



**Eötvös Loránd Tudományegyetem  
Informatikai Kar**

# **Webes alkalmazások fejlesztése**

---

## **11. előadás**

### **Alkalmazások felhőben**

---

**© 2015 Giachetta Roberto**  
**groberto@inf.elte.hu**  
**<http://people.inf.elte.hu/groberto>**

# Alkalmazások felhőben

## Számítástechnikai felhő

---

- A *számítástechnikai felhő (computational cloud)* egy olyan *szolgáltatás alapú* rendszer, amelyben
  - a szolgáltatásokat nem dedikált hardvereszközön, hanem a szolgáltató eszközein elosztva üzemeltetik
  - a felhasználók a szolgáltatásokat hálózaton (interneten) keresztül érik el
  - az üzemeltetés, működés részleteit a felhasználótól elrejtik
- A felhő *helyfüggetlen, skálázható, és nagy rendelkezésre állást* biztosít
  - hardvereszközök meghibásodása nem befolyásolja a teljesítményt és elérhetőséget

# Alkalmazások felhőben

## Felhő szolgáltatások

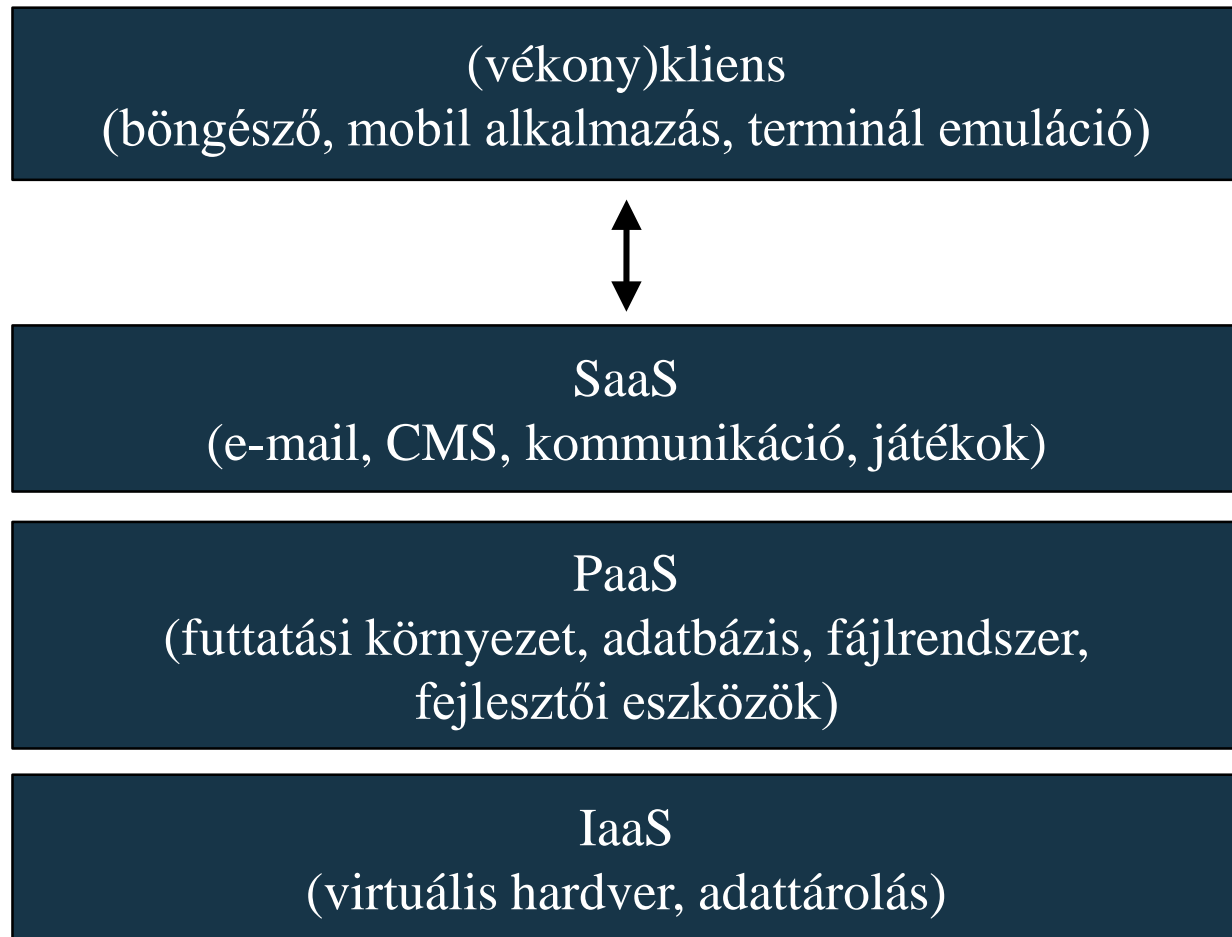
---

- A felhő alapú rendszerek hierarchikusan épülnek fel, és ennek megfelelően is nyújtják szolgáltatásaikat, három réteg adott:
  - *infrastruktúra szolgáltatás (Infrastructure as a Service, IaaS)*: biztosítja a hardver egy virtuális leképeződését (virtuális gépet), pl. *Amazon EC2, Google Compute Engine*
  - *platform szolgáltatás (Platform as a Service, PaaS)*: biztosítja szoftverek üzemeltetésének környezetét (operációs rendszert), pl. *Microsoft Azure, Google App Engine, Oracle Cloud*
  - *szoftver szolgáltatás (Software as a Service, SaaS)*: magát a szoftvert nyújtja szolgáltatásként, távoli elérésre (általában böngészőn keresztül)

