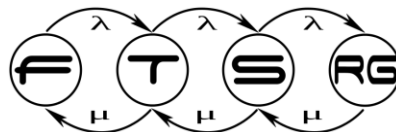


Strukturális modellezés

Szárnyas Gábor, Bergmann Gábor,
Gönczy László, Pataricza András,

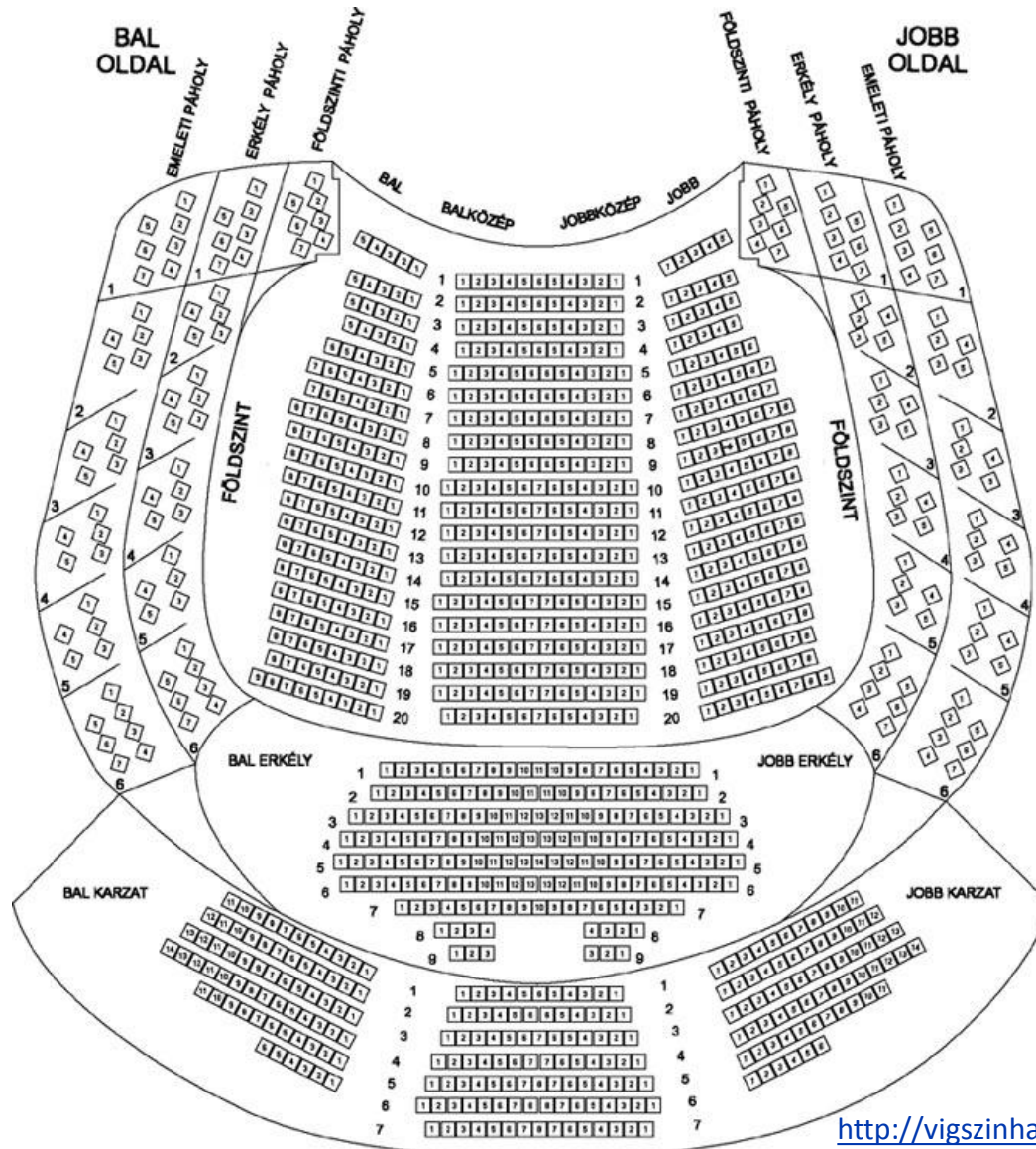
Budapest University of Technology and Economics
Fault Tolerant Systems Research Group



Miről lesz szó?

- Struktúra modellezés célja, alkalmazásai
- Dekompozíció
- Modell elemek leírása gráfokkal
- Tipikus kérdések (felépítés, elérhetőség)

