben érthető/világos = 1 Nem érthető/világos = 0		Állapot meghatározás: Érthető/világos = 2 Részben érthető/világos = 1 Nem érthető/világos = 0		Állapot meghatározás: Érthető/világos = 2 Részben érthető/világos = 1 Nem érthető/világos = 0		Állapot meghatározás: Érthető/világos = 2 Részben érthető/világos = 1 Nem érthető/világos = 0		A stratégia illesztés / Integráció szintje (Alignment / Integration) Integráció / nyomonkövethetőségi tényező 0-2; 0=Elégtelen 1= Átlagos 2=Teljes
	Szervezet	Információ	Információ-rendszer	Technológia infrastruktúra	"2" állapotok száma	"1" állapotok száma	"0" állapotok száma		
A szervezeti architektúrára vonatkozó kérdések eredménye									
1	Vajon a szervezeti architektúra küldetése (Mission), jövőképe (Vision), céljai (Goals) és célkitűzései (Objectives) érthetőek-e és világosak-e?	2	2	1	0	2	2	1	1
2	Vajon a szervezeti architektúra program kiterjedése, terjedelme érthető-e és világos-e?	2	2	2	2	2	4	0	0
3	Vajon a szervezeti architektúra kifejlesztése során előállított tervezési és fejlesztési termékek formája és funkcionális szolgáltatási szintje érthető-e és világos-e?	2	2	2	2	1	4	0	0
4	Vajon a szervezeti és információs stratégia érthető-e és világos-e?	1	1	0	0	1	0	2	2
6	Vajon a szervezetet vezárló és irányító alvek és erők érthetőek-e és világosak-e?	0	0	0	0	0	0	0	4
7	Vajon a szervezeti kulcsfontosságú mutatószámok, indikátorok érthetőek-e és világosak-e? (KPI, Key Performance Indicators)	1	1	1	1	1	0	4	0
8	Vajon a szervezeti kritikus fontosságú siker tényezői érthetőek-e és világosak-e? (Critical Success Factors)	2	2	1	1	1	2	2	0
9	Vajon a szervezeti kritikus fontosságú érintett felek meghatározása érthető-e és világos-e?	1	1	1	1	0	0	4	0
Részpontszám A szervezeti architektúra fejlesztés szervezeti környezetje									
10	Vajon a szervezettel kapcsolatban álló, együttműködő partnerek bevonásra kerültek érthető-e és világos-e?	2	2	1	2	2	3	1	0
11	Vajon a szervezettel kapcsolatban álló, együttműködő partnerekkel kapcsolatos szerződéses megállapodások érthetőek-e és világosak-e?	2	0	0	1	1	1	1	2
12	Vajon a szervezettel kapcsolatban álló, együttműködő partnerekkel kapcsolatos interoperabilitási (együttműködési) szabványok érthetőek-e és világosak-e?	0	1	2	2	1	2	1	1
13	Vajon a szervezettel kapcsolatos törvények, jogszabályok, hatósági szabályozások érthetőek-e és világosak-e?	1	1	0	0	0	0	2	2
14	Vajon a szervezettel kapcsolatban álló, együttműködő partnerek viszonylatában az információért viselt felelősség, tulajdonosi, kezelési és földolgozási jogosultságok érthetőek-e és világosak-e?	1	1	0	2	1	1	2	1
Részpontszám A szervezetet körülvevő, külső környezet									
11	Vajon a szervezeti architektúrával kapcsolatos funkcionális követelmények érthetőek-e és világosak-e?	1	1	2	2	1	2	2	0
12	Vajon a szervezeti architektúrával kapcsolatos nem funkcionális követelmények érthetőek-e és világosak-e?	1	1	0	1	0	0	3	1
13	Vajon a szervezeti architektúrával kapcsolatosan használt fogalmak érthetőek-e és világosak-e?	2	1	1	2	1	2	2	0
14	Vajon a szervezeti architektúrával kapcsolatos biztonsági követelmények érthetőek-e és világosak-e?	0	0	1	0	0	0	1	3
14	Vajon a szervezeti architektúrával kapcsolatosan az igazgatási, vezetési, irányítási követelmények érthetőek-e és világosak-e? (Governance Requirements)	1	1	1	1	1	0	4	0
Részpontszám Fogalmi, koncepcionális szint									
15	Vajon a lezárítandó termékek meghatározása megfelelő-e a logikai szintnek, érthető-e és világos-e?	1	2	1	2	1	2	2	0
16	Vajon a szervezeti architektúrával kapcsolatos logikai szintű terve vonatkozó kritikus fontosságú döntések érthetőek-e és világosak-e?	1	1	2	2	0	2	2	0
17	Vajon a szervezeti architektúrával kapcsolatos logikai szintű terve vonatkozó kritikus fontosságú döntések nyomonkövethetőségének módja mérhető-e és világos-e?	0	0	1	1	1	0	2	2
18	Vajon a szervezeti architektúra logikai szintjének megfelelő leíró módszerek és technikák érthetőek-e és világosak-e?	2	1	1	0	1	1	2	1
19	Vajon a szervezeti architektúra logikai szintjének megfelelőek-e az alkalmazott modellező eszközök?	1	1	1	0	1	0	3	1
20	Vajon a szervezeti architektúra logikai szintje megfelelő-e valamilyen szervezeti architektúra szabvány logikai szintjének?	1	0	1	2	1	1	2	1
Részpontszám Logikai szint									
21	Vajon a lezárítandó termékek meghatározása megfelelő-e a fizikai szintnek, érthető-e és világos-e?	1	1	2	2	1	2	2	0
22	Vajon a szervezeti architektúrával kapcsolatos fizikai szintű terve vonatkozó kritikus fontosságú döntések érthetőek-e és világosak-e?	1	2	2	2	1	3	1	0
23	Vajon a szervezeti architektúrával kapcsolatos fizikai szintű terve vonatkozó kritikus fontosságú döntések nyomonkövethetőségének módja mérhető-e és világos-e?	1	2	2	2	0	3	1	0
24	Vajon a szervezeti architektúra fizikai szintjének megfelelő leíró módszerek és technikák érthetőek-e és világosak-e?	2	1	1	0	1	1	2	1
25	Vajon a szervezeti architektúra fizikai szintjének megfelelőek-e az alkalmazott modellező eszközök?	0	1	1	0	1	0	2	2
26	Vajon a szervezeti architektúra fizikai szintje megfelelő-e valamilyen szervezeti architektúra szabvány fizikai szintjének?	1	0	1	2	1	1	2	1
Részpontszám Fizikai szint									
27	Vajon a szervezeti architektúrával kapcsolatos vonatkozó kritikus fontosságú tervezési döntések érthetőek-e és világosak-e?	0	0	1	1	0	0	2	2
28	Vajon a szervezeti architektúra fejlesztés által kiváltott szervezeti hatások következményei a szervezetben belül érthetőek-e és világosak-e?	1	1	2	2	1	2	2	0
29	Vajon a szervezeti architektúra fejlesztés költségvetéselvezetése gyakorolt következményei a szervezetben belül érthetőek-e és világosak-e?	0	0	0	0	2	0	0	4
30	Vajon a szervezeti architektúra fejlesztés biztonságra gyakorolt következményei a szervezetben belül érthetőek-e és világosak-e?	0	0	0	0	1	0	0	4
31	Vajon a szervezeti architektúra fejlesztés kiváltott hatásai a szervezeti architektúra igazgatására, vezetésére, irányítására a szervezetben belül érthetőek-e és világosak-e?	0	0	1	1	1	0	2	2
Részpontszám Architektúra átvalósítási szint									
		1	1	4	4				
Teljes pontszám									
	Összes szint	35	33	36	39				

10 A FOLYAMAT FOGALMA

A folyamat meghatározása nem egyszerű mivel a szervezés és vezetéstudományban, gazdálkodás tudományban és az informatikai különböző ágazataiban is találkozunk a folyamat fogalmával. Az értelmező kézi szótárakban is változatos meghatározásokat találunk. Különböző szakértők bár hasonlóan adják meg a fogalmat, mégis más-más nézőpontból fogják azt meg, mást hangsúlyoznak benne.

Szervezeti (vállalati, üzleti) folyamat

- „Folyamaton olyan szervezeti (vállalati, üzleti) tevékenységek sorát értjük, amelyek egymás mellé téve a vevő számára értéket jelentő eredményt produkálnak.” ([52], pp. 14.);
- „Szervezeti (vállalati, üzleti) tevékenységek egy olyan gyűjteménye, melyekhez egy vagy több input szükséges, amelyek a fogyasztó számára értékes kimenetet állítanak elő.” ([52], pp. 46.);
- „A szervezeti tevékenységek olyan strukturált, mérhető és mért gyűjteménye, melyek egy speciális terméket állítanak elő egy fogyasztó vagy egy piac számára. Időben és térben kiterjedő speciális tevékenységek egymáshoz rendelése, melyeknek van kezdete, vége, világosan meghatározott bemenete és kimenete: cselekvési struktúra.” ([34], , pp. 5.);
- „Egy vagy több olyan tevékenység, amely értéket növel úgy, hogy egy bemenetek halmazát alakítja át kimenetek halmazává (javakká vagy szolgáltatásokká) munkatársak, módszerek és eszközök kombinációjával.” ([109], pp. 75.);
- „A folyamat olyan koordinált tevékenységek sorozata, mely a fogyasztó igényeit elégíti ki.” ([38] pp. 9.);
- Egy bizonyos célra irányul és a feladatok olyan időbeli és logikai sorrendjét jelenti, amely szervezetek és szervezeti egységek közötti munkamegosztáson alapul és információ- és kommunikációtechnológiákat (ICT, Information and Communication Technology) használ fel a feladatok elvégzése során. A szervezeti stratégiából levezetett, előre adott *szervezeti folyamat* célokat megvalósító szolgáltatások állítja elő. Egy szervezeti folyamatot különböző részletezettséggel, változatos nézőpontokból lehet formálisan leírni. A leírás részletezettségének maximális lehetőségét akkor lehet el-

értnek tekinteni, ha a szóban forgó feladatot a munkatárs egy menetben, a munkaterület megváltoztatása nélkül tudja elvégezni.

Munkafolyamat („workflow”)

- olyan átalakítást végző folyamatokból (adat, termék, dokumentum stb.) áll össze, amelyek egy körmentes irányított gráfot alkotnak, általában egy kezdeti és egy befejezési ponttal.

Informatikai (adattranzformáló) folyamat

- A folyamatok olyan átalakító tevékenységek, amelyek a bemenő adatokat kimenő adatokká alakítják, közben az adatok módosulnak, aktualizálódnak. ([80]).

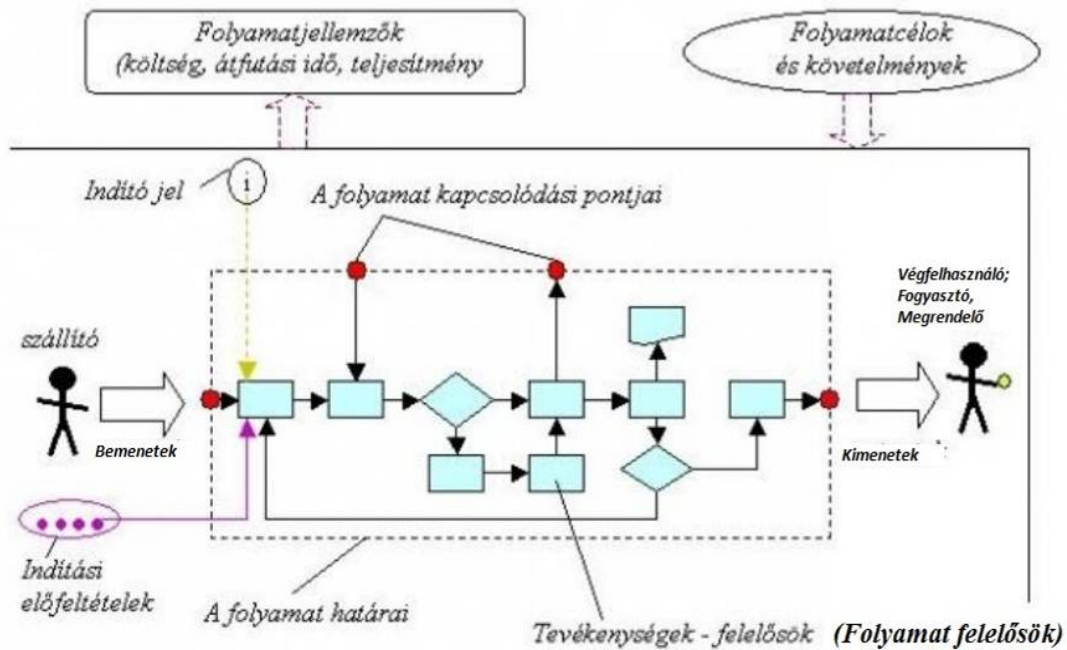
Az ábra (90. ábra) és a fenti definíciók megadják azt a keretet, mely a folyamatok fogalmának értelmezéséhez szükséges.

Ezek alapján a szervezeti (vállalati, üzleti) folyamat tehát cselekmények, tevékenységek strukturált, szabályozott lánc, aminek konkrét, akár több bemenete és több kimenet, eredménye van, a kimenetek a végfelhasználó vagy a fogyasztó számára értéket jelentenek, és kielégítik a szervezeti (vállalati, üzleti) környezet vagy piac igényeit. A folyamatok jól meghatározható kezdő- és végponttal rendelkeznek, lefutásuk során a folyamat láncba tartozó egyes tevékenységek nem csak egymás után, hanem egymással párhuzamosan is végrehajthatódnak.



90. ábra A folyamat és kapcsolatai

A legtöbb folyamat során döntéseket kell hozni, melyek hatással vannak a folyamat kimenetelére is, például bizonyos tevékenységeket nem kell végrehajtani. Fontos továbbá, hogy szervezeti (vállalati) szinten a folyamatok kapcsolódnak egymáshoz, nem egymástól elszigetelten mennek végbe.



91. ábra: A szervezeti, vállalati folyamatok teljes környezetükben [83]

Ez a komplexitás, melyet érzékeltet a 91. ábra is, eredményezte azt, hogy a szervezetek tudatosan elkezdtek foglalkozni folyamataikkal, azokat összehangolásával. A szervezés és vezetésstudományban, a gazdálkodástudományban kialakult a **folyamatmenedzsment**, folyamat kezelés fogalma, és tudományterülete. A gazdaságinformatikában, de általában az informatikában pedig **folyamatmodellezés** tudomány és szakmai területe. Ennek következményeként a szervezetek felismerték a *folyamatmenedzsment* és a *folyamatmodellezés* fontosságát.

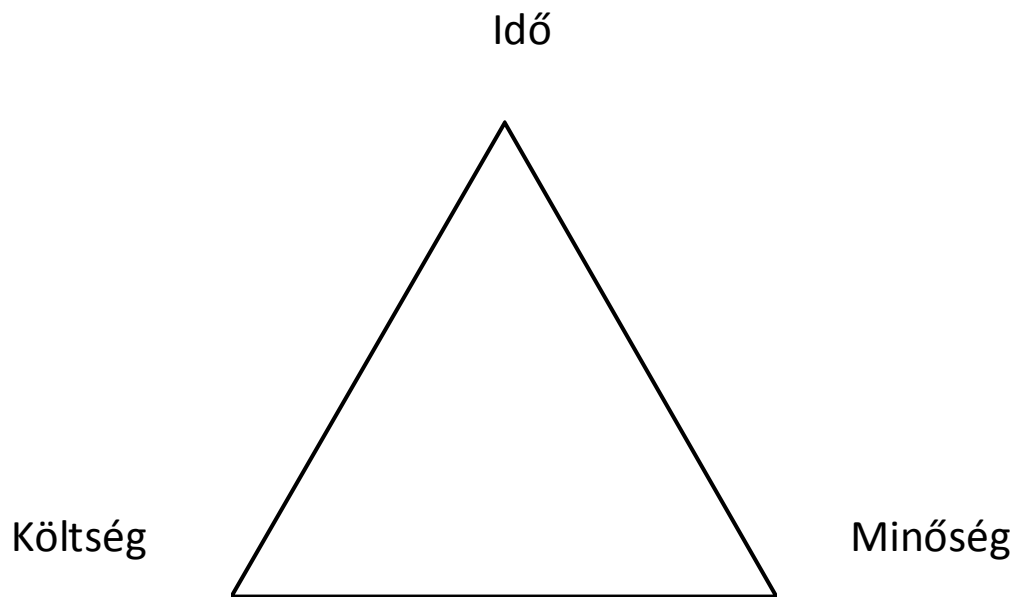
10.1 Szervezeti (vállalati, üzleti) folyamatmenedzsment – Business Process Management

A szervezeteknek, vállalatoknak először arra kellett ráébredniük, akár a piacon akár a közszolgáltatásban működnek, hogy nem a termékeikre és szolgáltatásaikra kell koncentrálniuk, hanem a saját folyamataikra. A 92. ábra szemlélteti, hogy a folyamatokra az idő, a költség és a minőség tényezői hatnak. Ezek egymástól függő és egymást gyengítő hatások.

Az a szervezet, vállalat, amely felismeri, kidolgozza, és folyamatosan javítja is folyamatait, azaz a működését, és a végtermékeit állandóan hatékonyabbá, gyorsabbá és olcsóbbá teszi, az hosszútávon megőrzi piaci részesedését, és esetleg még versenyelőnyre is szert tesz. Az ilyen cég a folyamatmenedzsment elvét alkalmazza. A folyamatmenedzsment alatt olyan vezetési rendszert értünk, amelyben a vállalati folyamatokban megjelennek a stratégia változá-

sai, a folyamatok teljesítményét mutatószámok segítségével mérik és vizsgálják, valamint a folyamatfejlesztésre is nagy hangsúlyt fektetnek [116]. A közigazgatásban és közszolgálatban a költség és a minőség adekvát mérőszámait kell alkalmazni, de az alap megközelítés azonos. A folyamatmenedzsment elmélettel kapcsolatban a következő elvárások fogalmazhatók meg [58]:

- **Tartósság:** A folyamatmenedzsmentnek egy szervezet, vállalat életében hosszú távon garantálnia kell az eredményes működést.



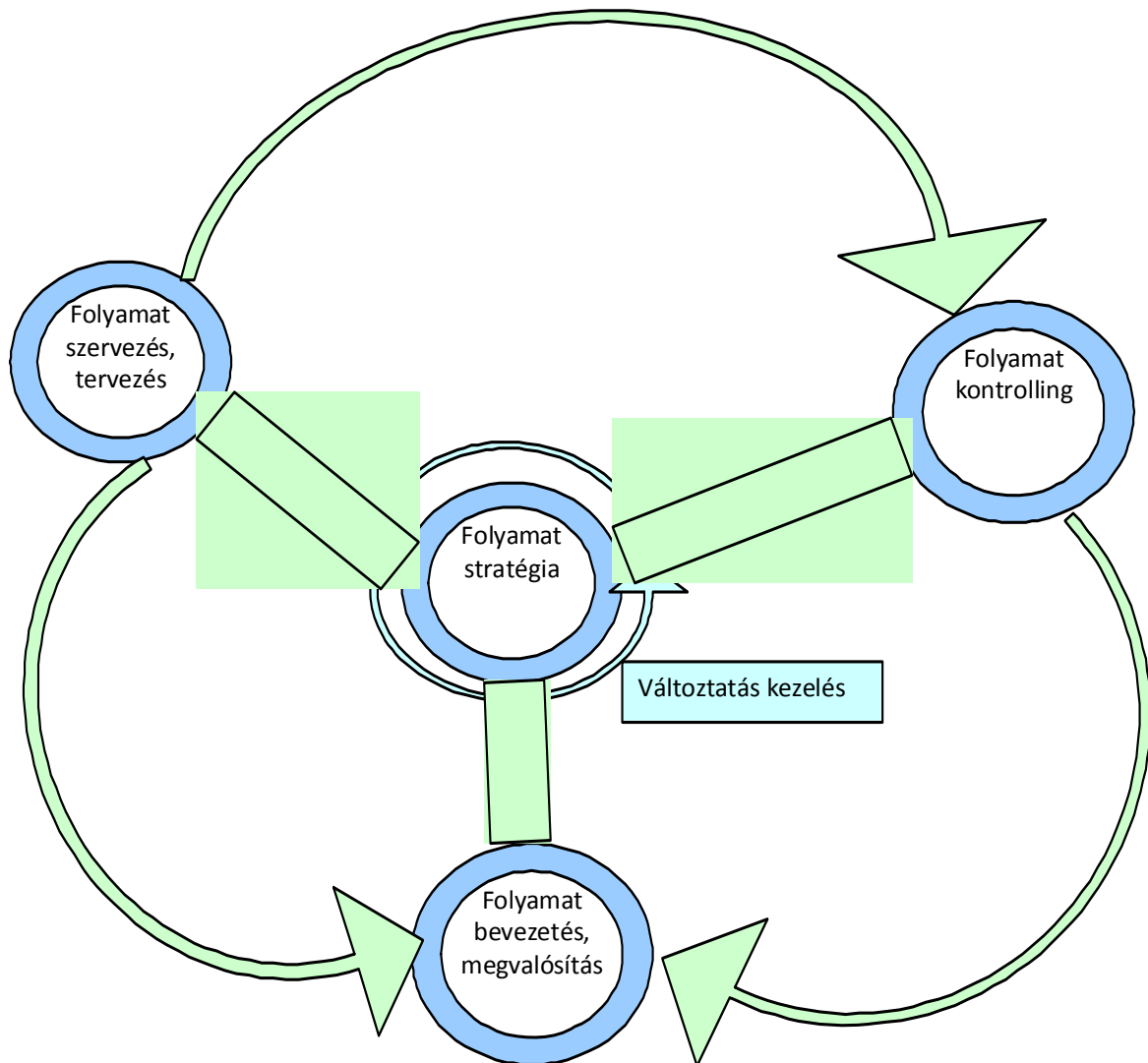
92. ábra A folyamatokat befolyásoló hatások – A klasszikus projekt háromszög

- **Minden ágazatban való alkalmazhatóság:** A folyamatmenedzsment elvnek iparágától, ágazattól, szektortól függetlenül mindenhol használhatónak kell lennie. Tehát a termelő-, és szolgáltatóvállalatoknál, sőt még a közszolgáltatási szféra szervezeteinél is alkalmazhatónak kell lennie.
- **Az eredmény szempontjából létfontosságú folyamatok teljesítményének növelése:** Így a szervezet, vállalat profiljának megfelelő és fontosabb folyamatokra nagyobb hangsúlyt kell fektetni.
- **Különböző szervezeti, vállalati helyzetekben való alkalmazhatóság:** Egy szervezet különböző célok miatt nyúlhat a folyamatmenedzsment eszközéhez. Ilyen szituációk lehetnek például a költségcsökkentés, a tervezési rendszer folyamatalapúra történő változtatása és a jövőbeli növekedés, a közszolgálatban politikai, adminisztratív, célok elérés, ügyfél barátságos kiszolgálás, szolgáltató állam céljainak elérése.

- **Fokozatos bevezetés:** Nem ajánlatos az egész folyamatmenedzsment elméletet egy nagy változtatási projektként megvalósítani, inkább a szervezeti ellenállás elkerülése érdekében érdemes kisebb lépésekben megismertetni a munkatársakkal a folyamat elvű gondolkodást.

A 93. ábra a folyamatmenedzsment rendszer legfontosabb elemeit mutatja be [116]:

- Az szervezeti (vállalati, üzleti) stratégiában bekövetkező változások lefordítása a folyamatokkal szembeni követelményekre: Az szervezeti (vállalati, üzleti) terv lebontása folyamatszintre. Célok, mutatószámok és tevékenységek, lépések meghatározása;
- A folyamatfejlesztések, tervezések és a folyamatszervezés végrehajtása;
- A folyamat célkitűzések teljesülésének ellenőrzése, mérése az előre definiált mutatószám rendszer alapján;
- A folyamatkezelés, -menedzsment integrálása a teljesítményértékelési rendszerbe, az eredmények összehasonlítása a tervekkel, és az eltérések alapján újabb lépések, tevékenységek meghatározása.



93. ábra A folyamatmenedzsment rendszer legfontosabb elemei

Az előbb leírt (93. ábra) folyamatmenedzsment rendszer sok előnnyel jár bevezetője számára. Az előnyök a következők szerint foglalhatók össze [116]:

- A mérőszámok és rendszeres értékelésük alapot jelent a működési beavatkozások megtérülési mutatóinak javításához.
- Az szervezeti (vállalati, üzleti) folyamatok folyamatos elemzése lehetőséget ad a hatékonyság és az eredményesség (*efficiency, effectiveness*) állandó javítására.
- Csökken a szervezetek érdekellentéte a szervezetek közötti, közös folyamat iránti érdekeltség, elkötelezettség miatt.
- Megvalósul a teljesítmények tárgyilagossabb összehasonlíthatósága.
- A vállalaton egéyzét átfogó, végigfutó folyamatoknak „gazdája”, felelőse lesz.

- Kialakul a működés ügyfél szempontú megközelítése és elterjed a folyamatelvű gondolkodásmód.

A folyamatmenedzsment tehát a folyamatok rendszeres finomítását, újjáalakítását jelentő menedzsment eszközrendszer, vagyis egy olyan irányítási rendszer, amely megvalósítja a stratégiában végbemenő változások lefordítását a folyamatok szintjére. A folyamatmenedzsment rendszer támogatja még a folyamatteljesítmények elemzését és kiértékelését, valamint a folyamatfejlesztések meghatározását és implementálását is. A folyamatok fejlesztése és átalakítása azonban egyáltalán nem egyszerű feladat.

10.2 Üzleti folyamatok újraszervezése – Business Process Reengineering

Szervezési szempontból a szervezeti (vállalati, üzleti) folyamatok teljesítményének növelésére több megközelítés, módszer van. E módszerek feladata a szervezet (vállalat) folyamatainak és szervezeti struktúrájának módosításával a költségoptimalizálás és a folyamat hatékonyságának és eredményességének növelése. Az egyik ilyen módszer a folyamatok állandó, folyamatos javítása (**Continuous Process Management – CPM**), aminek célja a vállalati hatékonyság és eredményesség folyamatos növelése az szervezeti (vállalati, üzleti) folyamatok állandó javítása, fejlesztése révén, melyben az egész szervezet részt vesz. Egy másik lehetőség a folyamat összemérés (*benchmarking*), aminek célja az adott vállalat iparágán belüli legjobb gyakorlatok összegyűjtése, majd ezen iparági standardokkal a vállalat saját folyamatainak összehasonlítása. Ehhez mutatószámokat kell definiálni a folyamatok hatékonyságára és eredményességére vonatkozóan. Ha eltérések vannak akkor az iparági, ágazati értékektől való negatív eltérések csökkentésére érdemes az iparági legjobb megoldásokat bevezetni/alkalmazni. Mivel a megoldások jelentősen különböznek, a közszolgálatban, közigazgatásban a versenyszféra egyes ágazataiban ezért általános szabályok nem léteznek, hanem a nemzetközileg bevált, és alkalmazható gyakorlat illesztése és befogadása az általános megoldás.

A folyamatfejlesztés egyik legtöbbször használt megoldása: az szervezeti (vállalati, üzleti) folyamatok újraszervezése (**Business Process Reengineering – BPR**). A BPR legtöbbször hivatkozott definíciója a következő:

Szervezeti (vállalati, üzleti) folyamatok újraszervezése (BPR) Az újraszervezés az üzleti, vál-

latalati folyamatok alapvető újragondolása, és drámai javulás elérése végett, radikális átszervezése azért, hogy a szervezetek olyan lényeges teljesítménymutatói érjenek jelentős javulást, mint pl. a költségek, a minőség, a szolgáltatás és a gyorsaság.[51].

