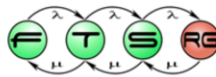


# Konfigurációkezelés

Kocsis Imre, Micskei Zoltán



Utolsó módosítás: 2016.03.29.

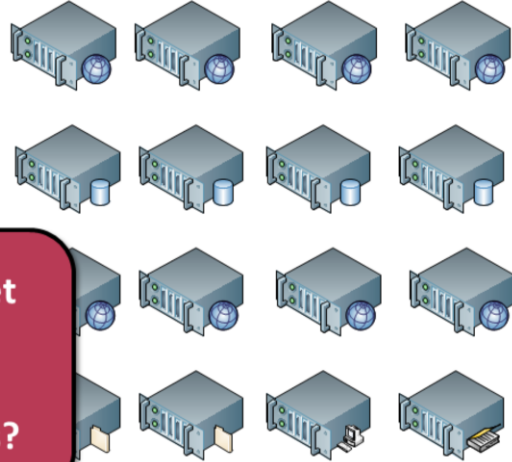
# Motiváció



Átírtuk a cache komponenst. Melyik szerverekre kell felrakni az új verziót?

50+ felhő alapú szerver esetén nem is triviális...

1. Munkavállalói emlékezet
2. „Kockás füzet”
3. Visio
4. (Konfiguráció) adatbázis?



Komplexebb vagy dinamikusan változó rendszerek esetén már pusztán az is egy nem triviális feladat, hogy megmondjuk, hogy jelenleg hány kiszolgálónk van és azok pontosan mit csinálnak.

Lehet egyszerűen csak nagyon sok gépünk és szolgáltatások (jó példa a CERN: <http://malandes.web.cern.ch/malandes/cc.html>) vagy használhatunk részben/egészben felhő alapú szolgáltatásokat, ott pedig a folyamatosan változó, az igényekhez igazodó foglalások és a megjegyezhetetlen gépnevek a gyakoriak.

# Motiváció



És ha nem frissek az adatok?

1. Odamegyünk
2. ssh/RDP/VNC/...
3. Házilagos „varázslat” ...
4. Platformtámogatás?



Bármilyen állapotfrissítés vagy beavatkozás nehézkes lehet, ha összetet és/vagy heterogén rendszerünk van. Akár üzemeltetnünk kell a rendszert, akár fejleszteni rá valami alkalmazást, ha nincs megfelelő automatizálási megoldás, akkor a telepítés és felügyelet nagyon nehézkes lesz. Az alapfunkciókat (távoli gépek összegyűjtése, állapot lekérdezése, egyszerűbb beavatkozások végrehajtása) pedig jó lenne mindenhol ugyanúgy megvalósítani, erre mutat megoldásokat az előadás.

# Konfigurációkezelés

Igények és hívások összegyűjtése

# 1. Konfiguráció-adatbázisok

**Igény: konfiguráció-adatbázisok**  
(nem triviális méretű rendszerek)

Területek és „silók” szerint több adatbázis lesz!

- HW
- Hálózati eszközök és topológia (~IP szintig)
- OS platformok és szoftverek – leltár
- SW licenszek
- OS és alkalmazás/kiszolgáló beállítások

Beállítások / Konfigurációk

**Ha tényleg adatbázis: modellezni is kell...**  
**Minden esetben új adatmodell?**  
**→ Common Information Model**



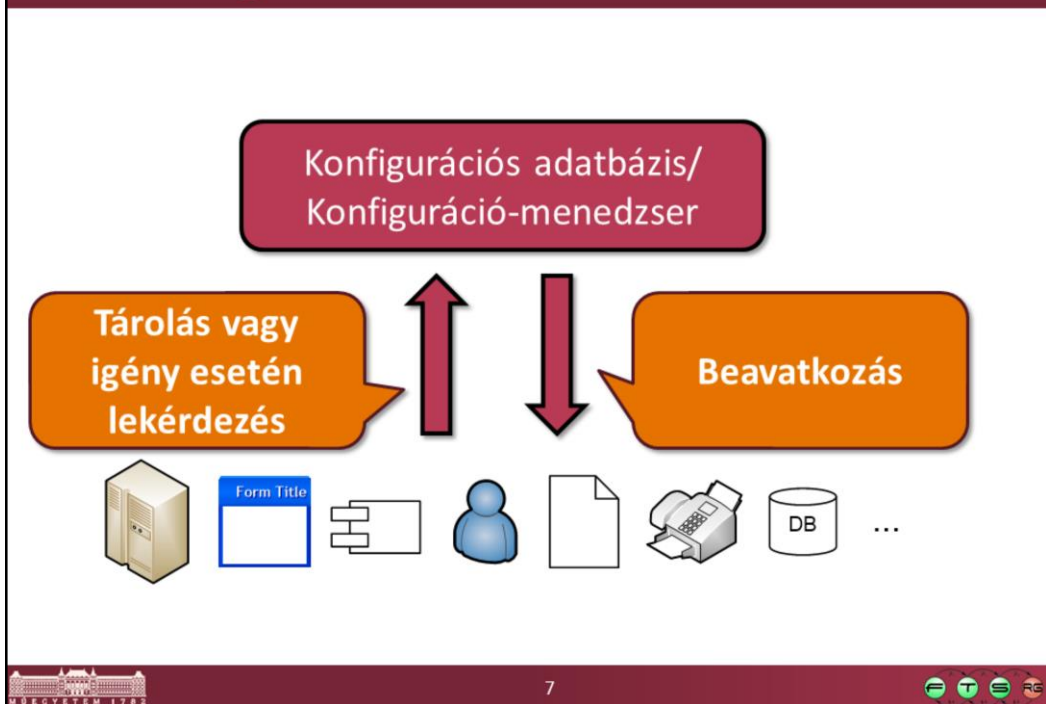
Az látszik, hogy a „konfiguráció” fogalma nem csak műszaki területről területre, de az egyes alkalmazási esetek között is változik, attól függően hogy milyen rendszeraspektusokat és azok közötti kapcsolatokat akarunk nyilvántartani.

## 2. Szabványosítás

Igény: konfigurációs adatok távoli  
lekérdezése/módosítása

- (A „konfigurációkezelés” erősen keveredik egyéb operatív feladatokkal, pl. monitorozás!)
  
- Nagy rendszerre nem skálázódik:
  - SSH + ad hoc parancsok
  - SSH + szkriptek
  - ...
  - Windows...?

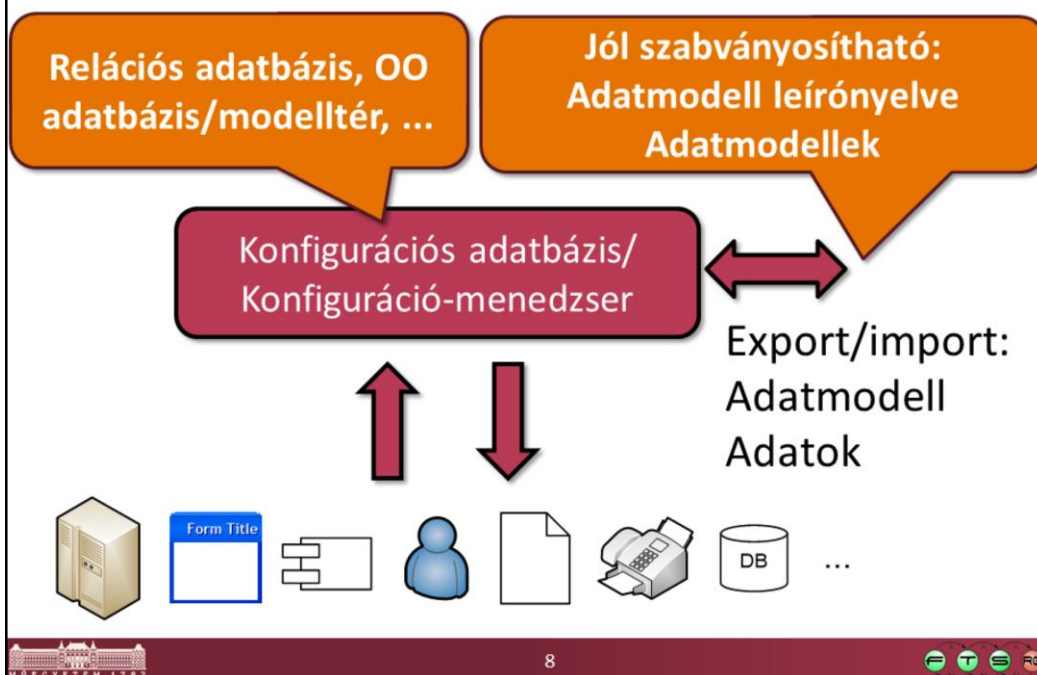
## Konfigurációkezelés – architektúra?