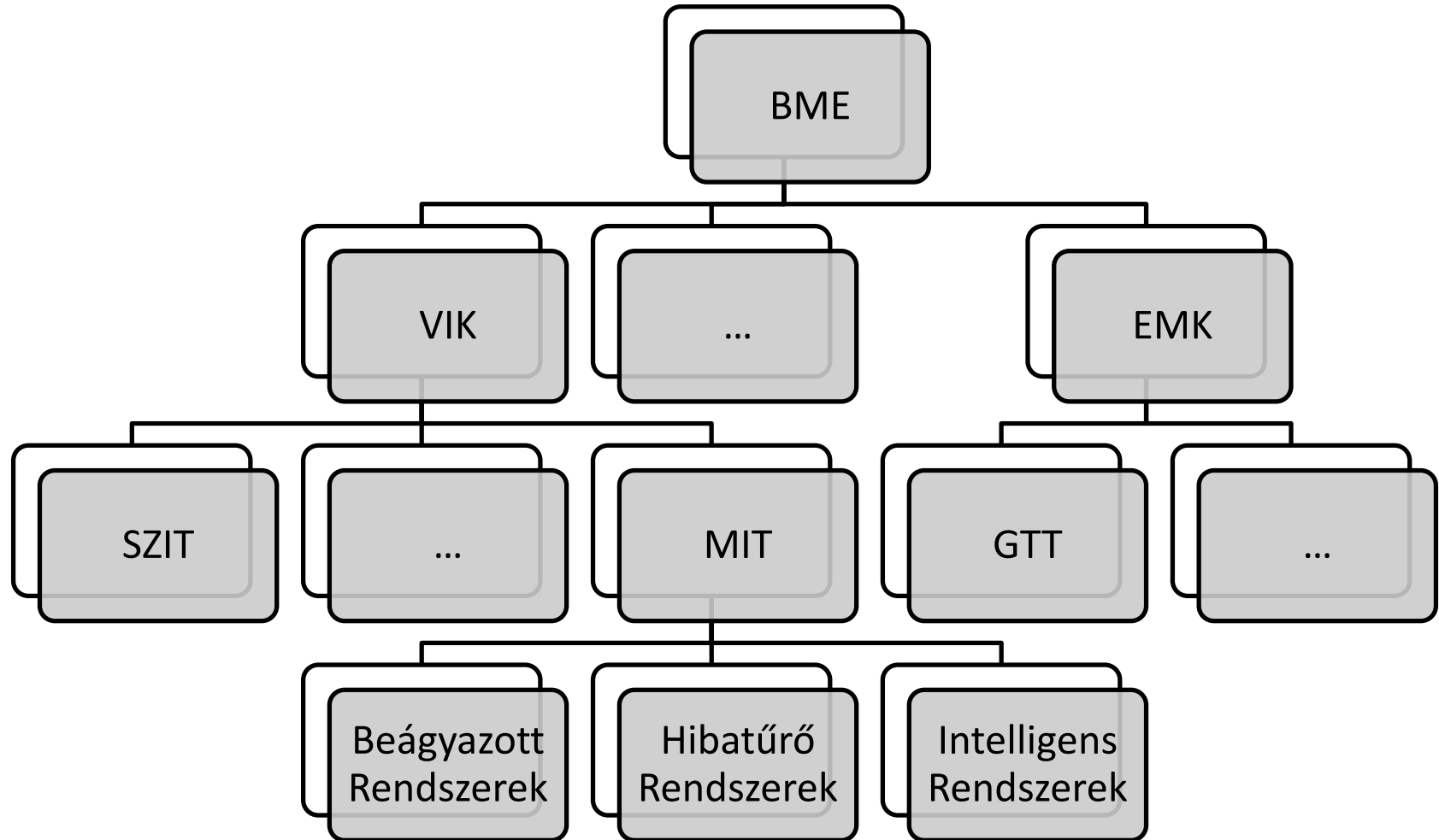
Illusztráció – Felépítési modellek



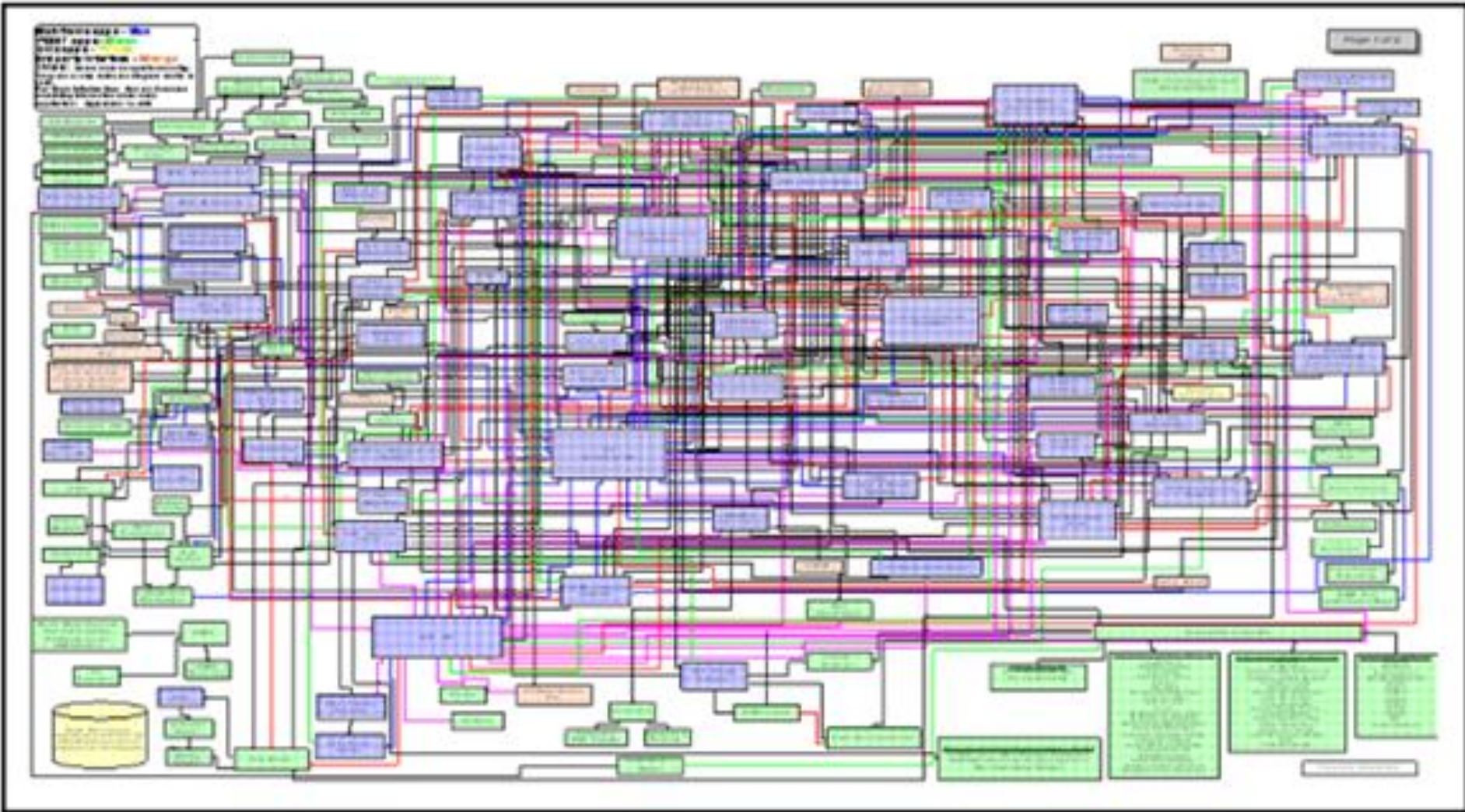
http://vigszinhaz.hu/images/upload/nezo_v1.jpg

Illusztráció – Felépítési modellek

Szervezeti felépítés (ld. tartalmazási hierarchia)



(... átlát a káoszon?)