# Alkalmazások felhőben

## Felhő szolgáltatások

---



# Alkalmazások felhőben

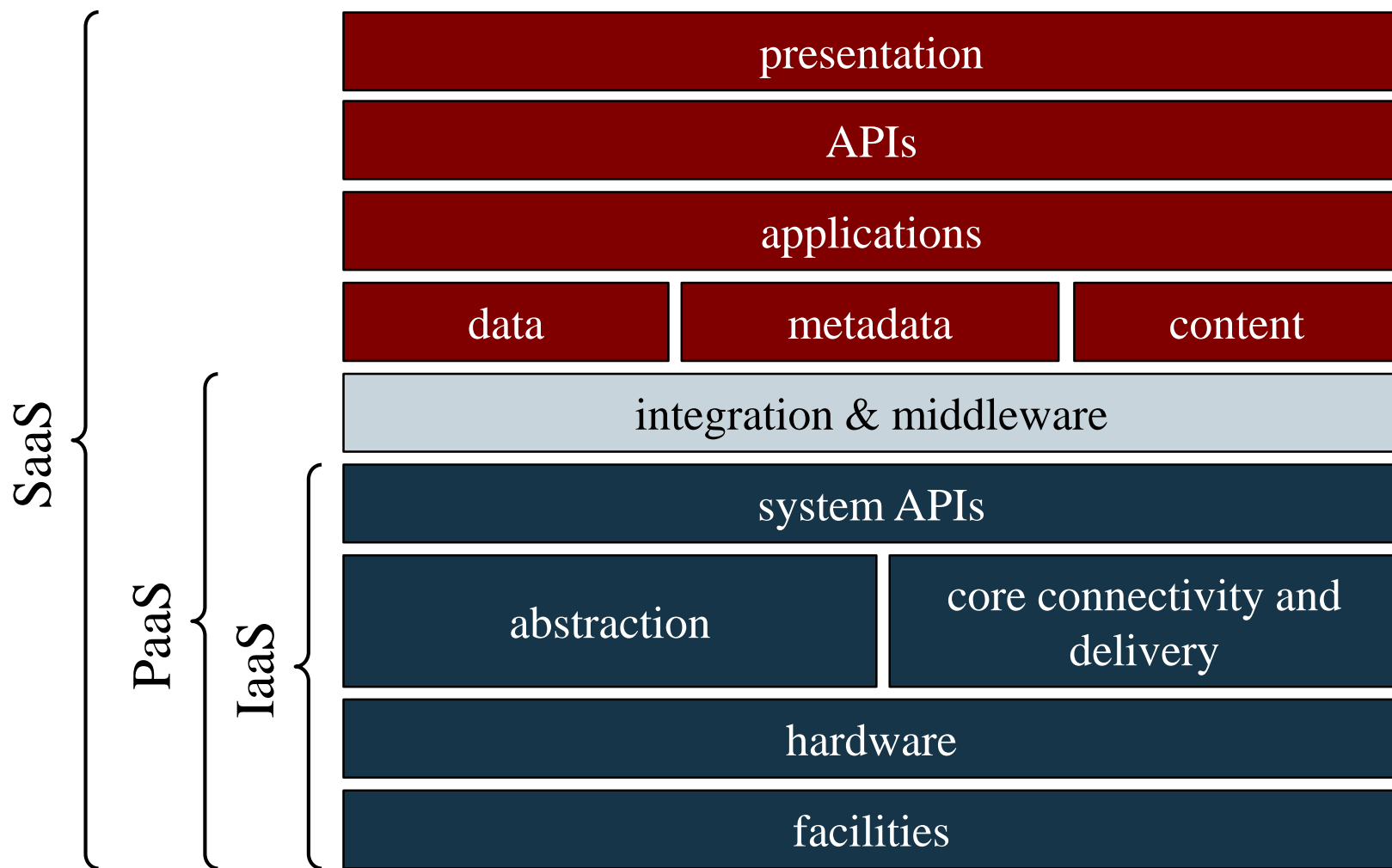
## Felhők megvalósítása

---

- A felhőkben harmadik fél vezérli az alkalmazást, tárolja az adatot, így számos kockázat felmerül (biztonsági kérdések, adatvesztés, szolgáltatás kimaradás, felelősség kérdése)
  - a felhőknek kimondottan kell ügyelnie a biztonságra, a felhasználói adatok kezelésére
- A felhők architekturális megvalósításában fontos szerepet játszik a *megosztott birtoklás (multitenancy)*, ahol a szoftver egy példánya szolgál ki tetszőleges sok kliens csoportot (szervezetet)
  - az alkalmazás virtuálisan particionálja adatait, valamint konfigurációját, így a csoportok egyedi, virtuális alkalmazásokat használnak

# Alkalmazások felhőben

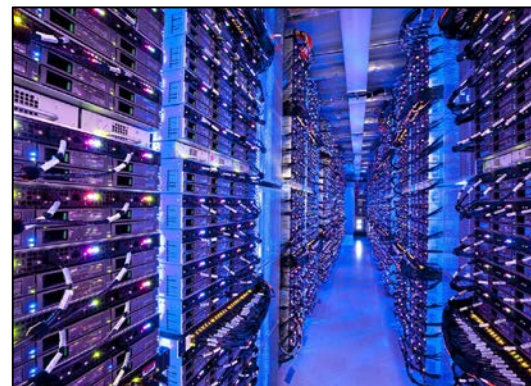
## Felhők megvalósítása



# Alkalmazások felhőben

## Felhők megvalósítása

- A felhők meghatározó tényezői:
  - *erőforrások* (adattárolás, hálózat, alkalmazások)
  - *elérhetőség* (publikus, privát, közösségi), *elhelyezkedés* (globális, lokális)
  - *infrastruktúra* (virtuálizáció, rugalmasság)
- A globális felhők *adatközpontokba* (*data center*) szerveződnek