A fogalom alapján megállapíthatjuk, hogy ez egy olyan innovatív változtatás, ami kiterjed a teljes szervezeti (vállalati, üzleti) folyamatrendszer fejlesztésére és folyamat alapúvá változtatja a szervezeti struktúrát. A BPR leginkább a folyamatok hatékonyságát, és informatikai támogatásukat hangsúlyozza. A BPR a vállalat korábbi szabályait is felrúgja, szakít a hagyományokkal, az új technológiai lehetőségeket kreatívan alkalmazza.

Bár a BPR elmélete, jelentős előnyöket ígér, a gyakorlatban a BPR eredményei egyáltalán nem meggyőzőek. A BPR projektek időtartama hosszú, akár 2-3 évet is eltarthatnak, a sikerességi arányuk 50 % közül mozog a jelentős piackutató cégek kimutatásai szerint. Az újrászervezés kockázata magas, és a radikális változás miatt a szervezeten belül általában jelentős az ellenállás.

10.3 BPM és BPR informatikai támogatása

Az szervezeti (vállalati, üzleti) folyamatok fontosságának előtérbe kerülésével párhuzamosan az informatika is egyre inkább elterjedtebbé és fontossá vált a szervezetek eredményességének és hatékonyságának javítása érdekében, ezért egyre szélesebb körben kezdték használni a szervezetek különböző funkcionális területein és szintjein. Az 1960-as évektől kezdve, az automatizált adatbevitel, majd az így bevitt, nagymennyiségű adatra épülő vezetői döntéstámogatás után a '90-es évekre az informatika a szervezetek egyre több szervezeti tevékenységét kezdte támogatni.

A BPR (a folyamatok újraszervezése) divathullám és a verseny fokozódása, a legtöbb szervezetet folyamatainak átvizsgálására és radikális átalakítására kényszerítette. Többen felismerték az információtechnológia, az informatika fontosságát, és az abban rejlő lehetőségeket az új folyamatok kialakításában. Manuális munkatevékenységek kerültek automatizálásra, különálló adatbázisokból központi adatbázisok keletkeztek, ha a folyamatlogika ezt kívánta meg. Az igazán jó újraszervezésekhez azonban nem elegendő mindenhova számítógépeket helyezni, hiszen az eleve rosszul működő folyamatok ettől nem lesznek jobbak, csupán hamarabb vezetnek rossz eredményre.

Az informatika fontossága egyre jelentősebb lett a folyamatok újra és átszervezése a sikeresség, eredményesség és hatékonyság tekintetében:

1. a fejlesztéseknek nem a feladatokra, hanem a várható eredményekre kell koncentrálniuk;
2. a sikeres újraszervezéshez meg kell nyerni a projekt igazi végrehajtóit;
3. a folyamatokat együtt kell kezelni az azokhoz tartozó információ- és adatfeldolgozó, azaz informatikai folyamatokkal;
4. a döntési pontok azok a helyek, ahol szabályozottan be lehet avatkozni folyamatokba, és ezeket a pontokat a munkafolyamatokhoz (*workflow*) és a munkafolyamatokon belüli egyes folyamatokhoz (*process*) kell igazítani;
5. az információt annak keletkezési helyén kell, és csak egyszer szabad rögzíteni.

Az informatika a folyamatok alapvető átalakítójának, sőt kényszerítő erőnek tekinthető, és a különböző partnerek közötti kommunikáció leghatékonyabb költségcsökkentő eszközeként jelenik meg.

A BPR során az informatika a szervezeti folyamatok megújulásában jelentős szerepet játszó, az új folyamatokban lényeges támogatást nyújtó funkció jelenik meg. Azonban a folyamatok újjászervezése önmagában nem elegendő, még akkor sem, ha kielégítő tervezés előzi meg azt. A folyamatok életciklusába azok tervezésén, kialakításán és bevezetésén túl beletartozik a *mérési, ellenőrzési, kontrolling* eljárások, illetve a megfelelő *mutatószámok* kialakítása. Az informatika jelentős szerepet játszik a folyamat-átalakítási projektek változáskezelési életszakaszában is.

A fentiekre építve „nőtte túl” a BPR-t (a folyamatok újraszervezését) a *folyamatmenedzsment*, amely a szervezet összes folyamatának összehangolását, egységes kezelését, és a meglévő folyamatok folyamatos fejlesztését, esetlegesen radikális átalakítását célozza meg. Ez az irányítási, igazgatási (menedzsment) szemlélet komplexebb informatikai támogatást igényel, és így – szintén az általános informatikai trendekkel megegyezően, mely szerint a korábban egy-egy részterületre összpontosító szoftverek egyre több funkciót nyújtanak az ügyfeleknek – a XXI. század elejére kialakultak a sok funkcionális támogatást nyújtó folyamatmenedzsment szoftverek (*multi-functional BPM software*), és számuk egyre nő.

A *folyamatmenedzsment* (BPM) szoftverek segítségével vizsgálhatók a folyamatok jellemzői. A folyamat leírásokat, ábrázolásokat grafikus formában jelenítik meg a felhasználók felé. Ennek megfelelően, a BPM szoftverek legáltalánosabb funkciói, a következők:

- a folyamatok modellezésének grafikus ábrázolása;

10.4 A folyamatmodellezés

- a szervezeti felépítés, az informatikai támogatás és az adatmodell grafikus ábrázolása;
- a folyamatok szervezeti rendszerének egységes kezelése, az összefüggések feltárása;
- a folyamat-szimuláció és a folyamatelemzés;
- közös munka támogatása;
- a folyamatok bevezetése és megvalósítása, szolgáltatások létrehozása folyamatmodellből;
- a folyamatok dokumentálása.

A legtöbb folyamatmenedzsment eszköz nem csupán a folyamatok feltérképezését és dokumentálását célozza meg, hanem a folyamatok kezelésének minden oldalát támogatni kívánja. Ezért a *folyamatmenedzsment* szoftverek (BPM szoftverek) részévé váltak olyan funkcionális szolgáltatások, mint például a kockázatkezelés és kockázat modellezés, vagy az *informatikai architektúra* tervezése. Emellett megjelentek a szakterületi folyamatok modellezésére is célzottan képes eszközök, mint például az informatikai infrastruktúrával kapcsolatos informatikai funkció (ITIL) folyamatok modellezésére alkalmas szoftverek.

10.4 A folyamatmodellezés

A folyamatmodellezés alapvető célja, hogy a szervezeti (vállalati, üzleti) folyamatok számára egy könnyen értelmezhető, egyértelműen megragadható szerkezetet hozzon létre. A folyamatmodellezés során meg kell határozni a folyamatmodellezésre vonatkozó stratégiai célokat, feladatokat, és a kapcsolódó erőforrások eszközeit. A folyamatmodellezésben az egyes munkafolyamatokra és azokon belüli egyes folyamatokra meg kell jelölni hogy ki, mit tevékenykedik és milyen eszközök segítségével. Az ily módon kialakított struktúra ad lehetőséget a folyamatok elemzésére, az idő, minőség és költség tekintetében

A modelleket kialakításhoz érdemes már a korábban kialakított módszertanokat használni. Ezekben a módszertanokban előre definiálták a folyamatmodellezési szabályokat, a használható objektumokat és az objektumok között létrehozható kapcsolatokat.

A modellek objektumokból állnak, amiket a modellezési munka során kell az ábrákon, a modellben elhelyeznünk és elneveznünk. Ilyen objektumok például az események, tevékenységek, szerepkörök, szervezeti egységek, dokumentumok, informatikai erőforrások stb. Ezeket az objektumokat az egyes, különböző szabványok, módszertanok előre meghatározott grafikus ábrákkal, szimbólumokkal jelölik. Az objektumok megrajzolása után a közöttük

fennálló kapcsolatokat is meg kell határozni, ezeket általában vizuálisan, nyilakkal vagy vonalakkal jelenítik meg. A folyamatok modellezésének és korrekt ábrázolásának fontos eleme az ún. folyamatkapcsolók, amelyek a folyamatok különböző kimeneteire vonatkozó lehetőségek érzékeltetésére szolgálnak. Az objektumokhoz és az azokat tartalmazó modellekhez is attribútumokat, jellemzőket, sajátosságokat, tulajdonságokat adhatunk meg.

A modellezési módszertanok általában lehetőséget biztosítanak egyedi objektumok létrehozására és használatára is, bár ezeket általában speciális osztályba sorolják be. Ezáltal a szabványos módszertanok nem sérülnek, mégis megteremtik a bővíthetőség és a testre szabhatóság lehetőségét. Az egyetlen bővítési szabály az, hogy az új elem nem sértheti meg az alapvető szabályokat, és nem használhat a többi elemmel összetéveszthető jelöléseket.

10.5 Folyamatmodellezési módszertanok

Az elemző és modellező informatikai eszközök egyik funkciója a különböző objektumok grafikus ábrázolása. Ennek a tevékenységnek támogatására területenként és funkcionális szolgáltatás tekintetében többféle modellezési módszertan terjedt el az évek során, melyek közül több is alkalmas a folyamatok modellezésére is. Egyes módszertanok csak a folyamatok egyes oldalainak, jellemzőinek modellezésére alkalmasak, például csak az adatáramlások és az adatfeldolgozó folyamatokat tudják modellezni, más módszerek pedig a folyamatok minden fontos jellemzőjét mind szervezési mind informatikai szempontból tudják a felhasználók felé megjeleníteni.

A különböző folyamatmodellezési módszertanok fő jellegzetességeik alapján a következő nagy csoportokba sorolhatóak:

- **Funkció központú** módszertanok, amelyek a vizsgált területet fekete doboznak tekintve, a vizsgált rendszer külvilág irányában nyújtott funkcióit vizsgálja. Ebbe a megközelítésbe sorolhatóak az adatfolyam diagram alapú módszerek ([80]) és IDEF0 ([54]). Ezek a leíró modellek a főbb szervezeti (vállalati, üzleti) tevékenységeket, szolgáltatásokat, szervezeti területeket azonosítják, és utána e tevékenységek bemeneteit, kimeneteit és felül nézetből, magas szinten a tevékenységek feldolgozási lépéseit ábrázolják.
- **Adat központú** módszertanok közé sorolhatóak az entitás-kapcsolat diagramok ([80]) és IDEF1x modellek ([119]). Ezek a modellek a munkafolyamatokon belüli adatfeldolgozási folyamatok adat- és információfeldolgozási feladataihoz szükséges adatszerke-

zetek modellezésére és ábrázolására koncentrálnak. A modell feladata egyrészt a szervezet (vállalat) lassan változó, statikus adat és információ szerkezetének feltárása, másrészt a munkafolyamatok és az adatfeldolgozó folyamatok információigényének kielégítése illetve az átalakított adatok helyes tárolásának visszatükrözése.

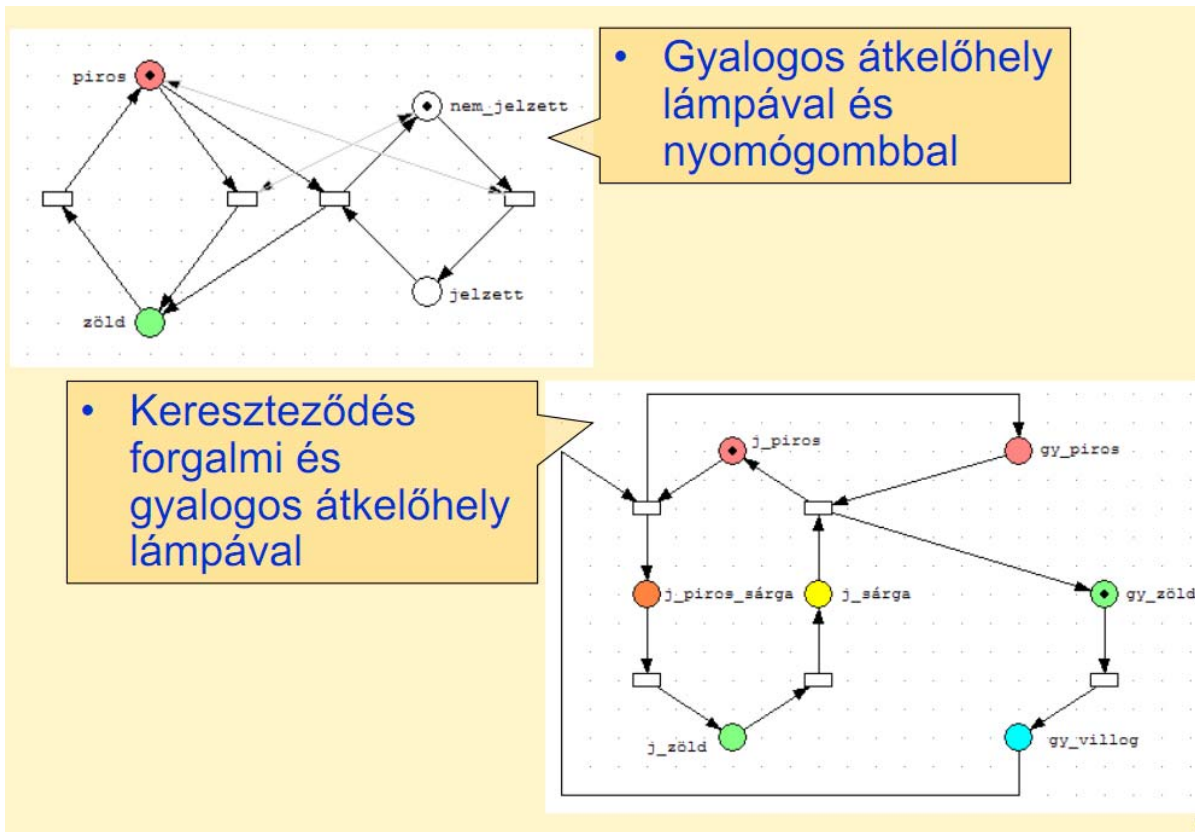
- **Objektum-orientált** módszertanok és modellező, leíró nyelveik. Az UML (Unified Modelling Language) mint leíró, modellező nyelv és ábrázoló, diagram technika szabványosított, nemzetközi szabványosítási testületek alakították ki az egységesített nyelvet. Az UML és az O-O módszertanok nem teljesen azonosak, de a gyakorlatban alkalmazott, szoftver eszközökkel támogatott módszerek egyre pontosabban betartják a szabványosított nyelv szabályait. Mivel az O-O módszertanokban nyilvánvalóan foglalkozni kell a létező és leendő rendszer funkcióival, munkafolyamataival, folyamataival ezért az információrendszerek ilyen oldalainak, nézőpontjainak leírására is több modellezési megoldás, diagram technika létezik az O-O módszertanokban és az UML nyelvben. Az objektumorientált szemlélet terjedésével ezek a modellek is megjelennek a szervezeti (vállalati, üzleti) munkafolyamatok és folyamatok szervezési, tervezési, ábrázolási technikái között. Céljuk absztrakt modellek létrehozása és magas szintű, újrahasznosítható szervezeti (vállalati, üzleti) munkafolyamat és folyamat mintázatok felismerése és megfogalmazása és azonosítása.
- **Szervezeti folyamat központú** módszertanok, közé sorolhatók az IDEF3, Gantt-diagram, PERT, Petri háló, eseményvezérelt folyamatlánc, BPMN. Ezek a módszertanok alapvető céljukként folyamatok ábrázolására jöttek létre. Megközelítésükben mégsem azonosak, a folyamatoknak különböző részeit hangsúlyozzák és ennek megfelelően más és más nézőpontokat részesítenek előnyben.

10.6 Petri háló

A Petri-háló diszkrét elosztott rendszerek matematikai ábrázolására szolgál. A Petri-hálókat az 196-ban Carl Adam Petri dolgozta ki PhD értekezésében. Ez a formális, matematikai alapú folyamatmodellező, diagram technikai, grafikus ábrázolás alkalmas a vasúti biztosító berendezések és a szervezeti (vállalati, üzleti) folyamatok leírására egyaránt.

A Petri-háló **helyekből** (place), **átmenetekből** (transition), **tokenekből** (token) és **irányított élekből** (arc) áll. Az élek kötik össze a helyeket az átmenetekkel és fordítva, ugyanakkor a helyek és az átmenetek saját csoportja között nincsen közvetlen éllel megvalósított kapcso-

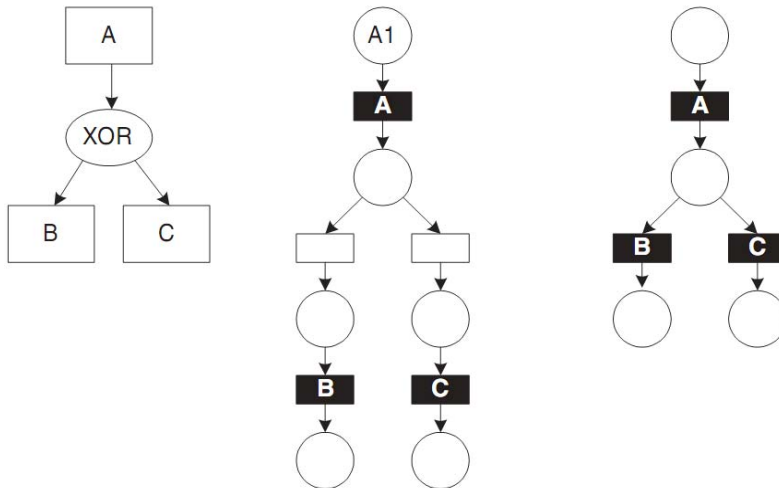
lat, azaz a Petri-hálókat irányított *páros gráfok*. Az egyes helyeken tetszés szerinti számú „token” fordulhat elő, a tokenek akkor kerülnek át a következő helyre, ha az átmenethez vezető élek mindegyikén az átmenet feltétele teljesül. Tokenek összessége a folyamat pillanatnyi *állapotát* reprezentálja. A 94. ábra egy egyszerű Petri-hálót, - köznap, közérthető helyzetre - ábrázol néhány tokennel.



94. ábra Egyszerű Petri-háló tokenekkel- klasszikus tankönyvi példa

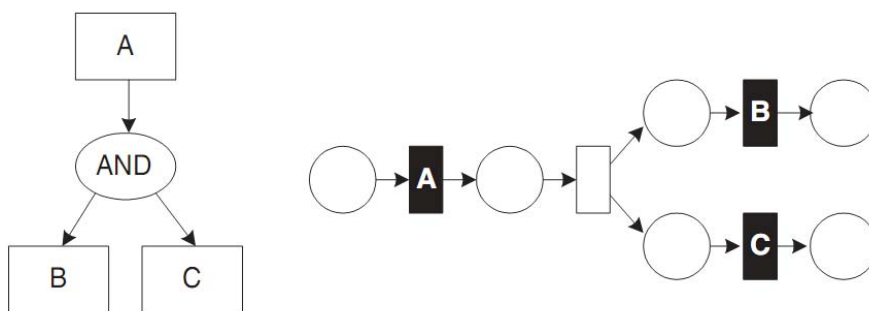
A Petri-hálókat alkalmasak a szervezeti (vállalati, üzleti) folyamatok vezérlési, irányítási rendszerének modellezésére. A különböző feltételes eseteket „AND” és „XOR” elágazásokkal és csatlakozásokkal (*split*, *join*) ábrázolni tudja (Ld. [21], [71]).

A legjelentősebb három *folyamatmenedzsment* ipari szabvány – a BPEL, a WSFL, és a BPMN – gyökere a Petri hálókat formális, matematikai elméletében található. Mindegyik szabvány használja a tokenek fogalmát a vezérlési folyamat szemantikájának megragadása érdekében, A BPMN specifikációja teljes egészében használja a token fogalmát. A WSFL és BPEL nyelvekben a halott utak eltávolításának (dead path elimination) problémája indított el szakmai vitát Petri hálókat elmélete által inspirálva.



95. ábra A XOR (logikai kizáró vagy) leképezése Petri hálóba

A Gantt- és PERT-diagramok kifejezetten a projekttervezésben, a projekt tevékenységek, folyamatok, munkacsomagok, kritikus utak tervezésében váltak be, a szervezésben, folyamatmenedzsmentben nem terjedtek el. A Petri-hálók pontos matematikai elméletének következetes alkalmazása a szervezésben és folyamatmenedzsmentben a kutatási, tudományos területeken kívül szintén nem vált széleskörűvé, azonban a Petri hálók elméletén alapuló folyamat illetve szolgáltatás leíró ipari szabvány nyelvek jelentős szerepre tettek szert, különösen amiatt, hogy a korszerű informatikai architektúrákban nevezetesen a SzOA-ban, közvetlenül alkalmazásra találhatnak.



96. ábra Az AND (logikai és kapcsolat) leképezése Petri hálóba

10.7 Eseményvezérelt folyamatlánc – Event-driven Process Chain (EPC)

Az **eseményvezérelt folyamatlánc** az egyik széles körűen alkalmazott folyamatmodellezési módszertan. A népszerűségének az oka az, hogy egyetlen egy modellben képes bemutatni a tevékenységek, események, logikai kapcsolók, szervezet lényeges alkotórészeit, adatszerke-

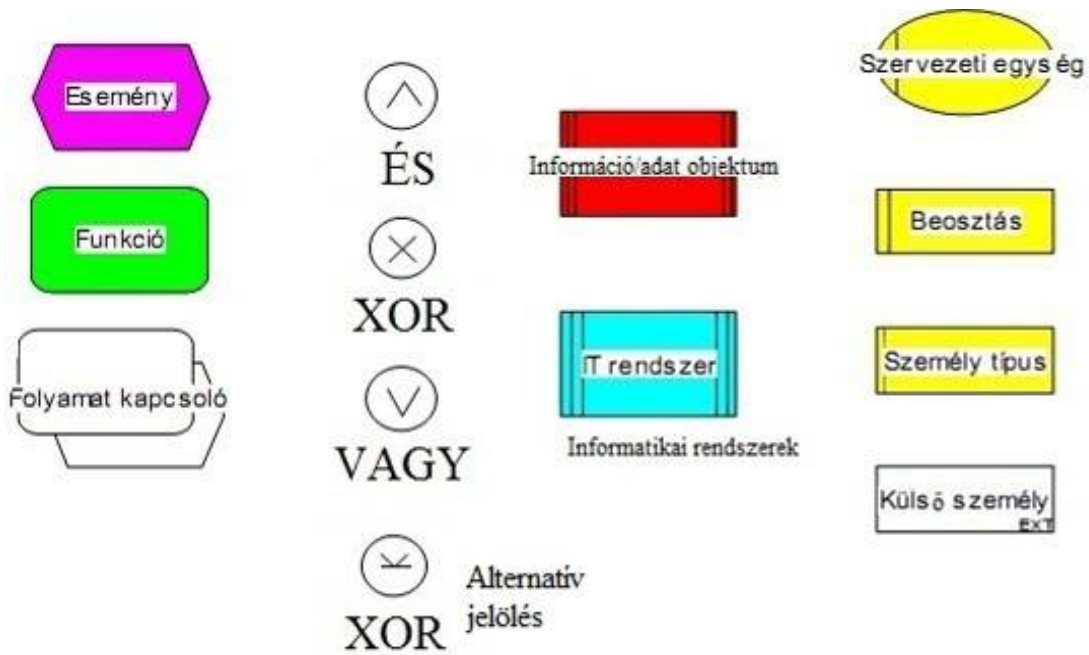
zet komponenseit, és alkalmazási rendszereket leíró objektumokat. Itt a hangsúly azon van, hogy nemcsak a tevékenységet ábrázolják, hanem hozzátartozó erőforrásokat is, vagyis a szerepköröket, informatikai rendszereket és az adat entitásokat. Ennek a folyamat modellező módszernek a gyökerei is a Petri hálókra nyúlnak vissza.

Az eseményvezérelt folyamat modellezés az 1980-as és 90-es években elterjedt adatfolyam diagram ábrázolásokkal szemben - amelyek deklaratív módon érzékeltették az adatok és információk áramlását a folyamatok között – a szervezeti (vállalati, üzleti) funkciók időbeli lefutását és logikai kapcsolatait egységes keretben kívánta megjeleníteni. Az *eseményvezérelt folyamatlánc* az SAP® vállalatirányítási rendszer egyik fontos alkotó eleme, ami elterjedtségének az oka, és ezért a gazdaságinformatikában jelentős szerepet játszik.

Az eseményvezérelt elnevezés onnan ered, hogy a funkciók és a funkciók végrehajtását kiváltó események ábrázolásából épül fel a folyamatmodell. Az objektumok idő dimenzióban és a köztük fennálló logikai kapcsolat dimenzióban kerülnek ábrázolásra folyamatábrában, szándék szerint korrekt módon tükrözve az objektumok között fennálló viszony rendszert. Az *eseményvezérelt folyamat* ábrázolást egy olyan félig- formális módszernek tekinthetjük, amelynek logikai-matematika alapjai a Petri hálókra vezethetők vissza de mind szintaktikai mind szemantikai értelemben jelentősen bővült, ezért gyakran kibővített *eseményvezérelt folyamatláncnak* is nevezik.

A folyamatdiagram teljesség szempontjából kétféle lehet. Az egyik az **egyszerűsített** modellezés, ami csak a lényeges időbeli és a logikai kapcsolatokat helyezi el a folyamatmodellben. A **kibővített** eseményvezérelt folyamatlánc egységes keretben, integrálva ábrázolja a funkciók és adatok közötti dinamikus, illetve a termék vagy szolgáltatás és szervezeti ábra közötti statikus kapcsolatokat. Ezeket az alap konstrukciókat egy sematikus ábrázolásban az ábrán lehet látni (98. ábra).

Az ábrán látható különböző színű és alakú alakzatok különböző folyamatmodellezési objektumokat jelentenek. Az alábbi ábrán (97. ábra) látható, hogy melyik szimbólum, diagram technikai jelölés mit jelent.



97. ábra Eseményvezérelt folyamatlánc objektumai

10.7.1 Az esemény

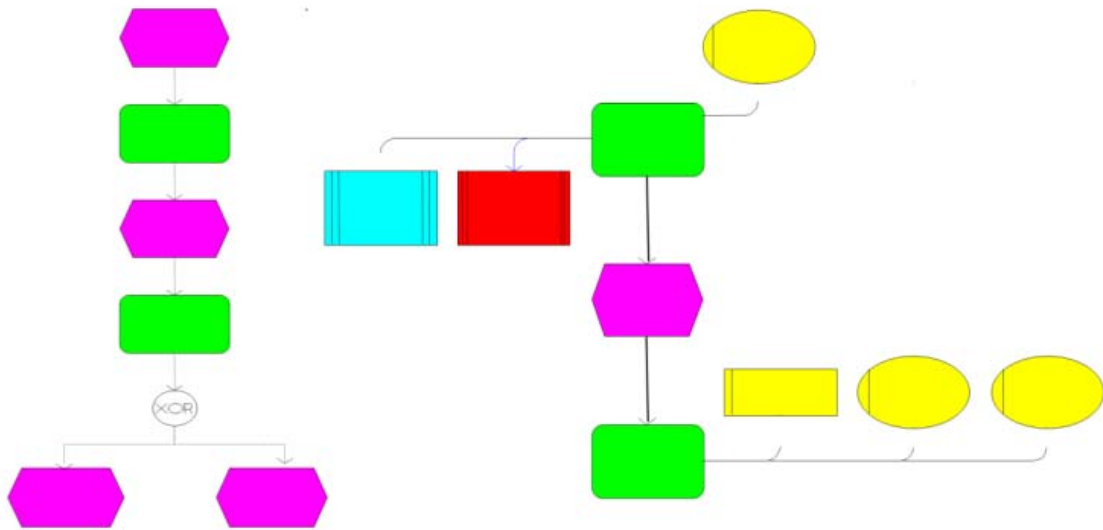
Szervezeti (vállalati, üzleti) esemény

- alatt olyan szervezeti vagy üzemgazdasági szempontból lényeges eseményt értünk, amely ilyen vagy olyan módon befolyásolja vagy vezérli a szervezet időbeni működését, tevékenységeinek lefolyását

Néhány az üzleti életből vett üzemgazdasági példa eseményre:

- Szerződést megkötöttek;
- Ajánlat érvénybe lépett;
- A megrendelést visszaigazolták;
- Átutalást előkészítették;
- A vevői reklamációt elutasították;
- Új ügyfél van (pl. egy előzetes adatbázis vizsgálat eredményeként).