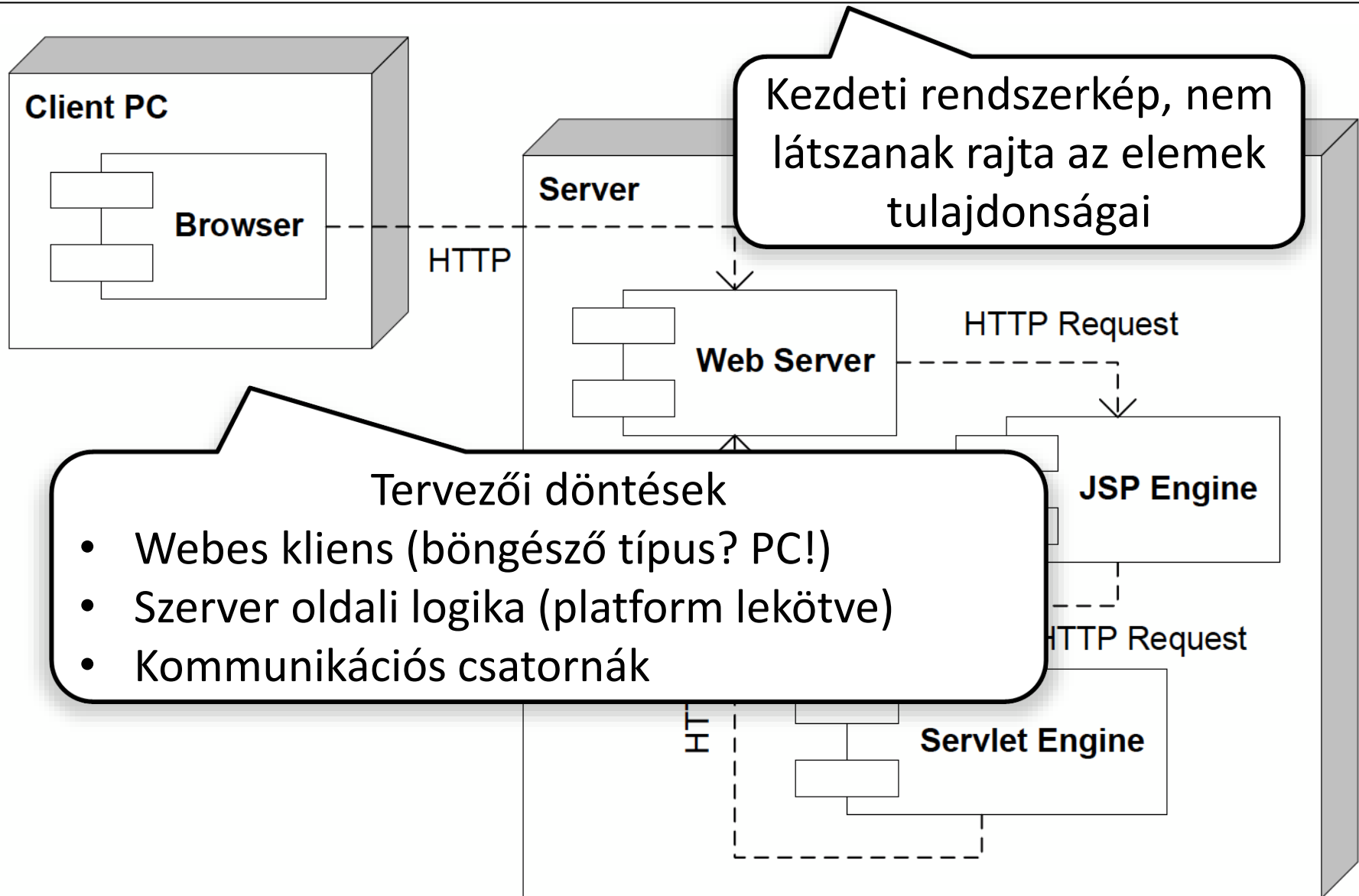


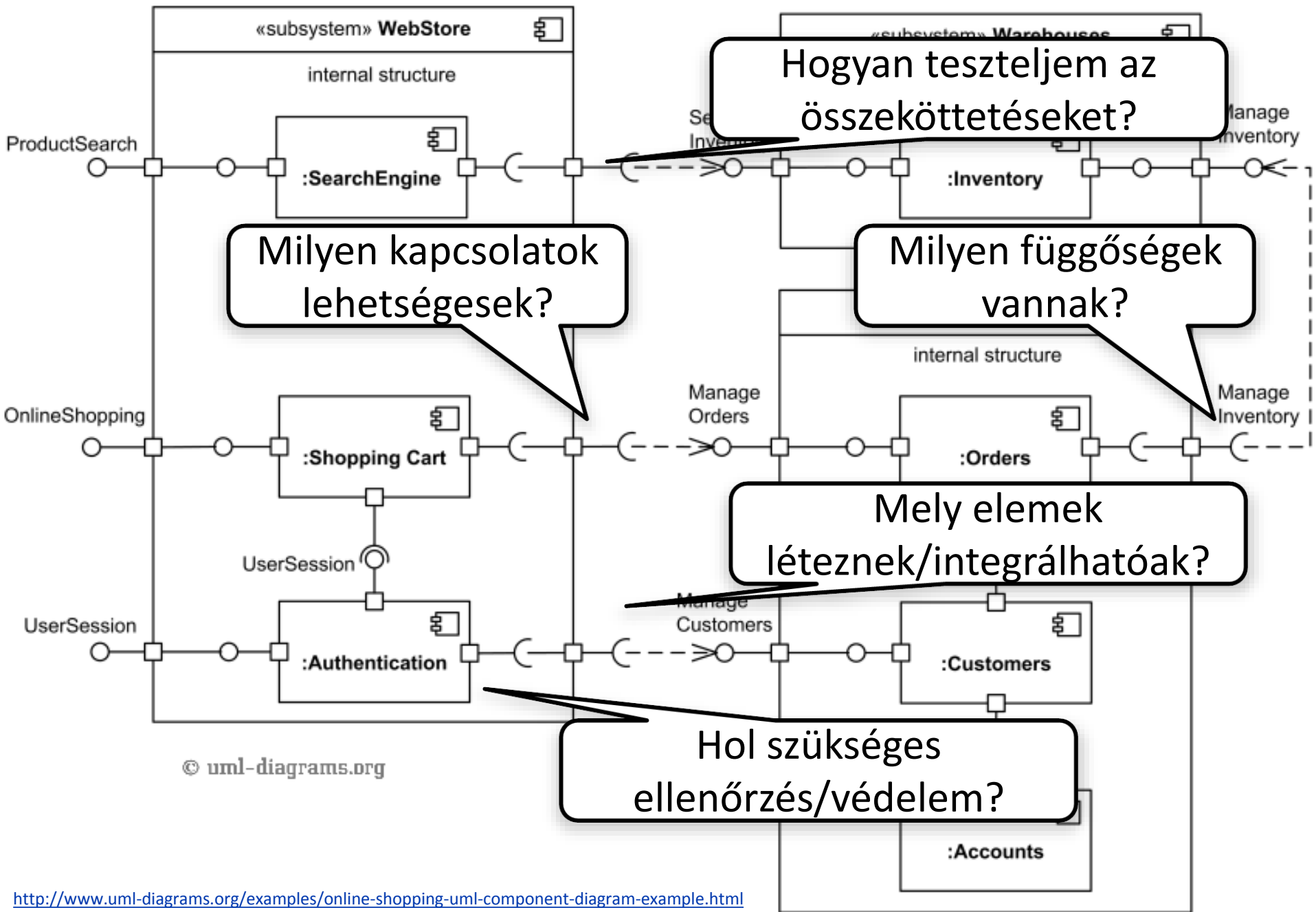
<http://www.mikethearchitect.com/2012/10/challenge-the-status-quo-and-advance-business-through-cloud-computing.html>

Strukturális modellezés célja

- Rendszer részekre bontása
 - Kisebb egységeket könnyebb megtervezni
 - Részegységek felhasználása
 - Általános célú komponensek használata
- Létező rendszer dokumentálása
 - “Rendszertérkép”
- Adatszerkezet megalkotása
 - Milyen információt kezelünk?
- Rendszer és specifikáció összevetése
 - Megfelel-e az elvárásoknak?

Hogyan épül fel a rendszerem?





<http://www.uml-diagrams.org/examples/online-shopping-uml-component-diagram-example.html>

Felépítési és viselkedési modellezés

■ Felépítési (*structural*)

- Statikus
- Rész és egész, összetevők
- Kapcsolatok, összeköttetések

■ Viselkedési (*behavioral*)

- Dinamikus
- Időbeli lefolyás
- Állapot, folyamat
- Reakciók a külvilágra

■ Nem fed le mindent, nem válik élesen szét...

Az autóban van kamera és kormányvezérlő

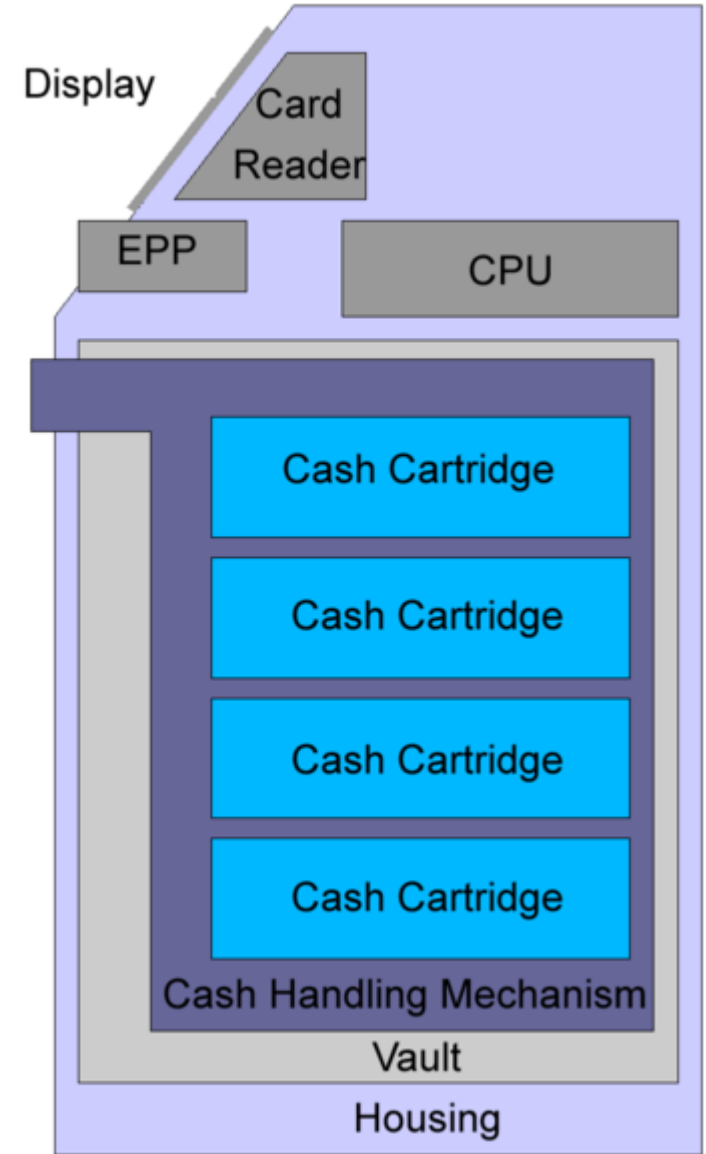
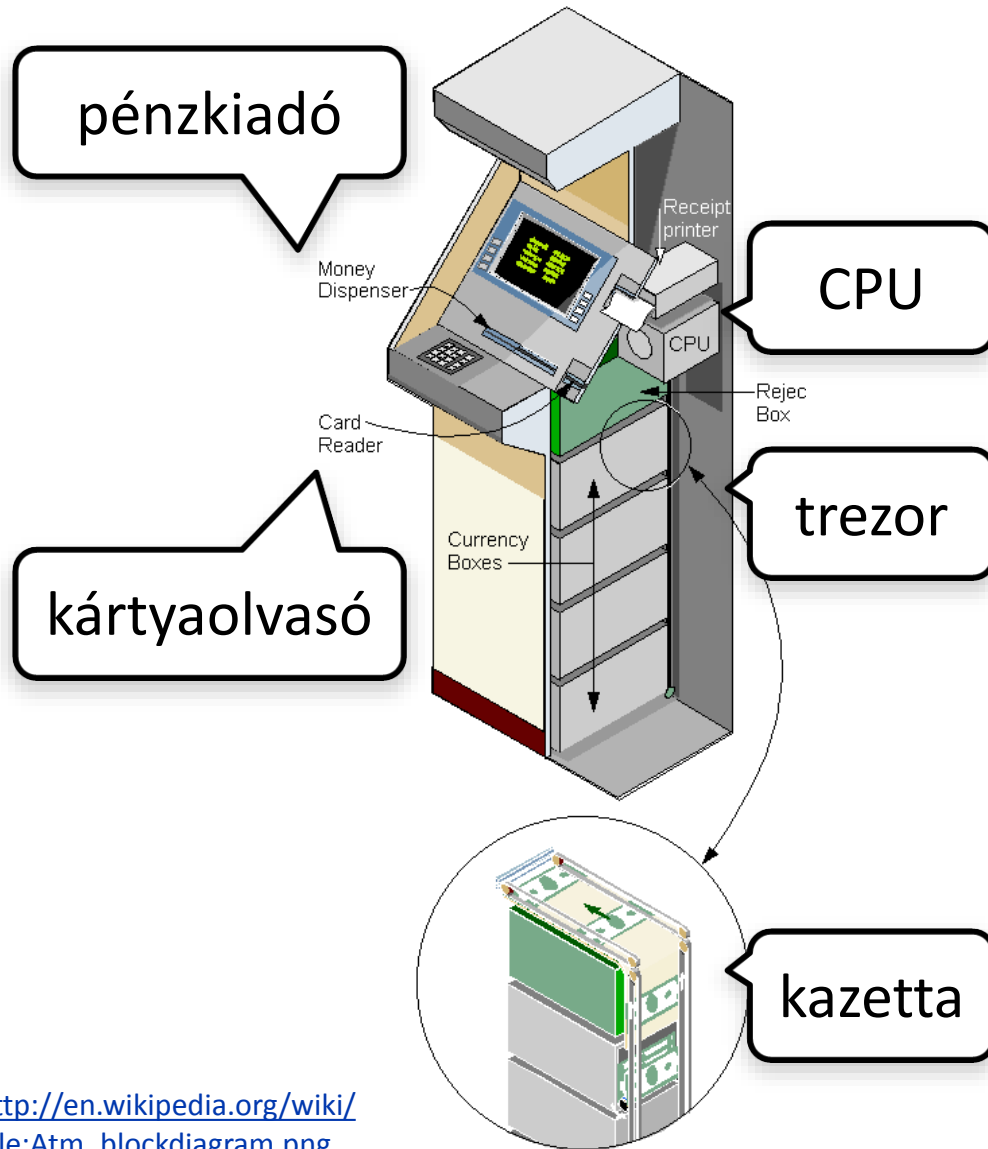
A kamera jeleket küld a sáv elhagyásáról (mennyit? mikor?)

