Logikai architektúra és az UML komponens diagramok

UML és az architektúra mintázatok alkalmazása

Experience is that marvelous thing that enables you to recognize a mistake when you make it again.

—F. P. Jones

A tapasztalat az a csodálatos dolog, amely lehetővé teszi azt, hogy felismerjük a tévedésünket akkor, amikor újra elkövetjük.

Logic is the art of going wrong with confidence.

—Joseph Wood Krutch

Logika az a tudomány, amelyben feltétel nélkül bízva teljesen félrevezető eredményre jutunk.

Definíció (Meghatározás)

Szoftver architektúra:

Több formája van. A közös vonás az, hogy a nagy léptékű, „Nagy gondolatok, ötletek" megfogalmazása jellemzi, az egész rendszer és a jelentősebb részrendszerek tekintetében: rendszerszervezés, az architektúrát befolyásoló szempontok, stílus/megközelítés, mintázatok, felelősségek, együttműködés, kapcsolatok és motivációk

A definíció variációi

A szoftver fejlesztésben az architektúra szót mint egy fogalomra utaló főnevet illetve mint a tervezési cselekményre utaló igét is használják (architektúra tervezés).

Az architektúra fogalma mint főnév magában foglalja a rendszer jelentősebb elemeinek szerkezeti összefüggéseit és szervezését.

Mint cselekmény részben a vizsgálódásra, elemzésre részben a a tervezési munkára utal.

Definíció (Meghatározás)

Architektúra vizsgálata, elemzése: a rendszer tervét befolyásoló funkcionális és nem funkcionális követelmények elemzését tartalmazza.

Néhány elem: Piaci tendenciák, teljesítmény, költség és a fejlődés töréspontjai.

Architektúra tervezés: a fentebbi követelmények, szempontok, mozzanatok összehangolása és összeegyeztetése, a szoftver tervezésében.

Az architektúra dimenziói és nézetei

Az általános dimenziók a következők.

A logikai architektúra a rendszerszervezésének miként jét szakmai terminológia fogalmaival írja le: rétegek, komponensek, osztályok, (adat)kapcsoló felületek és részrendszerek.

Az üzembe helyezett, letelepített architektúra a rendszert a rendszer folyamatok értelmében, a folyamatok és az adatfeldolgozó egységek (kiszolgáló gépek, CPU stb.) valamint a hálózati elemek összerendelése révén.

Mit tekintünk rétegnek?

"A réteg az osztályok, komponensek vagy részrendszerek olyan durva felbontású csoportosítása, amely a rendszer bizonyos fontosabb jellemzőinek kohéziójáéért felel."

A magasabb szintű rétegek az alacsonyabb szintű rétegek szolgáltatásait veszik igénybe.

Rendszer, funkció belső kapcsolatának, kohéziójának erőssége

Kohézió gyenge foka

Esetleges

Logikai

Időbeli

Kommunikációs

Eljárási

Funkcionális

Kohézió erős foka

Kohézió foka

Esetleges kapcsolat: a rendszert alkotó részeket nem köti össze semmi felismerhető kapcsolat.

Logikai kapcsolat: a rendszer részei, tevékenységei ugyanolyan típusúak, de valahányszor a rendszert aktiválják, mindig csak az egyik részét jelölik ki végrehajtandónak.

Időbeli kapcsolat: a rendszer részeit, tevékenységeit ugyanakkor kell végrehajtani, vagy ugyanakkor kell kezdeni, de mind különböző jellegű.

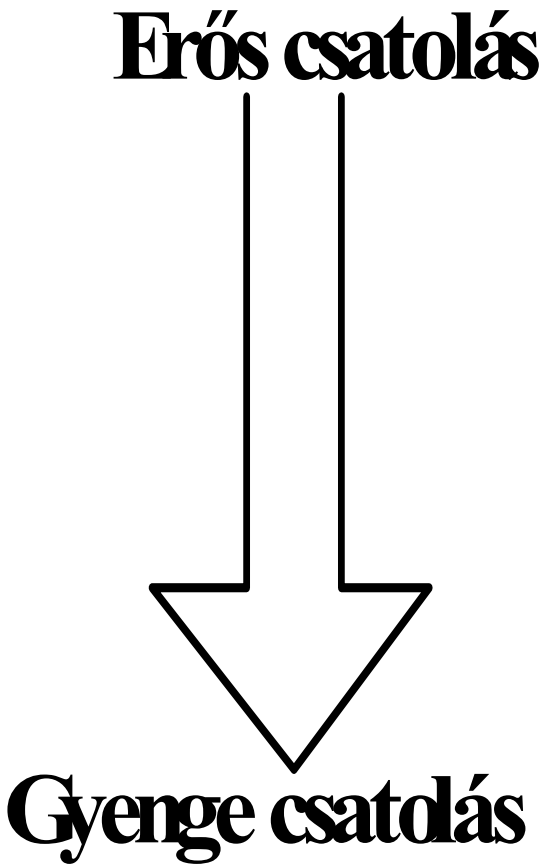
Kommunikációs kapcsolat: a rendszer részei, tevékenységei mind ugyanazt a dolgot ('entitást') használják.

Eljárási kapcsolat: a rendszer részeit, tevékenységeit mindig egy előre rögzített sorrendben kell végrehajtani.