Az *esemény* fogalmának meghatározáshoz a köznapi elfogadott értelmezésre támaszkodhatunk, üzemgazdasági körülmények között pedig a gazdálkodástudományban használt értelmezésre.



98. ábra Sematikus példa egyszerű és kibővített eseményvezérelt folyamatláncra

Az esemény fogalmának felfogását azonban ebben a megközelítésben szűkebben is kezelhetjük, mivel az esemény és a funkció között szoros kapcsolat áll fenn. Egy *funkció* végrehajtása egy vagy több újabb eseményhez vezethet (pl. ügy lezárva). Egy esemény tehát egy folyamat kiváltója vagy eredménye. Az eseményt egy kicsit konkrétabban abban a tekintetben lehet megragadni, hogy funkciók végrehajtásának eredményeként vagy azok előfeltételeként lehet leírni.

Vállalati, üzleti esemény

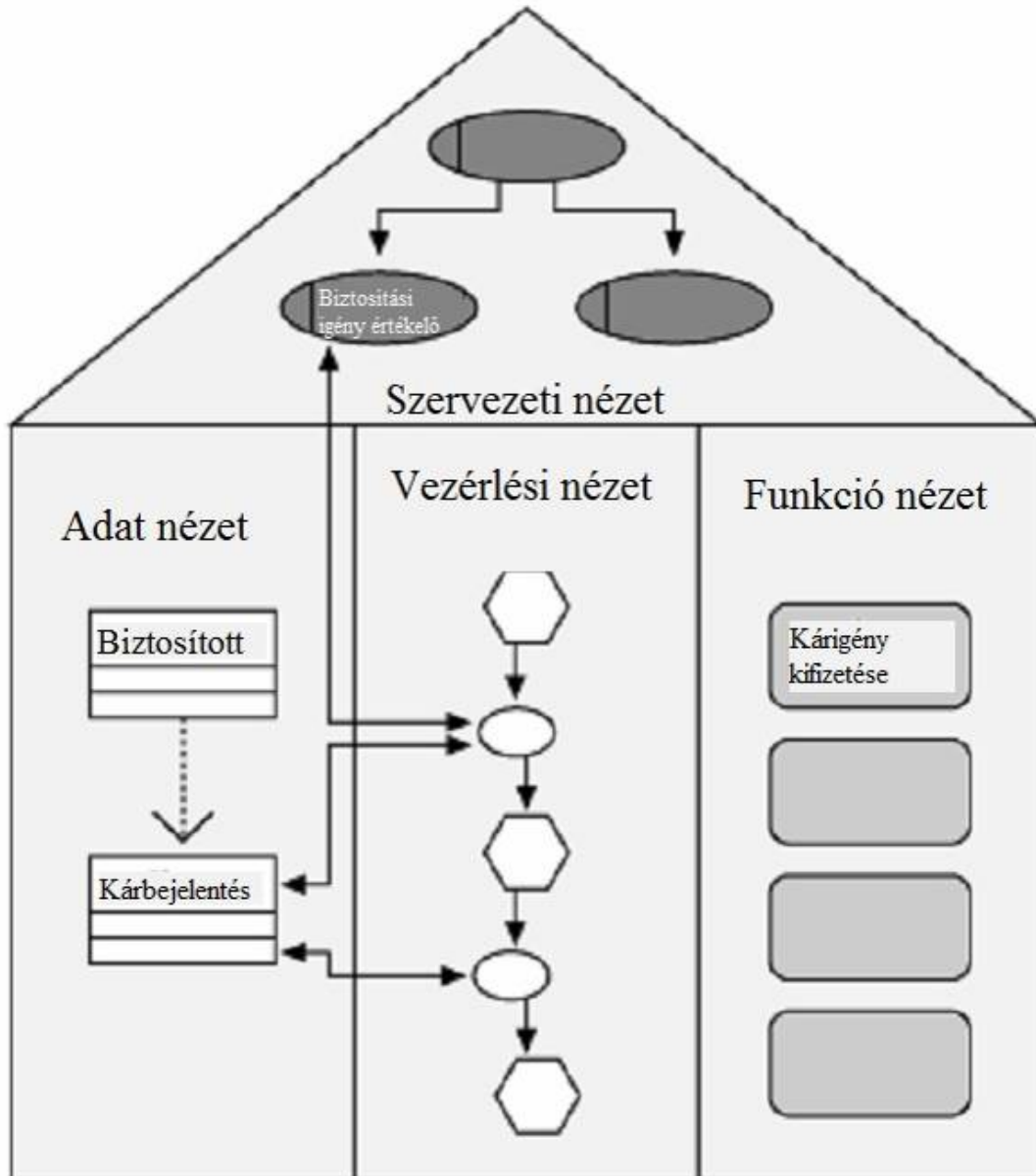
- akkor észlelhető ha egy objektum (pl. gyártási termék) jelenik meg a folyamatban vagy egy meghatározott attribútum jellemzőinek a megváltoztatása történik meg. [103].

Az események egy adott időpontra vonatkoznak, habár ezt általában nem szokták ábrázolásban rögzíteni. A jelenlegi diagram technikában ezt az időpontot csak azáltal érzékeltetik, hogy az esemény a feldolgozási (funkció) folyamatok és egyéb események láncolatában, a folyamat vezérlés egyik ágában helyezkedik el. Az SAP oktatási anyagában egy *esemény* úgy jelenik meg mint „(üzem)gazdasági szempontból fontos állapot, egy adott időpontban és megjelenés egy vagy több funkció végrehajtásának kiváltó oka”.

Ebben az értelemben az esemény a megelőző tevékenységek lehetséges **kimeneteinek**, **eredményeinek** összességéként ragadható meg, amelyre vagy amelyekre a munkafolyamat következő lépésében hivatkoznak vagy – mint derült égből a villámcsapás – egyszer csak a külvilágból megjelennek.

Eseménytér

- Az események összességét, halmazát egy szervezeten (vállalaton, vállalkozáson) belül és a szervezetet körülvevő környezetben a szervezet **eseményterének** nevezzük.



99. ábra Az eseményvezérelt folyamatlánc és azt információrendszer nézetek

Egy *esemény* ugyanakkor mindig egy **előfeltétel** is, amelyet valahogy valakinek ki kell elégítenie. A tevékenységek az eseményből indulnak és legtöbbször az esemény következményei. Ezeket a következményeket és tevékenységeket a *funkció* fogalmában tudjuk megragadni.

Az események a funkciókhoz hasonlóan összefuthatnak illetve szétághatnak. Vegyük például jól ismert gazdasági eseményt: „A vállalat nem ért el semmilyen nyereséget”. Ezt az eseményt meg lehet fogalmazni a következőképpen is : „Az értékesítés x%-al vissza esett”, „A költségek y €-ra rúgnak” stb. Az eseményeket ennek megfelelően a funkciók felbontottsági szintjéhez, lebontási finomságához kell igazítani.

Minden szervezeti (vállalati, üzleti) folyamathoz két esemény tartozik: a kezdő vagy **indító esemény** és a **lezáró esemény**. Az *indító eseménnyel* kezdődik a szervezeti folyamat, ami előtt semmilyen tevékenység a szóban forgó szervezeti folyamatra tekintettel nem történt, a *lezáró eseménnyel* pedig a folyamat befejeződik. Egy világos színű hatszög az esemény jele a diagramban (ld. 97. ábra).

10.7.2 A funkció

A **funkció** egy szervezeti (vállalati, üzleti) folyamatban, mint egy valamilyen szolgáltatást, teljesítményt nyújtó tevékenység ragadható meg. Azonban a modellező diszkrecionális joga, hogy a szóban forgó tevékenységet milyen terjedelemben ábrázolja, mint szervezeti funkciót. A funkciók azt is leírják, hogy mit kell meg tenni. Egy szervezeti folyamat összes olyan tevékenységét, amely valamilyen *szolgáltatást* nyújt a folyamaton belül, egyedi részfeladatokra kell bontani. Ezek a feladatok a munkatársakhoz tartozó mindennapi, üzemi („operatív”) tevékenységeket úgy fogják össze, mintha ez a feladat elem egy információrendszerrel kapcsolatos művelet, tranzakció illetve mint a gazdasági tevékenység egyik szervezeti funkciójának egyik építőeleme volna.

Funkció

- Egy szervezeti *funkció* egy szervezeti (vállalati, üzleti) **eljárásként** fogható fel. Egy ilyen eljárás időigényes történés, amelyet egy indító esemény vált ki és egy lezáró eseménnyel fejeződik be. Egy ilyen eljárás különböző lefolyású lehet az eljárás során előálló eredmények függvényében, elágazások, *visszaugrások* lehetségesek.

Egy *eljárást* a tevékenységek sorozatának lehet tekinteni, amelyet bizonyos feladatok végrehajtása érdekében végeznek el. Azt eljárást csak a szervezeti környezetben szabad értelmezni. A következő négy kategória ad segítséget az eljárás leírásához [28]:

- **Eseményfolyam** az előforduló események függvényében vezérli a feladatok aktiválását;

- **Adat- illetve objektumfolyam** modellezi azokat a bemeneti információkat illetve bemeneti objektumokat, amelyek a feladatok végrehajtásához szükségesek. Továbbá az eredmények felhasználását modellezi a rákövetkező feladatokban. Az objektum fogalma lefedi mind az anyagi mind az immateriális javakat a lehető legszélesebb értelemben;
- **Feladat végrehajtó** a szervezeten belüli munkahelyet, pozíciót reprezentálja és feladat elvégzését;
- **Erőforrások** a feladat elvégzéséhez szükséges anyagok, munkaeszközök vagy a *feladat végrehajtó*.

Ezt az átfogó meghatározást még azzal érdemes kiegészíteni, hogy más *eljárások* illetve *szervezeti folyamatok* egy vagy több *eljárást* tartalmazhatnak.

A *funkciókat* fel lehet bontani illetve egyesíteni lehet. A funkciók lebontásának mindenestre ott be kell fejeződnie, ahol olyan funkcióig jutunk le, amelyek a munka végzéskor egy menetben, a munka lefolyásában egyszeri alkalommal kerülnek feldolgozásra, valamint szervezési vagy gazdálkodási értelemben nincs értelme annak, hogy tovább bontsák.

Az így felfogott funkciók *erőforrásokat* és természetesen időt használnak fel. Ezért az erőforrások mindig valamilyen funkcióhoz kötődnek. Ebben a modellezési megközelítésben a felhasznált idő mennyiségét nem érzékeltetik, sem azt, hogy legalább / legfeljebb mennyi időre van szükség ahhoz, hogy a funkcióba tartozó tevékenységeket végre lehessen hajtani. A funkciót egy lekerített sarkú téglalappal jelölik (ld. 97. ábra).

10.7.3 Szervezeti egység

A szervezeti egység fogalmának segítségével rögzíteni lehet azt, hogy az egy funkcióba tartozó feladatokat a szervezetben hol végzik el. A szervezeti egységeket és funkciókat pedig egymáshoz lehet rendelni.

A mai világban a személyi állomány szintje a vállalkozásoknál és egyéb szervezeteknél nagyon alacsony, a szervezetek a klasszikus szervezeti felépítéstől általában eltávolodtak, a szervezeti egységek kicsik, ezért a szervezeti elemzésnek általában a munkahelyek szintjéig kell lehatolni azért, hogy a szervezeten belüli gyenge pontok feltárásának kimutatásához elegendő kifejező erő társuljon.

Egy ipari vállalat klasszikus szervezeti egységei a következők *Értékesítés, Humán erőforrás, Üzem, Hír és távközlés (informatika)*.



100. ábra Funkció és szervezeti egység összekapcsolása

A funkciók és a szervezeti egységek közötti kapcsolat nem irányított él, és azt jelenti, hogy a funkciót a szóban forgó szervezeti egység munkatársai fogják el intézni. Természetesen egy funkció több szervezeti egységhez kapcsolódhat. A szervezeti egység diagram technikai ábrázolását az ábra mutatja (ld. 97. ábra, 100. ábra).

10.7.4 Információobjektum

Ma semmilyen szervezet nem létezhet olyan adatállományok nélkül, amelyek magát a szervezetet, a szervezet tevékenységének vagy üzleti életének objektumait, továbbá a szervezetet körülvevő környezet objektumait modellezi, a maguk teljességében vagy bizonyos adatnézetekben, metszetekben. Ezeknek az adatállományoknak nagy jelentősége van a szervezeti munkafolyamatok, folyamatok lefolyása tekintetében. Az eseményvezérelt folyamatlánc esetében ez azt jelent, hogy a a funkciókhoz mindenkor a szükséges illetve a létrejövő információkat jelölni kell. Ezek az adatok lehetnek az ügyfél adatai, a korábbi ajánlatok adatai, az aktuális piaci árak vagy egyéb más információk, amelyek a szervezeti folyamat valamelyik szakaszában jelentőséggel bírnak.

Ebben a módszerben az ilyen adatokat információobjektumnak jelölik meg és összekapcsolják azzal a funkcióval, amelyiknek ezekre az adatokra szüksége van. Ezért például a "Megrendelés feldolgozás" funkcióhoz a következő információkat rendelik hozzá:

- Ügyfél;
- Ajánlat;
- Anyagok;
- Szerződési feltételek;
- Ügyfél megrendelés;
- Gyártási anyagszükséglet.

Az információobjektumok és a funkciók közti kapcsolat irányított élekkel ábrázolható. A funkcióknak szüksége van információkra az információobjektumokból, miközben olyan információkat állítanak elő a funkciók, amelyek viszont az információobjektumokhoz kerülnek. A grafikus megjelenítése az információobjektumoknak látható az ábrán (ld. 97. ábra). A funkciók és az információobjektumok közötti kapcsolatot nyilakkal jelölik, a nyilak irányítása visszatükrözi az ábrázolandó kapcsolat miatti tényleges információáramlást. Ha az információobjektumból történik információáramlás a funkcióba, akkor a nyílhegye a funkcióra mutat, ha a funkció állít elő információt, amely az információobjektumba áramlik be akkor a nyílhegye az információobjektumra mutat.



101. ábra Információáramlás a funkcióba

10.7.5 Logikai műveletek és vezérlés

A szervezeti munkafolyamatokban, a folyamatokban gyakran kell bizonyos tevékenységeket párhuzamosan végrehajtani azért, hogy valamilyen célt elérjenek. Bizonyos esetekben pedig alternatív lehetőségek vannak: vagy az egyik vagy a másik tevékenység vezet el a célhoz. Az eseményeknél hasonló a helyzet, alternatív események ugyanahhoz a tevékenységhez vezethetnek vagy pedig több esemény összessége jelenti azt a feltételt, amelynek alapján egy tevékenység elkezdődhet.

Ezek a strukturális jellemzők a mindenkori tevékenységeket és eseményeket egy bizonyos kapcsolatrendszerbe rendezi, az *eseményvezérelt folyamatláncban* pedig a funkciókat illetve az eseményeket.

Az *eseményvezérelt folyamatláncban* ezt a „kapcsolatrendszerbe rendezést” a logikai műveletek segítségével lehet modellezni úgy, ahogy formális nyelvekben az szokásos: a **kizáró vagy (XOR)**, a **vagy (OR)**, az **és (AND)**. Az alkalmazott grafikus szimbólumok az ábrán láthatók (ld. 97. ábra).

Események esetében a műveletek alkalmazhatóságának előfeltétele az, hogy legalább két esemény legyen, ekkor a jelentésük:

- **ÉS (AND) ()** : Az összes megelőző eseménynek be kell következnie, csak akkor folytatódik a folyamat és adja a vezérlést a következő lépésre;
- **VAGY (OR)**: legalább az egyik megelőző eseménynek be kell következnie, csak akkor folytatódik a folyamat és adja a vezérlést a következő lépésre;
- **KIZÁRÓ VAGY (XOR)**: pontosan egy eseménynek kell bekövetkeznie, csak akkor folytatódik a folyamat és adja a vezérlést a következő lépésre.

Egy eseményre vonatkoztatva egyébként az operátoroknak nincs értelmes jelentése (matematikai logikailag: ezek nem unáris hanem bináris operátorok).

Funkciók esetébe is a műveletek alkalmazhatóságának előfeltétele az, hogy legalább két funkció legyen, ekkor a jelentésük:

- **ÉS (AND) ()** : Az összes megelőző funkciónak végre kell hajtódnia, csak akkor folytatódik a folyamat és adja a vezérlést a következő lépésre;
- **VAGY (OR)**: legalább az egyik megelőző funkciónak végre kell hajtódnia, csak akkor folytatódik a folyamat és adja a vezérlést a következő lépésre;

- **KIZÁRÓ VAGY (XOR):** pontosan egy megelőző funkciónak kell végrehajtódnia, csak akkor folytatódik a folyamat és adja a vezérlést a következő lépésre.

A műveletek pontosabban jelentésének megértéséhez a vezérlési folyamat részleteinek ismerete szükséges. Pl. a művelet jelentése attól függhet, hogy vajon a hozzá kapcsolódó esemény egy funkció előtt vagy utána áll-e.

Szabály a műveletekre

- A logikai műveletek csak vagy eseményekre vagy csak funkciókra vonatkozhatnak. A műveletek paraméterei tehát csak homogének lehetnek, vegyesen nem fordulhatnak elő események és funkciók a műveletek paraméterei között..

Az eseményvezérelt folyamatláncban az áttekinthetőség végett a vezérlés folyamatát felülről lefelé haladva ábrázolják. Ezért a műveleteket a diagram jelölés technikában egy kettéosztott körlappal érzékeltetik, az alábbi módon:



A körlap felső részében jelölt művelet a körlap felsőrészébe belépő vezérlési folyamatra vonatkozik és azt határozza meg, hogy azok az események vagy funkciók, amelyekre vonatkozik a vezérlési folyam, hogyan kapcsolódnak egymáshoz, milyen a műveleti értelemben vett viszony közöttük. A körlap alsó része ugyanezt nyújtja a belőle kilépő nyilakkal ábrázolt vezérlési folyamra. Természetesen csak akkor kell jelölni a körlap egyik felében a műveleteket, ha a vonatkozó vezérlési folyamában legalább két elem (esemény vagy funkció) érintett. Ha csak egy nyíl érkezik vagy távozik a körlapból, akkor azon az oldalon nem kell jelölni a műveletet.

A logikai műveletekhez még két további fogalom kapcsolódik: az **elágazás** és a **csatlakozás**. Ha egy vezérlési ág lép be és utána több lép ki, akkor *elágazásról* beszélünk, ha több ág lép be és csak egy távozik, akkor *csatlakozásról* beszélünk.

10.8 Szervezeti folyamat modellezés jelölésrendszere- Business Process Modeling Notation (BPMN)

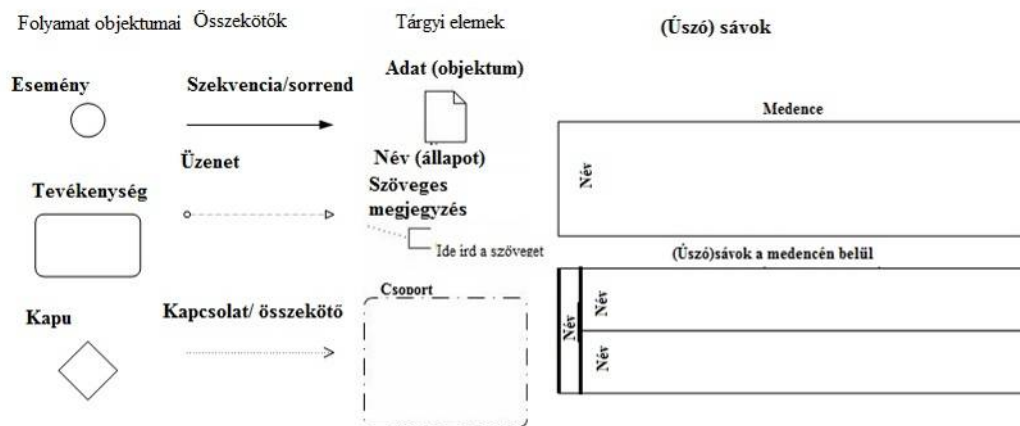
A szervezeti (vállalati, üzleti) folyamatok modellezésére a BPMN-t (Business Process Modeling Notation) először egy IBM alkalmazott (Stephen A. White) fejlesztette ki a szervezeti folyamatok diagram technikai, grafikus ábrázolására. A módszer publikálása után elindult

egy ipari „kezdeményezés” a „Business Process Management Initiative – BPMI” , egy egységesítés, szabványosítás irányába. Ezt követte az Object Management Group (OMG, www.omg.org) által vezetett, a jelentős ipari szereplőkből álló konzorcium létrejötte, amely érdekelt volt a jelölésrendszer továbbfejlesztésben egy gyártó illetve módszer tulajdonos független szabvány irányába. A BPMN diagramtechnika központi eleme a **szervezeti folyamat diagram, BPD** (Business Process Diagramm), mely többek között (úszó) sávokat tartalmaz az ábrázolás technikájában.

A kezdeményezés célja az volt, hogy egy könnyen érthető jelölési rendszert készítsenek az a szervezetke szereplőinek, az üzleti élet minden érintettjének, kezdve az szervezeti/üzleti elemzőkkel (Business Analyst), akik a folyamatok kezdeti tervezetét készítik el, amire alapozva a folyamat fejlesztők ezeknek a folyamatoknak a mind szervezeti mint informatikai bevezetésére képesek, és végül az üzletembereknek, akik ezeket a folyamatokat irányítani, ellenőrizni fogják. A BPMN-nek egy másik rendeltetése az, hogy teljesen leképezhető legyen a BPEL4WS nyelvre, amely a *Web szolgáltatások* szervezeti folyamatainak programnyelve (Business Process Execution Language for Web Services). Ez a leképezés valamilyen kapcsolaton, kapcsolófelületen („*interface*”) történhet, vagy a mai korszerű eszközökben valamilyen közvetlenül végrehajtható program kód előállításán keresztül. A BPEL4WS és hasonló XML alapú nyelvekben írt program kódok, amelyeket szervezeti folyamatok végrehajtására terveztek vizuálisan megjeleníthetők (részben vagy egészben) BPMN diagramban . Ezért a BPMN-t tekinthetjük egy olyan megoldásnak, amely hidat kíván építeni az szervezeti folyamatok tervezése, bevezetése és az informatikai megvalósítások technológiai részletei között.

A BPMN alapvetően egy vizuális nyelv, amiben az egyes diagramtechnikai szimbólumok kulcsszerepet játszanak, ezért a választott alakzatokat, ikonokat minden folyamatmodellezőnek ismernie és értenie kell. A folyamatábrák diagram technikai elemekből állnak, melyek alaptípusain változtatni nem szabad, és az esetleges bekerülő új objektumok sem nem kerülhetnek összeütközésbe a korábbi szabványos elemekkel, sem nem sérthetik a formai és tartalmi szabályokat (szintaktika, szemantika). Ezek az elemek könnyen megkülönböztethetők egymástól, és ezekhez az alakzatokhoz hasonlókat használ a legtöbb modellező. Például a tevékenységek téglalapok, a döntések csúcsra állított négyzet, vagy rombusz alakúak. Az alakzatok alapértelmezett formája nem változtatható, de szabadon színezhető.

Az alapvető folyamatmodellezési objektumok különálló kategóriákba tartoznak. Ezek a csoportok jól elkülönítik a hasonló funkciójú objektumokat, és így segítséget adnak a folyamatábra olvasójának, hogy könnyebben felismerje az alapvető objektum típusokat, és megértse az ábrát. [117].



102. ábra A BPMN alap diagram technikai elemei

A négy alapvető jelölési kategória a következő:

- A folyamat objektumai (esemény, tevékenység, (logikai) kapu); (flow objects, Events, Activities, Gateways);
- Összekötők (szekvencia folyam/sorrend, üzenetfolyam, asszociáció/kapcsolat/összekötő); (Sequence Flow, Message Flow, Association);
- Úszósávok (medence, sáv); (Pools, Lanes)
- Tárgyi elemek (adatobjektum, csoport, megjegyzés); (Artifacts, Data object, Group, annotation).

10.8.1 Folyamat objektumai

A folyamat objektumai csomópontok folyamatábrában, ezek az elemek mint fő grafikai elemek definiálják az üzleti folyamat viselkedését. Három fajtájuk van: az **esemény**, a **tevékenység** és a (logikai) kapu.

10.8.1.1 Esemény

Egy *eseményt* egy üres vagy egy szimbólumot tartalmazó kör jelöl. Az esemény valamilyen történést jelez, befolyásolja a folyamat lefolyását és valamilyen kiváltó oka („trigger”) vagy

eredménye („impact”) van. Az eseményeket megkülönböztethetjük a pozíciójuk (Start, Intermediate, End), a hatásuk (Catching, Throwing), és a típusuk (Unmarked, Timer, Error, Cancel, Compensation, Conditional, Signal, Multiple, Link, Message, Terminate³¹) alapján.

A „Start” esemény a folyamat elejét, az „End” esemény a folyamat végét jelöli. Minden folyamatnak tartalmaznia kell Start és End eseményt. Az „Intermediate Catching” esemény a folyamat közben történik, és a folyamat addig áll ezen a ponton, ameddig az adott esemény be nem következik. Az „Intermediate Throwing” esemény a folyamat közben történik, és egyből bekövetkezik, ahogy a folyamatban ehhez a ponthoz érünk. Más értelmezésben, a „Start” és „Catching” elemek indító jelleggel rendelkeznek, míg a „Throwing” és az „End” egy folyamat vagy egy tevékenységszál (akár csak átmeneti) végét jelölik.

Az ábrán (103. ábra) az eseményekre vonatkozó szimbólumokat látjuk csoportosítva, ezek közül néhányat részletesebben bemutatunk azért, hogy szemléltessük hogyan is működnek az események. A BPMN diagram technika és módszer részletesebb szabályrendszere megtalálható [117], [32] publikációkban. A „Message Start”-ot akkor használjuk, ha a folyamatot egy másik szereplőtől kapott üzenet indítja el. A „Message Catching” azt ábrázolja, hogy a folyamat addig várakozik, amíg az adott üzenet meg nem érkezik. Itt jól látszik tehát, hogy a „Start” és „Catching” jelentése hasonló: a folyamat akkor indulhat, ha egy üzenet érkezik. A „Message Throwing” azt mutatja, hogy a folyamatnak ehhez a pontjához érve egy üzenet küldése történik meg, míg a „Message End” a folyamat végét egy üzenetküldéssel jelzi.