A sávtartó rendszerben a kamera jeleit fogadva a kormányvezérlő beavatkozik (mikor/hogyan?)

Strukturális modell

- A rendszer felépítésére vonatkozó tudás
 - Milyen elemekből áll a rendszer?
 - Hogyan kapcsolódnak egymáshoz az elemek?
 - Milyen tulajdonságúak az elemek?

ATM struktúra és blokkdiagram



http://en.wikipedia.org/wiki/File:Atm_blockdiagram.png

How An Automated Teller Machine (ATM) Works

Withdrawing cash

1 After a user's ATM card and personal identification number (PIN) are approved, the ATM screen displays a list of options.

2 After the amount of a withdrawal is entered, the ATM opens one of four boxes by activating a door that rolls up to expose the face of a bill.

3 At the same time, a suction device pulls the correct number of bills.

4 The bills are placed on rollers and moved to a holding area until they are dispensed.

5 If the wrong number of bills is

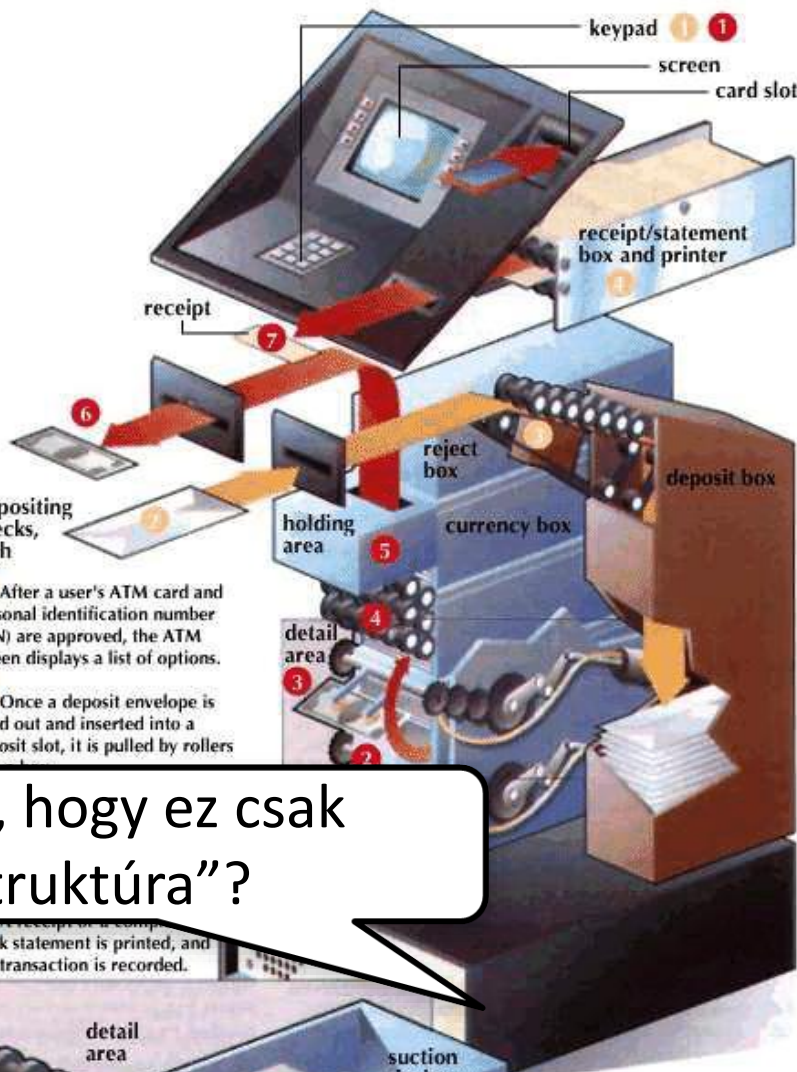
pull
AT
an
6
7

transaction is recorded.

1 After a user's ATM card and personal identification number (PIN) are approved, the ATM screen displays a list of options.

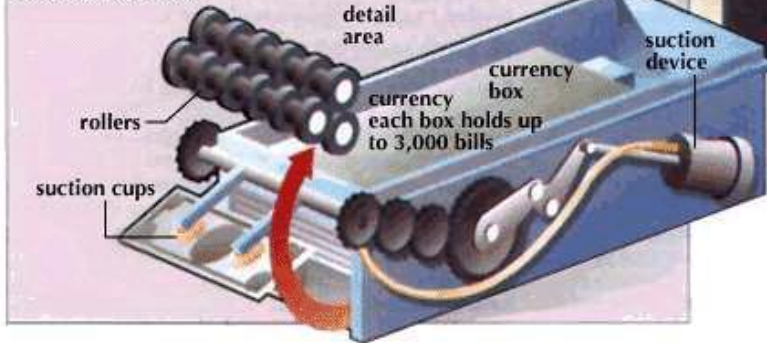
2 Once a deposit envelope is filled out and inserted into a deposit slot, it is pulled by rollers

receipt or a comp
bank statement is printed, and
the transaction is recorded.