# Alkalmazások felhőben

## Felhők jellemzői

---

- A felhőnek a következő tulajdonságokkal kell rendelkeznie alkalmazások futtatásához:
  - számos szolgáltatást biztosít (virtuális gépek, adattárolás, üzenetküldés, hálózat, egyszerű fejlesztési modell)
  - az erőforrások végtelen mennyiségének illúzióját biztosítja, amelyet csak a virtuális gépek száma korlátoz, vízszintesen skálázódik
  - biztosítja az erőforrások gyors növelését, csökkentését automatikusan, illetve manuálisan is
  - a költséghatékonyságra optimalizál, ezért megengedi a hardverhibákat, de a kieséseket nem (*commodity hardware*)



# Alkalmazások felhőben

## Szoftver a felhőben

---

- Egy szoftver két módon kerülhet kapcsolatba felhővel
  - a szoftver futhat lokálisan (helyi gépen, szerveren), de felhasználhat felhő által nyújtott szolgáltatást
    - általában adattárolás, autentikáció, közösségi funkcionalitás (pl. mobil alkalmazásoknál)
    - így a felhőt, mint háttérszolgáltatást használjuk (*Backend as a Service, BaaS*)
  - a futhat a felhőben lévő platformon (PaaS)
    - alkalmazkodnia kell a felhő környezethez (erőforráskezelés, szolgáltatás alapú működés, ...)
    - ugyanakkor működése nem feltétlenül kötött a felhőhöz