Ez most még az egy gépen belüli architektúra felvázolása lesz. Ilyenkor a konfigurációkezelés célja az adott számítógép konfigurációs elemeinek az összegyűjtése, tárolása és elérhetővé tétele.

Kell egy konfigurációs adatbázis, ami be tudja gyűjteni az információkat, és igény esetén be is tud avatkozni (=műveleteket hajt végre a rendszer elemein).

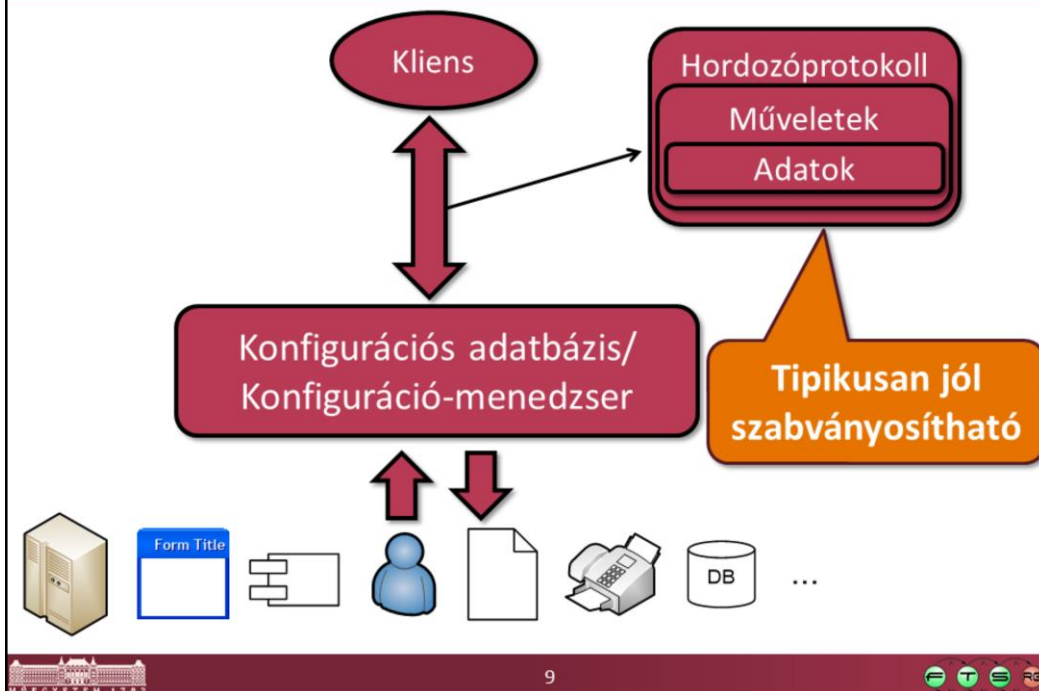
## Konfigurációkezelés – architektúra?



Ebben az adatbázisban lévő adatokat jó lenne, ha exportálni és importálni tudnánk könnyen, akár más gépre/más gyártójú adatbázisba átvinni.

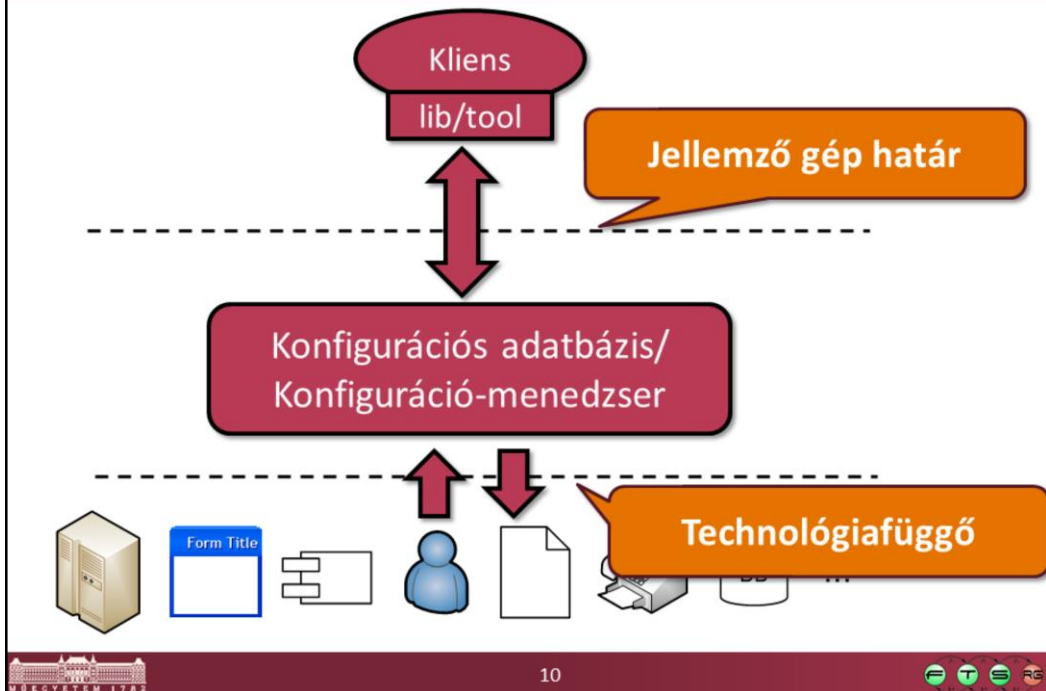


## Konfigurációkezelés – architektúra?



Ha megvannak már az adataink, le is kell kérdezni azokat valahogyan. Szeretnénk, ha nem kéne minden termékhez külön-külön klienst készíteni, és teljesen újramezteni egy-egy új platform használata esetén.

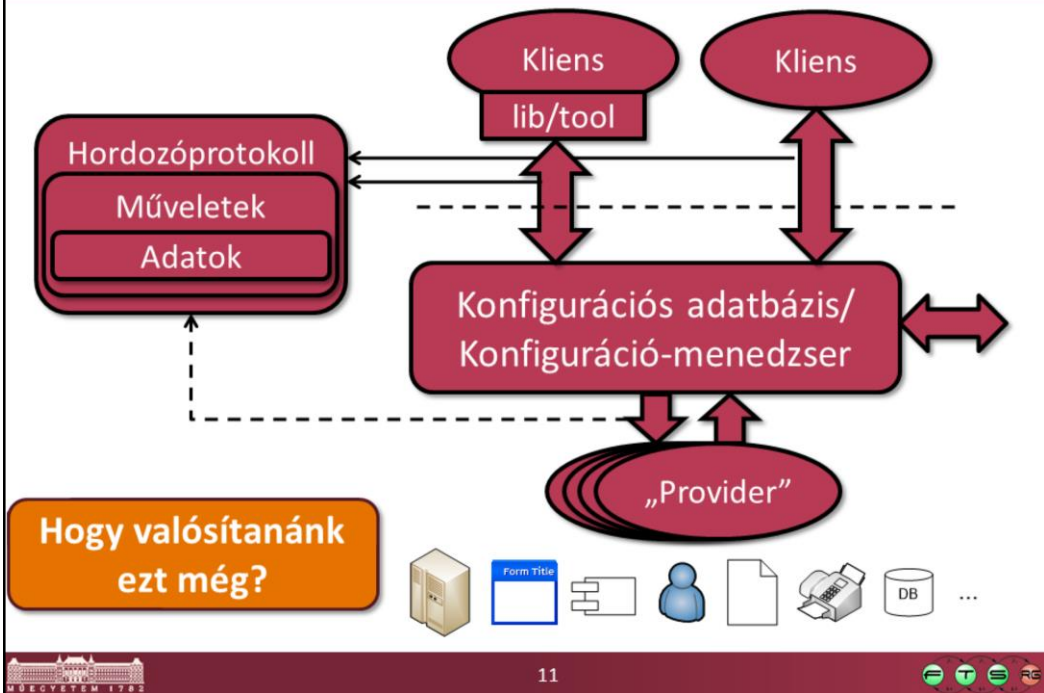
## Konfigurációkezelés – architektúra?



A csatlakozáshoz szükséges funkciókat tipikusan jól meg lehet valósítani egy megosztott könyvtárban vagy komponensben, ami például elfedi a távoli géppel való kommunikáció nehézségeinek egy részét.

A menedzselendő eszközök kezelése és lekérdezése általában már problémásabb, itt majd sokféle egymástól független technológiával fogunk találkozni.