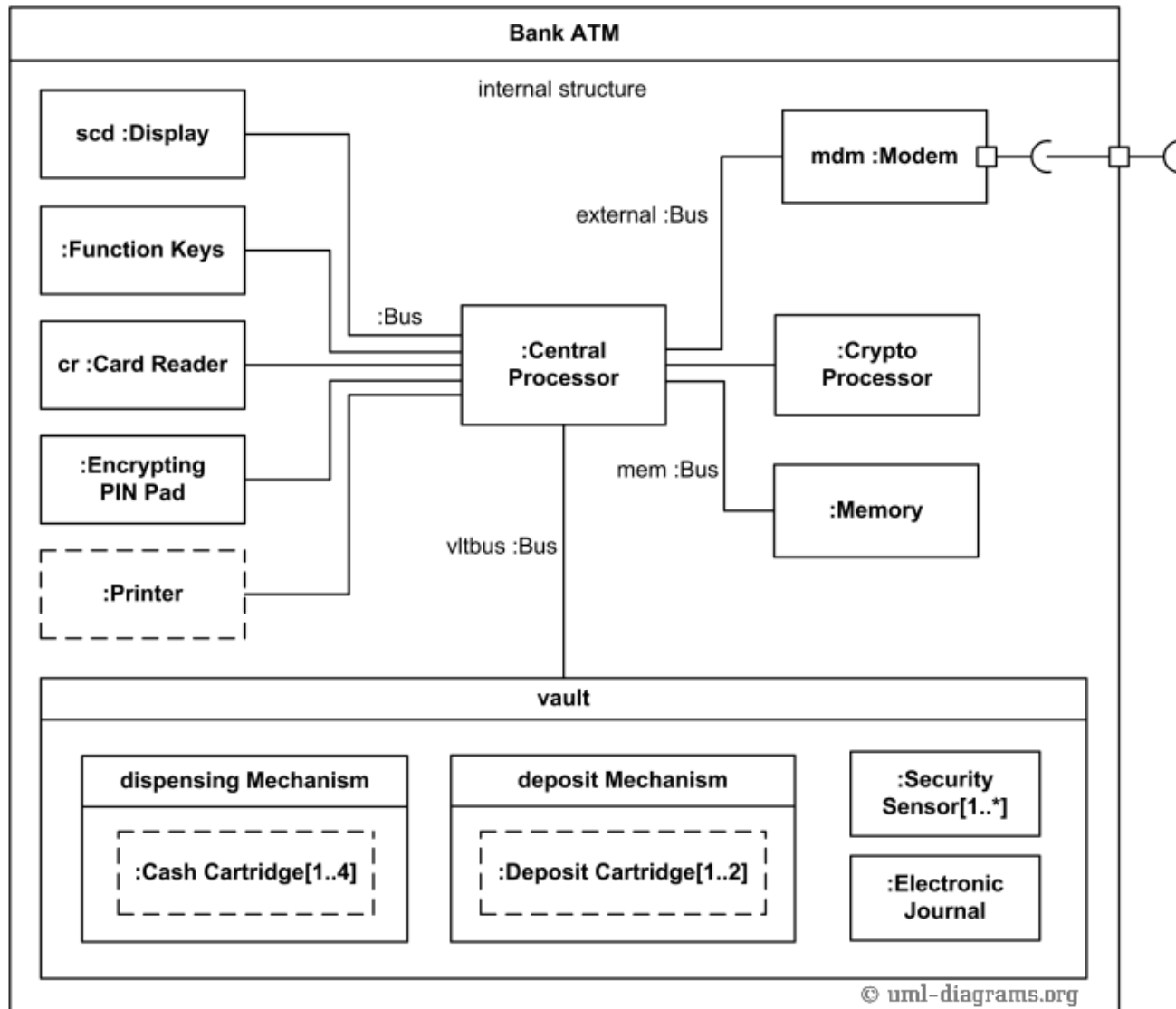


Biztos, hogy ez csak „struktúra”?