	Start		Intermediate		End	
	Catching	Throwing	Catching	Throwing		
Unmarked						Unmarked events
Timer						Timer events
Error						Catching or throwing an error
Cancel						Cancelled transactions or triggering cancellation
Compensation						Compensation handling or triggering
Conditional						Business conditions or business rules
Signal						Signalling across different processes
Multiple						Catching or throwing one out of a set of events
Link						Off page connectors
Message						Receiving and sending messages
Terminate						Terminate the process

103. ábra BPMN-ben az események típusai [117]

A „Timer” időpont, dátum, időtartam, és időközök jelölésére szolgál, a „Timer Start” a folyamat elindulását adott időhöz köti. A Timer Catching arra utal, hogy a folyamat addig várakozik, amíg az adott idő be nem következik.

A „Conditional Start”-tal tetszőleges feltételt szabhatunk a folyamat elindulásának és a folyamat csak a megadott feltétel teljesülésekor kezdődik. A „Conditional Catching” használatkor a folyamat addig várakozik, amíg az adott feltétel nem teljesül.

Az ábrán (103. ábra) látható az, hogy nem mindegyik eseménytípusnál kerül értelmezésre az összes lehetséges esemény kombináció. Nincs például értelme megszakítással kezdeni egy folyamatot, ezért nincs is „Terminate Start” esemény.

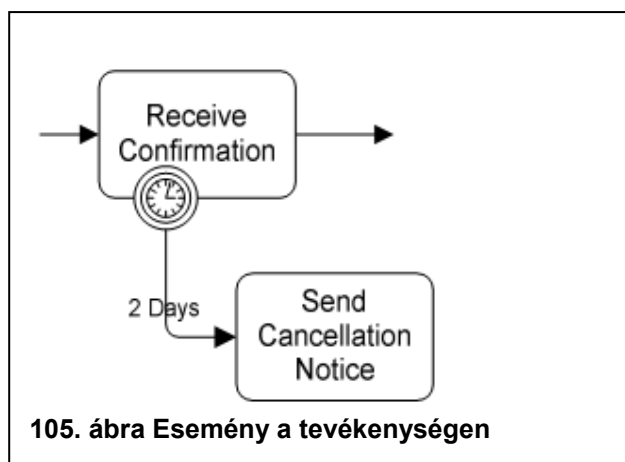
10.8.1.2 Tevékenységek

Egy *tevékenység* egy olyan feladatot ír le, amit a szervezeti folyamat végrehajtása során végeznek el. Lekerekített csúcsú téglalapokkal ábrázolják a tevékenységeket. Ha a téglalpnak az alja közepén egy pluszjel helyezkedik el, akkor azt al-, vagy részfolyamatnak nevezik, vagy összevont részfolyamatnak. (104. ábra) Az ilyen folyamatokat beágyazhatjuk a bonyolultabb tevékenységekbe, a részfolyamatok részletei nem látszódnak a szóban forgó diagramon, a részleteket egy alá rendelt ábrán fejtik ki. A folyamatok lehetnek ciklikusak, azaz egy bizonyos eredmény eléréséig ismételhetők.



104. ábra Kibontott részfolyamat, összevont folyamat (collapsed subprocess) és feladat (task)

A tevékenységeket és az eseményeket lehet kombinálni is, ez azt jelenti, hogyha egy ese-



105. ábra Esemény a tevékenységen

ményt egy tevékenységre helyezünk, akkor azzal befolyásoljuk bekövetkezést. Például, ha egy „Message Catchinget” helyezünk el egy Tevékenységen, akkor az azt jelöli, hogy a Tevékenység végrehajtása addig folyik, amíg az adott üzenet meg nem érkezik. Vagy ha egy „Timer Catchinget” tesszünk a Tevékenységre, akkor az azt jelöli,

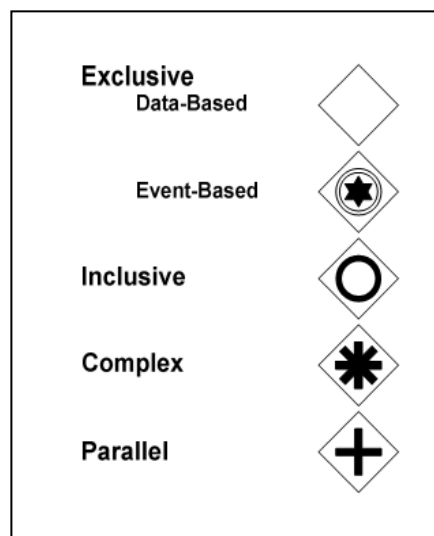
hogy a Tevékenység végrehajtása addig folyik, amíg az adott idő be nem következik. Az ábrán (105. ábra) egy olyan példa látható, amikor valamilyen megrendelés, pl. hotel szobafoglalás után megerősítést várnak, ha megerősítési szándék nem érkezik meg, akkor 2 nap elteltével elküldik a foglalás törléséről az értesítést.

10.8.1.3 Logikai kapuk

A folyamat objektumai közül az utolsó típus a **logikai kapu**. A *kaput* csúcsára állított négyzet (a kártya „káró” szimbólum) jelöli, aminek a közepén különböző alakzatok helyezkedhetnek el. A kaput döntések ábrázolására, különböző útvonalak összefuttatására, és elágazására

használjuk. A döntés sohasem a kapuban történik, hanem egy műveletben előtte, a kapu csak azt ábrázolja, hogy a döntési művelet után hányfelé és milyen módon folytatódhat a folyamat. A következő kapu típusokat különböztetjük meg ([32]):

- **Exclusive Gateway** (kizáró vagy kapu): a szervezeti folyamaton belül egy olyan hely ahol a lépések, tevékenységek sorozata, szekvenciája („Sequence Flow”) két vagy több alternatív útvonalon folytatódhat. Ez a helyzet teljesen hasonló az országúton az elágazáshoz érkezéshez. A szervezeti folyamat a végrehajtása során csak az egyik úton folytatódhat tovább. Két döntési mechanizmus létezik:

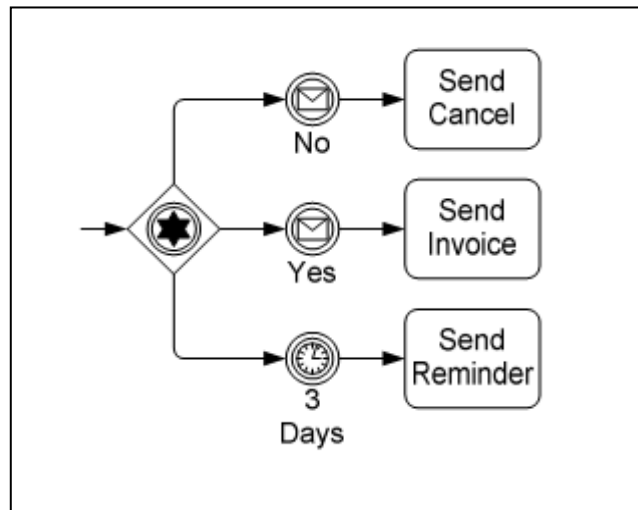


106. ábra A logikai kapuk típusai

- **Exclusive (Data-based):** Kizáró „VAGY”(XOR) adat alapú kizáró döntés, tehát a folyamat csak és kizárólag egy irányba mehet tovább valamilyen feltétel (adat kifejezés) kiértékelése után. Ez a típus általános kapuként is ismert, használható a négyzet belsejébe rajzolt jellel és a nélkül is.
- **Exclusive (Event-based):** Kizáró „VAGY”(XOR) esemény alapú kizáró döntés. A folyamat csak egy irányba mehet tovább, és a döntést valamilyen esemény bekövetkezéséhez kötik, aminek észlelése valamilyen üzenet megérkezése alapján történik. Ez a döntési típus a szervezeti folyamat egy olyan elágazását ábrázolja ahol az alternatív lehetőségek valamilyen eseményeken alapulnak, és nem megfogalmazott feltételrendszeren. Az ismétlődő közbenső esemény (Multiple Intermediate Event) szimbólumát használják ennek a kapunak a jelölésére. A kapu káró szimbóluma utáni esemény határozza meg a válasz-

tandó útvonalat, Az első bekövetkező esemény nyer („tüzel”, trigger), és a folyamat azt az útvonalat követi, amelyen ennek az eseménynek a szimbóluma szerepel

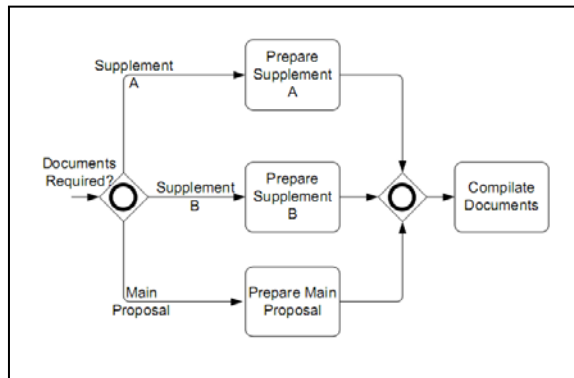
- **Csatlakozás (merging, join)** : Arra is használható ez a kapu, hogy az alternatív útvonalakból érkező tevékenységek sorozatát, szekvenciáját („Sequence Flow”) egy útvonalba fogják össze.³²



107. ábra Esemény alapú

- **Parallel**: A párhuzamosság kaput akkor lehet használni, amikor a folyamatban több egymással párhuzamos útvonal is létezik. Általában, a legtöbb helyzetben az elágazási pontoknál nem megkövetelt az alkalmazása. Főként módszertani, érthetőségi okok miatt célszerű használni. A káron belüli „+” jelzi ezt a kapu típust. Ezt a logikai kaput lehet a párhuzamos útvonalak szinkronizálására használni.
- **Inclusive**: Megengedő „VAGY”, tehát a folyamat egy vagy több irányba mehet tovább. Általában valahol a párja, egy másik VAGY kapu következik, amelyik az alternatív útvonalakat csatlakoztatja, összezsacornázza. A káró belsejében egy „O” van az „OR”-ra utalva.
- **Complex**: Olyan döntési helyzetet jelent, ahol bonyolult feltételek kombinációját lehet megfogalmazni. A káró szimbólumon belül egy csillag jelöli ezt az kaput. A szervezeti folyamat bonyolult viselkedési mintázatát lehet leírni mind az útvonalak elágazásánál mind a csatlakozásnál.

A logikai kapuk nagyon fontosak a folyamatban, a valóságos környezet leképezése érdekében, a tényleges szervezeti igények visszatükrözése végett.. A kapuk, a döntési helyzetek leírhatósága nélkül a folyamatok csak egyszerű, automatikusan végrehajtható folyamatok volnának..



108. ábra Elágazó és csatlakozó VAGY kapuk

10.8.1.4 Összekötők

A folyamatábrákban az **összekötők** kapcsolják össze a folyamat objektumait. Ezt nevezhetjük a folyamatdiagram csontvázának. Három típusa van, a szekvencia és üzenet, illetve az kapcsolat/összekötő/asszociáció.

10.8.1.5 Szekvencia

A tevékenységek végrehajtási sorrendjét jelöli. A jelölése egy egyszerű nyíl. A következő diagram elemek szerepelhetnek kezdő illetve végpontként: esemény, tevékenység és kapu. A kapu után megkülönböztethetjük az alapértelmezett és feltételekhez kötött lehetőségeket. A feltételes útvonalat a nyíl tövében elhelyezett rombuszsal vagy vonallal érzékeltetjük. Ezek futásidőben értékelődnek ki, és a döntéseknek vagy feltételeknek megfelelő ágra terelik a folyamatot. A feltételnek igaznak kell lennie ahhoz, hogy a lépés sorozat folytatódjon. Legalább az egyik úton folytatódnia kell a folyamatnak. A szekvencia nyíl nem lépheti át rész folyamat, vagy medence határát.

10.8.1.6 Üzenetfolyam

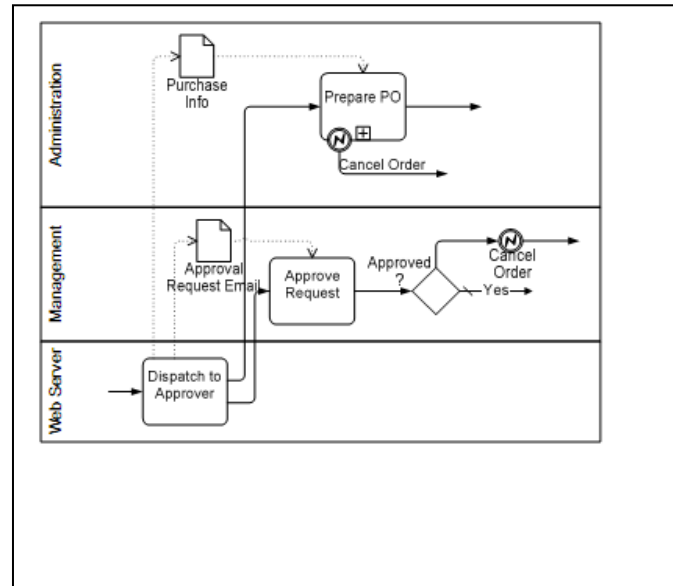
Az üzenet útját mutatja a *szervezeti folyamat* két *résztevője* (Participant) között, szaggatott nyíl a jele. A BPMN-ben a résztvevőket (szereplőket) külön-külön medence („Pool”) jelöli. Egy üzenetet ábrázoló nyíl hozzákapcsolódhat egy medence határvonalához, vagy a medencén belüli objektumhoz. Azonban az üzenet nyíl nem köthet össze két objektumot a medencén belül.

10.8.1.7 Kapcsolat, összekötő, asszociáció

Arra használják, hogy adatokat, szövegeket és más tárgyi elemeket („Artifacts”) kössenek össze a folyamat objektumaival (tevékenység stb.). Pontozott nyíl a jele, amely irányával

megmutatja, hogy a tevékenységekbe honnak kerülnek be a bemeneti adatok és hová kerülnek ki a kimeneti adatok. Az *asszociációkhoz szöveges megjegyzések* kapcsolhatók.

10.8.1.8 Úszósávok



109. ábra Sávokon belüli tevékenységek kapcsolatai [117]

Az **úszósáv** („swimlane”) fogalma a tevékenységek szervezését illetve felbontását támogatja. Az *úszósávok* két típusát különböztetik meg: a **medencét** („Pool”) és a **sávot** („Lane”). A medence a szervezeti folyamat diagram (BPD, BusinessProcess Diagram) szervezet-szervezet, vagy üzleti vállalkozás-üzleti vállalkozás (B2B, Business-to-Business) közötti kapcsolatot leíró helyzetben a kapcsolatot *résztevőit* tudja érzékeltetni. A sáv pedig a medencén belül elhelyezkedő objektumok csoportosítását, célszerű partíciókra bontását tudja megjeleníteni (102. ábra A BPMN alap diagram technikai elemei).

Egy olyan interaktív (B2B BPD) szervezeti folyamat diagramban, amely üzleti vállalkozás-üzleti vállalkozás kapcsolatát írja le, a résztvevők vagy az üzleti élet szereplő lehetnek (pl. megrendelő vagy eladó) illetve az üzleti élet valamilyen szereplője, jogalany lehet (pl. IBM vagy OMG). A medencét gyakran „fekete doboznak” foghatjuk fel és kezelhetjük, de tartalmazhat *folyamatot* is. A medencék közötti kölcsönhatást, üzenetek cseréjén keresztül lehet jelezni. A sorrendiséget érzékeltető szekvencia nyíl nem lépheti át a medence határát (vagyis egy folyamatnak teljes egészében a medencében kell maradnia).

A *sáv* a medencén belül levő objektumok csoportosítására szolgál, gyakran a szervezet igazgatási, vezetési szerepeit jelölik a sávokkal, de bármilyen folyamat jellemző ábrázolására felhasználható. A szekvencia nyíl átlépheti a sávok határát.

10.8.1.9 Tárgyi elemek (Artifacts)

A **tárgyi elemek** a szervezeti folyamat alap folyamat ábra szerkezet mellett további információkat jelenítenek meg. A legáltalánosabb tárgyi elem („Artifact”) az **adat** (dokumentum) objektum, amit asszociációval kapcsolunk a tevékenységhez, és azt szimbolizálja, hogy milyen adat szükséges vagy jön létre a tevékenység során. Az adat objektumnak lehet „állapota” („state”), amely azt mutatja meg, hogy a szervezeti folyamaton belül, hogyan történik meg az aktualizálása vagy megváltoztatása.

A **csoport** elem logikailag kapcsolódó objektumokat rendez össze egy csoportba - elemzési vagy dokumentációs célból – anélkül, hogy a szervezeti folyamat teljesítményét további peremfeltételekkel rontaná. – ahogy azt pl. a részfolyamat teszi. A csoportosításnak a szekvencia folyamam működésére nincs hatása, és jele szaggatott vonalú téglalap. A csoportosításnak nem jelent akadályt sem a medence sem a sáv, a határaikat át metszheti a csoport.

A megjegyzés („Annotation”) további lehetőséget nyújt információ, vagy magyarázó jegyzetek használatára, nagy előnye, hogy bárhol lehet használni a folyamatmodellen belül. A szervezeti folyamat modell bármelyik eleméhez hozzá lehet kötni egy asszociációval, kapcsolattal.

A BPMN rugalmasságát növelendő lehetővé tették azt, hogy a modellezők által definiált saját típusú modellezési objektumokkal is bővíteni lehessen a diagram technikai jelöléskészlet alaphalmazát („Artifacts”). A szabvány nem sérül a hozzáadott elemektől, viszont az szervezeti folyamat leírása és áttekinthetősége így válik teljessé.

BPMN 2.0 - Business Process Model and Notation

Activities

Task
A Task is a unit of work, the job to be performed. When marked with a symbol it indicates a Sub-Process, an activity that can be refined.

Transaction
A Transaction is a set of activities that logically follow a single start event and end with a specified transaction protocol.

Call Activity
An Event Sub-Process is placed into a Process or Sub-Process. It is activated when its start event occurs in the context of the start event (more interrupting) depending on the start event.

Call Activity
A Call Activity is a wrapper for a globally defined process or task that is reused in the current process.

- ### Activity Markers
- Markers indicate execution behavior of activities:
- Sub-Process Marker
 - Loop Marker
 - Parallel MI Marker
 - Sequential MI Marker
 - Ad-Hoc Marker
 - Compensation Marker
- ### Task Types
- Types specify the nature of the action to be performed:
- Send Task
 - Receive Task
 - User Task
 - Manual Task
 - Business Rule Task
 - Service Task
 - Script Task

Sequence Flow
defines the execution order of activities.

Default Flow
is the default branch when other conditions evaluate to false.

Conditional Flow
has a condition which determines whether or not the flow is used.

Gateways

Exclusive Gateway
When splitting, it routes the sequence flow to exactly one of the outgoing branches. When merging, it waits until all incoming branches to complete before triggering the outgoing flow.

Event-based Gateway
Sequence flow is routed to the subsequent event/task which happens first.

Parallel Gateway
When used to split the sequence flow, all outgoing parallel branches wait for all incoming branches to complete before triggering the outgoing flow.

Inclusive Gateway
When splitting, two or more branches are activated. All active incoming branches must complete before merging.

Complex Gateway
Complex merging or splitting. The sequence of all subsequent events starts a new process instance.

Parallel Event-based Gateway
Each occurrence of a subsequent event starts a new process instance.

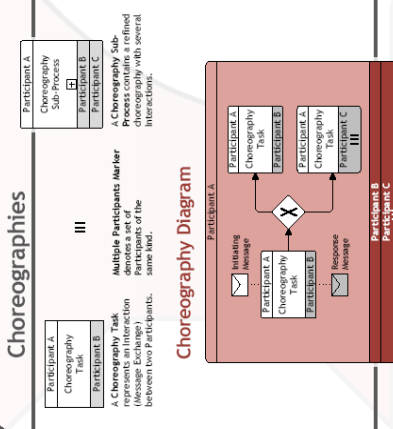
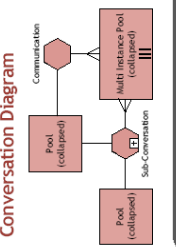
Parallel Event-based Gateway (Parallelization)
The occurrence of all subsequent events starts a new process instance.

Conversations

A Conversation defines a set of locally related message exchanges. When marked with a symbol it indicates a Sub-Conversation, a composing conversation element.

A Conversation Link connects Conversations and Participants.

A Filled Conversation Link connects Conversations and multiple Participants.



Events

	Start	Intermediate	End
None: Untyped events, indicate start point, state change, or trigger.			
Message: Receiving and sending messages.			
Timer: Clock time events, errors, time spans or timeouts.			
Escalation: Escalating to a higher level of process.			
Conditional: Reacting to changed business conditions or integrating business rules.			
Link: Off-page connectors, events in sequence flows.			
Error: Catching or throwing named errors.			
Cancel: Reacting to cancelled transactions or triggering cancellation.			
Compensation: Handling or triggering compensation.			
Signal: Signalling across different processes. A signal thrown can be caught multiple times, and a signal can trigger a set of events. Throwing all events defined.			
Parallel Multiple: Catching the occurrence of a set of parallel events.			
Terminate: Triggering the immediate termination of a process.			

Data

Data Input
A Data Input is an external input for the entire process. It can be read by an activity.

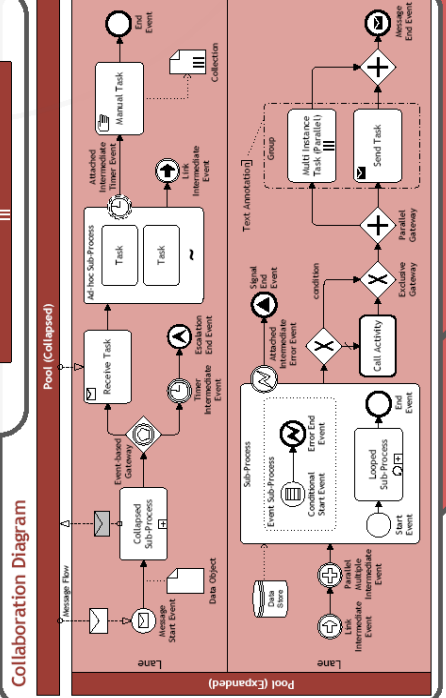
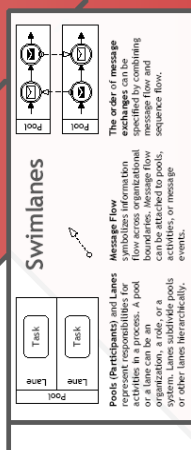
Data Output
A Data Output is a variable available as result of the entire process.

Data Object
Data Object represents information flowing through the process, such as business documents, e-mails, or letters.

Collection Data Object
A Collection Data Object represents a collection of information, e.g., a list of order items.

Data Store
A Data Store is a place where the process can store information, e.g., as a variable being called. It persists beyond the lifetime of the process instance.

Message
A Message is used to depict the contents of a communication between two Participants.

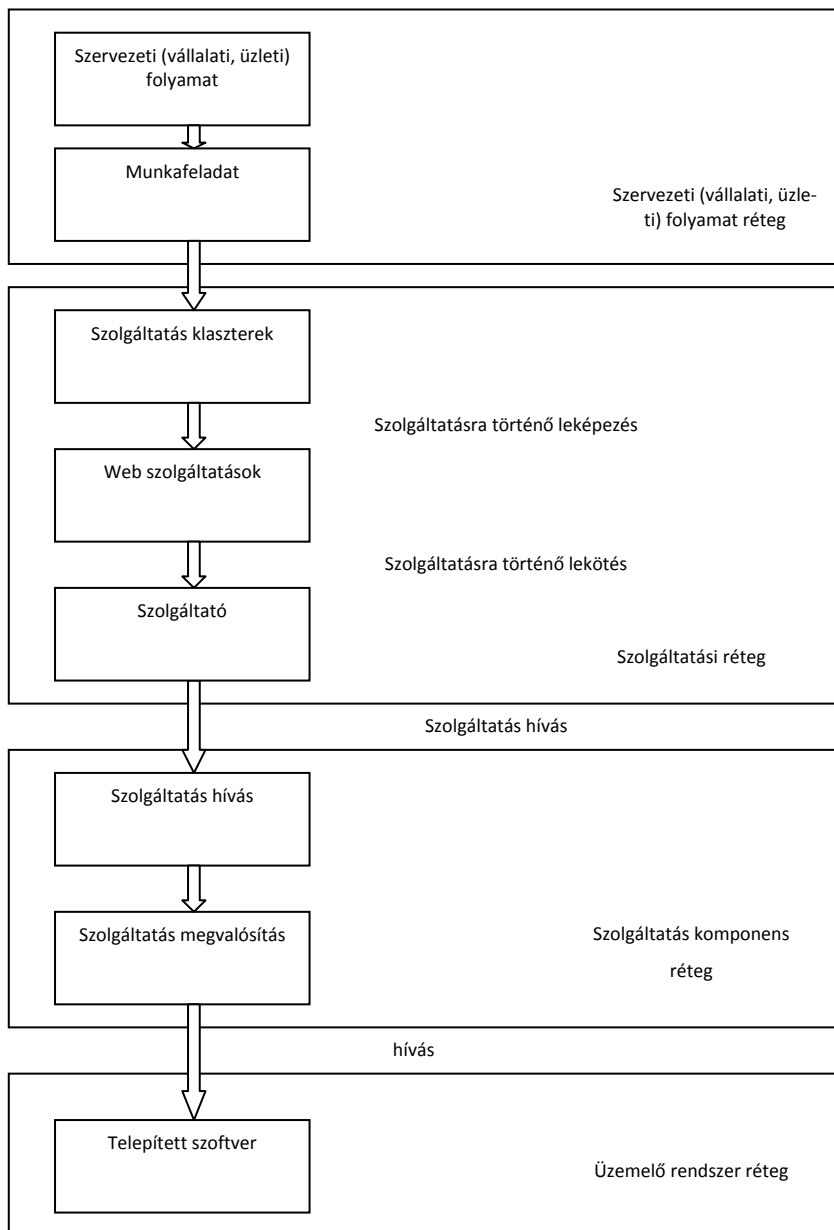


110. ábra A BPMN jelölésrendszer összefoglalása

10.9 SOA alapú BPM

A SOA alapú vállalati folyam modellezés célja, hogy a potenciális szolgáltatásokat azonosítsa, figyelembe véve azt, hogy a szolgáltatások között magas legyen a **kohézió** és a **függés** köztük kicsi legyen (laza csatolás). Ezáltal magas legyen újrafelhasználhatóság esélye,

Két stratégián keresztül modellezhetünk: felülről-lefele/felülnézetből (top-down), vagy alulról-felfelé/alulnézetből (bottom-up). A felülről-lefele módszer a vállalati *folyamatokat* lebontja *munkafeladatokká*, amíg nem találunk rá létező szolgáltatást, amely elvégzi azt. Az alulról-felfele módszer pedig létező komponensekből építi fel a vállalati folyamatokat.