Funkcionális kapcsolat: a rendszer részei, tevékenységei mind ugyanazt a célt szolgálják.

Csatolás: a függetlenség kritériumai

- Tartalmi
- Külső
- Vezérlési
- Közös rendszer
- Adat / információ



A csatolás ('coupling')

Tartalmi csatolás: a rendszer egy részét egy másik rendszer vagy részrendszer hívja meg és hajtatja végre a saját céljaira. Ennek megtételéhez vezérlési információk szükségesek, amelynek alapján a vezérlés a helyes időben tér vissza a kezdeményező rendszerhez.

Külső csatolás: az egyik rendszer vagy bizonyos rendszerek meg tudják változtatni egy másik rendszer működését az előbbiek irányítása alatt. Ezek a rendszerek nem közlik a megváltoztatott rendszerrel, hogy mi történik.

Vezérlési csatolás: a rendszer működését, tevékenységeinek végrehajtását egy vagy több rendszer irányítja. Az irányított rendszernek tudni kell, hogy kinél van a vezérlés pillanatnyilag.

Közös rendszer: két vagy több rendszer használ egy másik rendszert, aki viszont saját maga rendelkezik a saját erőforrásai felett.

Adat / információ csatolás: Ez a lehető leggyengébb csatolás. A rendszerek közti kapcsolat kimerül adat és / vagy információ cserében (esetleg valami anyag átadásában).

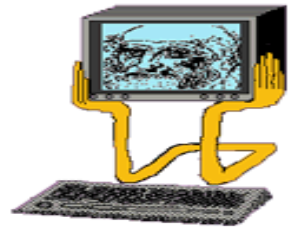
Architektúra mintázatok és mintázat kategóriák

Architektúra mintázatok: A nagyléptékű, nagy vonalú terv elkészítéséhez kapcsolódik, és tipikusan a fejlesztési folyamat korai iteratív lépéseiben alkalmazzák (a kidolgozás fázisában)

A tervezési mintázatok: az objektumok és az átfogó keretrendszerek közepes-léptékű és kis-léptékű terv készítéséhez kapcsolódnak.

Idiómák: a nyelvhez illetve az alacsony szintű, a megvalósításhoz kötődő tervekhez, megoldásokhoz kapcsolódnak.

Architektúra mintázatok : Rétegek ('layer')



A réteg mintázatok mögött meghúzódó elképzelés:

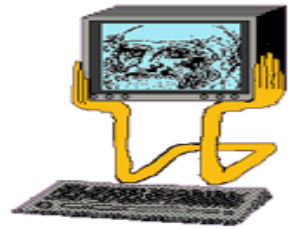
Szervezzük a rendszer nagy-léptékű logikai szerkezetét elkülönülő, az egymásért felelősséget viselő elemeket összefogó rétegekbe, amelyek világosan, és kohézív módon elhatárolják az illetékességeket úgy, hogy az alsóbb rétegek alacsony-szintű és általános szolgáltatásokat nyújtanak, a magasabb szintű rétegek sokkal alkalmazás specifikusabbak.

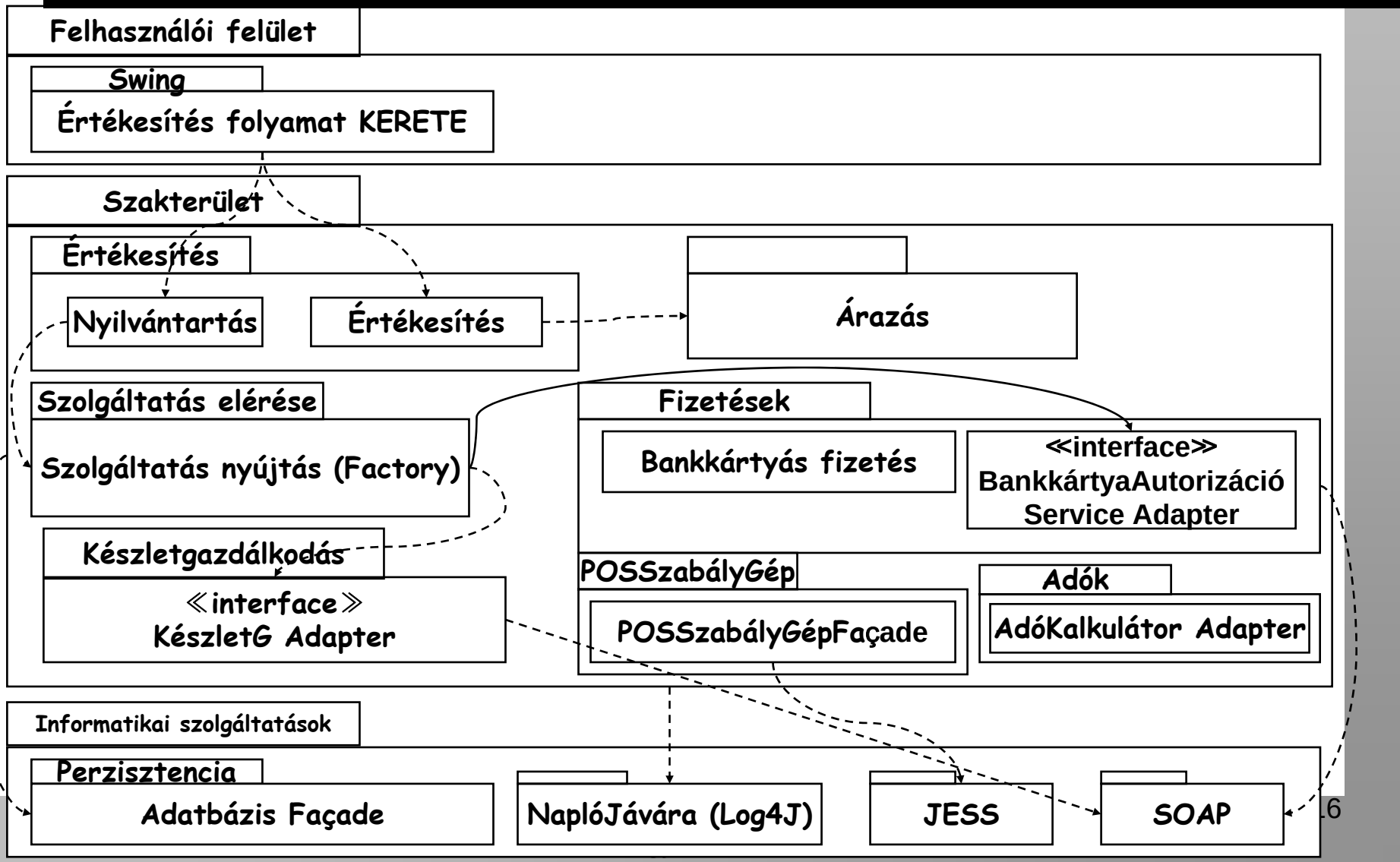
A rétegek közötti együttműködés és csatolás a magasabb szintű rétegektől az alacsonyabb szintű rétegek felé halad.