How the cash flows

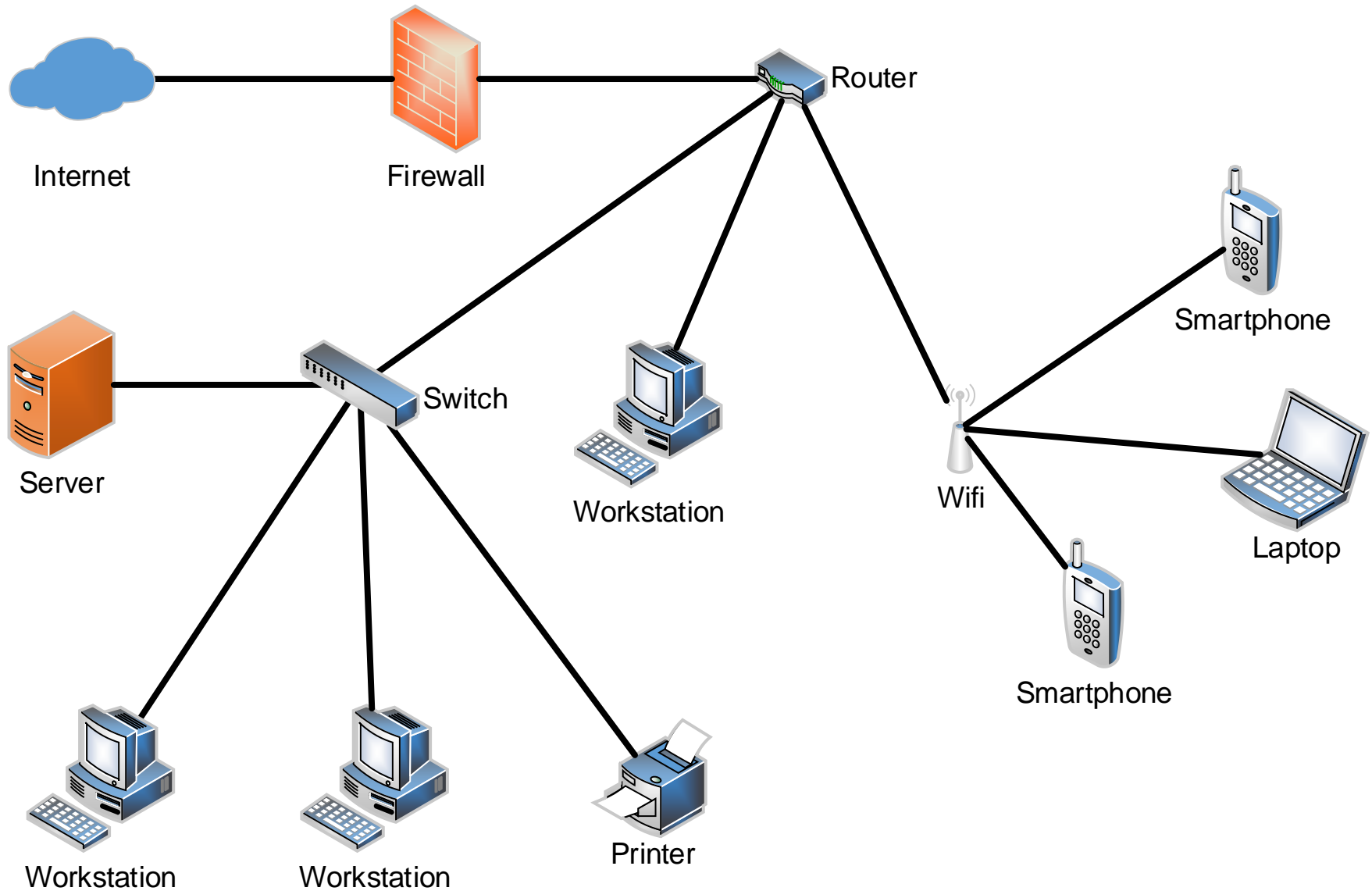


ATM struktúra

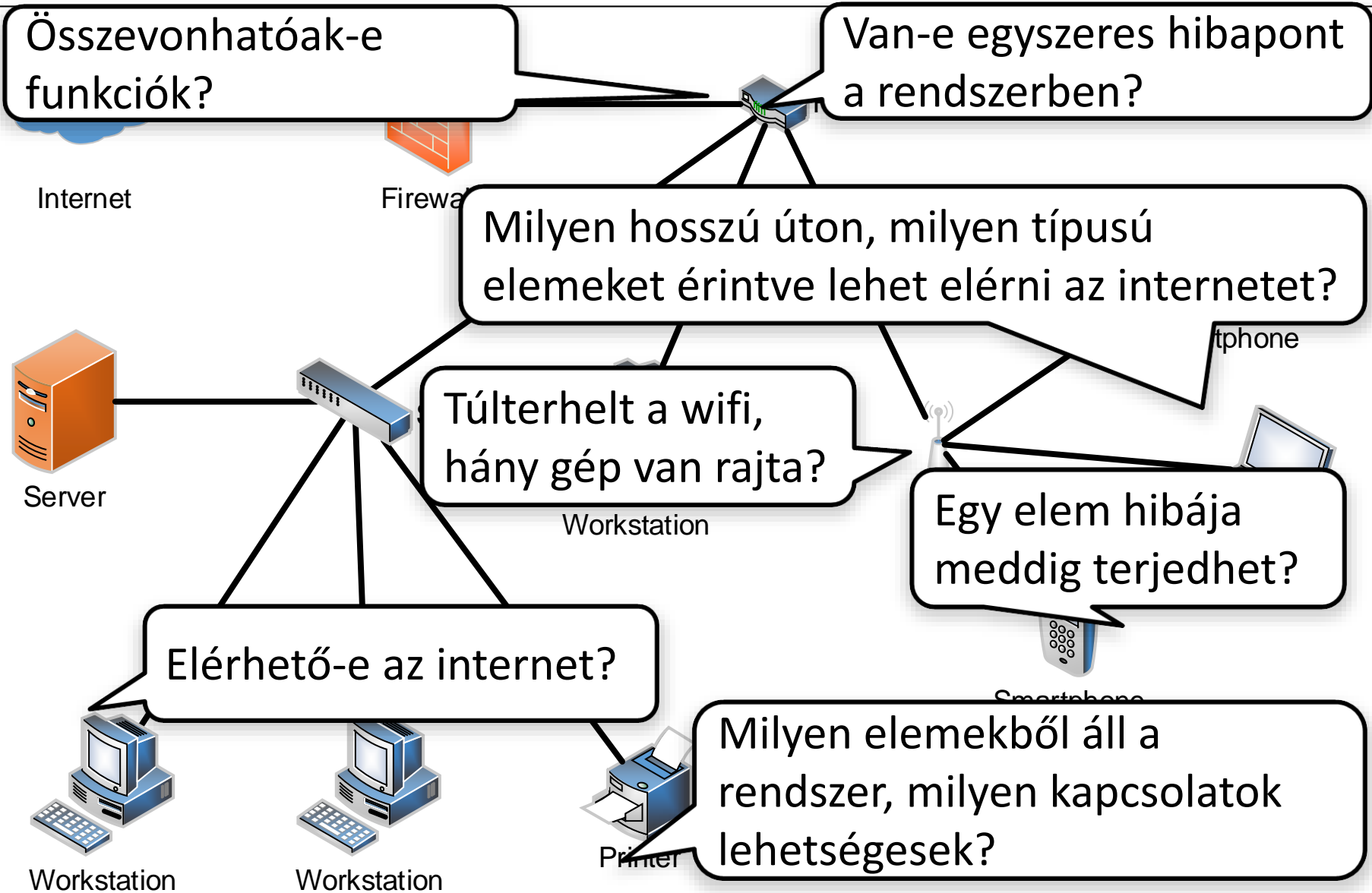


<http://www.pcmag.com/encyclopedia/term/38117/atm-machine>

Példa: (céges) hálózat



Példa: (céges) hálózat

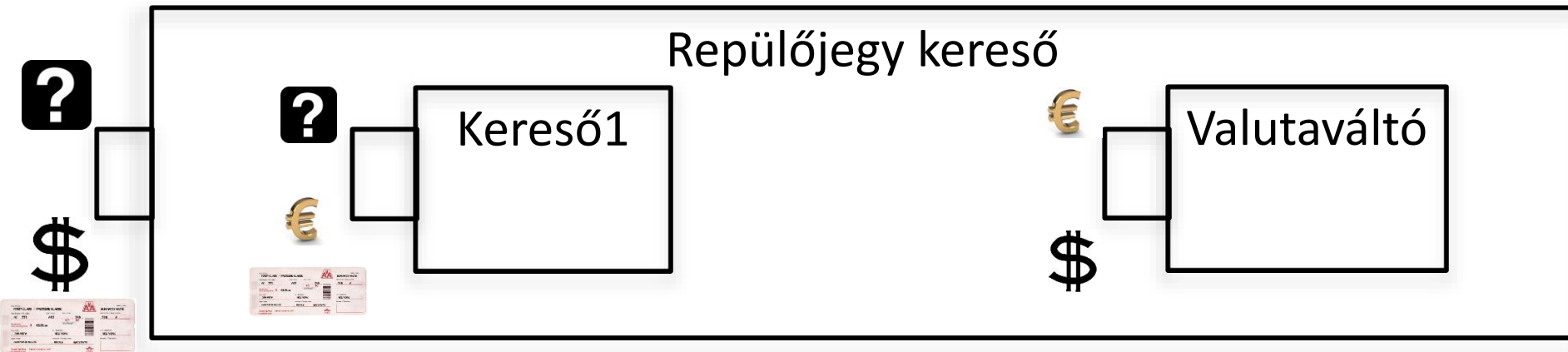


Dekompozíció

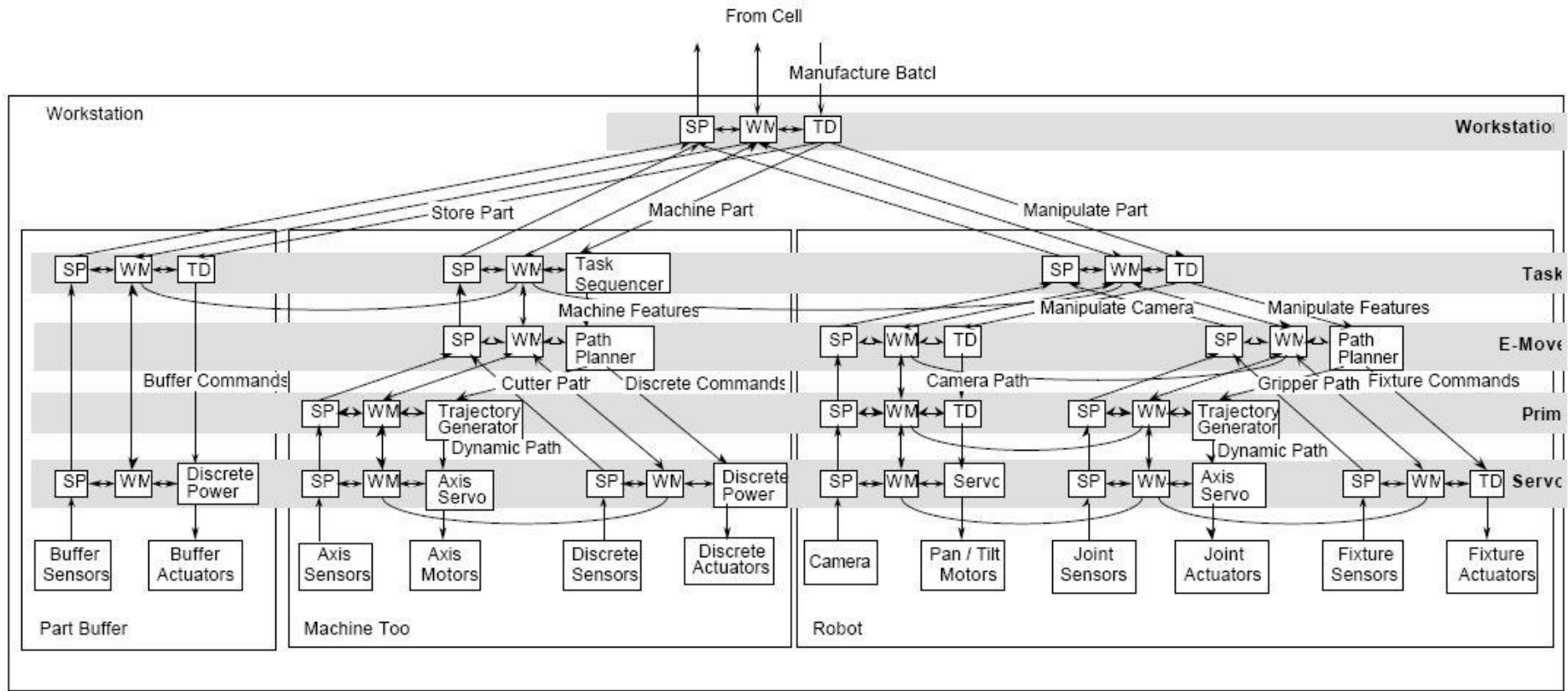
- **Decomposition** in computer science, also known as **factoring**, is breaking a complex problem or *system* into parts that are easier to conceive, understand, program, and maintain. (Wikipedia)
- A dekompozíció („faktoring”) tehát egy összetett probléma vagy *rendszer* kisebb részekre bontása, amelyek könnyebben érthetőek, fejleszthetőek és karbantarthatók.
- Funkcionális dekompozíció: ~blokkséma
- Hierarchikus dekompozíció: rész-egész viszony
- Alapja?
 - Elosztottság/hierarchia/algorithmika
- Hogyan ábrázoljuk/kezeljük általánosan?