# Az architektúra, amire szabványokat illesztünk



## 3. Folyamatok

**Igény:** megfelelő folyamatok  
Konfiguráció-változás bevezetése  
Eltérés monitorozása és kezelése

- Mennyire jól szabványosítható?
- „best practice” gyűjtemények: minták
  - Minimálisan: fogalmi keretrendszer
  - **ITIL** – „adapt & adopt”



Maga a technológiai segítség megvalósítása általában még kevés, legalább annyira fontos a megfelelő támogató folyamatok kitalálása és betartása. Erre ad például egy jó ajánlásgyűjteményt az ITIL.

## 4. CMDB

**Igény:** konfigurációs adatbázisok között kapcsolat

- Pl.: fizikai hely, IP-cím és szolgáltatás más DB-ben
- Egyesítés? Federáció?
- Metamodellek egyesítése?
- **Hiányzó kapcsolatok**
  - Az igazi hozzáadott érték...
  - ... cserébe nehéz feladat
- **ITIL v3: „Configuration Management DataBase”**



Federált adatbázis: olyan adatbáziskezelő rendszer (ha nagyon pontosak akarunk lenni: „metadatbázis-kezelő rendszer”), mely több autonóm adatbáziskezelő szolgáltatásait a saját felhasználói számára transzparens módon integrálja.

# Konfigurációkezelés - tematika

## 1. Alapok, modellezés, szabványok

- szabványos modellezés
- szabványos távoli hozzáférés
- eszközök

Házi feladat: CIM és WMI

## 2. Windows konfigurációkezelés

- WMI: alapok, architektúra
- WMI: eszközök, PowerShell
- WS-Management (WinRM)

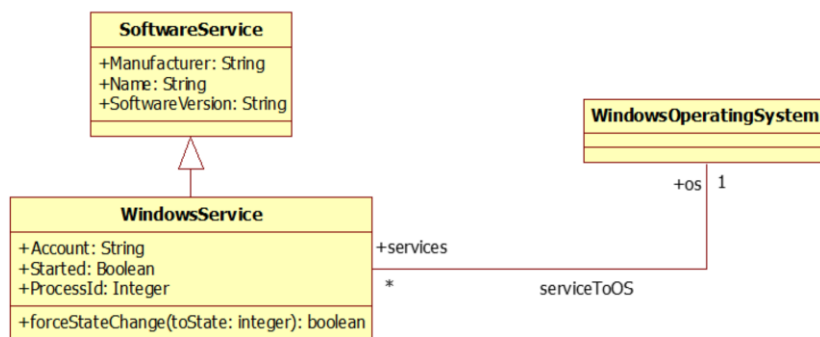
## 3. CMDB-k

- Az ITIL CMDB fogalma
- Funkcionális jellemzők
- Felderítés

A tantárgyban idén az 1. és 2. anyagokkal foglalkozunk. A HF2 keretében ezekhez a területekhez kapcsolódó feladatot kell majd megoldani.

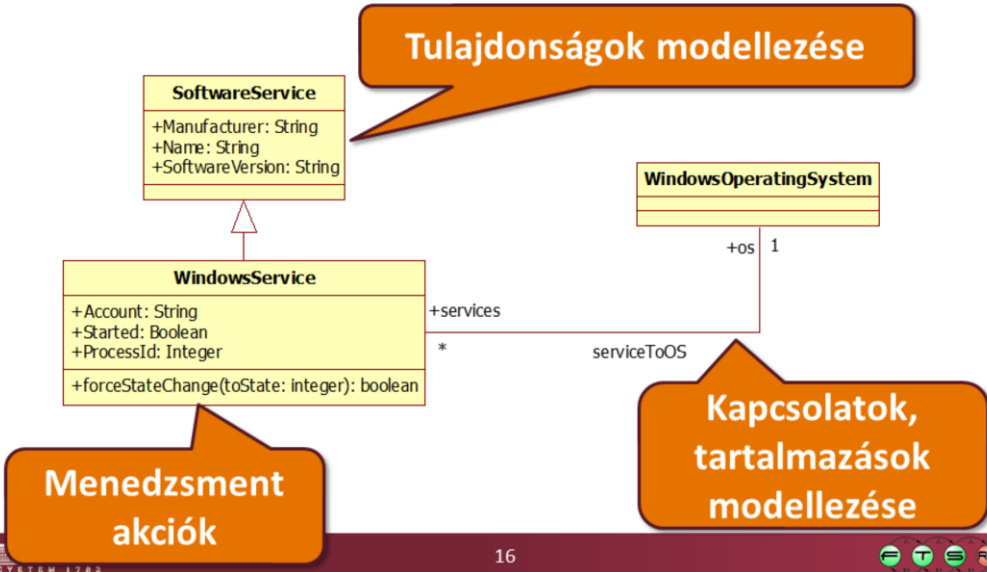
# Rendszermenedzsment és modellezés

- Rendszermenedzsment: OO szemlélet adódik
  - Különösen a konfiguráció-menedzsmentben



# Rendszermenedzsment és modellezés

- Rendszermenedzsment: OO szemlélet adódik
  - Különösen a konfiguráció-menedzsmentben





## Modellek megadása (emlékeztető)

1. Mi kell adatmodellek megadásához?
2. Mi kell egy modellezési nyelv precíz megadásához?

Kell(enek) metamodell(ek)

Kellenek:

- Absztrakt és konkrét szintaxis
- Jól formáltsági szabályok, szemantika

# CIM (Common Information Model)

<b>Kibocsájtó:</b>	Distributed Management Task Force (DMTF)
<b>Megalkotók:</b>	IBM, HP, EMC, VMware, Symantec, Microsoft ...
<b>Verziók:</b>	CIM Schema: 2.45.0 (2016) CIM Information Metamodel: 3.0.1 (2014)
<b>Cél:</b>	menedzsment információk objektum-orientált modellezése és szabványos modellek megadása

# A Common Information Model (CIM)

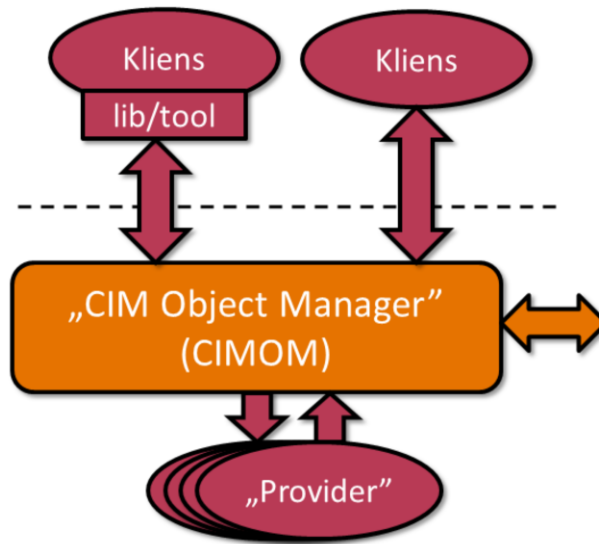
- Szabványos, kiterjeszthető IT adatmodell
  - Használják is: VMware ESX, HP, IBM termékek, Windows Management Instrumentation...
  
- Cél:
  - Különböző gyártók termékei együtt tudjanak működni
  - Ugyanazt értsük az adott fogalmak alatt
  - Könnyen kiterjeszthető legyen saját fogalmakkal



Rövid összefoglaló arról, hogy mi is a CIM:

- What is CIM? [http://dmtf.org/about/faq/cim\\_faq#C1](http://dmtf.org/about/faq/cim_faq#C1)

## A CIM jellemző alkalmazásai



A CIM adja meg, hogy

- a konfigurációs adatbázisunkban milyen típusú adatokat tároljunk,
- és ezeket hogyan lehessen exportálni és importálni.