Rétegek közötti és komponensek között csatolás

A rétegek és a komponensek közötti kapcsolatok érzékeltetésére a logikai architektúra nézetet ábrázoló diagramot érdemes készíteni

A komponensek közötti részleges csatolás





Rétegek közötti és komponensek közötti kölcsönhatás

A rétegekben elhelyezkedő objektumok , rétegeken átívelő kapcsolatainak és kommunikációinak a dinamikáját hangsúlyozza.

A kölcsönhatás (interaction diagram) diagram a logikai nézetre helyezi a hangsúlyt, a rétegek között együttműködésre és a komponensek határaira.

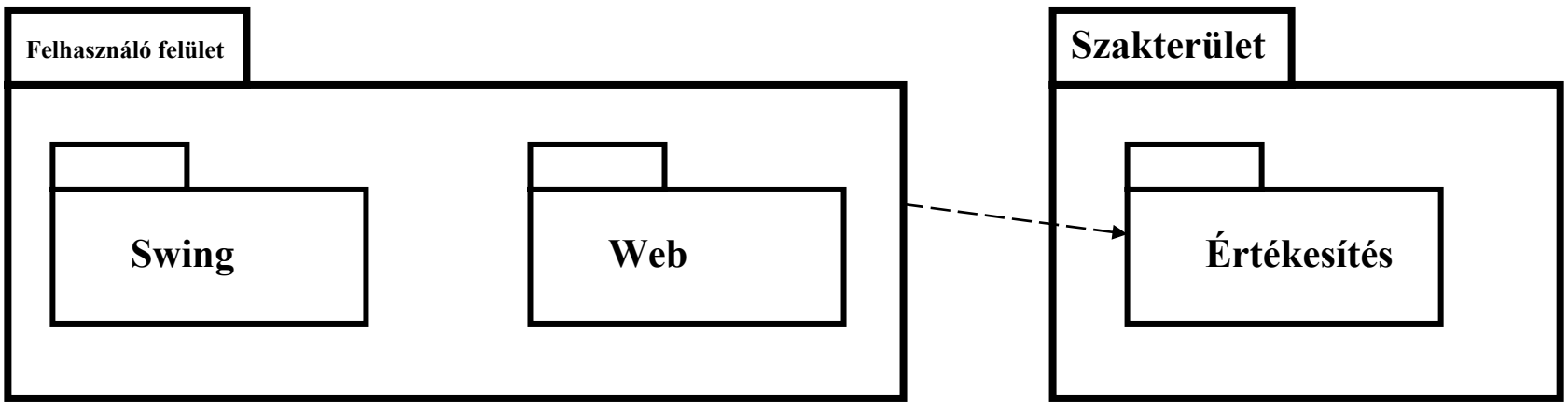
Façade : Homlokzat, olyan adatkapcsolati felület, amely eltakarja azt, hogy bizonyos körülményektől függően az adatfeldolgozás más jellegű lehet.

Komponens diagramok

Az UML komponens diagram gyakran jelzi a komponensek tartalmát, JAVA nyelvben, programozási környezetben ezt gyakran csomagnak (package) hívják.

A komponensek, csomagok természetesen egymásba ágyazhatók.

Komponens diagram



Logikai kontra folyamat kontra üzembe helyezett architektúra

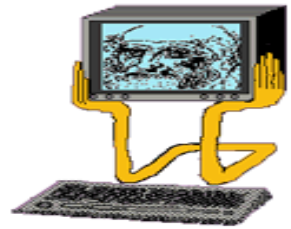
Az architektúra rétegek az architektúra logikai nézetét érzékeltetik.