111. ábra Folyamatszervezés SzOA segítségével felülnézetből

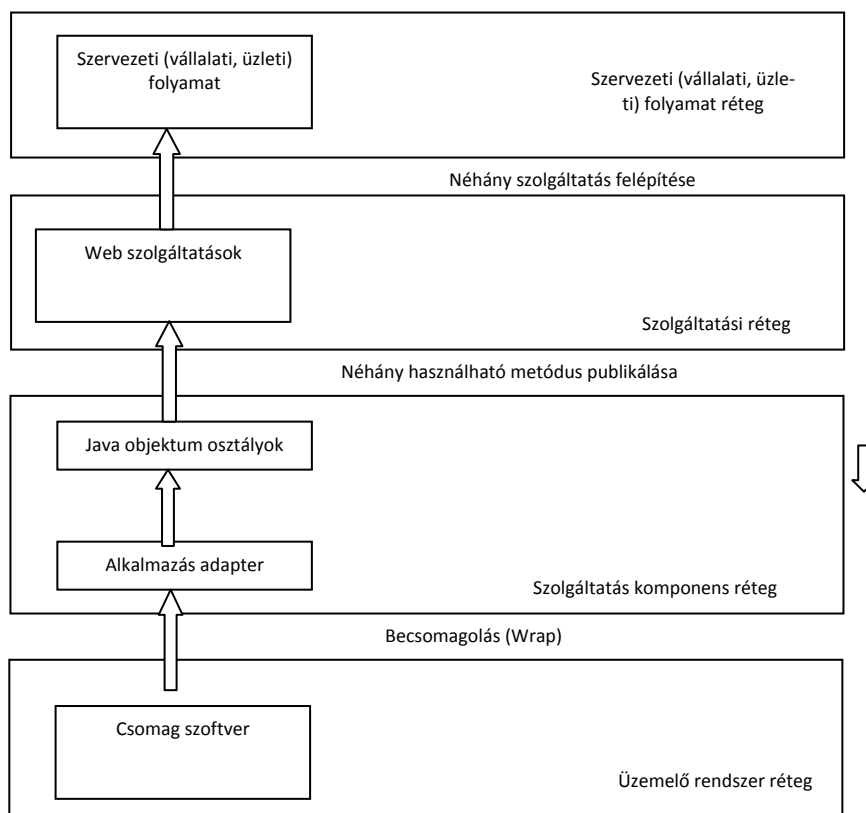
10.9.1 Felülről-lefele BPM

A felülről-lefele módszer a vállalati folyamatok rekurzív dekomponálásán, lebontásán alapul. Ennek eredményeképp részfolyamatokat vagy feladatokat kapunk. Addig bontjuk le a folyamatot, amíg nem lesz minden feladat „menedzselhető”, azaz már olyan egyszerű a feladat, hogy könnyen tudunk rá létező megoldást (akár szolgáltatást) adni.

Ennek a módszernek két előnye van. A vállalati folyamatok jól meg lehet szervezni, magas lesz az újrafelhasználhatóság. Másrészt ez illeszkedik legjobban a vállalati követelményekhez. Hátránya, hogy a dekompozíció, a folyamatok részekre bontása, a szolgáltatás transzformáció túl sok erőfeszítést igényel, így a vállalati célok és a létező szolgáltatások közti eltérés problémát okoz.

10.9.2 Alulról-felfele BPM

Alulról-felfele módszerrel a létező komponenseket építjük egy komplex vállalati folyamattá.



112. ábra Alulnézetből folyamatszerzés a SzOA segítségével

Két előnye van: a létező szolgáltatások vállalati folyamattá aggregálása a magas újrafelhasználhatóság miatt hasznos. Másrészt a validáció – a rendszer működés helyességének ellenőrzése – sokkal olcsóbb lesz, csak az integrációs teszttel kell többet foglalkozni. Hátránya, hogy az egész rendszerről nem kapunk átfogó képet.

10.9.3 A vállalati folyamatok és az IT implementáció közti rés áthidalása

10.9.3.1 BPM vállalati rendszerelemzésből

A BPM általában vállalati elemzőktől indul, akik megértik a vállalati folyamat forgatókönyvét és követelményeit. Számos modellező eszköz áll rendelkezésükre hogy a vállalati folyamatokat dokumentálják. Ezeket fel lehet használni, hogy más vállalati elemzőkkel tudjanak konzultálni.

10.9.3.2 Vállalati folyamat újratervezés SzOA-ban

A vállalati modellek és az informatikai fejlesztői modellek között eltérést találhatunk. Ennek oka, hogy a vállalati elemzőkről nem feltételezhetünk informatikai ismereteket. Például az elnevezési konvenciókat nem ismerhetik egy adott programozási nyelv esetében (pl. BPEL). Ennek következtében a vállalati modelleket közvetlenül nem tudják felhasználni a szoftverfejlesztők. Az ilyen modelleket a szoftvertervezőknek kell finomítani és újratervezni.

10.9.3.3 Általános tanácsok BPM újratervezéshez SOA-ban

A következő három általános tanácsot érdemes követnie egy SzOA tervezőnek azért, hogy a vállalati folyamat modellt SzOA alapú modellé alakítsa.

- A **particionálással** a vállalati folyamatokat funkcionális vállalati területekre bontjuk. Ezzel növeljük a kohéziót, és csökkentjük a függést (csatolást).
- A **szolgáltatások megállapítása** során a SOA tervező azonosítja azokat a szolgáltatásokat, amelyek teljesítik az egyes feladatokat.
- A **minták azonosításával** beépítjük azokat az általánosan elismert szabványokat, amelyekkel a hivatkozott szolgáltatások konfigurálhatók és újrakonfigurálhatók lesznek.

10.9.3.4 Újratervezési módszer BPM-hez SzOA-ban

A SzOA fontos eleme, hogy a megfelelő szolgáltatásokat megtaláljuk, amelyek a vállalati folyamatok feladatait ellátják. Erre létezik egy lépésről-lépésre módszer, amely megmutatja egy vállalati folyamat menedzselési életciklusát. Ennek lépései:

- Folyamat dekompozíció

- A vállalati folyamatokat kisebb, kontrollálható feladattá dekomponáljuk.
- Vállalati folyamat klaszterezés
 - A vállalati folyamatokat funkcionalitás szerint klaszterezhetjük. Így relatíve egymástól független szolgáltatásokat kapunk.
- Folyamat szelekció
 - A SzOA tervezők azonosítják azokat a webes szolgáltatásokat, melyek potenciálisan teljesítik a meghatározott vállalati folyamatokat.
- Adatmodellezés
 - Adatok és a kommunikációcsere modelljeit építjük fel.
- Szolgáltatás definíció
 - Elemezzük és megtervezünk a szolgáltatásokat. Három fontos lépést kell megtennünk.
 - Először is az azonosított meta-adatokat elemzzük.
 - Másodszor a informatikai biztonsági tervezők a kezdeti architektúra modellből meghatározzák a fontos biztonsági mintákat.
 - Végül a meta-adatokat összekapcsoljuk a szolgáltatásokkal, meta-adat szerkezet részletei definiáltak és már modellezve vannak.
- Vállalati logika finomítás
 - A definiált szolgáltatásokat iteratívan finomítjuk az interfészek alapján, és összekapcsoljuk a vállalati célokkal.
- Szolgáltatás implementáció
 - A vállalati folyamat modelleket transzformáljuk informatikai megvalósítási, kivitelezési modellé.
- Szolgáltatás telepítés
 - Az megvalósított szolgáltatásokat telepítjük alkalmazáserverekre, amik így már futtatható szolgáltatások lesznek.
- Monitorozás és rendszerfelügyelet
 - Napi monitorozást és rendszerfelügyeletet hajtunk végre a telepített szolgáltatásokon a megfelelő funkcionalitás biztosítása érdekében.
- Szolgáltatás karbantartás, napra készen tartás
 - Szükséges karbantartási munkákat végrehajtjuk a létező szolgáltatásokon.

10.9.3.5 Rugalmas vállalati folyamat integráció SzOA-ban

Létező alkalmazásokat integrálhatunk **wrapper** tervminta segítségével. Habár a *wrapper* mintának számos hátránya van. A vállalati modell változása ennek a *wrapper*nek az újratevezésével jár, és újratelepítést is igényel. Eredményképp az újrafelhasználhatóság korlátozott lesz.

10.9.3.6 Ontológia integráció

Ontológiát széles körben használják arra, hogy az információ struktúráját megosszák egymás között a szakértők, vagy a szoftverkomponensek. En struktúrát felhasználhatjuk az újrafelhasználhatósághoz. A BPM életciklushoz is illeszthető ez az ontológia. A definíciós fázis során a vállalati folyamatok definiálására használható.

10.9.3.7 Integrációs menedzser/bróker

Hét fő komponensből áll, amely a „plug-and-play” mechanizmust valósítja meg:

- Activity Parser
- Controller
- Event Capturer
- Access Control Utility
- Exception Handler
- Adaptation Layer
- Ontology Stores

A központi komponens a Controller, aminek a legfőbb feladata az Activity Parser által elemezett tevékenységek integrációja, és a megfelelő tevékenységek meghívása a tevékenység nevek alapján. Az Event Capturer a belső viselkedésre vonatkozó tevékenységeket kezeli. A Controller kivétel esetén gondoskodik a megfelelő tevékenységek végrehajtásáról. Az Access Control Utility megnézi, hogy az adott hatókörön belül érvényesen, a biztonsági feltételeknek megfelelően hívtunk meg egy tevékenységet. Az Exception Handler monitorozza a tevékenységeket futási időben. Az Adaptation Layer befogadja az új, integrálandó alkalmazásokat.

10.9.3.8 Integrációs tevékenység életciklusa

Három lépésből áll:

- először az új tevékenységet befogadjaz integrációs ontológiába, amit később a Controller és az Exception Handler felhasznál,
- másodsor a kiterjesztett részeket egy GUI eszközzel, vagy manuálisan hozzáadjuk egy létező integrációs ontológia tárolóhoz,

- harmadszor az Adaptation Layer-t fejlesztjük ki a megfelelő előredefiniált interfészeknek megfelelően.

Az utolsó lépést általában webes szolgáltatásokkal oldjuk meg, például *JavaBeannel*.

10.9.3.9 Vállalati folyamat monitorozás

Annak a bizonyítására, hogy a vállalati folyamat integráció holtpontmentes, számos formális verifikációs technika létezik, például a Petri háló. A Petri hálókat széles körben alkalmazzák, többek között gyártósorok és vállalati folyamatok verifikációjára. E folyamatok elosztott, konkurens, aszinkron irányítását is lehet modellezni Petri hálókkal.

10.9.3.10 BPM és integráció

Számos technika és módszertan létezik a vállalati folyamat menedzsment és integráció területén. Ez a SZKASZ a SZOA alapú vállalati folyamat menedzsmentre és integrációra fókuszált. Megmutattunk két megközelítést a vállalati folyamatok szolgáltatássá transzformálására: a fentről-lefelé, illetve az alulról-felfelé módszert. Ezután bemutattuk az ontológia alapú folyamat integrációt.

Vállalati folyamat integráció és folyamatok transzformálása SzOA megoldásokká fásaszto és sok hibalehetőséget rejtő feladat automatikus eszköz támogatása nélkül.

Megbízható vállalati folyamat integráció esetében három fontos teendő van.

- 1) Először is kell egy részletes **követelménylista** minden alkalmazás integrációs tevékenységről. Ez a lista nevekből, I/O paraméterek és azok formátumából, stb. áll.
- 2) Másodrészt az integráció megbízhatósága is egy fontos tényező. Az újonnan integrált alkalmazásnak nem szabad sok programozási erőfeszítést követelnie.
- 3) Harmadrészt szükség van egy olyan rugalmas/flexibilis mechanizmusra, amely a meta-adatokon, illetve ontológiákon alapul, amelyek segítségével a vállalati szabályok és feltételek viszonylag rugalmasan kifejezhetők és ábrázolhatók a számítógépben.

Ezek a módszerek egy általános és szabványos módszert adnak a vállalati folyamatok integrációjára.

11 FOLYAMATMODELLEZÉSI MÓDSZERTANOKAT ALKALMAZÓ ESZKÖZÖK

11.1 Adonis CE 3.9

Azt egyik eszköz az Adonis Community Edition. Az Adonis egy termékcsalád része, amit az osztrák BOC Group tanácsadó-, és szoftvercég fejleszt, és terjeszt. A BOC Group üzletpolitikája szerint, a Community Edition-ben tárja a nagyközönség elé az üzleti újításait, ezáltal gyakorlatilag a folyamatmodellező közösség végzi a szoftverújítások tesztelését, majd végül az ő tapasztalataikon és visszajelzésükön keresztül kerülnek be az új funkciók, újítások a végleges üzleti verziókba. Ez által az Adonis CE egy valós üzleti igényekre gyorsan reagáló, napra kész üzleti szoftver.

Az Adonis Community Edition-t tehát bárki szabadon használhatja, még az üzleti életben is, bár funkcionalitása nem teljes körű, mivel a modellek csak a szoftvert futtató gépen lévő adatbázisban tárolódnak, nem valósulhat meg a központi szervertes elérés, és a több felhasználós környezetet sem támogatja. [24]

Az Adonis szoftver eszköz család kereskedelmi célokat szolgáló tagja az Adonis Business Edition 4.0, amely magában foglalja a BPMN alapú szervezeti folyamat modellezést, illetve alapfunkcionalitásában is fejlettebb a közösségi változatnál.

Az Adonis saját, gyártó függő modellezési módszertana az EPC-hez hasonlít, de lehet benne BPMN szerint is modellezni. Viszont ez a BPMN diagram kicsit máshogyan viselkedik az igazi BPMN-t használó szoftverekhez képest. Például kijelölésnél nem mozgatja együtt a sávot és a benne elhelyezkedő objektumokat. Valamint még egy szembetűnő hátránya, hogy az azonos dokumentumokat nem engedi azonosan elnevezni, bár a státusz kezelése szabványszerű, így a BPMN eredeti dokumentum élelciklus kezelése sérül. Ez azért is érdekes, mert a BPMN modellben szereplő dokumentumokat egyértelműen meg lehet feleltetni az Adonis külön dokumentum modelljében szereplő elemeknek, azaz logikailag azonos entitást jelenítenek meg.

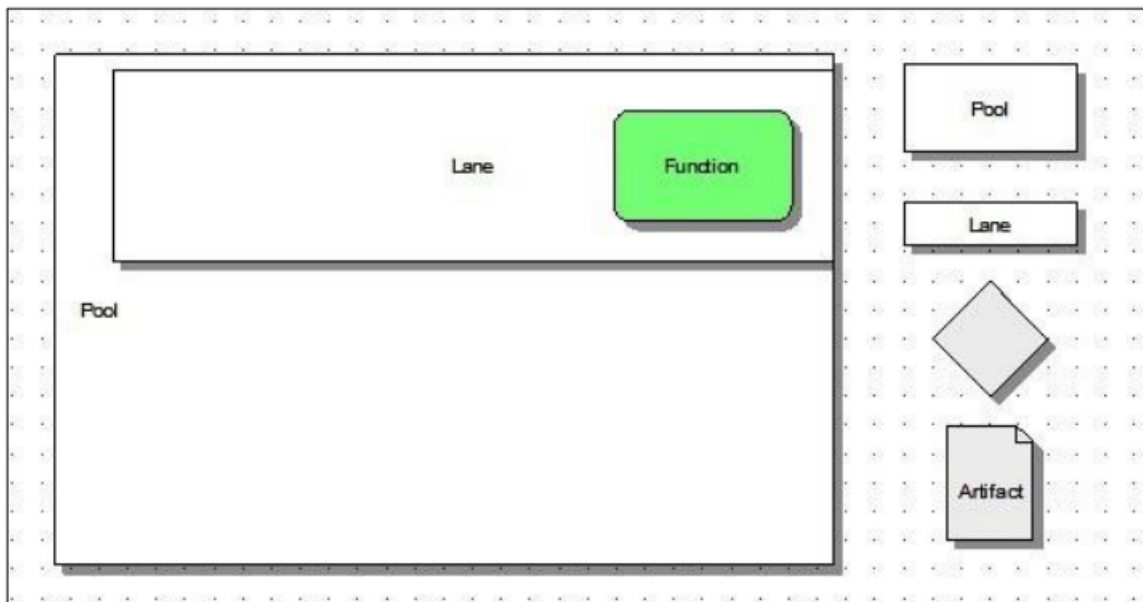
11.2 ARIS Toolset 7.0.2

Az ARIS termékcsaládot az IDS Scheer vállalat fejlesztette ki, amit a Saar vidéki egyetem professzora Dr. August-Wilhelm Scheer hozott létre egyetemi munkatársaival együtt. Az ARIS fejlesztésekor a cél az volt, hogy egy közös nyelvet hozzanak létre tanácsadók, menedzserek és informatikusok részére, akik így könnyebben megértik egymást és így hatékonyabban

tudnak közösen dolgozni. Az IDS Scheert 2009 júliusában vásárolta fel Németország második legnagyobb szoftvergyártója, a Software AG.

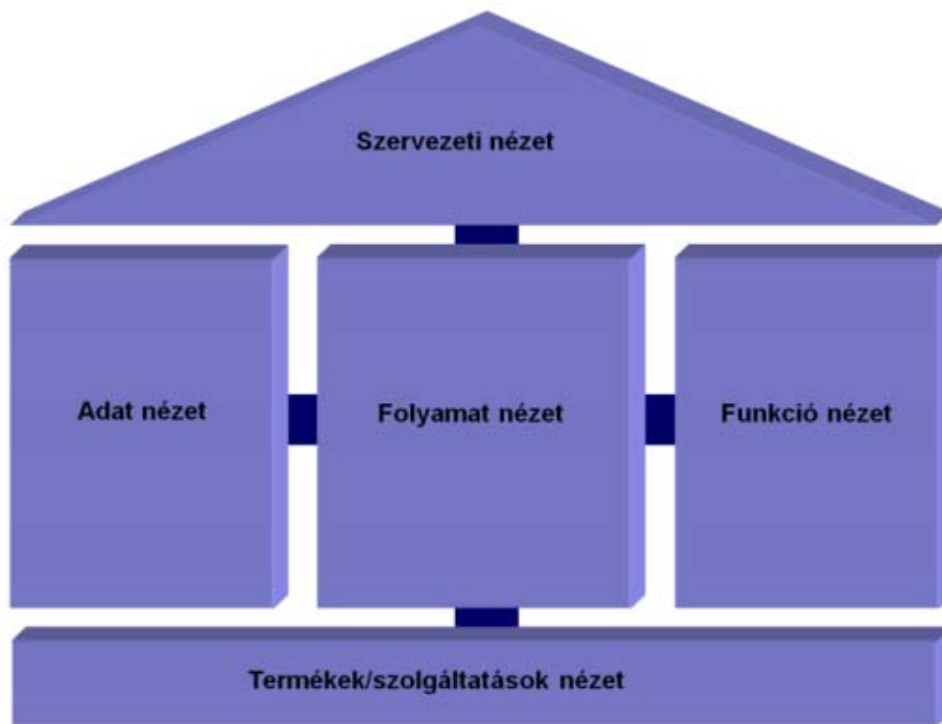
Az ARIS egy mozaikszó, ami a következő német szavak kezdőbetűiből áll össze: „Architecture of Integrated Information Systems”. Ez magyarul annyit jelent, hogy „Integrált Információ-rendszerek Architektúrája”. Ebből is láthatjuk, hogy az ARIS több mint egy egyszerű célszoftver, mert az ARIS vállalati információk integrált kezelését segíti elő azzal, hogy megfogalmazza a vállalati rendszerek felépítésének, architektúrájának az ajánlott, eredményes működését. A programcsomaggal emellett lehetővé válik a vállalati folyamatok leírása, és üzleti folyamatainak, valamint informatikai rendszereinek támogatása is.

Az IDS Scheer fejlesztette ki az EPC folyamatmodellezési módszertant, ezért alapértelmezetten így is modellez. Viszont az üzleti igények miatt kiegészítette a modellezési lehetőségét a BPMN diagrammal is. Aminek viszont a funkcionalitása hiányos, mert a dokumentumoknak nem kezeli a állapotát, nem rendelkezik az összes esemény típussal és csak egy fajta logikai kaput különböztet meg. Ráadásul az ARIS dinamikusan növekvő modellezési területében az úszósávok először csak egy objektum méretet vesznek fel, amit a modellezőnek kell növelnie. Ezek miatt a hátrányok miatt az ARIS-ban mindenképpen az EPC szerinti modellezés az ajánlott.



113. ábra BPMN objektumok ARIS-ban

Az eszközcsoport több alkotórészből áll, és ezek segítenek a folyamatok összehangolt kezelésében, vagyis folyamatok modellezésében, és a már működő folyamatok kiértékelésében. A vállalati struktúra lebontását felülnézetből, felülről-lefelé haladva („top-down”) módszerrel kell végezni. A szervezetek, vállalatok leképezésére az ARIS különböző nézetekbe csoportosítja a modell típusokat.



114. ábra ARIS ház

Négy különböző nézet van az ARIS felépítésében, ezeket ház formájában szokták ábrázolni (114. ábra):

- **Adat nézet:** A vállalati tranzakciókkal, folyamatokkal kapcsolatos adatokat a szervezet informatikai rendszerének kell kezelnie és tárolnia, a köztük fennálló összefüggéseket érdemes egy adatstruktúrában megtervezni, ezt támogatja az ARIS az adat nézetben.
- **Funkció nézet:** A szervezet életét funkciók szerint ábrázolja. Itt kerülnek leírásra a stratégiai célok, az ezek elérését célzó feladatok, és az azokat támogató alkalmazási rendszerek és azok alkotórészei, moduljai.
- **Szervezeti nézet:** Ebben a nézetben a vállalati irányítási és működtetési szervezeti egységeket ábrázolják. A szervezet különböző funkcióinak statikus kapcsolatát mutatják

be, melyek a vállalati tevékenységek végrehajtásáért felelősek. Valójában a szervezeti struktúrát definiálják a szervezeti ábra segítségével.

- **Termék / szolgáltatás nézet:** Itt kerülnek bemutatásra, strukturált formában a szervezet által előállított termékek, összetevőikkel együtt, illetve ide tartoznak a szolgáltató vállalatok által nyújtott szolgáltatások is.
- **Folyamat nézet:** Az ARIS ház közepén helyezkedik el. Jellegzetessége az, hogy a másik három statikus nézetet egyesíti egy dinamikus nézetben, a statikus nézetekben használt objektumokat kapcsolja össze. Ebben a nézetben már a szervezeti folyamatok szerepelnek, amik visszatükrözik a logikai és időbeli sorrendiséget is.

A szoftver a BPMN 1.1 verzióját támogatja.

11.3 Microsoft Office Visio Professional 2007

A Visio a Microsoft Office csomag általános célú rajzeszköze. Objektumalapú grafikák készítésére is alkalmas, a legtöbb diagramtípust és specifikus grafikai eszközrendszert tartalmazza, bár a legtöbbet kissé „lebutítva”.

Jó példa erre a folyamatmodellező eszköztára. Képes eseményvezérelt folyamatlánc (EPC), és BPMN-hez hasonló folyamatábrák rajzolására is, de a funkcionalitása valóban csak a rajzolásra terjed ki. Nem tarthat objektumait adatszótárban, repozitóriumban, így nem ismeri fel az azonos objektumokat sem. Nem veszi figyelembe az objektumok típusait sem, így történhet meg az, hogy tevékenységet szervezeti elem, azt pedig esemény követ a diagramon. (27. ábra)

A program arra is lehetőséget biztosít, hogy egy diagramfajtán egy másikdiagram típus vagy módszertan objektumát használjuk. Így például az eseményvezérelt folyamatlánc kiegészíthető egy Apple Mac-et ábrázoló piktogrammal, ha valamiért szükség lenne rá. Ugyanez mondható el a Visio Flowchartok-ról is, kiegészítve azzal, hogy azok nem követik a BPMN szabványt.

A Visio annak ellenére, hogy az egyik legnépszerűbb modellezési szoftver eszköz, igazából csak egy rajzeszköz, és ezáltal nem tudja folyamatok logikáját teljes körűen visszaadni. EPC-jébe gyakorlatilag bármilyen objektum beilleszthető, de hiányzik belőle a szabványos adatkör és informatikai rendszer, illetve a belső és külső szereplő objektuma.

12 IRODALOMJEGYZÉK

1. D. Garlan, M. Shaw, An Introduction to Software Architecture, Advances in Software Engineering and Knowledge Engineering, Volume I, World Scientific, 1993.
2. Mosley, Mark, et al. DAMA guide to the data management body of knowledge (DAMA-DMBOK guide). Technics Publications, 2010.
3. Szabó, Gy – Bagó, P. (Szabó et.al, 2011): Multinacionális vállalatok globalizált ERP modelljei, fejlődési tendenciák. *Vezetéstudomány*, Nr. 5/2011, 45-56. old, ISSN: 0133-0179. Budapest, 2011
4. ELTE (ELTE, 2010): ERP rendszerek globalizálódása, telepítési struktúrája nemzetközi cégeknél. *Kutatási Beszámoló*, Budapest, 2010,
5. Geschneidinger, W. (Geschneidinger, 2011): Anforderungen an ERP-Systeme. *ERP Management*, Nr. 1/2011, S. 61
6. Was kostet die Cloud? www.login2work.de (letöltve: 2012. 03. 28.)
7. Rosbach, M.-, Jung-Elsen, S. (Rosbach et al, 2011): ERP on demand. *ERP Management*, Nr. 3/2011, S. 51-52
8. Repschläger, J.-, Zarnekow, R. (Repschläger et al, 2011): IT-Outsourcing und Cloud-Sourcing Gemeinsamkeiten und Unterschiede. *ERP Management*, Nr. 1/2011, S. 48-51.
9. Fröschle, H-P. (Fröschle, 2011): Cloud Computing - Herausforderungen für IT-Management und –Betrieb. *ERP Management*, Nr. 1/2011, S. 45-46
10. von der Dovenmühle. T.-, Gómez, J. M. (Dovenmühle et al, 2011): Datenschutz beim Einsatz von Cloud Computing, *ERP Management*, Nr. 3/2011, S. 58-60
11. D-Grid-Projekts FinGrid (2009): Grid Computing in der Finanzindustrie. Publication 01/2009, *efinancelab* Frankfurt/M, Books on Demand GmbH, Nordenstadt.
12. Racskó P. (2012): A számítástechnikai felhő az Európai Unió egén. *Vezetéstudomány*, Nr. 1/2012, 2-16. oldal, ISSN: 0133-0179. Budapest, 2012
13. Weinman, J. (2012). *Cloudonomics: The Business Value of Cloud Computing*. Wiley.com.
14. The Open Group: *Building return on investment from cloud computing. A white paper, cloud business artifacts project*. Cloud Computing Work Group (2010).