Dekompozíció helyessége

- Részek teljessége ?= egész
- Ld később (pl. kommunikáló állapotgépek)
- Felépítés
 - Elemek megfelelő be/kimenettel dolgoznak?
 - Készen kapott elemek beilleszthetőek?
 - (Később: hibákat hogyan kezeljük?)



Példa: ipari vezérlő (Wikipedia)



Top-down tervezés

- Alaplépés: dekompozíció

Okmányirodai munkahely

Számítástechnikai eszközök

Kábelezés

POS terminál

Okmánynyomtató

PC munkaállomás

Monitor

Billentyűzet

Egér

Asztali gép

Fotófülke

Paraván

Szék

Fények

Kamera

Épített környezet

Íróasztal

Ügyfélablak

Szék

Zárható fiók

Bottom-up tervezés

- **Alaplépés: kompozíció**

Közösségi háló

Kiszolgáló infrastruktúra

Webszerver

DB szerver

Statikus tartalomkiszolgáló

Szerveroldali szoftver

Adatbázis

Alkalmazás

Hirdetések

Felügyelet

Webes felület

Felületterv (UX)

Grafika

Dinamika (JS)

Mobil felület

Top-down és bottom up

■ Top-down

- ☺ Részrendszer tervezésekor a szerepe már ismert
- ☹ „Félidőben” még nincsenek működő részek
- ☹ Részek problémái, igényei későn derülnek ki

■ Bottom-up

- ☺ Alrendszer önmagában kipróbálható, tesztelhető
- ☺ Részleges készütségnél is összeépíthető valami
- ☹ Nem látszik előre a rész szerepe az egészben

■ (Nem csak strukturális modellezésben...)

■ Meet in the middle?

STRUKTURÁLIS MODELLEK

Strukturális modell

- Mi volt a közös az eddigiekben bizonyos „dolgok” valamilyen „kapcsolata”
- **dolgok:** szoftver csomagok, személyek, repülőterek, területek
- **kapcsolata:** függőség, ismeretség, repülőjárat, része
 - szimmetrikus / aszimmetrikus
- Matematikai formalizmus: **gráf**
 - **csomópontok, élek és tulajdonságok**

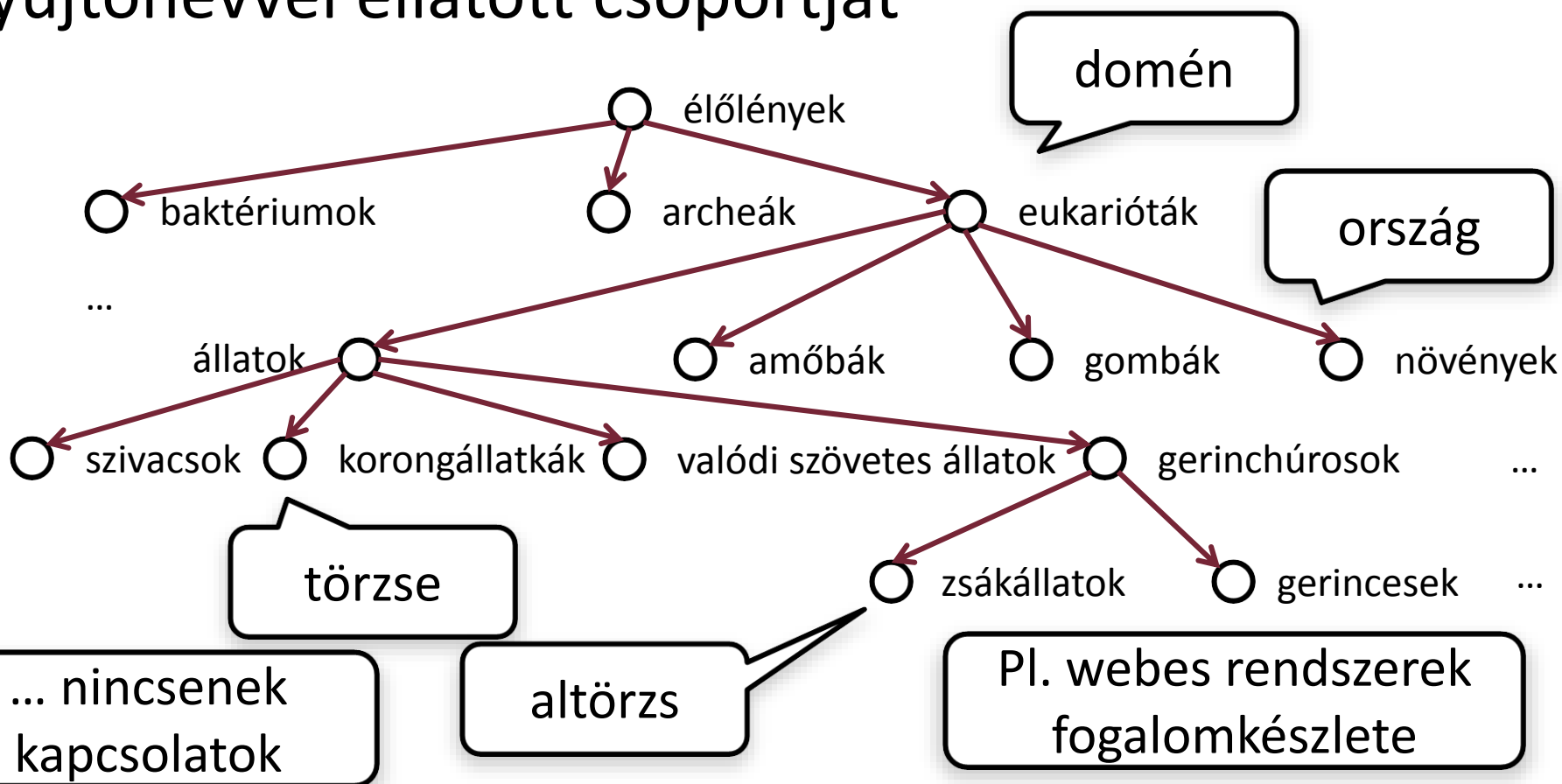
Definíciók

- Az informatikában mindennek több, gyakran ellentmondó definíciója van
 - rendszer = ?
 - modell = ?
- Gyakran ugyanarra a fogalomra több névvel is hivatkozunk
 - csomópont = csúcs
 - angolban: *node, vertex, object, concept*
 - él = kapcsolat
 - angolban: *edge, link, arc, connection, relationship*