Nem az üzembe helyezett, letelepített elemek és folyamatok viszonyát testesítik meg.

Az informatikai platformtól függően, az összes architektúra réteget ugyanabba a folyamatba, egy azon csomópontra is lehet telepíteni.

Vagy sok számítógép csomópontra szétosztva.

Terminológia: szint (tier), réteg (layer) és partíció

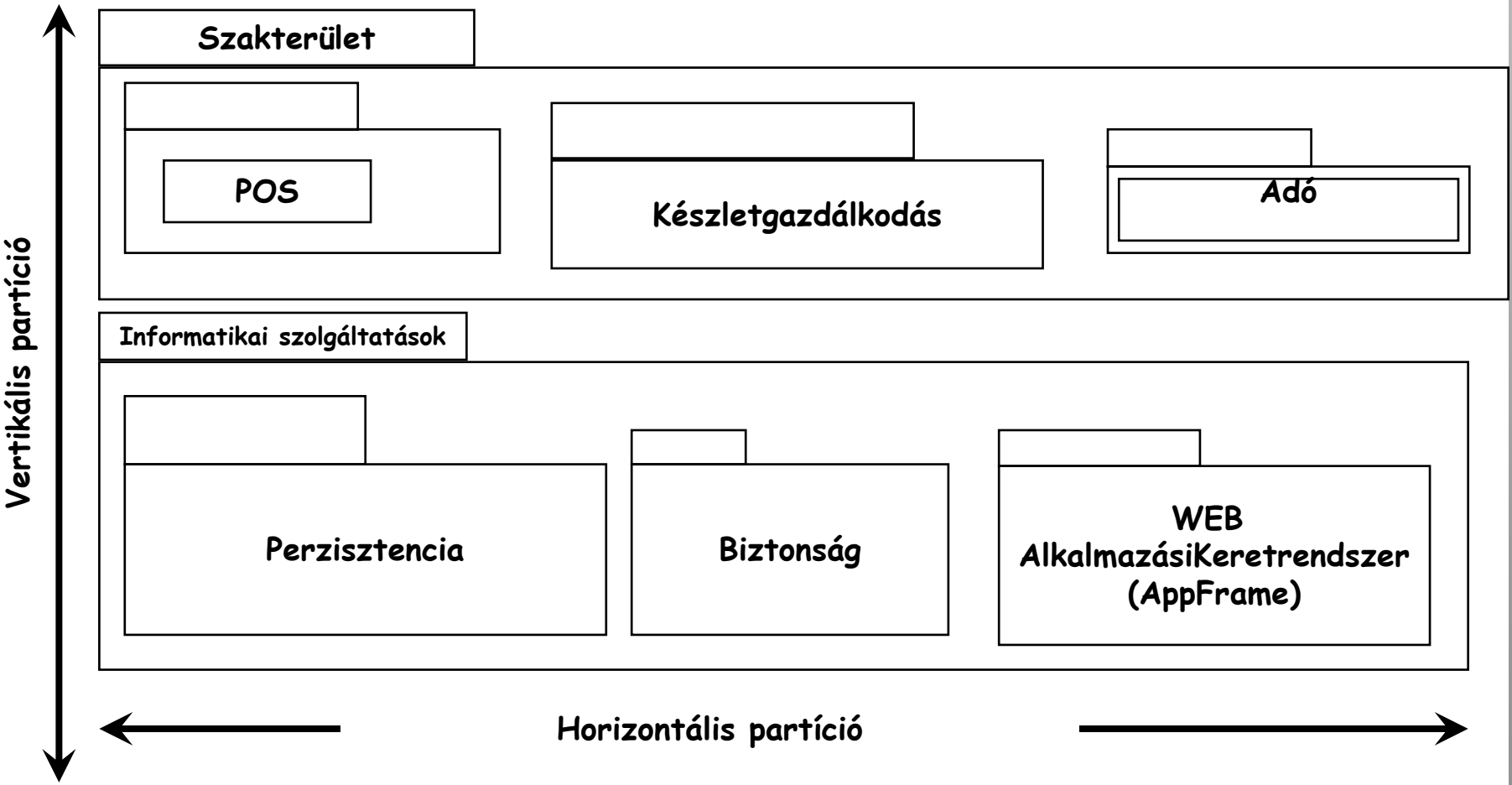


A szint a fizikai csomópontokra vagy csomópontok fürtjére utal, például „ ***ügyfél*** (*kliens*) szint"

Az architektúra rétegei az architektúra vertikális ('vízszintes') metszeteit érzékeltetik.

A partíciók a réteg viszonylag párhuzamos részrendszereinek horizontális ('függőleges') felosztását ábrázolják.

Rétegek és partíciók



Hogyan tervezzük meg az alkalmazási rendszer logikáját objektumokkal?

Készíthetünk egyetlen egy osztályt, amelye a teljes alkalmazás logikát tartalmazza, azonban ez alapjaiban sérti az objektum-orientált tervezés szellemét.

Létrehozhatunk olyan objektumokat, amelyeket a való világ alapján nevezünk el, és az alkalmazás logikájának megvalósítását rábízzuk.