# Alkalmazások felhőben

## Szoftver a felhőben

---

- A felhőben futó alkalmazás szolgáltatásokat biztosít, így jellemzően weblap, vagy webszolgáltatás
  - általában állapotmentes, kérés/válasz alapú, a kommunikációra szabványos csatornákat használ (*SOAP*, *REST*, *COBRA*)
  - lazán kapcsolt architektúrát definiál, biztosítja az aszinkron végrehajtást, háttérben frissíthető (frissítés hatására nem áll le a rendszer)
  - hatékony végrehajtást végez, nem pazarolja az erőforrásokat, ügyel a földrajzi eloszlásra, vízszintesen skálázódik (a rendszer skálázódása és hibajelenségei nem befolyásolják a felhasználói élményt)

# Alkalmazások felhőben

## Microsoft Azure

---

- A Microsoft saját felhő környezete az *Azure*, amelye infrastruktúra és platform szolgáltatásokat nyújt
  - a platform elsősorban weblapok (ASP.NET) és webszolgáltatások (WCF, ASP.NET WebAPI) számára, de számos más környezet is támogatott (PHP, Node.js, Python, ...)
  - adattárolás szolgáltatások (*data services*): bináris (*Blob Service*), relációs (*SQL Database*), séma független (*Table Service*) és nagy adatok (*HDInsight*)
  - alkalmazás szolgáltatások (*app services*): azonosítás (*Azure Identity*), e-mail (*SendGrid*), médiakezelés (*Media Service*), üzenetkezelés (*Service Bus, Azure Queue*), ...

# Alkalmazások felhőben

## Adatfeldolgozás felhőben

---

- A felhők összességében nagy számítási és adattárolási kapacitást tudnak biztosítani, így alkalmasak *nagy méretű adatok (Big Data) elosztott feldolgozására*
  - elsősorban olyan adatokra és feladatokra, amelyek jól eloszthatóak a rendszerben, pl. adatbányászat, indexelés, dokumentum feldolgozás, képfeldolgozás
  - a rendszer feldolgozó csomópontokból (*node*) áll, amelyek végrehajtják egy feldolgozási *munkafolyamat (workflow)* egyes lépéseit
  - biztosítani kell egy olyan programozási modellt, amely lehetőséget ad az adatfeldolgozás dinamikus párhuzamosítására, az elosztott végrehajtásra