# Miket tartalmaz a CIM?

- **CIM Specification**
  - Szöveges dokumentum embereknek
  - Hogyan kell értelmezni ezt az egészet
- **CIM Metamodel**
  - Metamodel a későbbiekben definiált elemekhez
- **CIM Schema**
  - Konfigurációs adatokhoz modellek
  - CIM Metamodelben definiált elemek példányai
- **Managed Object Format (MOF)**
  - konkrét szintaxis CIM Schema elemekhez

# Miket tartalmaz a CIM?

- **CIM Specification**
  - Szöveges dokumentum embereknek
  - Hogyan kell értelmezni ezt az egészet
- **CIM Metamodel**
  - **Metamodell a későbbiekben definiált elemekhez**
- **CIM Schema**
  - Konfigurációs adatokhoz modellek
  - CIM Metamodelben definiált elemek példányai
- **Managed Object Format (MOF)**
  - konkrét szintaxis CIM Schema elemekhez

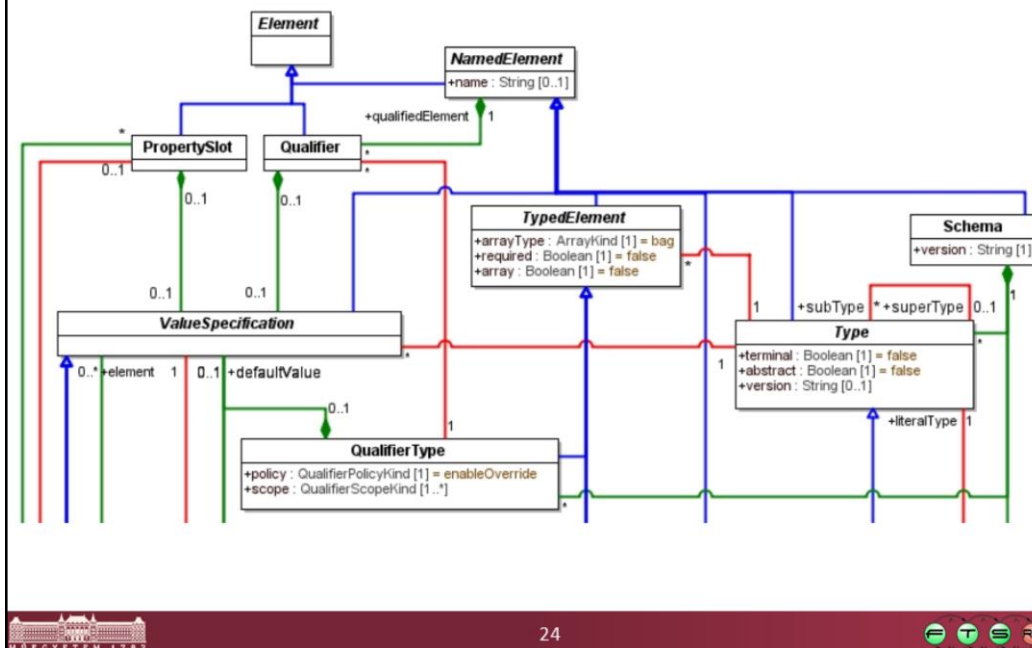
# CIM Metamodel

- Szokásos fogalmak
  - Osztály, példány, metódus, tulajdonság, asszociáció
- CIM specifikus fogalmak
  - Séma, minősítő (qualifier)...
- (3.0.0-tól kezdve) UML metamodel + OCL kényszerek segítségével van definiálva



- A CIM Meta Schema az UML-től kicsit eltérő fogalmakat használ néha, mert egyrészt más területre dolgozták ki, másrészt részben korábbi is, mint az UML 2.0.
- (Régebben CIM Meta Schema néven hivatkoztak a Metamodelre.)

# CIM Metamodel (részlet)

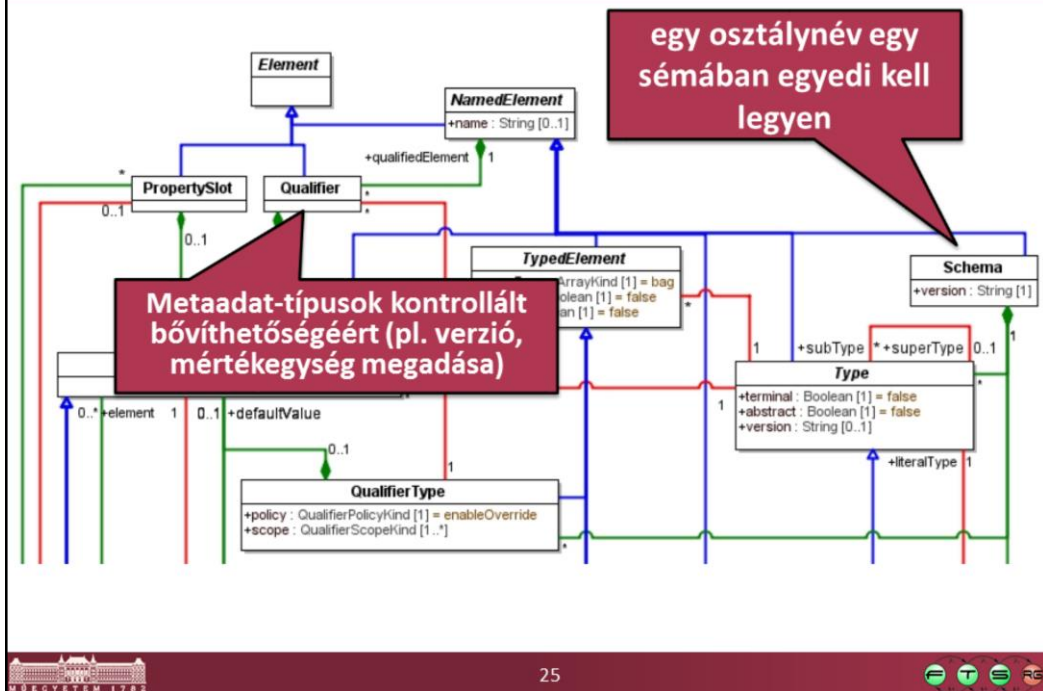


A teljes Meta Schema megtalálható a CIM Metamodel PDF-ben:

- <http://dmtf.org/standards/cim>
  - És innen: CIM Metamodel (DSP0004)
  - 3.0.0-s verzióban: 31. oldal, Figure 1 – Overview of CIM Metamodel



## A CIM Metamodel (részlet)



Néhány fontosabb elem definíciója (idézet a szabványból):

- **NamedElement**: A NamedElement is an abstract metaelement that models elements that have a name.
- **Schema**: A schema provides a context for assigning schema unique names to the definition of elements.
  - Fully-qualified class names are in the form <schema name>\_<class name>
- **Class**: A class models an aspect of a managed resource. A class is a type that has zero or more properties, methods, and qualifiers and may define local structures and enumerations .
- **Qualifier**: A CIM qualifier is meta data that provides additional information about the element on which the qualifier is specified. The specification of a qualifier on an element defines a value for the qualifier on that element.

## CIM Metamodel – adattípusok

- Tulajdonságok, referenciák, paraméterek, visszatérési értékek, minősítők: van típusuk
  
- Csak a beépített típusok vagy azok tömbjei:
  - {u/s}int{8/16/32/64}, string, boolean, real{32/64}, Datetime, char 16
  - <classname> ref
  - NULL (de nem asszociációban)

# CIM Metamodel - minősítők

## ■ Qualifier (minősítő)

- Kicsit hasonlít az UML szerotípiára
- Csak osztályokon szerepelhet, példányon nem!

## ■ Gyakori minősítők:

- Abstract                      absztrakt osztály jelölése
- Description                    elem leírása
- In, Out                         paraméter irány megadása
- Key                              mik a kulcsok az osztályban
- PUnit                            mértékegység megadása
- Version                         verziószám

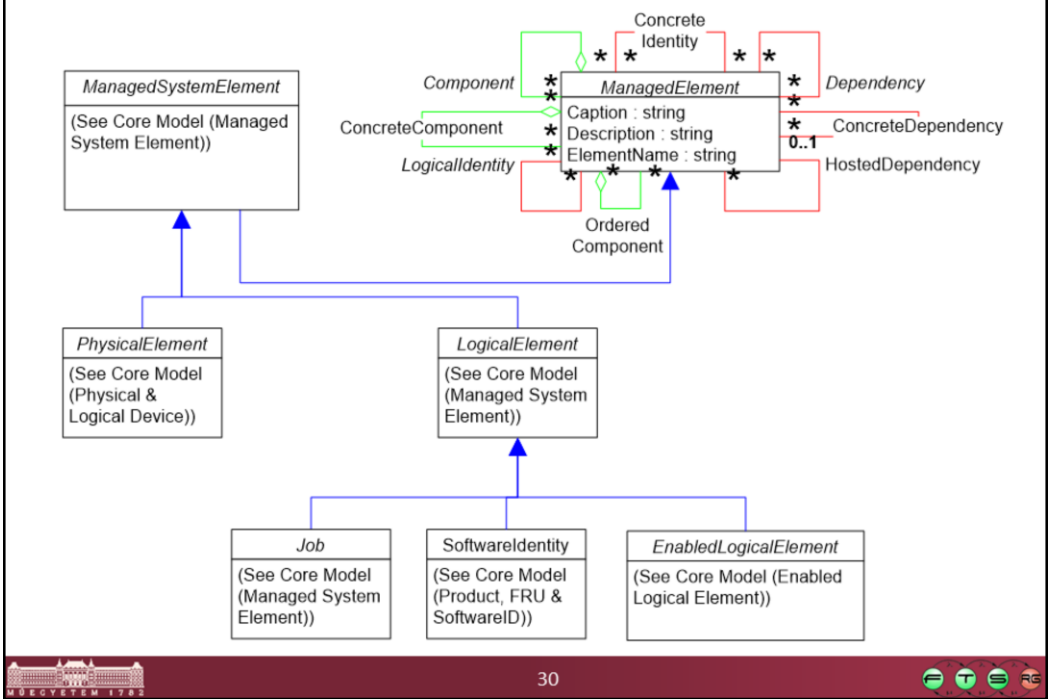
# Miket tartalmaz a CIM?