Sok ügyességet és tapasztalatot igényel a megfelelő objektumok kiválasztása, és felelősségek kijelölése

Szakterület rétege és a szakterület modellje

Ezek nem ugyanazok a dolgok. *A szakterület modell* a való világ modelljét ábrázolja, míg a *szakterület rétege* a szoftver architektúrát.

Azonban a szakterület modellje inspirálja a szakterület réteget, és a fogalom alkotás forrása, valamint az osztály elnevezések kiindulópontja.

Ne keverjük össze a probléma megfogalmazását, a megoldással

Információrendszerek

Az információrendszerek architektúra rétegeit a *három-szintű architektúra* (*three-tier architecture*) fogalmával írták le.

A *három-szintű architektúra* az adatkapcsolati felületből, az alkalmazási logikából és az adattárolási szolgáltatásokból áll.

A ***három-szintű architektúra*** minőségi jellegzetessége:

Az alkalmazási logikai leválasztása egy elkülönített *köztes logikai szintbe* .

Az *adatkapcsolati, felhasználó felület* et ezzel megszabadítják az alkalmazási logikához köthető feldolgozási feladatoktól.

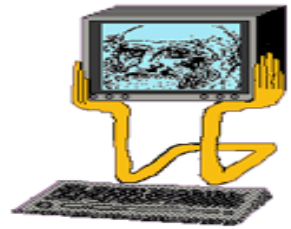
Az *adattárolási szolgáltatások* az adatok tartós, perzisztens, adatbázis jellegű megőrzését jelentik.

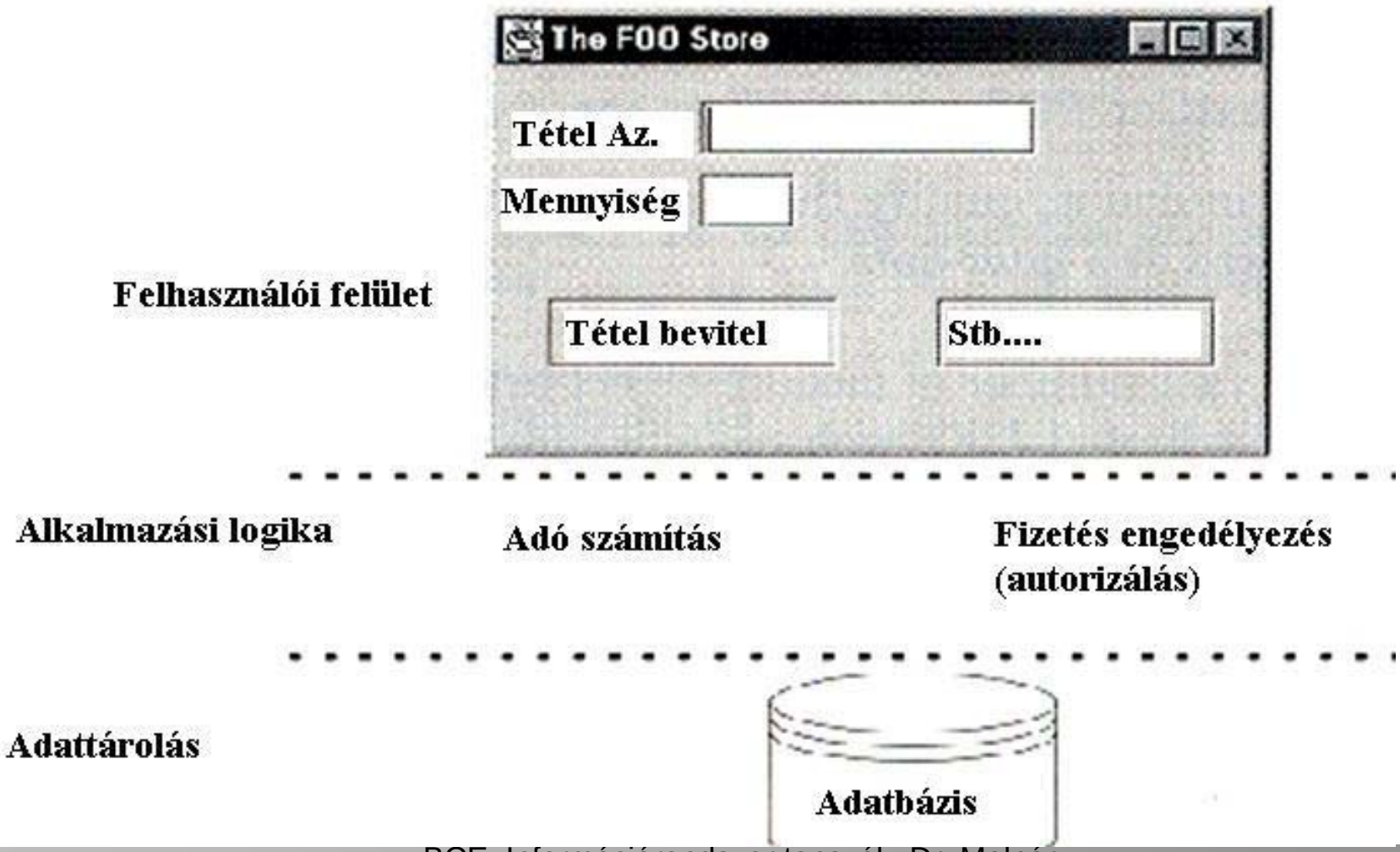
Információrendszerek (folyt.)