15. Holtz, P. Cloud biztonság, <http://www.bitport.hu/biztonsag/cloud-biztonsag-szakertoi-cikk-holtzl-peter> (letöltve: 2012.08.07.)
16. Bundesamt für Sicherheit in der Informationstechnik(BSI): *Eckpunktepapier. Sicherheitsempfehlungen für Cloud-Computing-Anbieter.*
https://www.bsi.bund.de/SharedDocs/Downloads/DE/BSI/Mindestanforderungen/Eckpunktepapier-Sicherheitsempfehlungen-CloudComputing-Anbieter.pdf;jsessionid=07F82607FB8632F14F0DC0A96F9BD4F9.2_cid286?_blob=publicationFile (letöltve: 2012.10.30.)
17. Repschläger, J., Zarnekow, R.: *IT-Outsourcing und Cloud-Sourcing – Gemeinsamkeiten und Unterschiede.* ERP Management 7 (2011) 1, 48-51. old.
18. Voas, J.; Zhang, J.: *Számítási felhő: New Wine or Just a New Bottle?* Computer org/IT PRO March/April 2009. ©Published by IEEE.
19. Ali Arsanjani, Toward a pattern language for Service-Oriented Architecture and Integration, Part 1: Build a service eco-system, <http://www.ibm.com/developerworks/webservices/library/ws-soa-soi/index.html> , 2011-08-21
20. Amdahl, G., Blaauw, G., Brooks, F. Jr: Architecture of the IBM system/360. *IBM J. Res. Develop.* 8(2) (1964) Architecture Framework Forum, <http://www.architectureframework.com/frameworks/modaf/> , (2011-08-25)
21. B. Kiepuszewski, A. H. M. ter Hofstede, W. M. P. van der Aalst, "Fundamentals of Control Flow in Workflows," *Acta Informatica*, 39(3):143-209, 2003
22. Becker 2004, Jörg Becker, *Referenzmodellierung: Grundlagen, Techniken und domänenbezogene Anwendung*, Physica Verlag, Heidelberg, 2004 (http://books.google.com/books?id=CdUQzres9EC&printsec=frontcover&hl=hu&source=gbs_slider_thumb#v=onepage&q&f=false)
23. Bieberstein-Bose-Fiammante-Jones-Shah, *Szolgáltatás-orientált architektúra*, PANEM, Budapest, 2009.
24. BOC Group (2006): ADONIS version 3.9 – User’s Manual
25. Booch, Rumbaugh, and Jacobson, *The UML Modeling Language User Guide*, Addison-Wesley, Reading, MA, 1999.
26. Breuer H. [1995]: *Informatika*, SH Atlasz, Springer-Verlag Budapest, Berlin

27. Brooks, F. *The Mythical Man-Month: Essays on Software Engineering*. Addison-Wesley, 1975.
28. Bullinger, Hans-Jörg; Fähnrich, Klaus-Peter: *Betriebliche Informationssysteme. Grundlagen und Werkzeuge der methodischen Softwareentwicklung*. Berlin, 1997
29. Casteleyn, et al., *Engineering Web Applications*, Springer-Verlag Berlin Heidelberg 2009
30. Daniel M. Brandon, editor, *Software engineering for modern Web applications : methodologies and technologies*, IGI Global, 2008.
31. Daniel Minoli, *Enterprise Architecture A to Z Frameworks, Business Process Modeling, SOA, and Infrastructure Technology*, Auerbach Publications, Taylor & Francis Group, ISBN 978-0-8493-8517-9, 2008
32. Daniel Minoli, *Enterprise Architecture A to Z Frameworks, Business Process Modeling, SOA, and Infrastructure Technology*, Auerbach Publications, Taylor & Francis Group, ISBN 978-0-8493-8517-9, 2008
33. Davenport, T. H. – Short, J. E. (1990): *The New Industrial Engineering: Information Technology and Business Process Redesign*, in *Sloan Management Review*, 1990. Summer
34. Davenport. T. H. (1993): *Process Innovation: Reengineering Work Through Information Technology*, Harvard Business School Press, Cambridge, MA
35. Davis 2001, Rob Davis *Business process modelling with ARIS: a practical guide*, Springer-Verlag London Limited 2001, (http://books.google.com/books?id=H2eKSmA-A7kC&printsec=frontcover&lr=&hl=hu&source=gbs_similarbooks_s&cad=1#v=onepage&q&f=false)
36. Davis 2007, Rob Davis and Eric Brabänder, *ARIS Design Platform Getting Started with BPM*, Springer-Verlag London Limited 2007, ISBN-13: 978-1-84628-612-4
37. Davis 2008, Rob Davis, *ARIS Design Platform, Advanced Process Modelling and Administration*, Springer-Verlag London Limited 2008, ISBN: 978-1-84800-110-7
38. Deák Csaba, dr., (2005): *Üzleti Folyamatok Újjáalakítása*, Miskolci Egyetemi Kiadó
39. Dewayne E. Perry, Wolf, A., "Foundations for the Study of Software Architecture" *ACM SIGSOFT Software Engineering Notes*, Vol. 17, No. 4, October 1992, pp. 40-52.

40. Dijkstra, E.W., The Structure of "THE" Multiprogramming System, *Communications of the ACM* (1968), Volume 11, Number 5, Pages 345–346.
41. Dirk Matthes, Enterprise Architecture Frameworks Kompendium, © Springer-Verlag Berlin Heidelberg 2011, e-ISBN 978-3-642-12955-1, ISSN 1439-5428
42. Dobák Miklós (2006): Szervezeti Formák és Vezetés, Akadémiai kiadó
43. DoD Architectural Framework, Version 1.0, Volume I, "Definitions and Guidelines. "
44. Emilia Mendes · Nile Mosley (Eds.), Web Engineering, Springer-Verlag Berlin Heidelberg, 2006,
45. Eric Newcomer, Greg Lomow, Understanding SOA with Web Service, Addison Wesley Profession, 2004, ISBN: 0-321-18086-0
46. Eve Andersson, Philip Greenspun, and Andrew, Grumet , Software engineering for Internet applications , Massachusetts Institute of Technology, 2006
47. Gábor András és munkatársai (2007): Üzleti Informatika, Aula Kiadó
48. Gerti Kappel ... [et al.], Web engineering , ISBN-13: 978-0-470-01554-4, John Wiley & Sons Inc.
49. Ghazi Alkhatib and David Rine, editors, Web engineering advancements and trends : building new dimensions of information technology, IGI Global, 2010, ISBN 978-1-60566-719-5
50. Global360 (2008): Insight360 Process Designer – Getting Started
51. Hammer, Michael – Champy, James (1993): Reengineering the Corporation. A Manifesto for Business Revolution
52. Hammer, Michael – Champy, James (2000): Vállalatok Újraszervezése, Panem Könyvkiadó
53. Hammer, Michael (1990): Re-engineering Work: Don't Automate, Obliterate, in Harvard Business Review, 1990. július-augusztus
54. Hanrahan, Robert P. (N/A): The IDEF Process Modeling Methodology, <http://www.stsc.hill.af.mil/crosstalk/1995/06/IDEF.asp> Letöltve: 2010. március 16.
55. Helmut Schmalen, Általános üzleti gazdaságtan, 2002, Axel-Springer Budapest Kiadó, 963-7880-89-5
56. Homonnay Gábor , Alkalmazási rendszerek , 2003 , Műszaki Könyvkiadó,
57. IDS Scheer AG (2005): ARIS Platform 7.0 – Quick Start Guide

58. IFUA Horváth & Partner Vezetési és Informatikai Tanácsadó Kft. (2008): Folyamatmenedzsment a Gyakorlatban – Árbevétel-növelés és Költségcsökkentés Tartósan Jó Folyamatteljesítménnyel, Alinea Kiadó
59. Intalio | Works (2010): BPMS, <http://www.intalioworks.com/products/bpm/> Letöltve: 2010. március 20.
60. Isabel Seruca, José Cordeiro, Slimane Hammoudi, Joaquim Filipe, *Enterprise Information Systems VI.*, 2006, Springer Verlag, ISBN-10 1-4020-3674-4
61. ISO, Information technology -- Open Distributed Processing -- Unified Modeling Language (UML) Version 1.4.2 ISO/IEC 19501:2005, http://www.iso.org/iso/iso_catalogue/catalogue_tc/catalogue_detail.htm?csnumber=32620, 2011-09-02
62. Iványi A. (szerk.) [2006]: Informatikai angol-magyar szótár, Tinta Kiadó
63. Jack Wout, et al. *The integrated architecture framework explained : why, what, how.* Springer, 2010. ISBN 978-3-642-11517-2, DOI 10.1007/978-3-642-11518-9
64. Jeffrey Zeldman, *Designing With Web Standards*, New Riders Publishing, 2003, ISBN, 0-7357-1201-8
65. Jim Conallen, *Building Web Applications with UML*, Second Edition, Addison Wesley, 2002, ISBN, 0-201-73038-3
66. John A. Zachman (2009): The Zachman Framework: The Official Concise Definition <http://test.zachmaninternational.com/index.php/the-zachman-framework> , 2011-08-18
67. Kaisha Tec (2009): BPMN - Business Process Modeling Notation 1.2 Poster, http://www.active-flow.com/download/Documents/Poster_BPMN.pdf Letöltve: 2010. március 16.
68. Lankhorst, M., et al. (eds.): *Enterprise Architecture at Work: Modelling, Communication and Analysis.* Springer, Berlin (2005), ISBN-10: 3540243712
69. Li, Qing – Chen, Yu-Liu (2009): *Modeling and Analysis of Enterprise and Information Systems – From Requirements to Realization*, Springer Berlin Heidelberg
70. Lombardi (2009): *Process Mapping 101 – A Guide to Getting Started*
71. M. Havey *Essential Business Process Modeling*, O'Reilly , 2005, ISBN: 0-596-00843-0

72. Marc Lankhorst et al., *Enterprise Architecture at Work*, 2005, Springer-Verlag Berlin Heidelberg, ISBN-10 3-540-24371-2
73. Martin Op 't Land, Erik Proper, Maarten Waage, Jeroen Cloo, Claudia Steghuis, *Enterprise Architecture, Creating Value by Informed Governance*, Springer-Verlag Berlin Heidelberg, ISBN 978-3-540-85231-5, 2009
74. Mathias Weske, *Business Process Management, Concepts, Languages, Architectures*, © Springer-Verlag Berlin Heidelberg 2007, ISBN 978-3-540-73521-2 ,
75. McDermott J. (1982). R1—A rule-based configurer of computer systems. *Artificial Intelligence*, 19 (1), 39–88.
76. MEGA (2010): MEGA Process (BPMN) – User Guide
77. Metastorm (2008): Metastorm BPM Release 7.6 – Designer Manager's Edition Overview
78. Molnár B., [1997]: *Bevezetés a rendszerelemzésbe*, in: Gábor András (szerk.) „Információmenedzsment”, Aula Kiadó, 1997, pp 107-239.
79. Molnár Bálint, 'Ismeretszerzés', in: Futó Iván (szerk.) „Mesterséges Intelligencia”, Aula Kiadó, 1999, pp 665-708. http://www.mtaita.hu/KADSbev9_1.PDF , <http://www.mtaita.hu/CommonKADS.PDF>
80. Molnár Bálint, *Rendszerelemzés*, in: Gábor András (szerk.) „*Információmenedzsment*”, Aula Kiadó, CD-melléklet, 1996–98, <http://www.mtaita.hu/hu/Publikaciok/Ssadm1.pdf> (2011-08-29)
81. Molnár Bálint (2002): *Bevezetés a rendszerelemzésbe*, A rendszerszervezés alapjai, Műszaki Könyvkiadó, 2002. (Kiadói azonosító: MK-00275), <http://www.muszakikiado.hu/details.php?details=MK-00275>, <http://www.mtaita.hu/Bevezetes.pdf>
82. Molnár, B.; Kő, A. & Vas, R. 2003Az információs társadalom tudástranszfer innovatív módjaiAz Informatikai és Hírközlési Minisztérium. <http://www.mtaita.hu/hu/Publikaciok> (2011-08-29)
83. Németh Balázs, dr., (2008): *Folyamatmenedzsment Megvalósítása a Vállalati Gyakorlatban*, in *Minőség és megbízhatóság*, 2008. 42. évf. 1. szám, p. 27-31.
84. Object Management Group [OMG] (2009): *Business Process Model and Notation Version 1.2*
85. OMG Formally Released Versions of UML , <http://www.omg.org/spec/UML/> ,2011-09-02

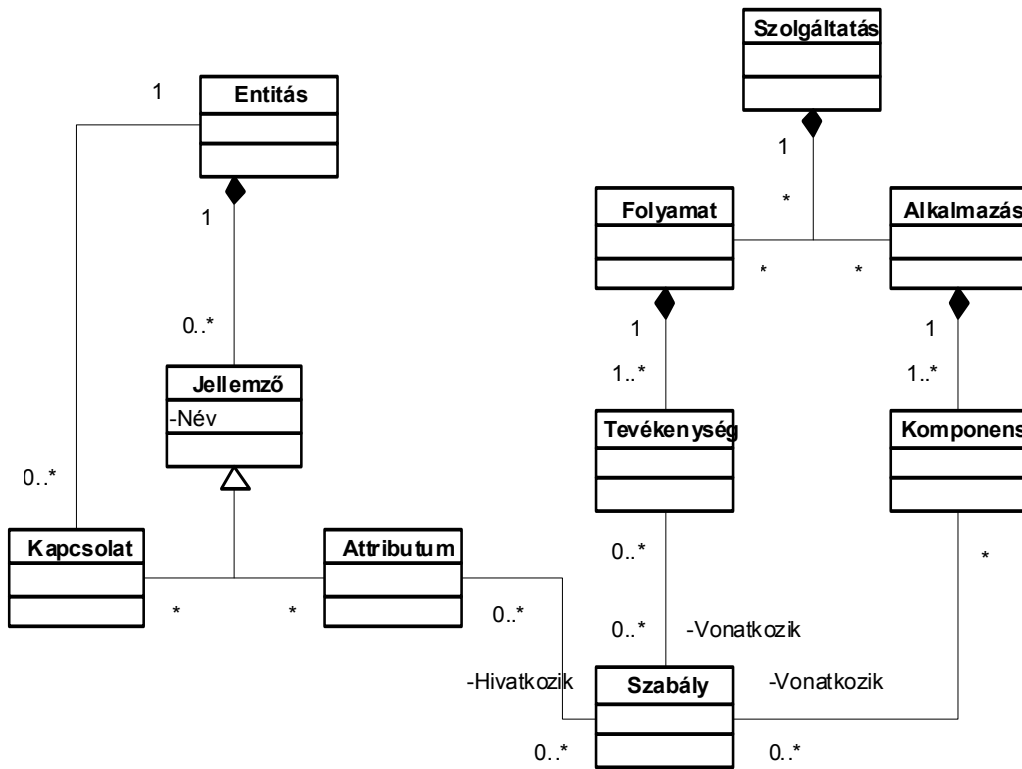
86. Open Group, TOGAF: The Open Group Architecture Framework, TOGAF® Version 9, <http://www.opengroup.org/togaf> , 2010.
87. *Oxford Dictionary of English*, new edn. Oxford University Press, Oxford (2005), ISBN-13: 9780198610571, <http://www.oxfordadvancedlearnersdictionary.com/dictionary>
88. Perks, Col., Beveridge, Tony, *Guide to enterprise IT architecture*, Springer-Verlag New York., ISBN 0-387-95132-6, 2003 .
89. Portugal 2006, Victor Portugal and David Sundaram , *Business processes : operational solutions for SAP implementation*, 2006, Idea Group Inc., ISBN 1-59140-615-3
90. Quick start your Enterprise Architecture (EA) with TOGAF 9 reference content and ARIS http://www.ids-scheer.com/en/ARIS/ARIS_Reference_Models_/ARIS_TOGAF/171464.html
91. R. Heeks, *Implementing and Managing eGovernment*, SAGE Publications, 2006, ISBN 0 7619 6792 3
(http://books.google.com/books?id=hRzAnMulatUC&printsec=frontcover&hl=hu&source=gbs_slider_thumb#v=onepage&q&f=false)
92. Raffai Mária (1999): *BPR – Üzleti Folyamatok Újjászervezése*, Novadat Kiadó
93. *Recommended Practice for Architectural Description of Software Intensive Systems*. Technical Report IEEE P1471-2000, The Architecture Working Group of the Software Engineering Committee, Standards Department, IEEE, Piscataway, New Jersey, USA (2000), ISBN-10:0738125180. <http://www.ieee.org>
94. Rossi, G.; Pastor, O.; Schwabe, D.; Olsina, L. (Eds.): *Web Engineering: Modelling and Implementing Web Applications*, ISBN 978-1-84628-922-4, Springer, 2008
95. Russel, S. J., Norvig, P., „Mesterséges intelligencia – modern megközelítésben”, Panem – Prentice Hall, Budapest, 2000, 1093 old. (Az eredeti mű: *Artificial Intelligence. A Modern Approach*” Prentice Hall, Inc., 1995.)
96. San Murugesan, Yogesh Deshpande: *Web Engineering : Managing Diversity and Complexity of Web Application Development*, Springer, 2001, ISBN 978-3540421306,
97. Scheer , August-Wilhelm Scheer, Wolfram Jost, Karl Wagner - *Von Prozessmodellen zu lauffähigen Anwendungen: ARIS in der Praxis*, ISBN: 3540234578,
98. Scheer 1994, August-Wilhelm Scheer, *Business Process Engineering Study Edition: Reference Models for Industrial Enterprises*, Springer-Verlag ,1994

99. Scheer 1997, August-Wilhelm Scheer, *Wirtschaftsinformatik: Referenzmodelle für industrielle Geschäftsprozesse*, Springer-Verlag, 1997
(<http://books.google.com/books?id=LKHTM-9dqF4C&printsec=frontcover&hl=hu#v=onepage&q&f=false>)
100. Scheer 2001, August-Wilhelm Scheer, *ARIS - Modellierungsmethoden, Metamodelle, Anwendungen*, Springer-Verlag, 2001,
(<http://books.google.com/books?id=Of12tWrCtxcC&printsec=frontcover&lr=&hl=hu#v=onepage&q&f=false>)
101. Scheer 2002, August-Wilhelm Scheer 2002, *ARIS - vom Geschäftsprozess zum Anwendungssystem*, Springer-Verlag Berlin, 2002,
(<http://books.google.com/books?id=eayNqsed1QMC&printsec=frontcover&hl=hu#v=onepage&q&f=false>)
102. Scheer 2002, August-Wilhelm Scheer, Wolfram Jost, *ARIS in der Praxis*, Springer-Verlag Berlin, 2002,
(<http://books.google.com/books?id=fl8SvLlgAHkC&printsec=frontcover&lr=&hl=hu#v=onepage&q&f=false>)
103. Scheer, August-Wilhelm: *ARIS – Modellierungsmethoden, Metamodelle, Anwendungen (3. Auflage)*, Berlin, 1998
104. Shaw, M., Garlan, D.: *Software Architecture: Perspectives on an Emerging Discipline*. Prentice-Hall, Englewood Cliffs (1996), ISBN-10: 0131829572.
105. [Simha R. Magal](#) (Grand Valley State University), [Jeffrey Word](#) (SAP), *Essentials of Business Processes and Information Systems*, ISBN 978-0-470-23059-6, Wiley , 2010
106. Sowa, J., Zachman, J.: Extending and formalizing the framework for information systems architecture. *IBM Syst. J.* **31**(3), 590–616 (1992)
107. Szerkesztette: Heteyi József: *ERP rendszerek Magyarországon a 21. században*, 2004, Computer Books, Budapest, ISBN: 963 618 318 X
108. Szerkesztő: Heteyi József: *Pénzüntézetek és állami intézmények információs rendszerei Magyarországon*, Computer Books, Budapest, ISBN: 963 618 291 4
109. Tenner, A.R., De Toro, I.J. (1998): *BPR – Vállalati Folyamatok Újraformálása*, Műszaki Könyvkiadó

110. Thomas A. Curran, Gerhard Keller, Andrew Ladd, *SAP R/3 Business Blueprint: Understanding the Business Process Reference Model*, 1998, Prentice Hall, ISBN 0*13-521147-6
111. Thomas Erl, *Service-Oriented Architecture Concepts, Technology and Design*, 2005, Pearson Education
112. Tibco (2009): *Tibco Business Studio – Process Modeling User’s Guide*
113. TOGAF— *TOGAF Version 9, The Open Group Architectural Framework*, The Open Group, 2009, <http://www.togaf.org>
114. USA Government: Clinger–Cohen; *IT Management Reform Documents Act* (1996). http://www.cio.gov/it_management_reform_act_Feb_1996.html
115. Várkonyi, L., *Szolgáltatás Orientált Architektúra, Jelen és jövő című előadás*, 2005 Letöltve: 2011-08-23 , http://www-05.ibm.com/hu/news/events/2005/akademia/dn_0928/2.pdf
116. Vida Gábor (2006): *Folyamatcontrolling. Controlling Portál*, http://www.controllingportal.hu/temp/Vida_G_Folyamatcontrolling.pdf Letöltve: 2010. március 8.
117. White, Stephen A. (2005): *Introduction to BPMN*, http://www.bpmn.org/Documents/Introduction_to_BPMN.pdf Letöltve: 2011-09-07.
118. Wikipedia (2010a): *Business Process Modeling*, http://en.wikipedia.org/wiki/Business_process_modeling Letöltve: 2010. március 16.
119. Wikipedia (2010b): *IDEF3*, <http://en.wikipedia.org/wiki/IDEF3> , Letöltve: 2010. március 5.
120. William L. Oellermann, Jr., *Architecting Web Services*, Apress , 2001, ISBN:1893115585
121. Williams, S. (1967): *Business Process Modeling Improves Administrative Control*, in *Automation*. December, 1967, pp. 44 - 50.
122. Woojong Suh (ed.): *Web Engineering: Principles and Techniques*, ISBN 978-1591404330, IGI Global, 2005.
123. xAF working group: *Extensible Architecture Framework version 1.1 (formal edition)*. Technical Report (2006).

124. Zachman, J.: A framework for information systems architecture. IBM Syst. J.
26(3) (1987)

13 ÁBRA ÉS TÁBLÁZAT MELLÉKLETEK

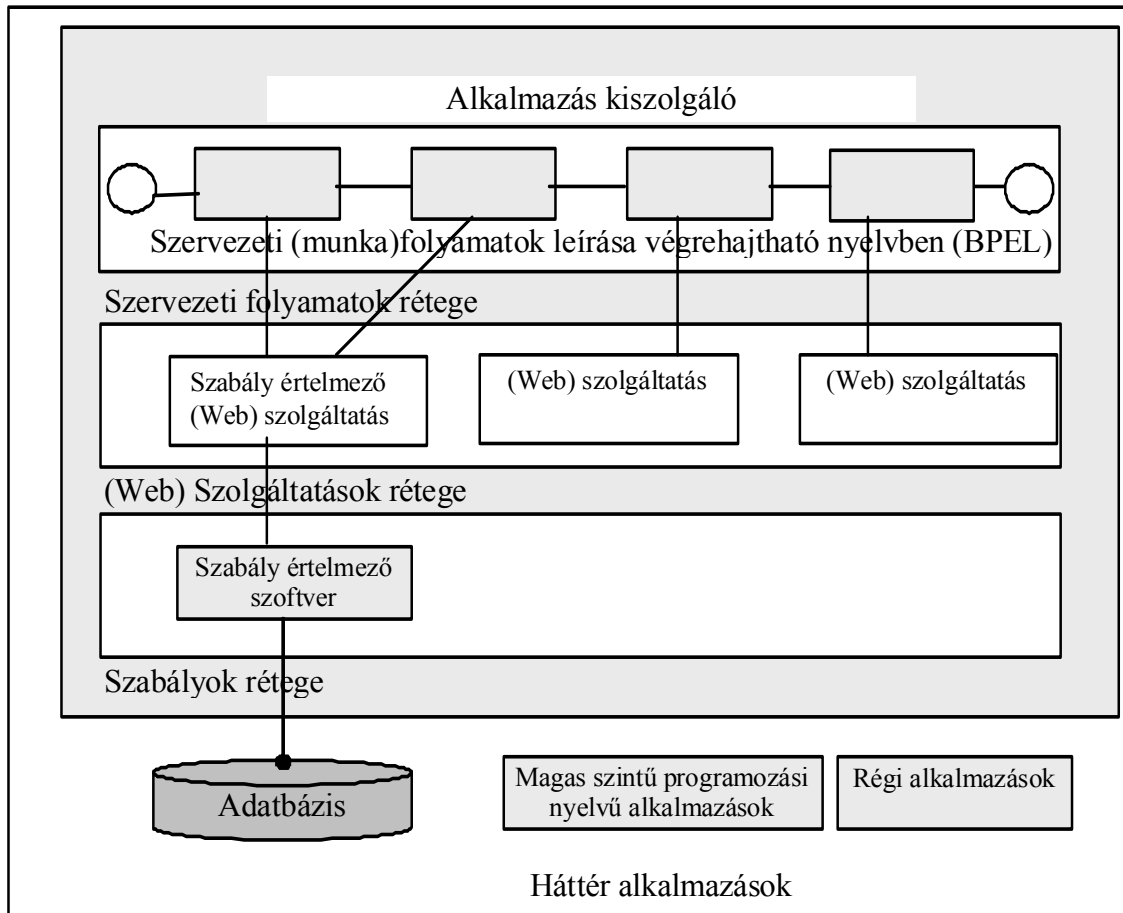


115. ábra A szabályalapú folyamatmenedzsment elemeinek metamodellje

Folyamatmodellezés és menedzsment	Munkafolyamat leírást értelmező szoftver (workflow, engine)	Szervezeti tevékenység monitoring
Tevékenység alapú költség kalkuláció	Szervezeti szabály értelmező szoftver	
Szimuláció	Integráció	
Kiegyensúlyozott mutató számrendszer	Integrált dokumentum kezelő rendszer	

↑

116. ábra Folyamat automatizálás főbb komponensei



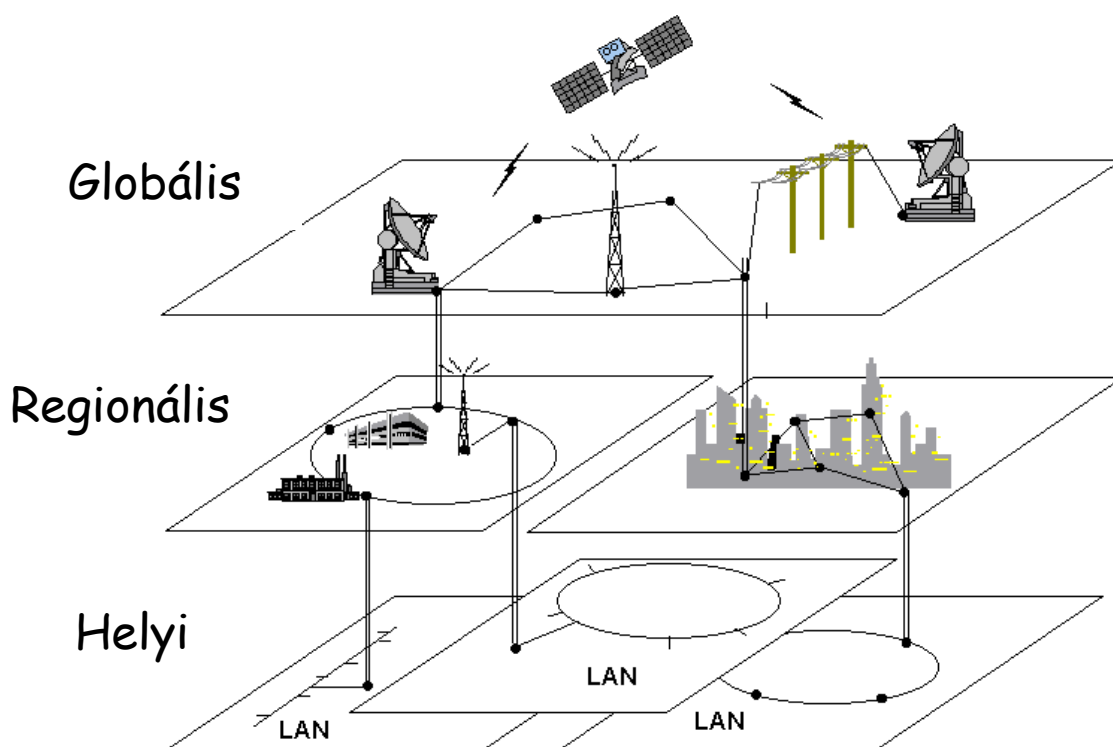
117. ábra Folyamat automatizálás szabály és szolgáltatás központú alapon.