- **CIM Specification**
  - Szöveges dokumentum embereknek
  - Hogyan kell értelmezni ezt az egészet
- **CIM Metamodel**
  - Metamodell a későbbiekben definiált elemekhez
- **CIM Schema**
  - **Konfigurációs adatokhoz modellek**
  - **CIM Metamodelben definiált elemek példányai**
- **Managed Object Format (MOF)**
  - konkrét szintaxis CIM Schema elemekhez

## A CIM Schema szintjei

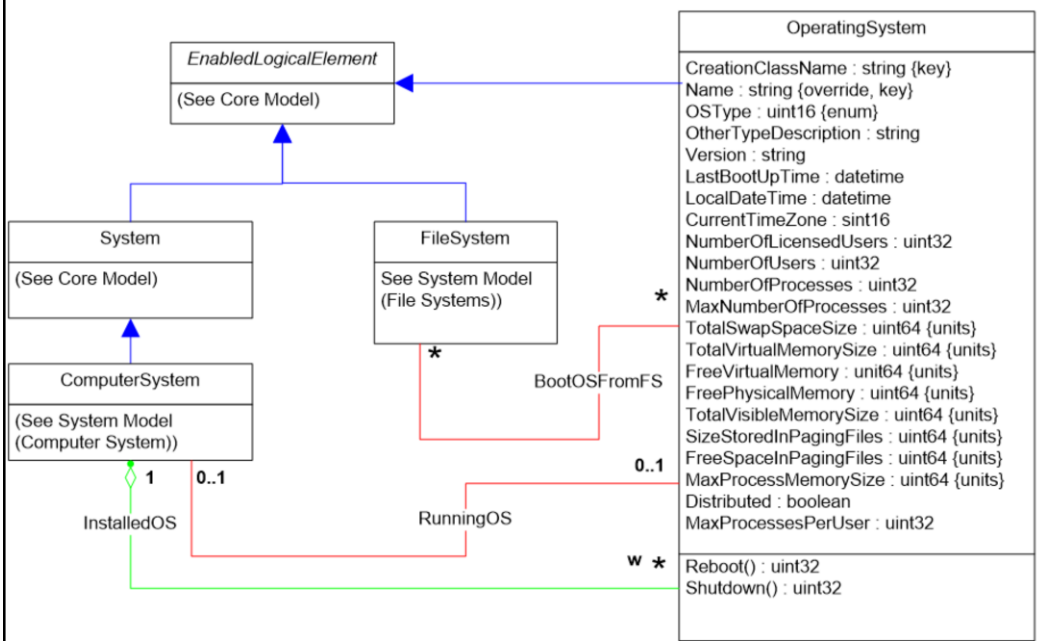
- Három szintbe szokás szervezni
  - Figyelem: ezek nem metaszintek!
  - Az egyes szintek elemei között öröklés van általában
- **Core Model:** általános fogalmak
  - Példák: ManagedElement, Setting, Location, FRU
- **Common Model:** szokásos területek technológia-független modelljei
  - Példák: Application, Database, Device
- **Extension Schemas:** gyártóspecifikus kiterjesztések
  - Példák: SAP\_SWProduct, HP\_BladeEnclosureCS

# Példa: CIM\_Core részlet



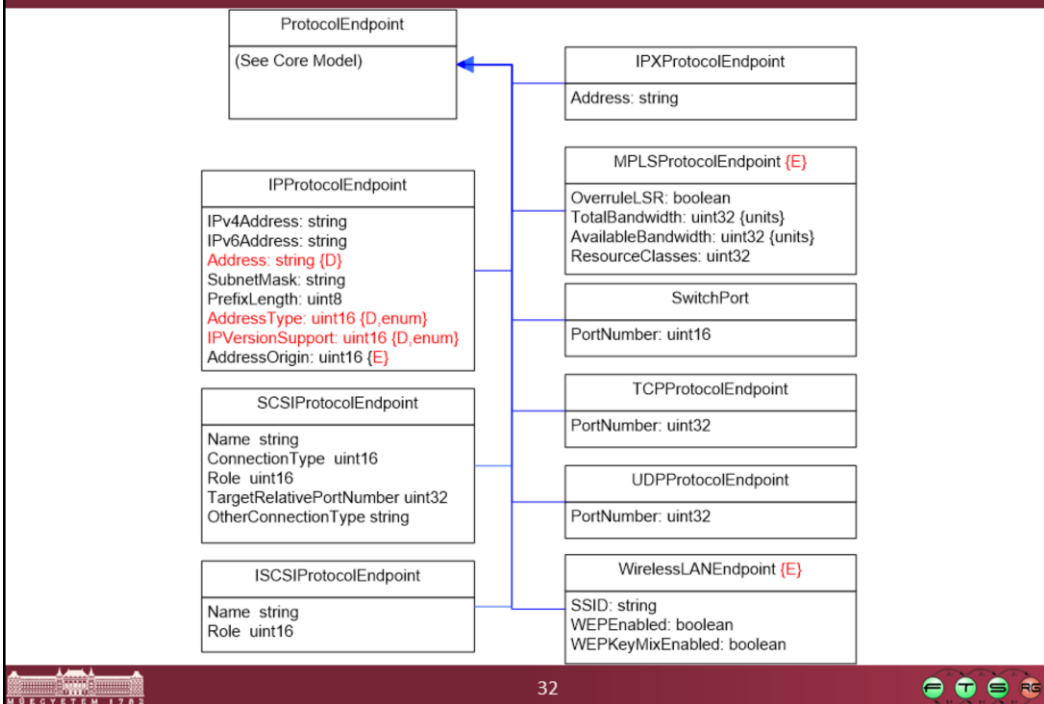
Itt nagyon általános, alapvető fogalmak szerepelnek még csak, amik a későbbiek csoportosításához lesznek hasznosak.

# Példa: CIM\_System részlet (Common)



Common model: Itt már konkrét menedzselhető elemek és tulajdonságaik jelennek meg, de még platform- és gyártófüggetlen módon.

## Példa: CIM\_Network részlet (Common)



Érdeemes megnézni, hogy például csak a hálózati végpontok esetén mennyi mindent kell modellezni, ami később konkrét rendszerek leírása esetén kellhet.



# A CIM Schema (v2.45) struktúrája

## CIM\_Schema (összesen ~1600 osztály!)

- CIM\_Application (139 osztály)
- CIM\_Core (193 osztály)
- CIM\_Database (19 osztály)
- CIM\_Device (401 osztály)
- CIM\_Event (30 osztály)
- CIM\_Interop (33 osztály)
- CIM\_IPsecPolicy (25 osztály)
- CIM\_Metrics (21 osztály)
- CIM\_Network (282 osztály)
- CIM\_Physical (47 osztály)
- CIM\_Policy (56 osztály)
- CIM\_Support (61 osztály)
- CIM\_System (198 osztály)
- CIM\_User (116 osztály)

A teljes modell hatalmas.