A középső szint kommunikál az adattráolási és háttér réteggel.

Egy példa a 3-szintű architektúrára

Példa:



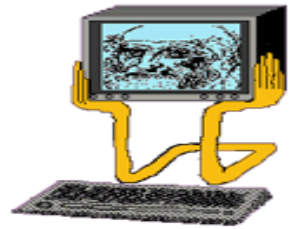

The FOO Store
Tétel Az.
Mennyiség
Tétel bevitel
Stb....
Felhasználói felület
Alkalmazási logika
Adó számítás
Fizetés engedélyezés
(autorizálás)
Adattárolás
Adatbázis

Két-szintű architektúra terv

Ebben az architektúra tervben, az alkalmazási logika az ablakok definíciójában helyezkedik el, és ez azt jelenti, hogy közvetlenül olvassa és írja az adatbázist.

Nincs köztes réteg, amely elválasztaná az alkalmazási logikától.

A modell-nézet elválasztásának elve



Az alapelv szerint: a modell (szakterületi) objektumoknak nem lehet közvetlen ismerete a nézet (megjelenítési) objektumokról.

Továbbá, a szakterület fogalom osztályainak magukba kell foglalniuk az alkalmazási logikához kapcsolódó információkat és az ebből származó viselkedési szabályokat.

A modell-nézet elválasztásának szükségessége

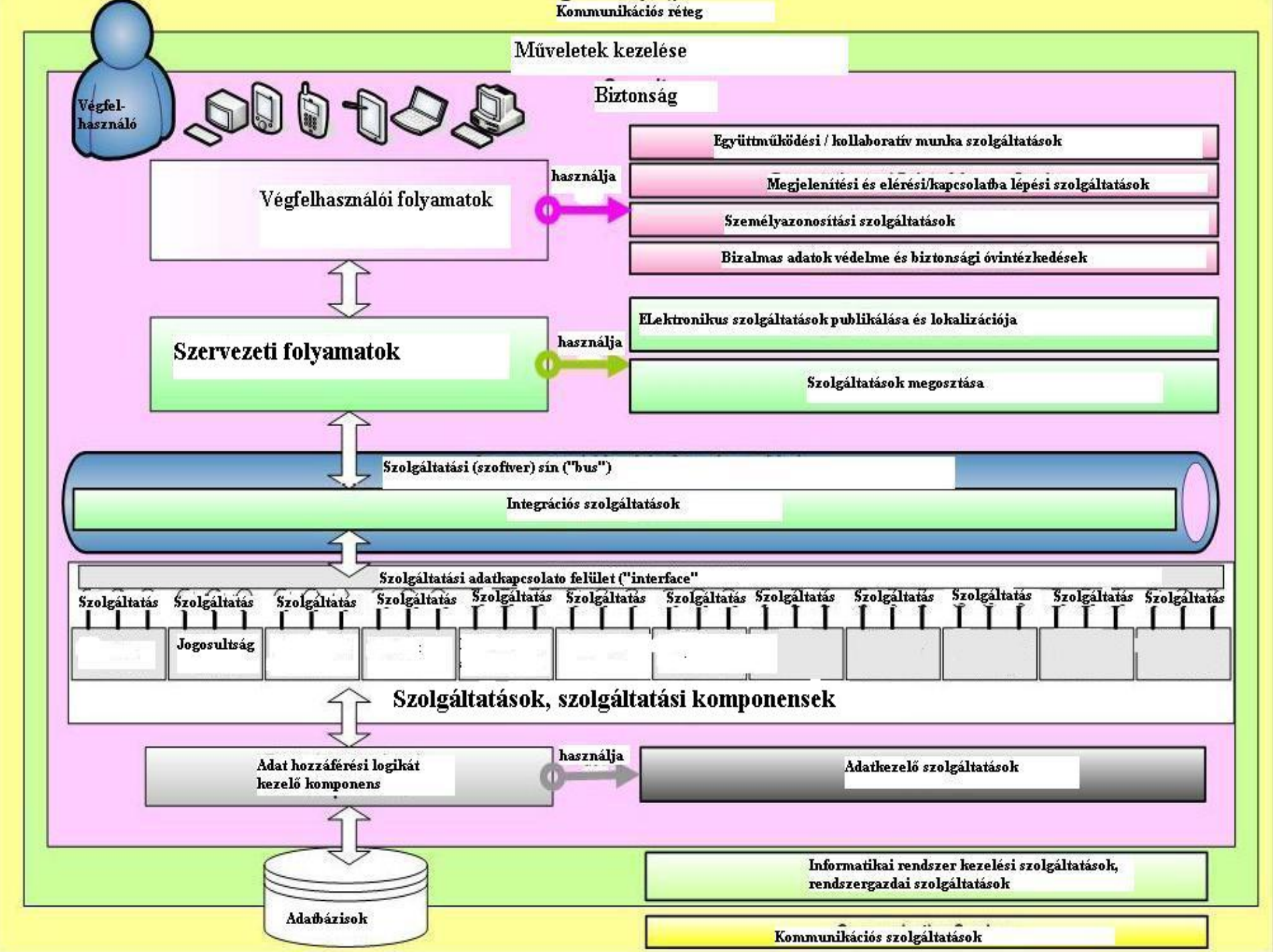
- A *modell definíció* kohézióját kell támogatni az elválasztásnak, amely a szakterület folyamataira koncentrál és nem felhasználói, adatkapcsolati felületre.
- A modell és a felhasználói felület fejlesztésének elválasztását lehetővé kell tenni.
- A felhasználói felületre vonatkozó követelmények változásának hatását lehetőleg mérsékelni kell a *szakterület architektúra rétegére*.
- Az újabb rendszer nézetek megjelenése esetén azok könnyedén hozzákapcsolhatók legyenek a már létező *szakterület architektúra réteg*hez anélkül, hogy befolyásolnák a szakterület architektúra rétegét.