# Alkalmazások felhőben

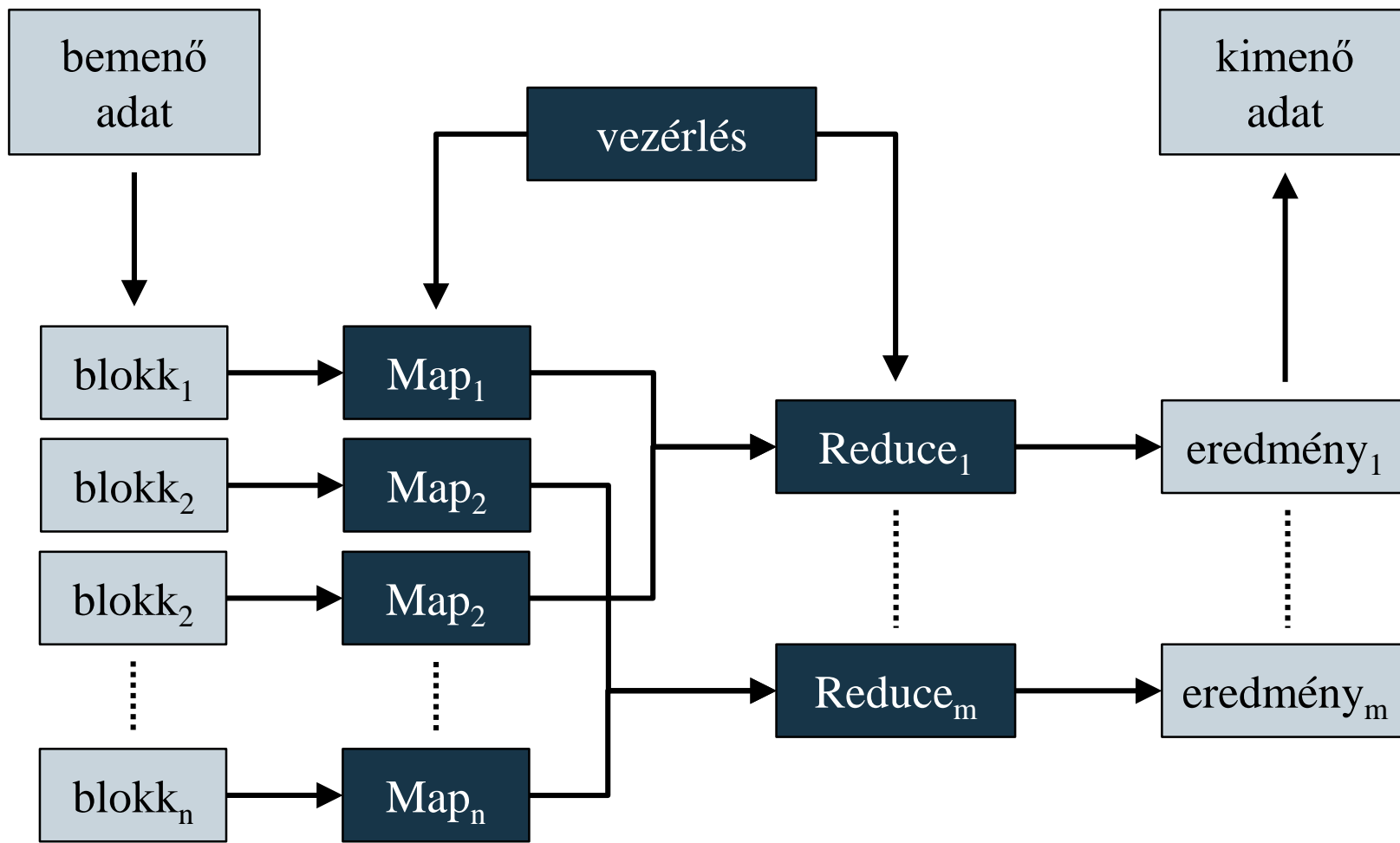
## A MapReduce modell

---

- A *MapReduce* programozási modell egy egyszerű feldolgozási modell, amely öt lépésben végzi el a munkafolyamatot:
  1. *Prepare*: felosztja a beérkező adatokat azonos méretű blokkokra, mindegyikhez egy kulcsot rendel
  2. *Map*: a kapott kulcs/érték párokat párhuzamosan transzformálja (szűri, rendezi, ...) köztes eredményekké
  3. *Shuffle*: a köztes eredményeket átrendezi, csoportosítja kulcs szerint és újabb kulccsal látja el
  4. *Reduce*: a kapott kulcs/érték párokat párhuzamosan kiértékeli
  5. *Produce*: az eredményeket begyűjti és összesíti

# Alkalmazások felhőben

## A MapReduce modell



# Alkalmazások felhőben

## A MapReduce modell

---

- Pl. szeretnénk megszámolni dokumentumok gyűjteményében minden szónak az előfordulási számát:
  1. felosztjuk a dokumentumokat, minden blokk egy dokumentumot kap (a kulcs a dokumentum neve)
  2. a dokumentumban minden megtalált szóra egy 1-es értéket adunk (a kulcs a szó, az érték 1), ez a *Map* lépés
  3. csoportosítunk a szó szerint, minden szóhoz egészek egy sorozata tartozik
  4. összeadjuk a sorozat értékeit, így megkapjuk minden szóra az összes előfordulás számát, ez a *Reduce* lépés
  5. az eredményeket összesítjük (szavanként)

# Alkalmazások felhőben

## A MapReduce modell

---

```
Map(String key, String value):
```

```
    // key: dokumentum neve
```

```
    // value: dokumentum tartalma
```

```
    foreach (String word in value):
```

```
        EmitIntermediate(w, "1");
```

```
        // a kulcs a szó, az érték 1 lesz
```

```
Reduce(String key, Enumerable<String> values):
```

```
    // key: a szó
```

```
    // values: a szóra kapott értékek sorozata
```

```
    int result = 0;
```

```
    foreach (String v in values):
```