24. Táblázat A folyamat menedzsment négy szintje

Vezetési szint	Idő horizont	Költség hatás	Döntés típusa	Használandó módszer
Valós idejű	Másodpercek- órák	Alacsony	Humán és egyéb erőforrások irá- nyítása	Vezérlés elmélet
Üzemi/ műkö- dési	Órák- napok	Korlátozott	Erőforrások kije- lölése, hozzá- rendelése	Ütemezés elmé- let (CP? PERT, stb.)
Taktikai	Napok-hónapok	Magas	Erőforrások ka- pacitás és költ- ségtervezése	Kiszolgálási, sorban állási modellek

Stratégiai	Hónapok-évek	Nagyon magas	Folyamat tervezés és erőforrás típus tervezés, hozzákapcsolás	Költségvetési, pénzügyi modellek, multi-kritériumos döntés elemzés
------------	--------------	--------------	---	--

Infrastruktúrák

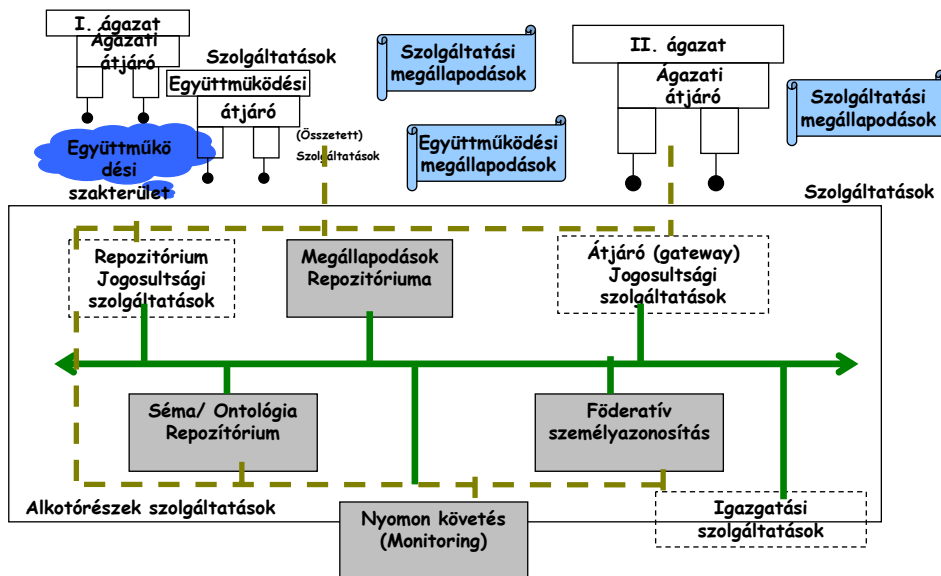


118. ábra Általános kommunikációs infrastruktúra

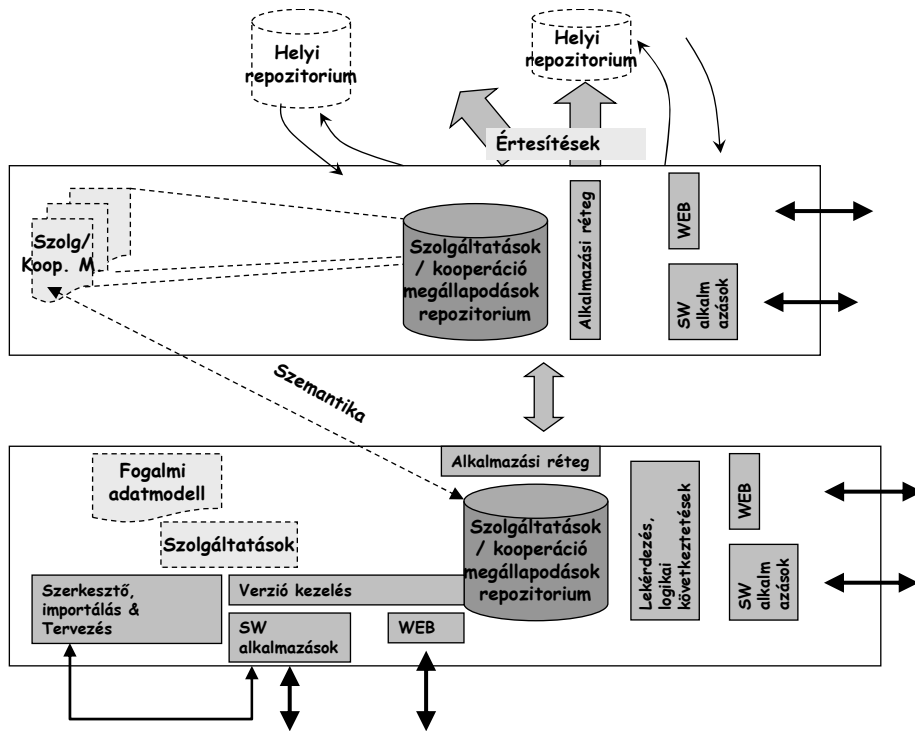
E-kormányzati célok elérése

Cél	Kapcsolattartás (információcsere)	Tranzakciónális	Transzformációs
Szolgáltatás nyújtás	Kormányzati szolgáltatások széles köre	Kormányzati szolgáltatások nagy része	„Push”. Kormányzati szolgáltatások döntő része
Belső hatékonyság	Ágazatok tervezik a polgár-centrikus szolgáltatásokat és integrálják	A szolgáltatások áttervezése a polgárok igényei szerint Ágazatközi együttműködés	Ágazatközi együttműködés bevált gyakorlat Egyéni szolgáltatási szükségletek kielégítése
Közösségi részvétel	Polgárok informálása és bevonása	Ügymenetek elektronizálása, e-szavazás. „Push”	A polgárok on-line bevonása általános.

119. ábra E-kormányzati célok elérése



120. ábra Egy referencia architektúra



121. ábra Közigazgatási szolgáltatások és jogosultságok kezelése megállapodásokon keresztül

14 FÜGGELÉK – ALAPFOGALMAK

Informatika (Informatics)

- alatt, az *információ* rendszeres és automatikus – elsősorban számítógépek segítségével történő – rögzítésével, tárolásával, feldolgozásával és továbbításával foglalkozó tudományt értjük [78], [26].

Információtechnológia

- Az angolszász eredetű, röviden IT (**Information Technology**), illetve a német eredetű információtechnika (Informationstechnik, Informationstechnologie) fogalma az informatikában a technológiai elemekre helyezi a hangsúlyt, vagyis a hardver, szoftver, távközlési, telekommunikációs és hálózati elemekre, amelyek a számítógépek fogalmát jelentős mértékben kiterjesztették [78].

15 FÜGGELÉK: KIS ANGOL – MAGYAR SZÓTÁR

A következő szószedet/kisszótár a témához tartozó fontosabb szakkifejezések angol-magyar szószedetét tartalmazza. Célja az angol nyelvű szakirodalom olvasásának segítése. Az összeállításnál támaszkodtunk a 2006-ban megjelent angol-magyar informatikai szótárra [Ld. Iványi 2006, 62].

Acceptability (User's)	Elfogadhatóság (a leendő felhasználók, megrendelő részéről)
Acceptable error rate	elfogadható hibaarány
Acceptance testing	(felhasználói) átvételi teszt
Access control	hozzáférési jogosultságok ellenőrzése
Access violation	hozzáférési jogosultságok megsértése
Accountabilities and responsibilities	feladat és felelősségi kör
Accountabilities and responsibilities	feladat és felelősségi kör
Accountability	számonkérhetőség, felelősségre vonhatóság,
Accounts (user)	felhasználói fiókok
Accredit	bevizsgál
Accreditation	bevizsgálás jóváhagyás
Accrediting systems for security	Rendszer biztonsági bevizsgálása
Accrual-based	eredmény szemléletű
Accuracy	szabatosság pontosság helyesség
Accurate	pontos
Acid test	Lakmusz teszt (Igen-nem), döntő próba
Acquisition	beszerzés
Action list	(megteendő) intézkedés lista

Action plan	intézkedési terv
Activity goal	tevékenység célja
Adequacy	megfelelés
Adjustment	igazítás kiigazítás módosítás
Administration (system)	rendszerfelügyelet
Administrative level	a napi működés irányításának szintje
Adoption	(módszer, eljárás) befogadása alkalmazása
Advantage	előny
Adverse opinion	függő záradék (negatív vélemény)
Advisable	ajánlatos
Agile	serény, tevékeny, mozgékony, fürge, gyors, élénk, agilís
Agile method	Agilis rendszerfejlesztési módszertan
Agreed-upon service levels	megállapodásban rögzített szolgáltatási szintek
Alert	éber, riasztás (ha főnév)
Align	illesztés, összehangolás,
Allocate (resources)	Elkülönít (pl. pénzügyi eszközöket, költségeket)
Amendment	kiegészítés módosítás
Analysis of cause for the problem	probléma okának elemzése
Appearance (product: system, software)	Megjelenítés
Appetite for risk	kockázat vállalási hajlandóság képesség
Application (software)	alkalmazás (szoftver)
Application layer	Alkalmazási réteg (OSI protokoll, TCP / IP)
Application system	Alkalmazási rendszer
Appraise	minősítés (szakmai előmenetel szempontjából, személyzeti lapon) személyzeti értékelés

Approach	megközelítés, eljárásrend
Appropriateness	megfelelőség
Approval	jóváhagyás
Approval standards.	átadás, átvételi eljárás szabályai előírásai szabványai
Approve	elfogad jóváhagy
Arithmetic	számszaki
Assess	értékel felmér
Assessment	értékelés
Assessment	értékelés (CMM környezetben, folyamat képesség érettségének értékelése), felmérés
Asset	vagyon(elem), eszköz (számviteli értelemben)
Assign (task)	kijelöl (valakit feladat elvégzésére), átruház
Assigned data classifications	osztályozott adatok
Assurance	értékelés biztosíték nyújtás garancia adás szavatolás
Assurance Guide	Értékelési útmutató
Assurance process	értékelési folyamat
Assurance professional	értékelési szakértő
Assurance review	értékelés felülvizsgálata szemlézése
Assurance services	Értékelési biztosíték nyújtási megerősítő szolgáltatások
Assurance, to provide reasonable assurance	Értékelés azért, hogy ésszerű biztosítékot adjon / nyújtson
Assure	szavatol biztosítékot ad garanciát nyújt megerősít
Attestation	hiteles könyvvizsgálat
Attribute	jellemző
Audit	auditálni ellenőrizni vizsgálni
Audit Committee	Informatikai Biztonsági és Ellenőrzési Bizottság Számvevői Bizottság Ellenőrzési Bizottság

Audit by exception	Rendkívüli, kivételes események által vezérelt auditálás
Audit committee (IT)	Informatikai ellenőrzési bizottság
Audit function	ellenőrzési részleg ellenőrzési szervezet, szervezeti egység
Audit Guidelines	Ellenőrzési Kézikönyv
Audit Trail	nyomon követési napló [Blanko 2002]
Audit Trail	ellenőrzési napló ellenőrizhetőségi napló tevékenység naplózása informatikai tranzakciók naplózása nyomkövetési napló nyomkövetési feljegyzések
Audit trail	Számviteli, pénzügyi ellenőrzésnél ellenőrzési nyomvonal (PM, ÁSZ, Bankárképző)
Auditor	auditor információrendszer ellenőr informatikai rendszer ellenőr informatikai ellenőr vizsgáló revizor
Auditor (IT)	(Információrendszer) ellenőr
Authenticate	hitelesítés
Authentication	hitelesítés <a kérelmező, illetve az informatikai rendszerbe bejelentkezni kívánó személy azonosságának megállapítása> (felhasználó) hitelesítés
authorising	
authority	hatáskör
Authorization	a leendő felhasználók felhatalmazása jogosítványok, engedélyek kibocsátása jogosultságok kibocsátása jogosultságok megadása
Availability	rendelkezésre állás
aware	tisztában lenni vm-vel
Awareness	veszélytudat tudatosítás tudatosság

Back Door	Hátsó ajtót nyitó program
Background check	biztonsági ellenőrzés (Id. Államigazgatás, B. és C. típusú nemzetbiztonsági ellenőrzés)
Backlog	Hátralék (munka, feladat)
Back-up	(adatok) mentése mentések
Back-up site	tartalék telephely <rendszerek helyreállítására szolgáló>
Balanced Scorecard	Kiegyensúlyozott stratégiai mutatószám rendszer
Bandwidth	sávszélesség
Base line	viszonyítási alap
Baseline (configuration)	termék mérföldkő
Benchmark	összehasonlításkor használt mérték
Benchmark incrementally	fokozatosan végezve az összehasonlítást
Benchmarking	ipari normákkal összehasonlítás informatikai ipar teljesítmény szintjeivel összevetés összehasonlító értékelés
Benefit	haszon
Benefit Management	pénzügyi haszon menedzsment
Best practices	bevált (iparági) gyakorlat
Blank (information criterion)	biankó (kritérium, feltétel)
Breach	szabálysértés, vm-nek a megsértése
Briefing	tájékoztató
Budget	Éves pénzügyi terv, költségvetés
Build (IT)	kivitelez, kivitelezés
Business case	üzleti terv
Business continuity management	a folyamatos üzemvitel fenntartása

Business continuity plan	üzemvitel, üzletvitel folyamatossága fenntarthatóságának tervezése (vállalati) működés folyamatossági terv
Business culture	szervezeti kultúra
Business disruption	üzleti tevékenység leállása szünetelése üzletvitel szünetelése üzleti szolgáltatások leállása, szünetelése, üzemszünete
Business entity	jogi személyiségű társaság társas vállalkozás vállalkozás, vállalat
Business focus(ed)	vállalat központú
Business Goals for IT	Az informatika üzleti céljai
Business initiative	üzleti javaslat
Business process owner	üzleti / szervezeti folyamat felelőse
Business process re-engineering	a szervezeti / üzleti folyamatok újra szervezése; áttervezése
Business Sponsors	finanszírozó (vállalati / üzleti terület) vezetők
Business unit	szervezeti egység (divízió, főosztály, osztály, stb.)
Bypass	megkerüli (szabályokat, előírásokat, hatóságot)
Bypass change management process	megkerülik a változáskezelési eljárásokat
CAATS (Computer assisted audit techniques)	számítógéppel támogatott auditálási módszerek
Capability	(szolgáltatási) képesség
Capacity management	Kapacitásgazdálkodás
Causal analysis	okszági elemzés; ok feltátás
Certificate	tanúsítvány
Change control	változtatás ellenőrzés kézben tartás

Change control (procedure)	Változtatás engedélyezési (eljárás)
Change management	változás / változtatás kezelés
Change request	változtatási kérelem
Charging (service fees)	(szolgáltatási díjak) felszámolása ráterhelése
Checklist	ellenőrző lista
Chief architect	Fő architektúra tervező (Informatika!)
CISA	okleveles információ-rendszer auditor
Classification	(adat)osztályozás
Clearance	biztonsági ellenőrzés
CMM, Capability Maturity Model for Software	szoftverrendszer fejlesztési képesség érettségi modellje
Coaching	személyes képességfejlesztés mentori gesztori tevékenység
COBIT	az informatika és a kapcsolódó technológiák ellenőrzési eljárásainak célkitűzései Az Információhoz és kapcsolódó technológiához tartozó kontroll célkitűzések
Code of conduct	etikai szabályok
Collating	egybevetés egyeztetés összeolvasás
Commission	hivatalosan megbíz
Commissioning	projekt erőforrással ellátása
Communicated	közölt, tájékoztatást adott
Communicating	ismertet tájékoztat közöl; tájékoztatást/ ismerttetést ad nyújt; terjeszt információt terít
Compensating control	kiegészítő ellenőrzési mechanizmus
Competency profile	szakértelem (leírása)
Compliance	megfelelőség (ld. ISO 9000, ISO 27000, ISO 9126, stb.), könyvitel: szabályosság

Compliance (with standards)	megfelelőség (a szabványoknak és egyéb előírásoknak) külső előírások, követelmények betartása
Compliance checking	megfelelőség ellenőrzés
Compliant	(előírásoknak) megfelelő, szabvány szerinti
Component	(alkotó/rendszer) elem, (szoftver) komponens
Component based approach	komponens-alapú megközelítés
Comprehensive	mindenre kiterjedő
Compromise	(az előírások, követelmények, elvek szabályok) megsértése (kriptográfiai kulcs) nyilvánosságra kerülése
Computer based documentation	elektronikus dokumentáció
Conceptual design	Fogalmi szintű terv
Concern	aggály
Condensed	tömör
Confidentiality	titkosság bizalmas (és titkos) adatok kezelése bizalmas (és titkos) adatkezelés
Confidentiality	Bizalmas (és titkos) adatok kezelése, bizalmas jelleg
Configuration baseline	alapkonzfiguráció, báziskonfiguráció (hardvernél, operációs-rendszerrel) termékmérföldkő (rendszerfejlesztési projektekben)
Configuration data management	konfiguráció adatainak kezelése
Configuration documentation	konfiguráció dokumentáció / dokumentációja
Consistency	szabványokkal összhangban álló
Consistent	(ön)ellentmondásmentes egymással összhangban álló

Console log	operátori konzol terminál naplója
Console log	konzol napló
Context	kontextus, szöveggörnyezet
Contingency planning	rendkívüli eseményekre vonatkozó tervek katasztrófa elhárításra vonatkozó tervek
Contingency test	rendkívüli helyzet kezelésének tesztelése
Continous service	folyamatos szolgáltatás a szolgáltatás folyamatossága
Continuity plan	üzemvitel, üzletvitel folyamatossága fenntarthatóságának tervezése
Contractually negotiated liabilities	szervződésben kikötött kötelezettségek
Control	irányítási és ellenőrzési eljárás(ok) ellenőrzési mechanizmus (Informatika irányítása és audit)
Control (másodszor, de más szemszögből!)	irányítás vezérlés ellenőrzés kézben tartás felügyelet (létesítménygazdálkodás) . (Kibernetikai, műszaki, informatikai területeken)
Control framework	irányítási és ellenőrzési keretrendszer
Control issues	irányítási és ellenőrzési mechanizmussal kapcsolatos ügyek
Control loop	ellenőrzési ciklus vezérlési ciklus irányítási ciklus
Control loop	ellenőrzési ciklus
Control measures	ellenőrzési intézkedések
Control Model	Vezérlési Modell Irányítási Modell
Control objectives	ellenőrzési és irányítási célkitűzések ellenőrzési és vezetési célkitűzések

Control Objectives	irányítási és ellenőrzési célkitűzések
Control Objectives for Information and Related Technology	az informatika és a kapcsolódó technológiák ellenőrzési eljárásainak célkitűzései
Control Observations	az ellenőrzési és irányítási mechanizmusok megfigyelése
Control practices	ellenőrzési és irányítási eljárások
Control provisions	ellenőrzési és irányítási előírások
Control requirement	ellenőrzési követelmény
Control statement	irányítási és ellenőrzési eljárások célkitűzéseinek megfogalmazása
Control totals	„mindösszesen” ellenőrző összeg
control-based	ellenőrzési alapú
Convert site	telephely átalakítása
Corporate asset	vállalati eszközök vállalat vagyona
Corporate data model	vállalati adatmodell
Corporate resource	vállalati erőforrás
Correct	korrigál helyesbít, kijavít
Corrective action	helyreigazító tevékenység
Corroborative evidence	megerősítő bizonyíték
Corruption of data	adatok károsodása
Cost benefit analysis	költség haszon elemzés
Cost management	Költség gazdálkodás
Cost monitoring	költségek nyomon követése
Cost performance	költségvetési terv teljesítése
Cost/performance ratios	költség/teljesítmény értékarány
Cost-accounting	költségelszámolás költségkönyvelés
cost-effective	költség-takarékos gazdaságos

Costing	költségtervezés
Covenant	szerződés
Critical	kritikus fontosságú, létfontosságú
Critical Success Factor	kritikus sikertényező (KST)
criticality	kritikus fontosság
Custodian of data	adatfeldolgozó felelősség (Lásd 1992. évi LXIII. Törvény a személyes adatok védelméről és a közérdekű adatok nyilvánosságáról.)
Custody of data	az adatokról való gondoskodás az adatokért való felelősség az adatok őrzése
Customer	Ügyfél
Customer (BSC)	Vevői nézőpont (Kiegyensúlyozott mutató számrendszer)
Cut-off points (operations)	leállítási pontok (rendszer) [Üzemeltetésben]
Cyber threats	kibertérből származó fenyegetések
Cyberspace	Kibertér
Damage	káresemény
Dashboard	műszerfal
Database Repository	adatszótár
Data classification schema	Adatosztályozási séma (biztonsági szempontból)
Data control	az adatok ellenőrzése (adat bevitelnél és adatkimenetnél)
Data control quality	az adatok minőségének ellenőrzése
Data dictionary	adatszótár
Data ownership	adat-felelős
Data ownership policy	1. Adatfelelősökre / adatkezelőkre vonatkozó irányelvek
Data redundancy	adat redundancia
Data validation	adat helyesség ellenőrzése (tartalmi)

Data verification	adat (formai) helyesség ellenőrzése
Decision support systems	döntés támogató rendszerek
Defects of data	adathibák
Deficiency	gyengeség hiba probléma
Deficiency of process	a folyamat hiányossága
Defined	szabályozott
Defined process	szabályozott folyamat
Deliverables	leszállítandó (projekt) termékek
Delivery process	szolgáltatás nyújtás folyamata
Demand	kereslet
Denial of service attack	túlterheléses támadás a szolgáltatások felfüggesztésére irányuló támadások
Deploy	telepít
Design (system, software)	(rendszer / szoftver) terv ; (műszaki dologra)
design specification (application, sw acquisition development)	szoftver specifikáció
Detection method	feltérési (audit), észlelési (biztonság) módszer
Detection risk	feltérési kockázat (audit), észlelési kockázat (biztonság)
Deviation (from)	jelentős eltérés (vmitől)
Direction	útmutatás
Disaster recovery plan	Katasztrófa utáni helyreállítási terv
Discharge	felment
Discipline	szakterület
Disclaimer	Felelősségelhárítás
Disclosure	nyilvánosságra hozás, feltárás, (nyilvánosságra hozatal, ha

	disclose to the public), felfedés
Discrepancy	az egyezés hiánya, eltérés, különbözőség
Discretionary authority	saját belátására bízott döntési jog (kijelölt vezető)
Dispose	megszabadul, rendelkezik
Disruption	üzemzavar, üzemszünet
Distraction	figyelemelterelés
Distribution	(szoftver) terítés ellátás elosztás
Divestiture	jogfosztás
Domain	szakterület
Domain expert	A terület szakértője
Driver	szervezeti, üzleti paraméter tényező hajtóerő, ösztönző tényező
DRP	katasztrófa-helyreállítási tervezés
Due diligence	kötelező gondosság elvárható gondosság
Due professional care	kötelező gondosság elvárható szakmai gondosság
Effective	eredményes
Effectively	ténylegesen eredményesen hathatósan hatásosan
Effectiveness	eredményesség
Efficacy	hatásos (mint gyógyszer) eredményes eredményesen vált ki hatást
Efficient	hatékony
Electronic business	Elektronikus üzletvitel
Electronic commerce	Elektronikus kereskedelem
Electronic signature and certification	az elektronikus aláírás és a tanúsítványa
Embezzlement	hűtlen kezelés (jogilag, és könyvvizsgálat szempontjából)
Emergency Changes	rendkívüli változtatás

Emergency situations	vészhelyzet veszélyhelyzet rendkívüli helyzet
Empowerment	hatáskörök átadása feladatok átruházása
Enabler(s)	(a lehetőségek kihasználását) lehetővé tevő tényezők
Encryption	rejtjelezés sifírozás algoritmikus adatvédelem kriptográfiai adatvédelem titkosítás
End-to-end	végponttól-végpontig
Enforced (process)	betartatható betartatott (folyamat)
Engagement	kötelezettségvállalás
Enhancement of process	folyamat továbbfejlesztése
Ensure	biztosít
Entity	entitás
Environmental controls (facility management)	a környezet felügyelete (létesítménygazdálkodás) a környezet figyelése, megfigyelése
Error-prone	meghibásodásra hajlamos
escalation path	
Escalation procedures	felterjesztés (probléma) továbbítás (probléma) felsőbb szintre küldés szolgálati út a felterjesztésre felterjesztési szolgálati út
Escrow agreement	letéti szerződés (szoftver, algoritmus, kriptográfiai kulcsra vonatkozóan)
Established methodology	elfogadott módszertan
Evaluation	kiértékelés
Ex-ante	előzetes
Excellence	kiválóság
Exception	kivétel, rendkívüli esemény

Exception analysis	rendkívüli események elemzése
Exception handling	rendkívüli események kezelése
Exception reports	rendkívüli eseményekről történő jelentés
Executive	felső vezető/vezetés
Executive Summary	Vezetői összefoglaló
Expertise	szaktudás
Ex-post	(megteendő) intézkedés lista
Exposure	veszélyeztetettség
Expected error-rate	várható hibaarány
Extent of breaches of continuous service	A folyamatos szolgáltatásban bekövetkezett üzemszünet kiterjedése
Facilitated assessment	irányított értékelési eljárás
Facilities	létesítmények (az összes IT erőforrásnál)
Facilities management	létesítmény gazdálkodás
Facility environmental control	létesítmények / berendezések környezet felügyeleti eljárásai (pl. térfigyelő rendszerek)
Failure	meghibásodás
Fallback	visszatérés
Feedback	visszacsatolás
Fiduciary	pénzügyi megbízhatóság, pénzügyi bizalom
Field work	helyszíni ellenőrzés (audit munkánál)
File	állomány, fájl
Financial (BSC)	Pénzügyi nézőpont
Findings	feltárt tények megállapítások
Findings during audit	A feltárt hiányosságok (az auditálás, ellenőrzés során) megállapítások a vizsgálat során
Finger pointing	Egymásra mutogatás (egymás vádolása)
Fire alarm	tűzjelző
Focus areas	Kiemelt fontosságú területek (informatikai irányítás)

Follower	követő (stratégia), technológiailag követő
Follow-up	utóellenőrzés utómunkálatok elvégzésének leellenőrzése folyamatos nyomon követése az auditálási javaslatok végrehajtásának utómunkálatok
follow-up activities	Utómunkálatok (projektirányításban, minőségbiztosításban)
Framework	Keretrendszer
Fraud	csalás
Full life-cycle project methodology	Teljes rendszerfejlesztési-életciklusra vonatkozó projektmenedzsment módszertan
Function points	Funkció pontok (alkalmazási rendszerek méretének mérésére)
Gap analysis	Eltérés elemzés Fennálló különbség elemzése
Going concern concept	működés folyamatosságának elve
Go-live	éles indulás
Good practice	bevált gyakorlat
Governance	irányítás
Granularity	finomság, részletesség
Guidelines	Útmutató
Hacking	számítógépes betörés
Health and safety	munkahelyi egészség és biztonság
Health and Safety Authority	Munkavédelmi Felügyelőség Munkahelyi biztonság és egészség felügyelő hatósága
Help-desk	(informatikai) ügyfélszolgálat
Housekeeping	rend fenntartó eljárások (biztonsági, informatikai értelemben). Windows housekeeping: a biztonsági rendtartás betartása.
Human resources management plan	személyzeti terv

Identity Management	Személy azonosítás kezelése
Image systems	dokumentumok elektronikus képét rögzítő és tároló rendszerek
Impact	hatás
Impacts	hatások, következmények
Imperatives	alapkövetelmény
Implementation	Megvalósítás
Impractical	(gyakorlatban) használhatatlan
Improvement	továbbfejlesztés
Inception (of project)	Projekt indítása
Inception report	Projekt alapító dokumentum
Incident (security, ITIL)	esemény, rendkívüli esemény
Incompatible	inkompatibilis
Inconsistency of data	ellentmondásos adatok adatokban ellentmondások vannak nincs összhang az adatok között
Inconsistency of data	ellentmondásos adatok inkonzisztens adatok
Incorporate	figyelembe vesz (pl. döntések során szempontot)
Incremental measurement scale	Fokozatosan növekedő mérési skála
Independent assurance	független értékelés, független megerősítés
Indicator	mutató(szám)
Industry leaders	iparági vezető szakemberek / vállalatok
Information architecture	információ-architektúra
Information assets	informatikai eszközök informatikai vagyon információ vagyon
Information control environment	informatikai ellenőrzés környezete
Information criteria	információ-kritériumok