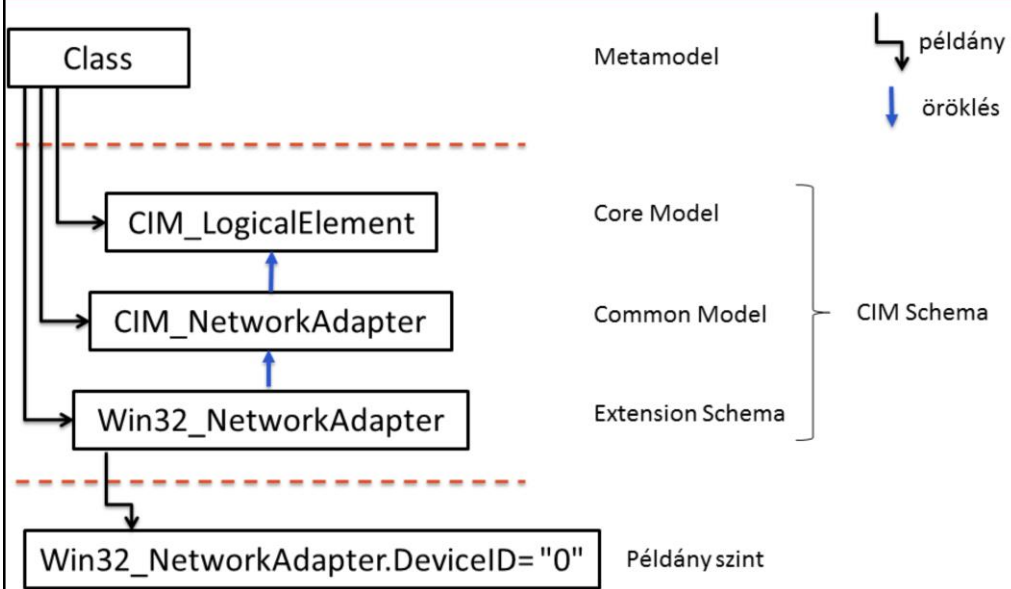
(Viszont: egy eszköz „CIM megfelelése” pusztán a képesség MOF állományok betöltésére és exportálására)



Ráadásul a modellt folyamatosan frissítik, kb. 4-6 havonta publikálnak egy friss verziót, ami 20-30 új osztályt ad hozzá a modellhez.

A teljes modell a <http://www.dmtf.org/standards/cim> oldalról tölthető le szöveges és grafikus formában is.

# CIM Metamodel és Schema viszonya



- <http://www.dmtf.org/standards/cim>
- Specification: terminológia, metamodell
- Schema leírások: PDF ábrák

# Miket tartalmaz a CIM?

- **CIM Specification**
  - Szöveges dokumentum embereknek
  - Hogyan kell értelmezni ezt az egészet
- **CIM Metamodel**
  - Metamodel a későbbiekben definiált elemekhez
- **CIM Schema**
  - Konfigurációs adatokhoz modellek
  - CIM Metamodelben definiált elemek példányai
- **Managed Object Format (MOF)**
  - konkrét szintaxis CIM Schema elemekhez

# Managed Object Format (MOF)

- Szöveges leírónyelv
  - Felfogható a CIM Metamodel konkrét szintaxisaként
  - CIMOM import/export megvalósítása
- Metamodellek és modellek egy .mof állományban is lehetnek!
- (A szintaxisra itt nem térünk ki részletesen)

## Példa: CIM kapcsolóosztályok

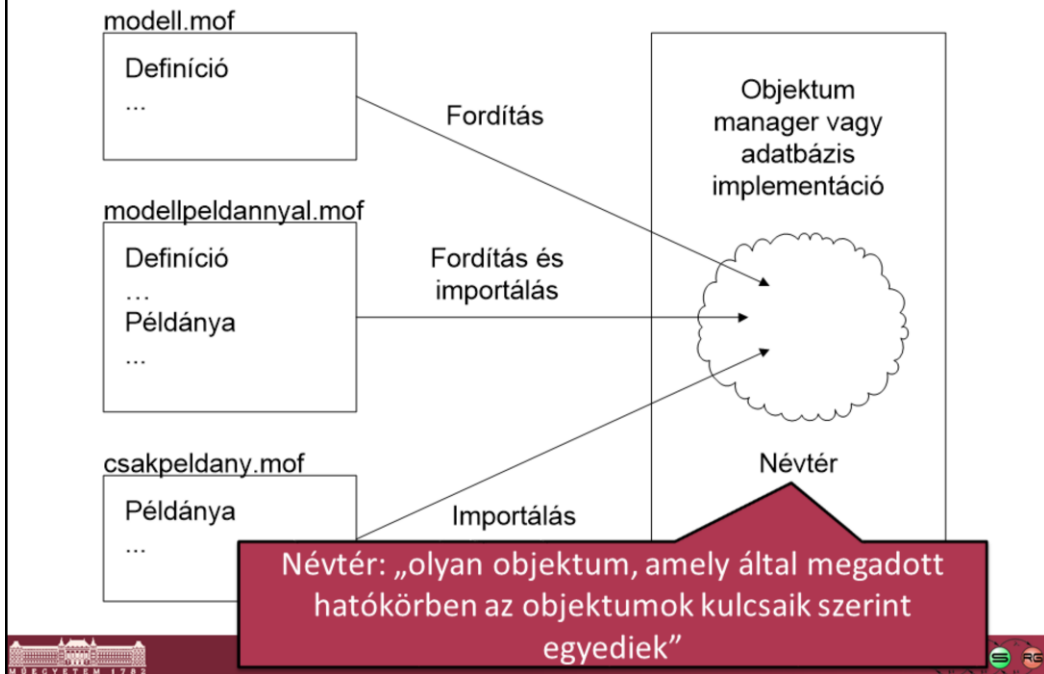
### Asszociáció megadása:

```
[Association]
class CIM_RunningOS : CIM_Dependency {
    CIM_OperatingSystem REF Antecedent;
    CIM_ComputerSystem REF Dependent;
};
```

### Kompozíció megadása:

```
[Association, Aggregation, Composition]
class CIM_ComputerSystemResource {
    [Aggregate]
    CIM_ComputerSystem REF GroupComponent;
    CIM_SystemResource REF PartComponent;
};
```

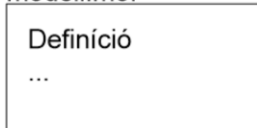
# MOF állomány alapú adatcsere



MOF esetén egy MOF fájlban lévő osztály- és típusdefiníciók importálását fordításnak (compile) szokták hívni.

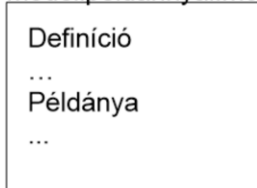
# MOF állomány alapú adatcsere

modell.mof



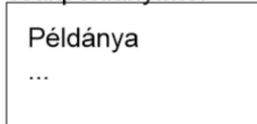
Fordítás

modellpeldanyal.mof



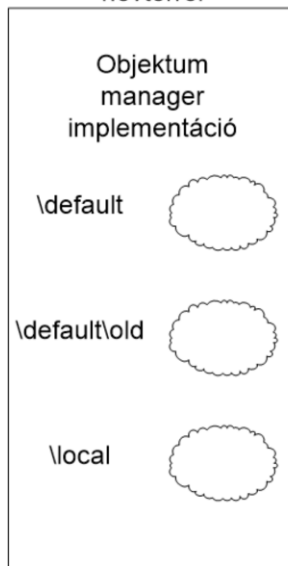
Fordítás és  
importálás

csakpeldany.mof



Importálás

Implementáció több  
névtérrel





- CIM Schema MOF leírása
  - Pl.: CIM\_Location
  - Minősítők [ ] között
- Wbemtest.exe
  - root\cimv2 névtérhez csatlakozás
  - CIM\_LogicalDevice osztály leszármazottai
    - Osztály definíció (minősítők, tulajdonságok, metódusok)
    - MOF megnézése
    - Példányait megnézni, tulajdonságok értékei, MOF megnézése

## CIM összefoglaló

- A Common Information Model:
  - Definiál egy nyelvet menedzsment adatok objektum orientált modellezésére
  - Megadja modelleknek egy igen tág, konkrét technológiáktól független, hierarchikus, kiterjeszhető halmazát
- A modellek UML-ben jól vizualizálhatóak
- A modellek (sémák) MOF és grafikus formátumban is elérhetőek

# Web Based Enterprise Management (WBEM)

**Kibocsájtó:** Distributed Management Task Force (DMTF)  
**Cél:** CIM-et támogató rendszermenedzsment  
protokollok

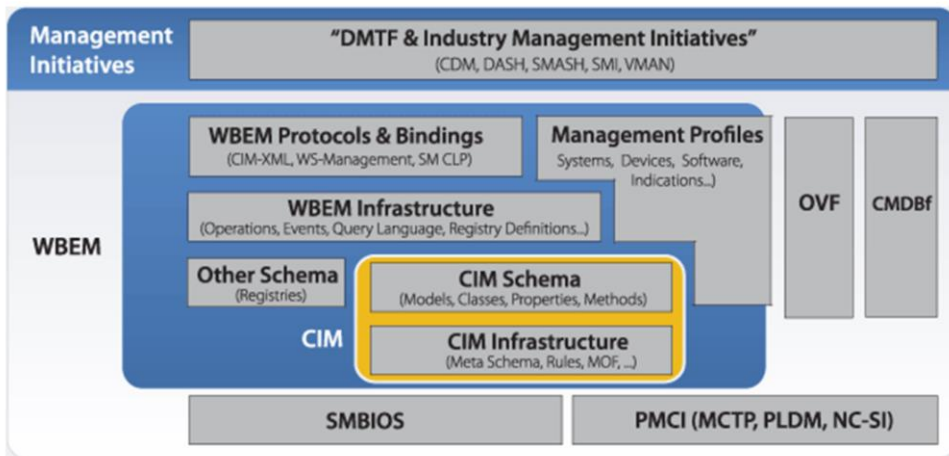
## Szabványos adatmodelltől a protokollokig