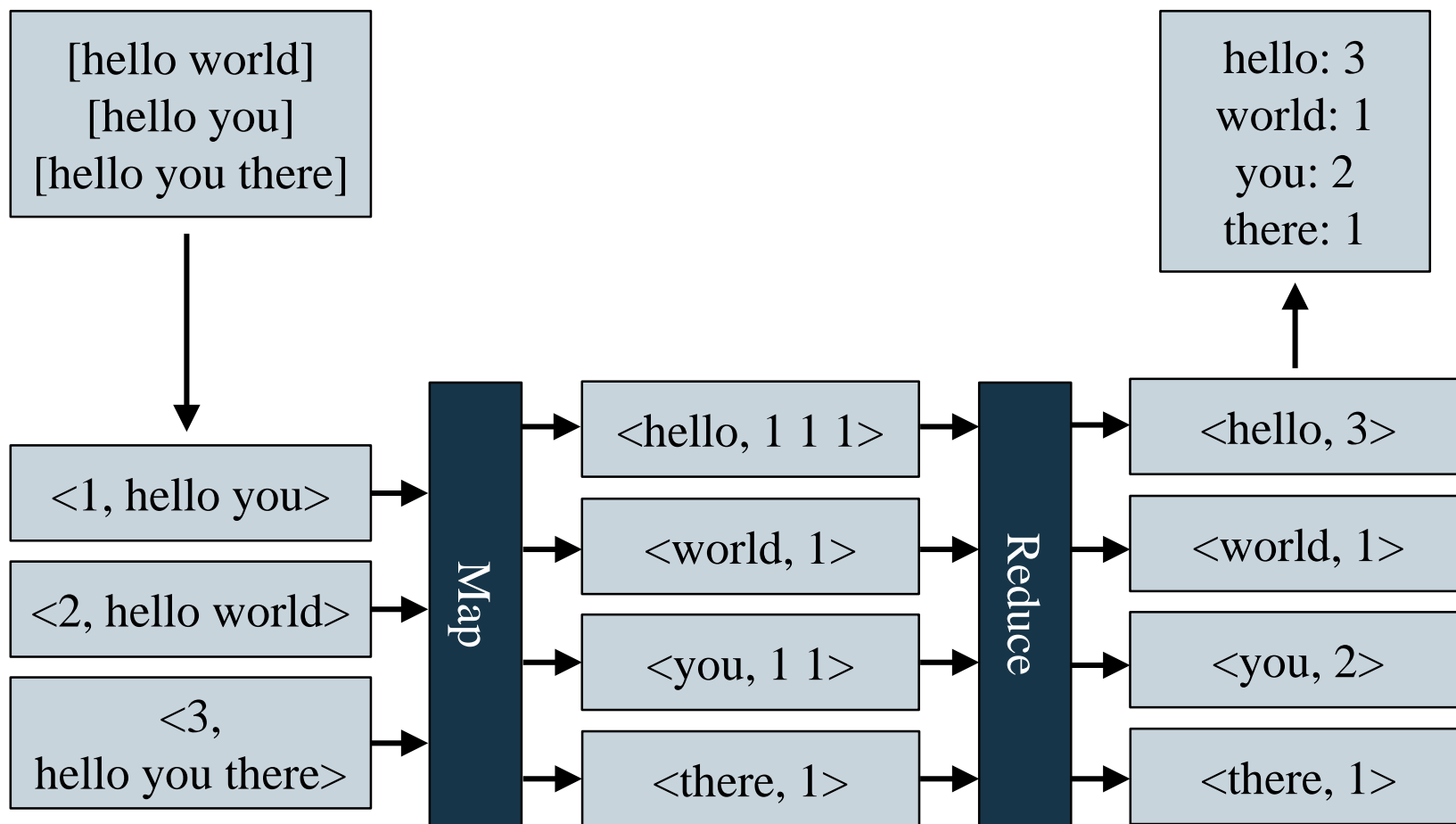
```
        result += AsInt(v);
```

```
    Emit(AsString(result)); // szövegesen adjuk meg
```



# Alkalmazások felhőben

## A MapReduce modell



# Alkalmazások felhőben

## Apache Hadoop

- Az *Apache Hadoop* egy, a MapReduce programozási modellre épülő szoftver keretrendszer
  - 3 fő modulból alkotja:
    - *Hadoop MapReduce*: adatfeldolgozó
    - *Hadoop Distributed File System (HDFS)*: elosztott fájlrendszer, amely felel az adatok elosztásáért az egyes node-ok között
    - *Hadoop YARN*: erőforrás kezelő és folyamat ütemező
  - Java-ban íródott, de alkalmas bármilyen nyelvű, MapReduce alapú alkalmazás futtatására (*Hadoop Streaming, REST API*)