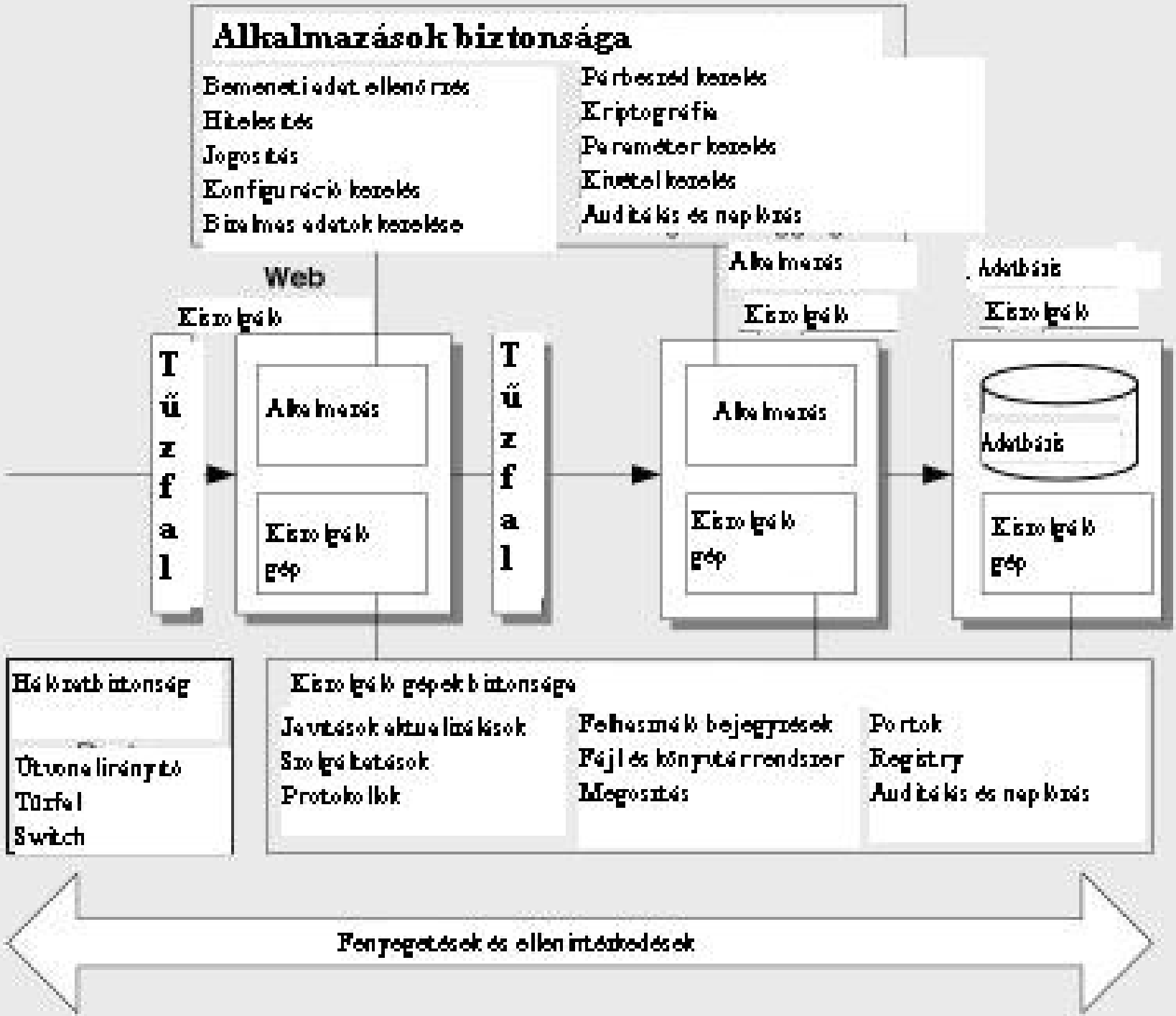
Folytatás..

Ez a megközelítés lehetővé teszi azt, hogy egyszerre, egy időben több szimultán nézeten keresztül lehessen látni a modell objektumait.

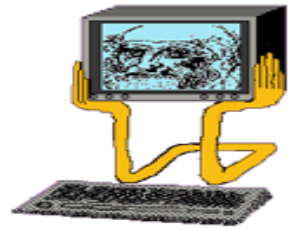
A modell rétegben a folyamatok végrehajtása a felhasználó felülettől függetlenül történhet.