Information management	információmenedzsment
Information model	Információmodell
Information security	informatikai biztonság információbiztonság
Information system	Információrendszer
Information system access	információrendszerek hozzáférés jogosultságai
Information system security	információrendszerek biztonsága
Information Systems Audit and Control Association	Információrendszer Ellenőrök Egyesülete
Information systems planning	Információrendszer projekt portfólió tervezése
Inherent IT risks	az informatikával együtt járó belső (inherens) kockázatok
Initiative	Javaslat előterjesztés kezdeményezés
Innovator	technológiai élenjáró
Institute	intézményesít bevezet
Insurance codes	biztosítási szabályok biztosítási előírások
Insurance policy	biztosítási kötvény
Integration testing	integrációs tesztelés
Integrity	sértetlenség, sérthetlenség <adatok, információk esetében> összhang <információrendszerek, rendszerelemek, adatelemek között> adatok, rendszerek épsége
Integrity (még egyszer,	becsületesség, tisztesség <emberekre, alkalmazottakra> er-

más szemszögből)	kölcsi tartás <emberekre, alkalmazottakra>
Intellectual property	szellemi tulajdon
Interface	kapcsoló felület, csatoló, interfész
Internal (BSC)	Működési folyamatok nézőpont (balanced scorecard)
Internal audit	belső ellenőrzés
Internal control (system)	belső irányítási és ellenőrzési rendszer
Internal control environment	belső irányítási és ellenőrzési rendszer környezete
Internal control review	belső irányítási és ellenőrzési rendszer felülvizsgálata
Internal control system	belső ellenőrzési és irányítási rendszer
Internal controls	belső ellenőrzési és irányítási eljárások, eljárásrend, mechanizmus belső szabályzatok
Internal controls	belső irányítási és ellenőrzési rendszer
International Standard Guidelines	a nemzetközi szabványok útmutatói
International Standard Guidelines	Nemzetközi szabványok útmutatói
Interoperability	együttműködési képesség
Inventory (of configuration elements)	(konfigurációs elemek) leltára
Investment management	Informatikai beruházások kezelése
IT	informatika
IT Assurance	informatika értékelése
IT Control Diagnostic	Az informatikai irányítási és ellenőrzési rendszer felmérése
IT disruption	(informatikai) üzemszünet
IT governance	Informatika irányítása
IT initiative	informatikai részleg/funkció javaslata

IT management	Informatikai vezetése
IT operative plan	informatikai üzemeltetési terv
IT organisational standards	Informatikai részleg szervezeti működési szabályzata
IT production	Informatikai szolgáltatás (nyújtása)
IT project plan	informatikai projekt terv
IT Quality Plan	informatikai minőségügyi terv
IT strategic plan	informatikai stratégiai terv
IT tactical plan(s),	informatikai taktikai terv
IT value management	informatikai érték menedzsment
Job (operations)	Kötegelt adatfeldolgozási folyamat (Üzemeltetés)
Job Scheduling	A kötegelt adatfeldolgozási műveletek (jobok) ütemezése
Key Goal Indicator	kulcsfontosságú célmutató (KCM)
Key Performance Indicator	kulcsfontosságú teljesítménymutató (KTM)
Knowledge management	tudásmenedzsment
Knowledge-based system	ismeretalapú rendszer
Lead times	átfutási idő
Leading edge	élenjáró (technológiai szempontból)
Learning / Innovation (BSC)	Tanulási és fejlődési Nézőpont
Least access as required	a szükségesnél nem több hozzáférési jogosultság
Legal entity	Jogi személy(iségű társaság)
Leveraged resources	kiaknázott felhasznált <erőforrások>
Leveraging	kiaknáz, kihasznál, hasznosít kezdeményez felhasználás ösztönzése

Liabilities	kötelezettségek
Life cycle methodology	(Rendszerfejlesztési) életciklus módszertan
Local building codes	Helyi építési szabályzat / szabályok
Locations of wiring used	kábelek elhelyezése (kábelszekrények, kábelrendező, csatlakozó aljzatok, stb)
Low-profile site	Az informatikai telephelyre vonatkozó információk elrejtése Nem feltűnő, fedett telephely
Maintain	karbantart napra készen tart fenntartja <működőképességet, aktualitását stb.> napra készen tart: szabványt dokumentumot, adatbázist, adatot karbantart: hardver, hálózati elemek, szoftver rendszeres állag fenntartó, és egyéb hibák kijavítása
Malicious	kártékony (szoftver), ártó (szándék)
Manage quality	Minőség irányítása
Management Awareness Diagnostic	A vezetői tájékozottság felmérése
Management Guidelines	Vezetői útmutató
Management processes	Vezetési/irányítási folyamatok
Management reporting	Vezetőknek szóló jelentések
Management tools	Vezetési eszközök Rendszergazdai eszközök
Mass storage media	tömeges adattárolásra alkalmas elektronikus adathordozók
Master file	törzsadatállomány
Material	lényeges
Measure	mértékek, metrikák
Measurement-driven	mérték-alapú
Measures	intézkedések
Measuring metrics	a metrikák (mérési eszközök, eljárások, módszerek) kalibrálá-

	sa
media	adathordozók (adatkezelésnél). Görög (latinban) többes szám!
medium	adathordozó (adatkezelésnél). Görög (latinban) egyes szám!
Migration strategy	Migrációs stratégia (adat, adatállományok, stb.), áttérés stratégiája (technológia)
Minutes	jegyzőkönyv
Misuse	visszaélés
Mitigation	enyhítés
Monitor production	Nyomon követik az élesben üzemelő rendszert
Monitoring	nyomon követ megfigyel figyelemmel kísér monitoroz
moral	erkölcs
morale	munkahelyi légkör
Need to know basis	csak annyit tudjon mindenki, amennyi feltétlenül szükséges szükséges ismernie elv
need to know basis	ismeret szükségessége, hogy csak annyit tudjon mindenki, amennyit feltétlenül szükséges
Network layer	Hálózati réteg (OSI protokoll, TCP / IP)
non-compliance	meg-nem-felelősség nem megfelelősség
Non-disclosure agreement	titoktartási nyilatkozat
Non-fiduciary	hűtlen (kezelés)
Non-opinion	függő záradék
Non-repudiation	letagadhatatlanság
Off-line	Kapcsolat nélküli módban
Off-site storage	Külső adattárolási telephely
On-going	folyamatos
On-the-job training	munka közbeni képzés

Operating system	operációs rendszer
operation management (IT)	üzemeltetés vezetése (informatika)
Operational management	(vállalati) működés irányítása / vezetése; adminisztratív vezetés
Operational plans	működési tervek
Operations	üzemeltetés
Operations personnel	üzemeltetési személyzet
Organizational learning	Szervezeti tanulás
Outsourcing	külső szolgáltatóhoz történő kihelyezés <szervezeti tevékenység, funkció, feladat, folyamat> kihelyezés <szolgáltatásoké>
Outsourcing Contracts	Szolgáltatás kihelyezési szerződés kiszervezési szerződés
Owners of the business processes.	szervezeti folyamat felelősei
Ownership (Facility management)	tulajdonlás, tulajdonjog
Packaging (software)	szoftver csomag kialakítása
Participative reviews.	Közösen végzett felülvizsgálat
Pattern	mintázat, minta
Peer	társszervezet, partner (egyenrangú)
Penetration test	behatolás próba
Perform readiness test	Készre jelentési tesztelés végrehajtása
Performance	Teljesítmény
Performance management	Teljesítménymenedzsment
Performance needs	Teljesítmény igények
Performance require-	elvárt teljesítmény

ments	
Periodic reporting	Rendszeres / időszakos jelentések
Personnel safety	Munkavédelem (személyek, személyzet)
Phase	fázis (pl. RUP szerint végrehajtott projekteknél, a negyedek)
Physical layout	alaprajz fizikai elhelyezkedés tervrajza
Piecemeal (applications)	szigetszerű (alkalmazások)
Plan	tervkészítés (idő dimenzióban)
Plan, build, run, monitor	tervez, kivitelez, működtet, nyomon követ
Planning (system, solutions)	Kivitelezés tervezése (rendszer, megoldások)
Policy	irányelv
Policy statement	irányelv célkitűzéseinek megfogalmazása
Portfolio management	portfolió menedzsment
Positive assurance	megerősítő értékelés
Positive control framework	Pozitív visszacsatolási keretrendszer
Post implementation review	A megvalósítást követő felülvizsgálat
Preventive maintenance	megelőző karbantartás
Privacy (laws, regulation)	A személyes adatok védelmére vonatkozó (törvények és jogszabályok)
Privacy protections	Személyes adatok védelme
Private information	személyes jellegű adatok
Privilege	(kiemelt) jogosultság
proactive	kezdeményező jellegű kezdeményez, kezdeményezően reagál
Pro-active process	Megelőző jellegű folyamat
Probability proportional	a volumennel arányos valószínűség

to size	
Problem escalation	problémák felterjesztése (az illetékeseknek)
Process owner	folyamat felelős
Process ownership	folyamat felelősség
Process-oriented	folyamat-központú
Production environment	éles környezet
Productive time	Munkaidő (a termelés ideje)
Program	Alkalmazási /számítógép program
Programme management	Program irányítás (projektek összessége)
Project boundary	Projekt határa (szervezeti egységek, folyamatok értelmében megvont határ az adott projektre)
Project inception	Projekt megalapítása
Project master plan	projekt felsőszintű terve
Project Owners/Sponsors	Projekt felelősök / szponzorok, finanszírozók
Project Quality Manager/Coordinator	Projekt minőségügyi felelős / koordinátor
Project scope	projekt terjedelme (a kijelölt , leszállítandó, előállítandó termékekből áll)
Project Sponsors	Projekt finanszírozók
Project Team Leader	Projekt munkacsoport vezetője
Protect asset	Vagyon védelme
Provide reasonable assurance	biztosítékot ad / nyújt; garantál; szavatol
Public enterprise	Közzolgálati feladatokat ellátó szervezet
Push (technology)	Információ kézhez juttatása

	Felhasználóhoz juttatása
Qualitative	kvalitatív
Quality	minőségügy
Quality /compliance re-view	(minőségi) szemle
Quality assurance	minőségbiztosítás
Quality Assurance Coordinator	Minőségbiztosítási koordinátor
Quality assurance group	minőségbiztosítás csoport
Quality Assurance Review	Minőségbiztosítási felülvizsgálat
Quality control	minőség ellenőrzés
quality improvement	Minőségjavítás
Quality management	minőségügy, -irányítás
Quality Management	Minőségirányítás(i szervezet)
Quality management function / business unit / entity	minőségirányítási / minőségügyi szervezeti egység (adminisztratív funkció)
Quality manager	Minőségügyi vezető
Quality plan	Minőségügyi terv
Quality practices	minőségügyi eljárások
Quality procedure	minőségügyi módszerek
Quality Review	minőségi szemle
Quantitative	kvantitatív
Reactionary	(valamire) reagáló

Reactive	(késve, utólag) reagál
Reactive development	tűzoltás jellegű fejlesztés
Reality check	megalapozottság vizsgálata
Realizing benefits	A hasznok realizálása
Reasonable	ésszerű
Recommendation (solutions)	ajánlás
Recovery site manager	a tartalék telephely vezetője
Recruit	toboroz
Recuperate	helyrehoz
Refinement	finomítás
Refreshing	frissítés(az egész anyagot, pl. adatbázist, operációs rendszert, teljes tervet, stb. lecseréljük)
Regulatory authority	szabályozó hatóság (pl. NHH)
Regulatory obligations	Hatósági szabályozási kötelezettségek, követelmények, rendeletekben előírt kötelezettségek
Release	kibocsátás (szoftver, verzió, alkalmazási rendszer stb.)
Release planning	kibocsátás tervezése
Reliability	megbízhatóság
Remedial action	kiigazító tevékenység; korrigáló, kijavító tevékenység, kiküszöbölő tevékenység
Remediation action plans	kockázat kiküszöbölési terv
Remote monitoring capabilities	Távoli beavatkozási és vezérlési lehetőség
Repeatable but intuitive	ismétlődő de ösztönös
request for change	Változtatási kérelem

Request for Proposal	1. Ajánlattételi felhívás 2. Felhívás ajánlattételre
Requests for change	Változtatási kérelem
Requirement analysis	Követelmény elemzés
Requirement definition	Követelmény meghatározás
Requirement specification	Követelmény specifikáció
Residual risk	maradványkockázat
Resilience	alkalmazkodási képesség (a környezethez, meghibásodás, vagy vészhelyzet esetén)
Respond	reagál
Response time	Válaszidő (felhasználó kérésére milyen gyorsan reagál a számítógép)
Responsibility	hatáskör feladatkör
Responsiveness	reagálóképesség
Resumption planning	(üzleti) tevékenység helyreállítására / folytatására vonatkozó tervezés
Retrieve	visszakeres
Revenue	árbevétel
Review	felülvizsgálat, szemle (projektirányításban és fejlesztésben)
Rework	átdolgozási munka
Risk analysis	kockázatelemzés
Risk appetite	kockázat vállalás (vállalási képesség)
Risk assessment	kockázatfelmérés /értékelés
Risk avoidance	kockázatelkerülés
Risk evaluation	kockázatkiértékelés
Risk identification	kockázatazonosítás/- felismerés
Risk limits	kockázatvállalási határértékek
Risk management	kockázatkezelés

Risk reduction	kockázatcsökkentés
Risk remedial	kockázat kiküszöbölés
Risk scenarios	Kockázattal kapcsolatos forgatókönyvek
Risk tolerance.	kockázat tűrőképesség
Risk tolerant	kockázat tűró
Roadmap	megvalósításhoz vezető út terve
Road-tested	kipróbált
Robust	hibatűrő (rendszer) robosztus
roll-back	visszaállítás
Roll-out phases	teljes körű bevezetés fázisai
Root-cause analysis	A problémák gyökerét feltáró elemzés (azaz az oksági fán végig mennek a gyökeréig)
Rudimentary	kidolgozatlan
Run-to-run	adatfeldolgozás különböző szakaszai közötti adatátadás
Safe	biztonságos
Safeguard	óvintézkedés
Safety	munkavédelem (ellenőrzés, auditálás, igazgatási területeken)
Scalable	skálázható
Schedule	ütemterv
Scope of variable	a változó hatóköre
Scorecard	mutatószám rendszer
Seamless	zökkenőmentes; akadálymentes
Security	biztonság
Security administration	(informatikai) biztonsági felügyelet, szolgáltatás
Security Baseline	alap biztonsági szint biztonsági kiindulási helyzet
Security Baselines	biztonsági minimumok (követelmények, előírások)

Security certification	folyamat: biztonsági tanúsítás; eredmény: biztonsági tanúsítvány; személyre vonatkoztatottan: biztonsági igazolás
Security exercises	biztonsági gyakorlatok
Security guard	biztonsági őr
Security policy	biztonsági irányelvek
Security policy statements	biztonsági szabályzat
Security practices	biztonsági eljárások biztonsági eljárásrend
Security rules	biztonsági szabályok
Security standards	biztonsági szabályzat
Segregation (of duty)	a feladatkörök szigorú elválasztása
Segregation (of duty)	A feladatkörök elkülönítése
Sensitive information, data	bizalmas információ, adat kényes információ, adat
Sensitive output	bizalmas kimeneti adatok
Sensitive position (within an organisation)	Bizalmi munkakör
Sensitive problem	kényes probléma (<biztonsági szempontból>)
Sensitivity assessment	rendszerbiztonsági kockázat felmérés
Service agreement / contract	szolgáltatási szerződés
Service level agreements	szolgáltatási szint megállapodás (SzSzM)
Service level manager	szolgáltatási szint felelős
Service offerings	szolgáltatás kínálat
Settlement	peren kívüli egyezség
Shared functions and re-	közösen használt funkciók és erőforrások

sources	
Sign off	jóváhagyás (aláírással hitelesítve) elfogadás átvétel (informatikai rendszer) igazolás (pl. teljesítési jegyzőkönyv)
Single point of failure	egyetlen forrásra visszavezethető hibalehetőség
Skill	szakmai képesség szakmai gyakorlat
Social engineering	emberek bizalmára és/vagy hiszékenységére építő kapcsolat-építés (gyakran csalásra, vagy előnyszerzésre használják)
Software coding	szoftver programozás program kód írása
Software distribution	Szoftver terítés ellátás elosztás
Software engineer	szoftvertervező mérnök
software engineering	szoftvertervezés
Software Engineering Institute	Szoftver Tervezési Intézet
Software Release	(új) szoftver (változat) kibocsátás
Sophisticated	kidolgozott
Sound	megalapozott
Sourcing strategy	forrásbiztosítási stratégia
Specification	Specifikáció szabatos műszaki leírás
Specify	specifikál pontosan részletekbe menően leír pontosan meghatároz
Spoofed email	manipulált elektronikus levél más nevében hamisított elektronikus levél
Stage	szakasz (projekt)
Stakeholders	érdekelt felek
Standard	Szabvány (műszaki, informatikai értelemben) szabályozás szabályzat
State-of-the-art	A (műszaki/ informatikai) tudomány mai állása szerinti
Status report	helyzetjelentés

Statutory safeguards	törvényben meghatározott biztosítékok jogi garanciák
Storage	adattároló
Strategic alignment	Stratégia illesztése (nem „integration”, ami az összehangolás)
Strategic options	a stratégia alternatívái
Strategic plan	stratégiai terv
Striking a balance	az egyensúly megtalálása az ésszerű kompromisszum megtalálása
Substantive testing	helyesség és sértetlenség ellenőrzése (tranzakcióké, adatoké stb.) alapvetően lényeges tulajdonságok vizsgálata; tesztelése
Succession plans	Munkaerő felvétel tervezés utánpótlás tervezés
Supervision	1. Számonkérés (munkafeladatok) Ellenőrzés (munkáé)
Supplier	szállító
Support (of product)	termékkövetés
Support services	A felhasználókat segítő szolgáltatás (az informatikai ügyfélszolgálattal kapcsoltban!)
Sustain	alátámaszt
Sustainable	életképes (megoldás) fenntartható (fejlődés)
Symptom	Jelenség (informatikában!)
System Administration	rendszeradminisztráció
System implementation	rendszer megvalósítás
System management	rendszerfelügyelet
Systematically	módszeresen, szisztematikusan, rendszerszerűen
Tamper	szabotálás módosítás(engedély nélkül) hamisítás megbolygatás
Task force	konkrét feladatra létrehozott munkacsoport
Technological developments	műszaki fejlesztések

Technology planning	a technológia tervezése
Technology research	a technológia fejlődés figyelemmel kísérése
Test plan entry and exit criteria.	teszt indítási és befejezési kritériumok kezdeti és befejezési kritériumok
The IT policies and procedures	Informatikai irányelvek és eljárások
Third-party	Külső fél (szállító, szolgáltató) [az informatikai és a többi szervezeti egységtől különböző]
Threats (security)	fenyegetés fenyegetettség (biztonságot)
Threshold	küszöb, határérték
Threshold model	Lépcsős modell
Throughput	tranzakció feldolgozási képesség áteresztőképesség
Timelines of development effort	a fejlesztési mérföldkövek időbeosztása
to assure	biztosítani, garantálni, szavatolni
to incrementally benchmark	fokozatosan végezve az összehasonlítást
Token	(biztonsági) hardver eszköz (intelligens kártya, USB stb.)
Tolerance	tűrőképesség tolerancia
Total cost of ownership	a tulajdonlással járó összes költség
Track	nyomkövetés
Trade-off	kompromisszum
Transfer price	Belső elszámoló ár
Transition to production status	Az éles üzemi állapotra való áttérés
Transmit	továbbít
Transparency (costs)	átláthatóság (költségek, költségvetés)

Transportation layer	szállítási réteg (OSI protokoll, TCP / IP)
Trend	tendencia
Trojan horse	trójai faló típusú program
Trusted	megbízható
uberrimae fidei (in the utmost good faith)	a legteljesebb jóhiszeműség
Ultimate deliverable	a végeredményként /végtermékként leszállítandó termék (projekt végterméke)
Unauthorized	jogosulatlan
Unavailability	használhatatlanság
Unbiased	elfogulatlan
Under pinning contract (UC)	alátámasztó szerződések
Understanding	megértés
Uninterruptible Power Supply	Szünetmentes áramforrás
Unit cost	fajlagos költség darabár egységár
Update	aktualizálás napra készre tesz
Upgrade (of software)	frissítés (szoftver)
Uptime	üzemidő
User Account	felhasználói fiók
User account management	a felhasználók nyilvántartásának kezelése
User interface	felhasználói felület
User interface standards	felhasználói felület szabványa
Utilisation	kihasználtság
Utilities	kiegészítő, illetve karbantartó szolgáltatásokat nyújtó szoftverek; pl. adatmentés, FTP, Telnet stb.

Utilities (facilities management)	közművek
Validate	helyességet igazolja jogi érvényességet igazol az előírásoknak való megfelelést, elfogadhatóságot hivatalosan igazolja helyességet megvizsgálja és igazolja
Validate technology	A technológia alkalmazhatóságát megvizsgálja
Value delivery	Érték előállítás érték termelés érték létrehozása
Value delivery	Érték előállítás
Value Drivers	érték teremtő tényezők / vállalati belső értékteremtő tényezők
Value proposition	érték előállítására ajánlat
Variable costs	Változó költségek
Vendor	szoftver szállító szerződéses partner
Verify	Helyességet és a pontos megfelelést ellenőriz . Helyességet és a pontos megfelelést igazolja
Vulnerability	sebezhetőség
Workflow	Munkafolyamat (szervezés)
workload schedule modification	munkaterhelés-ütemezés módosítás
Workshop	műhelymunka
Wrap-up Procedures	Folyamatlezárási eljárások

15.1 Rövidítések

B2B	Business To Business
B2C	Business To Consumer
BAM	Business Activity Monitoring
BPEL	Business Process Execution Language

BPM	Business Process Management
BPMS	Business Process Management Solution
BPMN	Business Process Modelling Notation
BSD	Business Services Directory
COBOL	COmmon Business Oriented Language
EJB	Enterprise Java Beans
ESB	Enterprise Service Bus
G2B	Government-To-Business
G2C	Government-To-Citizen
GUI	Graphical User Interface
HCA	Hub-Centered Architecture
HCD	Hub-Centered Design
HCDP	Hub-Centered Design Patterns
JAVA J2EE	Java 2 Platform, Enterprise Edition
OWSM	Oracle Web Service Manager
SOA	Service Oriented Architecture
SOAP	Simple Object Access Protocol
SSL	Secure Socket Layer
TCP/IP	Transmission Control Protocol/Internet Protocol
UDDI	Universal Description, Discovery, and Integration

WSDL	Web Service Definition Language
XML	Extensible Markup Language
TACACS	Terminal Access Controller Access-Control System

Lábjegyzetek/Végjegyzetek

¹ *“The term architecture is used here to describe the attributes of a system as seen by the programmer, i.e., the conceptual structure and functional behavior, as distinct from the organization of the data flow and controls, the logical design, and the physical implementation.”*

² Business/IT alignment problem

³ Magyar nyelven az üzleti / nagyvállalati architektúra (business / enterprise architecture) fogalmának használata a nem nyereség központú (non-profit), közszolgálati, köz- és államigazgatási területeken viszsztatetszést kelt, ezért az általánosabb szervezeti architektúra fogalma elfogadhatóbb és ajánlhatóbb.

⁴ Itt a specifikáció a szabványokra illetve a szabványok műszaki mellékleteire, kiegészítéseire utal.

⁵ Kormányzati nyílt rendszerek összekapcsolási leírása. Ld. <http://www.itb.hu/ajanlasok/a7/>

⁶ The Internet Engineering Task Force (IETF). <http://www.ietf.org/>

⁷ Felhívás bírálatra – vitára bocsátott munkaanyag, javaslat.

⁸ Information Technology Management Reform Act of 1996 (ITMRA), http://en.wikipedia.org/wiki/Clinger-Cohen_Act

⁹ Netherlands Architecture Forum (NAF)

¹⁰ the activity of governing a country or controlling a company or an organization; the way in which a country is governed or a company or institution is controlled

¹¹ http://www.kipling.org.uk/poems_serving.htm . „Hat derék szolgát tartok. Ők tanítottak meg engem mindenre, amit csak tudok.”

¹² <http://zachmanframeworkassociates.com/Standards/protected/framework-graphic-3>. 2011-08-18

¹³ www.cio.gov

¹⁴ http://en.wikipedia.org/wiki/Enterprise_Architecture_framework

¹⁵ Ld [86].

¹⁶ <http://www.opengroup.org/architecture/togaf9-doc/arch/chap42.html> , Evaluation Criteria and Guidelines

¹⁷ <http://www.db.opengroup.org/sib.htm>

¹⁸ <http://pubs.opengroup.org/architecture/togaf9-doc/arch/chap35.html>;

http://people.inf.elte.hu/molnarba/Web_technologiak_IR-ben/El%5ad%e1s_anyag/legyzet_Tank%26nnyv/V%e1llalati_Architekt%26far%e1k_20110920.pdf

¹⁹ <http://www.eyeos.org/>

²⁰ A kulcsok generálása, létrehozása magasabb biztonsági szint érdekében csak megfelelően elkülönített szoftver és hardver eszközben történhet. Ugyanis a jelenlegi elterjedt operációs rendszerekből a veremben („stack”= operációs adathalma) keletkező kriptográfiai kulcsokat automatikusan továbbítják az Interneten keresztül: ez mind UNIX/LINUX mind a Windows változatokra igaz a patrióta törvény következtében.

²¹ http://hu.wikipedia.org/wiki/Elektronikus_al%C3%A1%C3%ADr%C3%A1s

²² Az EU tagállamokban az a megoldás terjed, hogy a minősített aláírás létrehozására alkalmas tanúsítványt olyan eszközre, adathordozóra helyezik, amelyeknek a biztonsági besorolása egy kicsit gyengébb, ezért a tanúsítvány és az adathordozó eszköz (intelligens kártya, USB) együttese csak fokozott biztonságúnak tekintendő, de ennek jogérvényességét kiegészítő szabályozással biztosítják.

²³ OASIS (Organization for the Advancement of Structured Information Standards): <https://www.oasis-open.org/>

²⁴ <http://openid.net/connect/>

²⁵ <http://oauth.net/>

²⁶ http://en.wikipedia.org/wiki/Web_Access_Management

²⁷ Create, Read, Update, Delete; Létrehoz, Olvas, Aktualizál, Töröl.

²⁸ <http://www.axis.hu/consulting/index.htm>

²⁹ <http://www.omg.org/>

³⁰ Az architektúra fogalmak esetében *tier*-t **szintnek**, a *layer*-t **rétegnek** fordítjuk, hogy a két fogalom magyarul is megkülönböztethető legyen.

³¹ Az angol elnevezéseket azért tartjuk meg, mert a rendelkezésre álló, beszerezhető eszközök mind „angolul beszélnek”, magyar nyelvű eszköz ezen a területen nem várható, a használhatóság és az egyes elemek összerendelhetősége érdekében ezért a kialakult angol szakkifejezéseket használjuk.

³² A csatlakozás tartalmi viselkedése, szemantikája az alkalmazott BPMN verziójától függ, a szabvány fejlődése során módosítások történtek.