# Alkalmazások felhőben

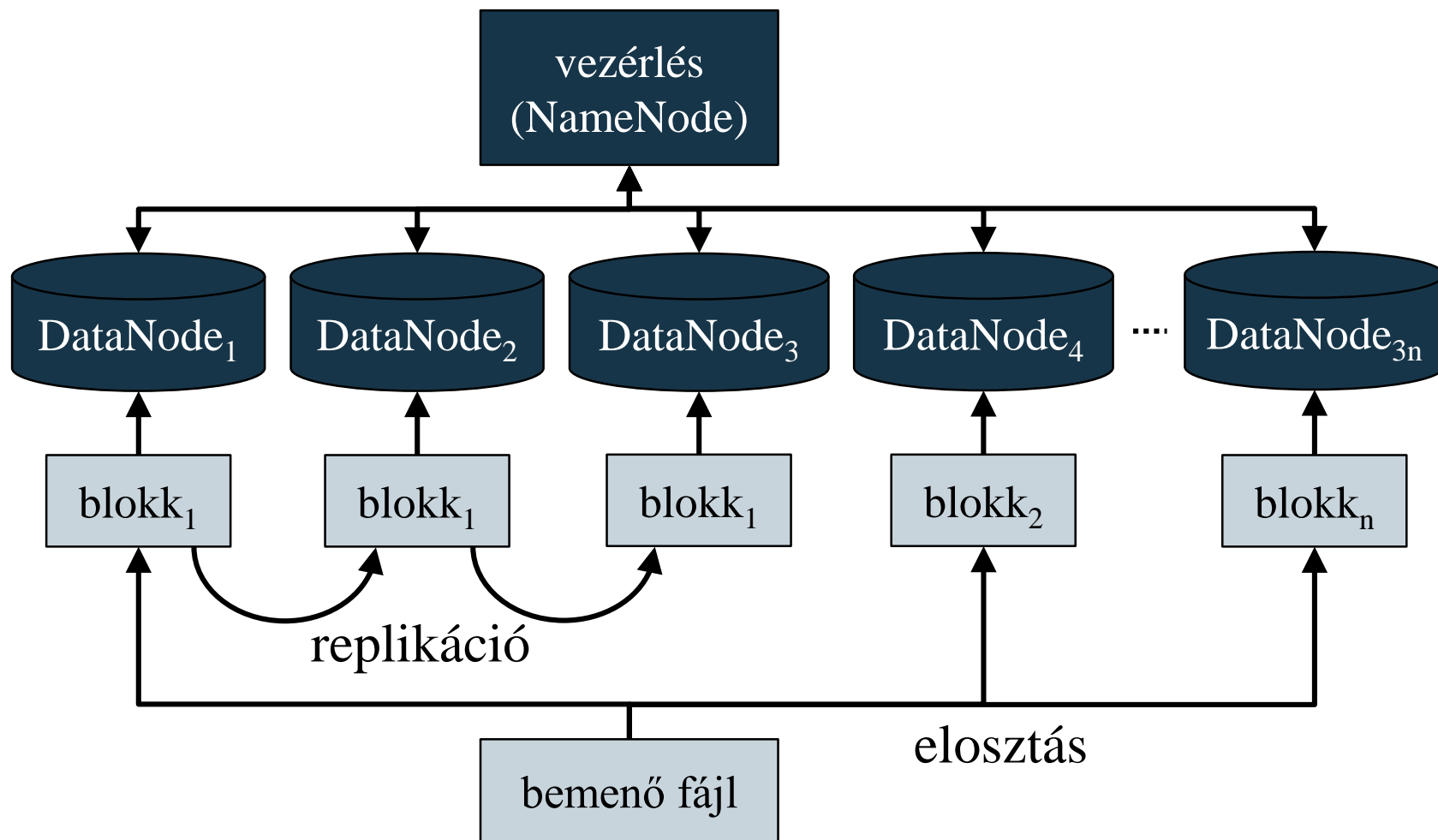
## Apache Hadoop

---

- A Hadoop nagyon jól skálázható, akár több ezer node alkotta rendszert is felépíthetünk
- A HDFS fájlrendszer célja a Hadoop architektúra kiszolgálása, ügyelve a rendelkezésre állásra
  - hatékonyan kezeli a nagy, homogén szerkezetű fájlkat, amelyeket azonos méretű blokkokra bont
  - olvasásra optimalizál (*write once, read many times*)
  - képes kezelni az egyes gépek kiesését, az egyes fájlrendszer csomópontok (*DataNode*) folyamatosan jelzik jelenlétüket
  - replikálja a tartalmat (alapértelmezetten 3 példány), így kerüli el az adatvesztést