- Az alap CIM szabvány sok mindent **nem definiál**:
- **Hozzáférési protokoll**
  - Egy CIMOM, mint kiszolgáló milyen kéréseket értelmezzen?
  - Hogyan válaszoljon?
  - Milyen protokollon keresztül legyen elérhető?
- **Lekérdezőnyelv?**
- **Szolgáltatás-felderítés?**

# Web Based Enterprise Management

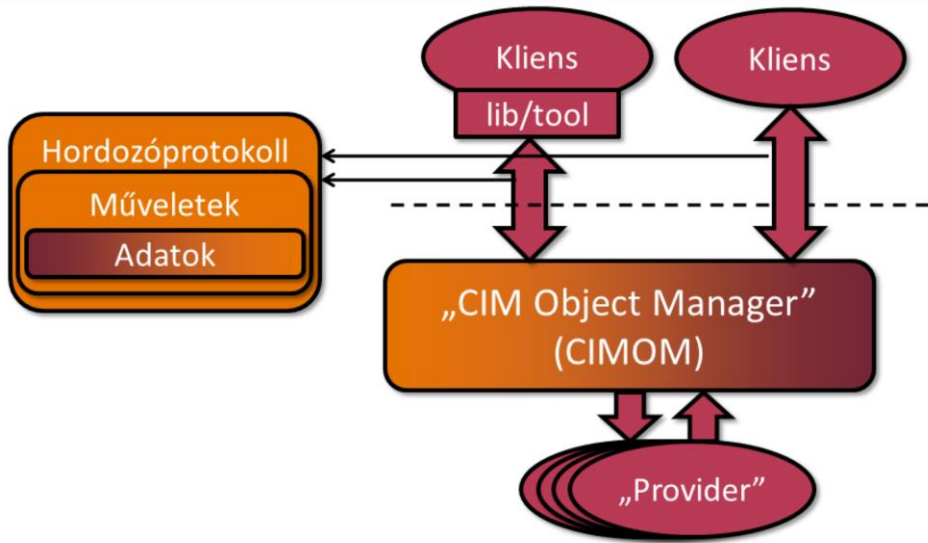
- A WBEM nem egy szabvány, hanem szabvány-készlet
  - **Protokoll:** CIM-XML **vagy** WS-Management
    - 2013-tól: új CIM-RS (CIM RESTful Services)
  - **Lekérdezőnyelv:** CIM Query Language (CQL)
  - **Szolgáltatás-felderítés:** WBEM Discovery Using the Service Location Protocol (SLP)

# DMTF Technologies Diagram



forrás: <http://dmtf.org/standards/stackmap>

# WBEM



# CIM-XML

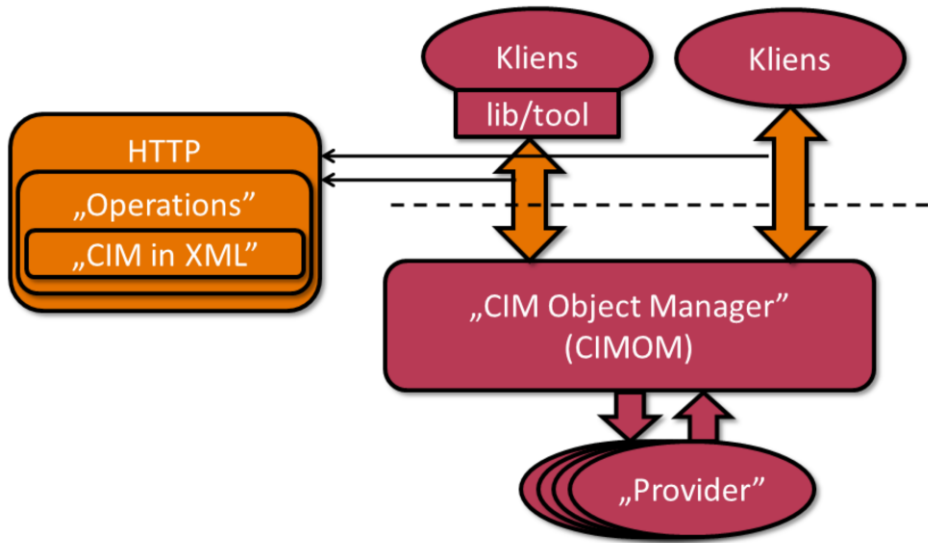
<b>Kibocsájtó:</b>	Distributed Management Task Force
<b>Verzió:</b>	XML DTD: 2.4.0 (final, 2014) Representation of CIM in XML: 2.4 (final, 2014) CIM Operations over HTTP: 1.4 (final, 2014)
<b>Cél:</b>	CIM hordozó és interop. protokoll



# CIM-XML

- **Cél:** XML-ben reprezentált CIM adatok HTTP felett
  
- **Részei:**
  - **Információ-reprezentáció XML-ben**  
*(Representation of CIM in XML)*
  - **Műveletek definíciója**  
*(CIM Operations over HTTP)*
  - **Lekérdezéshez nyelv**  
*(CIM Query Language Specification)*

# WBEM: CIM-XML



## Példa: egy tulajdonság lekérdezése

```
<?xml version="1.0" encoding="utf-8" ?>
<CIM CIMVERSION="2.0" DTDVERSION="2.0">
  <MESSAGE ID="87872" PROTOCOLVERSION="1.0">
    <SIMPLEREQ><IMETHODCALL NAME="GetProperty">
      <LOCALNAMESPACEPATH>
        <NAMESPACE NAME="root"/>
        <NAMESPACE NAME="myNamespace"/>
      </LOCALNAMESPACEPATH>
      <IPARAMVALUE NAME="InstanceName">
        <INSTANCENAME CLASSNAME="MyDisk">
          <KEYBINDING NAME="DeviceID">
            <KEYVALUE>C:</KEYVALUE></KEYBINDING>
          </INSTANCENAME>
        </IPARAMVALUE>
        <IPARAMVALUE NAME="PropertyName">
          <VALUE>FreeSpace</VALUE> [...]
        </IPARAMVALUE>
      </IMETHODCALL></SIMPLEREQ></MESSAGE>
    </CIM>
```



Miért érdemes megnézni egy ilyen üzenetet:

- Valószínűleg nekünk kézzel nem kell megírni, arra megvannak a megfelelő programok/könyvtárak.
- De hibakeresés esetén van, hogy csak az segít, hogy ha megnézzük, hogy mi megy át a hálózaton. Ilyenkor jó, ha tisztában vagyunk, hogy a CIM hogyan képződik le XML-re.

## Példa: lekérdezésre válasz

```
HTTP/1.1 200 OK
Content-Type: application/xml; charset="utf-8"
Content-Length: xxxx
Ext:
Cache-Control: no-cache
Man: http://www.dmtf.org/cim/mapping/http/v1.0 ; ns=73
73-CIMOperation: MethodResponse
```

```
<?xml version="1.0" encoding="utf-8" ?>
<CIM CIMVERSION="2.0" DTDVERSION="2.0">
<MESSAGE ID="87872" PROTOCOLVERSION="1.0"><SIMPLERSP>
  <IMETHODRESPONSE NAME="GetProperty">
    <IRETURNVALUE>
      <VALUE>6752332</VALUE>
    </IRETURNVALUE>
  </IMETHODRESPONSE>
</SIMPLERSP></MESSAGE></CIM>
```

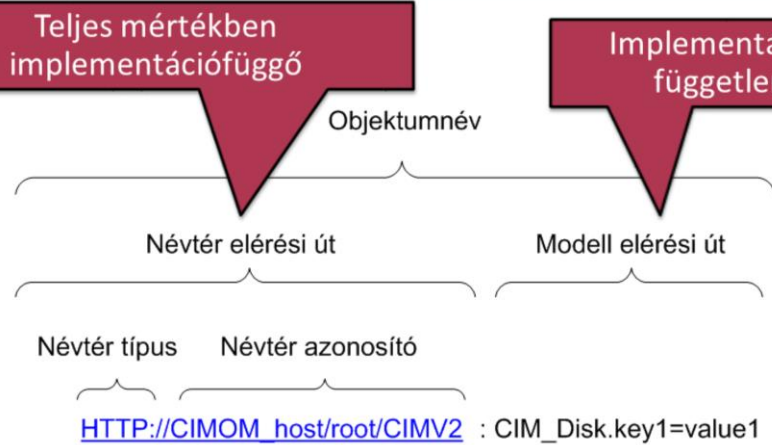
## CIM-XML műveletek (részlet)

- **Osztály**
  - GetClass
  - EnumerateClasses
  - EnumerateClassNames
  - ...
- **Kapcsolatok**
  - Associators
  - AssociatorNames
  - References
- **Példány**
  - GetInstance
  - EnumerateInstances
  - EnumerateInstanceNames
  - GetProperty
  - ...
- **Lekérdezés**
  - ExecQuery



Ezek az úgynevezett **intrinsic** metódusok, amik maguk a CIM-műveletek, ezeken kívül még a menedzselt elem is definiálhat saját metódusokat, amiket meg lehet rajta hívni.