A modell réteg könnyen hordozható más, felhasználói felületet megvalósító keretrendszert használó környezetbe.

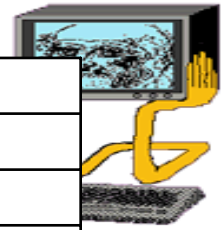

Kommunikációs réteg
Műveletek kezelése
Biztonság
Végfel-használó
Együttműködési / kollaboratív munka szolgáltatások
Végfelhasználói folyamatok
használja
Megjelenítési és elérési/kapcsolatba lépési szolgáltatások
Személyazonosítási szolgáltatások
Bizalmas adatok védelme és biztonsági óvintézkedések
ELektronikus szolgáltatások publikálása és lokalizációja
Szervezeti folyamatok
használja
Szolgáltatások megosztása
Szolgáltatási (szoftver) sín ("bus")
Integrációs szolgáltatások
Szolgáltatási adatkapcsolato felület ("interface"
Szolgáltatás
Jogosultság
Szolgáltatások, szolgáltatási komponensek
Adat hozzáférési logikát kezelő komponens
használja
Adatkezelő szolgáltatások
Adatbázisok
Informatikai rendszer kezelési szolgáltatások, rendszergazdai szolgáltatások
Kommunikációs szolgáltatások


Alkalmarások biztonsága
Bemeneti adat ellenőrzés
Hitelesítés
Jogosítás
Konfiguráció kezelés
Bizalmas adatok kezelése
Párbeszéd kezelés
Kriptográfia
Paraméter kezelés
Kivétel kezelés
Auditálás és naplózás
Web
Kiszolgáló
Alkalmazás
Kiszolgáló
Adatbázis
Kiszolgáló
Tűzfal
Alkalmazás
Kiszolgáló gép
Tűzfal
Alkalmazás
Kiszolgáló gép
Adatbázis
Kiszolgáló gép
Hálózatbiztonság
Útvonalirányító
Tűzfal
Switch
Kiszolgáló gépek biztonsága
Javítások aktualizálások
Szolgáltatások
Protokollok
Felhasználó bejegyzések
Fájl és könyvtárrendszer
Megosztás
Portok
Registry
Auditálás és naplózás
Fenyegetések és ellenintézkedések

Az informatikai szolgáltatások logikai architektúrális sajátosságainak jellemzése



Információ rendszer:			
Alkalmazási rendszer típusa: Szervezeti szintű alkalmazás			
Informatikai szolgáltatás kategória	**Alkategória**	**Szükséges-e**	**Megvalósító/támogató technológia**
Adatkezelés	Adatbázis kezelő rendszer	Igen	Bevált adatbáziskezelő rendszer / OEP stratégiai döntés alapján kiválasztott
	Repozitórium (Adatszótár)	Igen	
Biztonsági szolgáltatások	Személyazonosítási és hitelesítési szolgáltatások	Igen	PKI infrastruktúra, X.509 tanúsítvány, SAML – Security Assertion Markup Language egy XML, XACML - eXtensible Access Control Markup Language (XACML), SPML a Service Provisioning Markup Language (SPML), SOAP -a SOAP (Simple Object Access Protocol), Minősített és fokozatt biztonsági szint
	Auditálási lehetőségek biztosítása	Igen	
	Hozzáférés ellenőrzés, nyomon követés, kézben tartás	Igen	LDAP, Szerep-alapú jogosultság kezelés (RBAC)
	Biztonságirányítás	Igen	ISO/IEC 27000, ISO/IEC TR 13335
Címtár és szolgáltatás lokalizáció szolgáltatás	Címtár szolgáltatások	Igen	LDAP,
Szolgáltatási minőség	Rendelkezésre állás	Igen	24/7/365 , 99,95%
	Adaptálhatóság	Igen	Skálázhatóság, kiszolgáló gépek



Hálózati szolgáltatások	Adat kommunikáció	Igen	TCP/IP
	Elektronikus levél	Nem	
	Távoli eljárások (Remote processes)	Nem	
	Elosztott adatbázis kezelés	Nem	
Tranzakció feldolgozás	Tranzakciókezelése	Igen	
Felhasználói felületek	Karakter alapú	Nem	
	Ablakos, grafikus rendszer	Igen	MS Windows
Adatcsere	Dokumentum	Nem	
	Grafikus, digitális kép elem	Nem	
	Elektronikus adat	Igen	
Grafikus és digitális kép szolgáltatások	Grafikus objektum kezelés	Nem	
	Rajzolás	Nem	
Rendszer és hálózatfelügyelet	Hálózat felügyelet	Igen	Hálózatfelügyeleti szoftver
	Teljesítmény kezelés	Igen	Hálózatfelügyeleti szoftver
	Konfiguráció kezelés	Nem	Manuális
	Meghibásodások kezelése	Igen	Hálózatfelügyeleti szoftver
	Mentés, visszaállítás	Igen	Manuális
	Biztonságirányítás	Igen	Manuális
	Felhasználói felület készítési szolgáltatások	Igen	

Operációs rendszer	Kernel szolgáltatások	Igen	szabványos operációs rendszer
Szoftver tervezési szolgáltatások	Programozási nyelvek	Igen	
	Specifikáció, tervezési, kivitelezési, rendszerépítési szolgáltatások	Igen	