# Alkalmazások felhőben

## Apache Hadoop



# Alkalmazások felhőben

## Apache Hadoop

---

- Az alap Hadoop rendszer számos további modullal egészülhet ki, amelyek az alap modulokra épülve bővítik a funkcionalitást, pl.:
  - adatbázis-kezelés (*HBase, Accumulo, Cassandra*)
  - speciális adatfeldolgozás (*Spark, Stratosphere, Giraph, GraphX*)
  - SQL alapú lekérdezések (*Hive, MRQL, Shark, Phoenix*)
  - adatelemzés (*Mahout, MLlib, MLbase*)
  - üzenetkezelés (*Thift*)
  - elosztott rendszer koordináció (*ZooKeeper*)

# Alkalmazások felhőben

## Hadoop alkalmazások .NET keretrendszerben

---

- A Windows Azure is támogatja a Hadoop architektúrát (*HDInsight*), illetve a .NET alapú alkalmazásfejlesztést (*.NET SDK for Hadoop*)
  - létrehozhatunk MapReduce tevékenységeket (`Microsoft.Hadoop.MapReduce`)
  - írhatjuk, olvashatjuk a HDFS fájlrendszert (`Microsoft.Hadoop.WebClient`)
  - futtathatunk LINQ alapú lekérdezéseket Hive adatbázisokon (`Microsoft.Hadoop.Hive`)
  - menedzselhetjük az Azure felhőben futó a felhőben futó tevékenységeket (`Microsoft.WindowsAzure.Management.HdInsight`)

# Alkalmazások felhőben

## Hadoop alkalmazások .NET keretrendszerben

---

- A HDFS fájlrendszerhez a `WebHDFSClient` segítségével férhetünk hozzá, pl.:

```
WebHDFSClient client = new WebHDFSClient(...);
await client.CreateDirectory("/myDirectory");
    // könyvtár létrehozása (aszinkron)

await client.ClientFile("c:\myFile.txt",
    "/myDirectory/myFile.txt");
    // fájl létrehozása a lokális fájl másolásával

List<String> fileList = (await
    client.GetDirectoryStatus("/myDirectory")).
    Files.Select(f => f.PathSuffix).ToList();
    // könyvtár tartalmának lekérdezése
```

# Alkalmazások felhőben

## Hadoop alkalmazások .NET keretrendszerben

---

- MapReduce folyamatok végrehajtásához meg kell valósítunk a Map és Reduce műveleteket, majd ezeket egy Hadoop feladatként kell konfigurálnunk
  - a Map tevékenységet a **MapperBase** osztályból származtatva, a **Map** metódus felüldefiniálásával adjuk meg
  - a Reduce tevékenységet egy **ReducerCombinerBase** leszármazott osztály a **Reduce** metódusában adjuk meg
  - a feladat egy **HadoopJob** leszármazott osztály, ahol megadjuk a végrehajtás paramétereit (a felüldefiniált **Configure** metódusban)
  - a kommunikáció szöveg alapú, mindent konvertálnunk kell



# Alkalmazások felhőben

## Hadoop alkalmazások .NET keretrendszerben

---

- Pl.:

```
public class WordCountMapper : MapperBase {
    public override void Map(string inputLine,
                            MapperContext context)
    {
        // felüldefiniáljuk a Map metódust
        String[] words = inputLine.Split(' ');
        // elvégezzük a tevékenységet
        foreach(String word in words)
            context.EmitKeyValue(word, "1");
        // továbbítjuk az eredményt
    }
}
```

# Alkalmazások felhőben

## Hadoop alkalmazások .NET keretrendszerben

---

```
public class WordCountReducer :
    ReducerCombinerBase
{
    public override void Reduce(String key,
        IEnumerable<String> values,
        ReducerCombinerContext context)
    {
        Int32 result =
            values.Sum(val => Int32.Parse(val));
        // végrehajtjuk a műveletet
        context.EmitKeyValue(key,
            result.ToString());
        // továbbítjuk az eredményt
    }
}
```

# Alkalmazások felhőben

## Hadoop alkalmazások .NET keretrendszerben

---

```
public class WordCountJob :
    HadoopJob<WordCountMapper, WordCountReducer>
    // sablonban megadjuk a két típust
{
    public override HadoopJobConfiguration
        Configure(ExecutorContext context)
    {
        HadoopJobConfiguration config =
            new HadoopJobConfiguration();
        config.InputPath = "/myDirectory";
        config.OutputFolder = "/myResultDirectory";
        return config;
        // megadjuk a bementi/kimeneti könyvtárat
    }
}
```