# CIM objektumok megnevezése



A terminológia angolul:

- Object name
- Namespace Path
- Model Path
- Namespace type
- Namespace handle

## WBEM (CIM-XML): eszköztámogatás

- OpenPegasus
  - The Open Group fejleszti, nyílt forráskódú
  - Red Hat Enterprise Linux, HP WBEM Services...
  
- SBLIM („sublime”)
  - Standards Based Linux Instrumentation
  - összefogó projekt
  
- ...

## Standards Based Linux Instrumentation

- Nyílt forrású, ingyenes projekt
  - Eredetileg IBM fejlesztés
  - Több részprojekt
- CIMOM: **SFCB** (Small Footprint CIM Broker)
- Linux **CMPI Provider**-ek (lásd CMPI)
  - cmpi-base, cmpi-network, cmpi-service, cmpi-syslog ...
- Kliens: **Java CIM Client**
  - JSR 48 implementáció (~ WBEM Java API)
- Kliens: **wbemcli** (parancssori)



# wbemcli

- Egyszerű parancssori CIM kliens
- Szintaxis:
  - `wbemcli <op> objectPath`
- Műveletek (op)
  - gc – get class, gi – get instance
  - ei – enumerate instances
  - ...
- ObjectPath: CIM objektum teljes neve
  - `<scheme>://[user:pwd@]<host>:<port>/<namespace>[/...]>:<class-name>[.<key=value[,...]>]`

# DEMO wbemcli

- **Környezet:**
  - OpenPegasus, openlmi providerek és wbemcli
  - Tipikus portok: wbem-http (5988), wbem-https (5989)
- **YAWN (Yet Another WBEM Navigator)**
- **wbemcli**
  - Szolgáltatott osztályok listázása
  - CIM\_OperatingSystem
  - CIM\_UnixProcess; XML nézet



58



## Telepítés

- Csomagok: openlmi, sblim-wbemcli, yawn
- OpenPegasus indítása/leállítása: `systemctl start tog-pegasus.service`
- Milyen portol figyel az OpenPegasus?
  - `sudo netstat -t -l -p | grep cimserver`

## YAWN

`http://localhost/yawn`  
Osztályok és példányok listázása

## wbemcli

`man wbemcli` (ami fontos: operációk)

Lássuk mit tud  
- `wbemcli`  
- Kiírja a helpet

Próbáljuk egy biztosan létező alap osztályt lekérdezni

- A netstat segítségével megfigyelt porton nézzük
- Használjuk az objektum megnevezésre a szabványos `objectPath`-t, az alap osztályok a `root/cimv2` névtérben szoktak lenni
- `wbemcli gc 'http://localhost:5988/root/cimv2:CIM_OperatingSystem'`
- Eredmény: `username/password required`

Adjunk meg felhasználónevet és jelszavát is

```
wbemcli gc 'http://root:LaborImage@localhost:5988/root/cimv2:CIM_OperatingSystem'
```

Jó lesz, csak nehezen olvasható

- Az `nl` kapcsoló megadásával új sort szúr be minden tulajdonság után
- `wbemcli -nl gc 'http://root:LaborImage@localhost:5988/root/cimv2:CIM_OperatingSystem'`

Most nézzünk meg ennek az osztálynak a példányaait:

```
wbemcli -nl ei 'http://root:LaborImage@localhost:5988/root/cimv2:CIM_OperatingSystem'
```

Működik! Nézzük, mit lehet még lekérdezni:

```
wbemcli -nl ei 'http://root:LaborImage@localhost:5988/root/cimv2:LMI_FileSystemCapabilities'
```

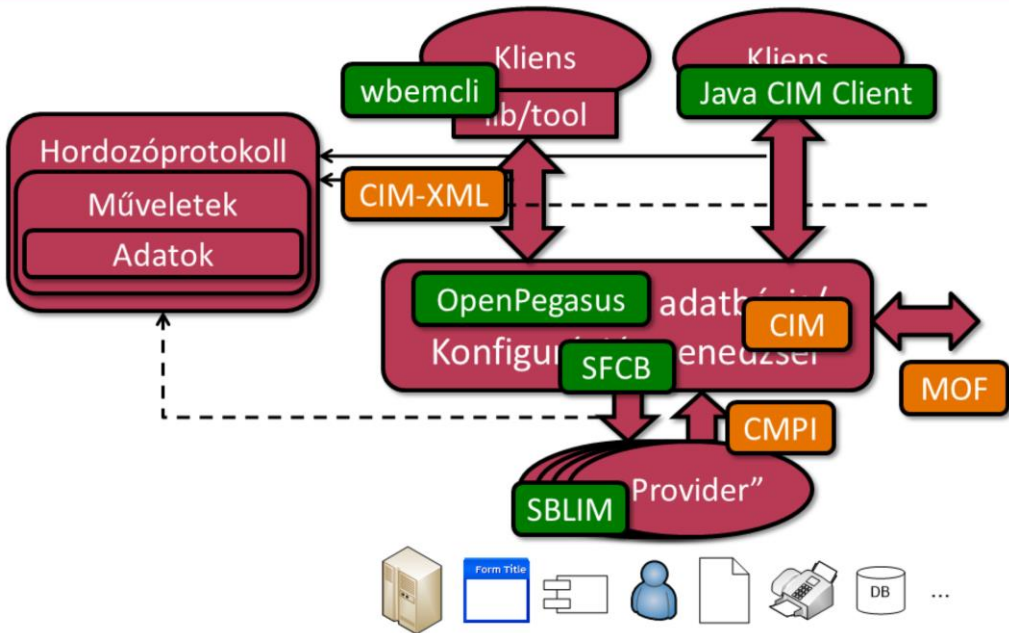
Kérdezzünk most csak le egy konkrét példányt:

- Ehhez a kulcs tulajdonságait meg kell adni, ez ennél az osztálynál az `InstanceID`
- `wbemcli -nl gi 'http://root:LaborImage@localhost:5988/root/cimv2:LMI_FileSystemCapabilities.InstanceID="LMI:LMI_FileSystemCapabilities:btfs"'`
- Így már csak egy konkrét példányt kapunk vissza

Nézzük meg, hogy ilyenkor hogyan néz ki a CIM-XML kérés és válaszüzenet

```
wbemcli -nl -dx gi 'http://root:LaborImage@localhost:5988/root/cimv2:LMI_FileSystemCapabilities.InstanceID="LMI:LMI_FileSystemCapabilities:btfs"'
```

# Összefoglalás



## Linkek

- DMTF: Common Information Model szabványok
  - <http://dmtf.org/standards/cim>
  - CIM FAQ, [http://dmtf.org/about/faq/cim\\_faq](http://dmtf.org/about/faq/cim_faq)
- Web Based Enterprise Management
  - <http://www.dmtf.org/standards/wbem/>
  - CIM-XML protocol
  - Web Services for Management
    - <http://www.dmtf.org/standards/wsman/>
- Implementációk
  - SBLIM <http://sblim.sourceforge.net/wiki>
  - OpenPegasus, <http://www.openpegasus.org/>



<http://dmtf.org/standards/cim>  
[http://dmtf.org/about/faq/cim\\_faq](http://dmtf.org/about/faq/cim_faq)  
<http://www.dmtf.org/standards/wbem/>  
<http://www.dmtf.org/standards/wsman/>  
<http://sblim.sourceforge.net/wiki/>  
<http://www.openpegasus.org/>