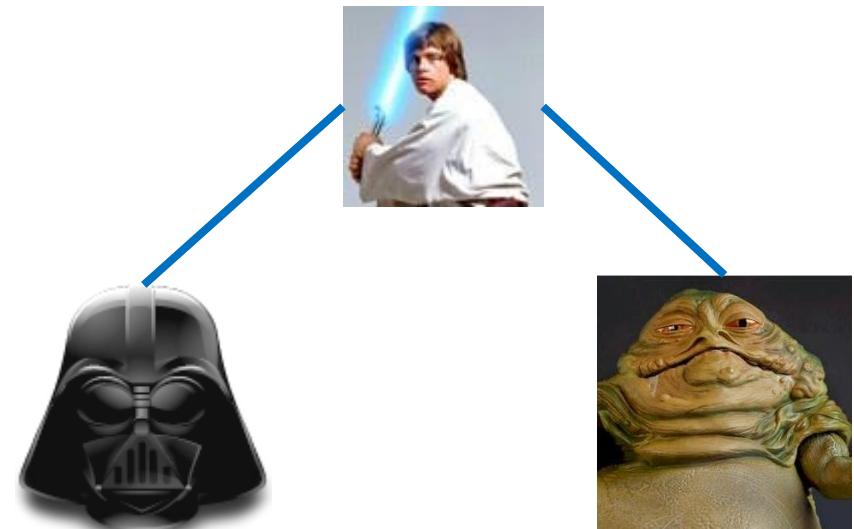
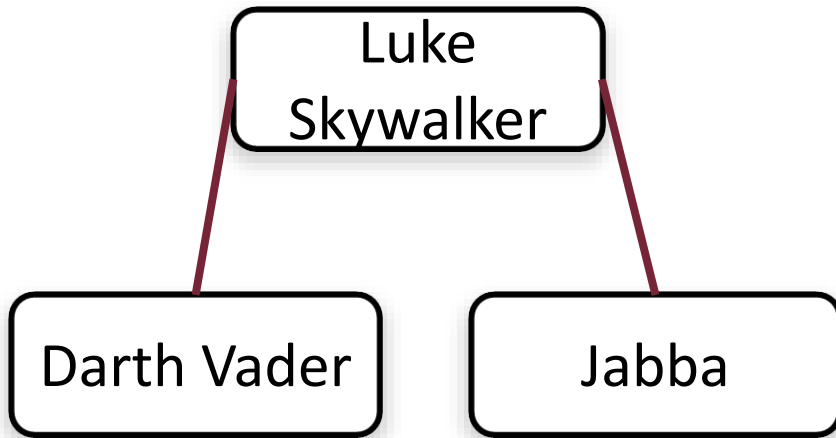
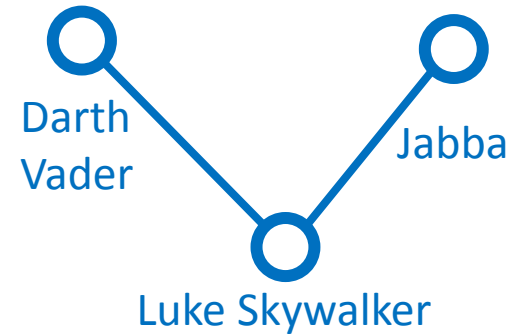
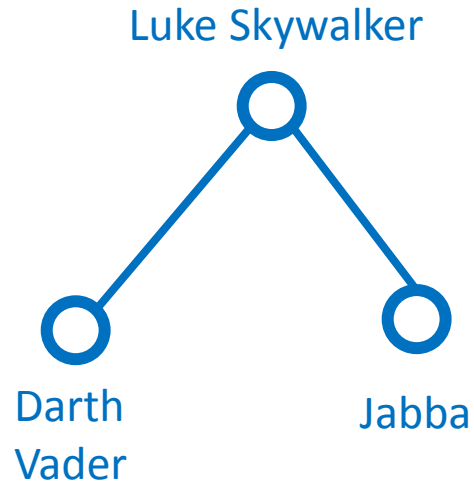
fontos a fogalmak
precíz definiálása

Taxonómia

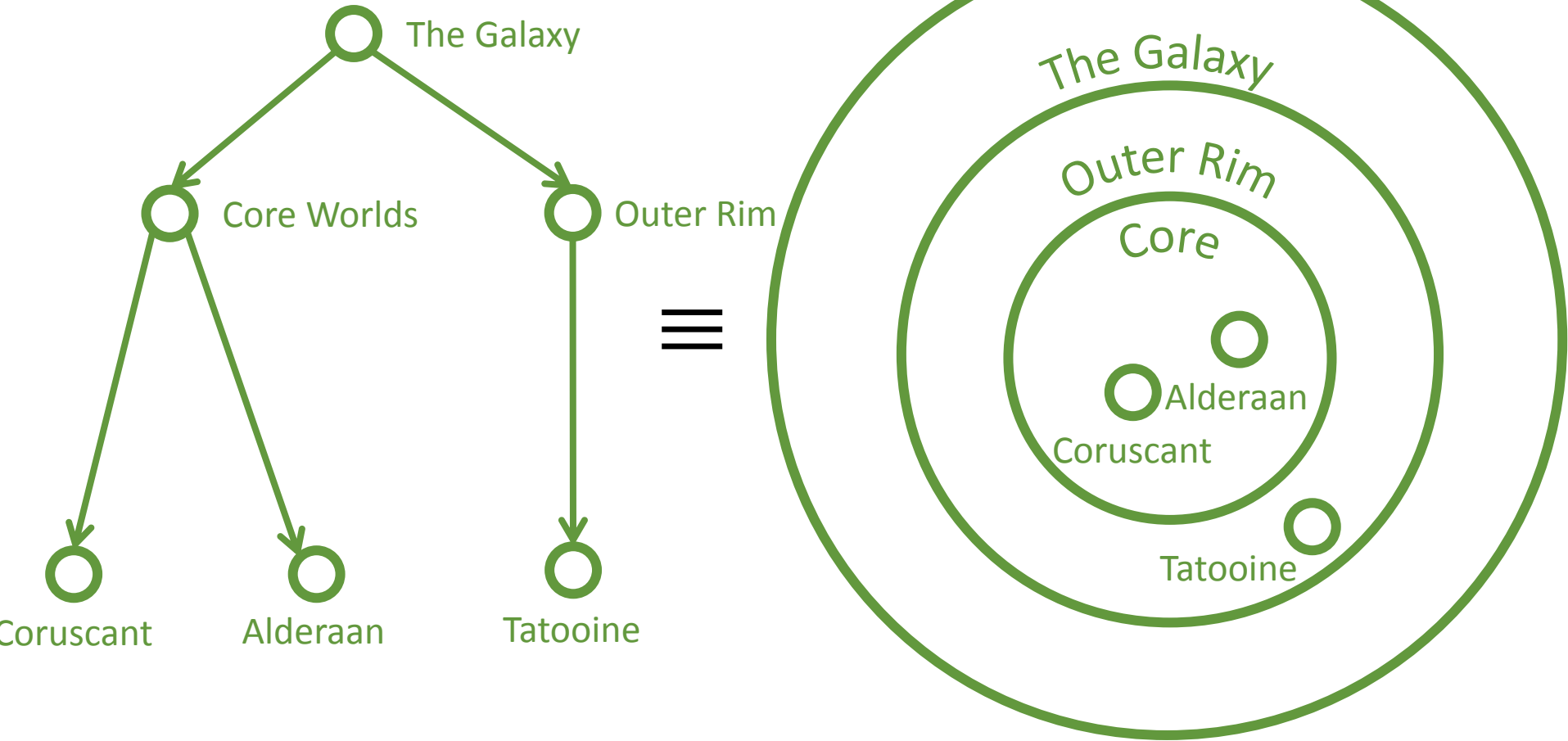
„A biológiai rendszertanban **taxonnak** nevezik az élőlények egyazon kategóriába sorolt és közös gyűjtőnévvel ellátott csoportját”



Modell (gráf) \neq diagram

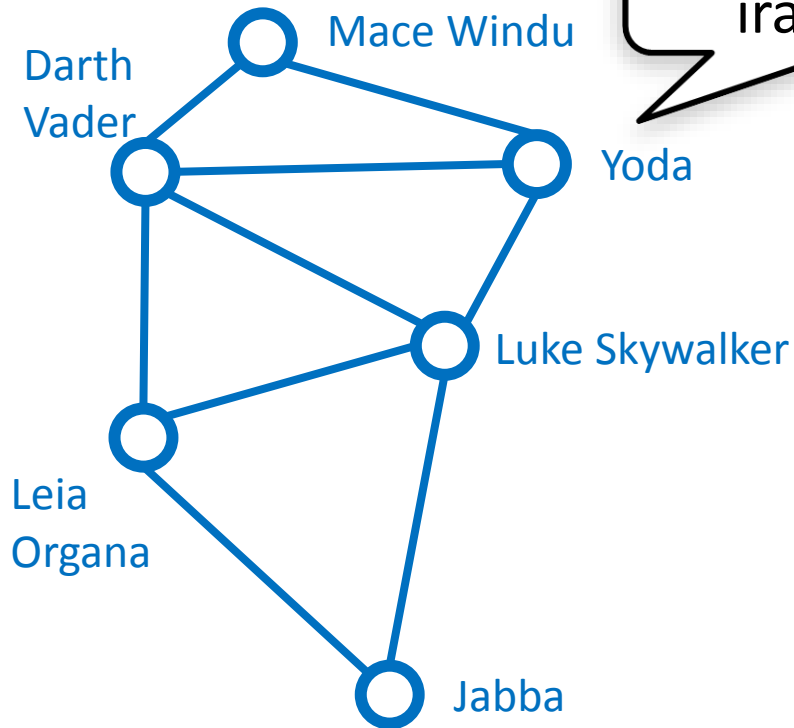


Hierarchia ábrázolása



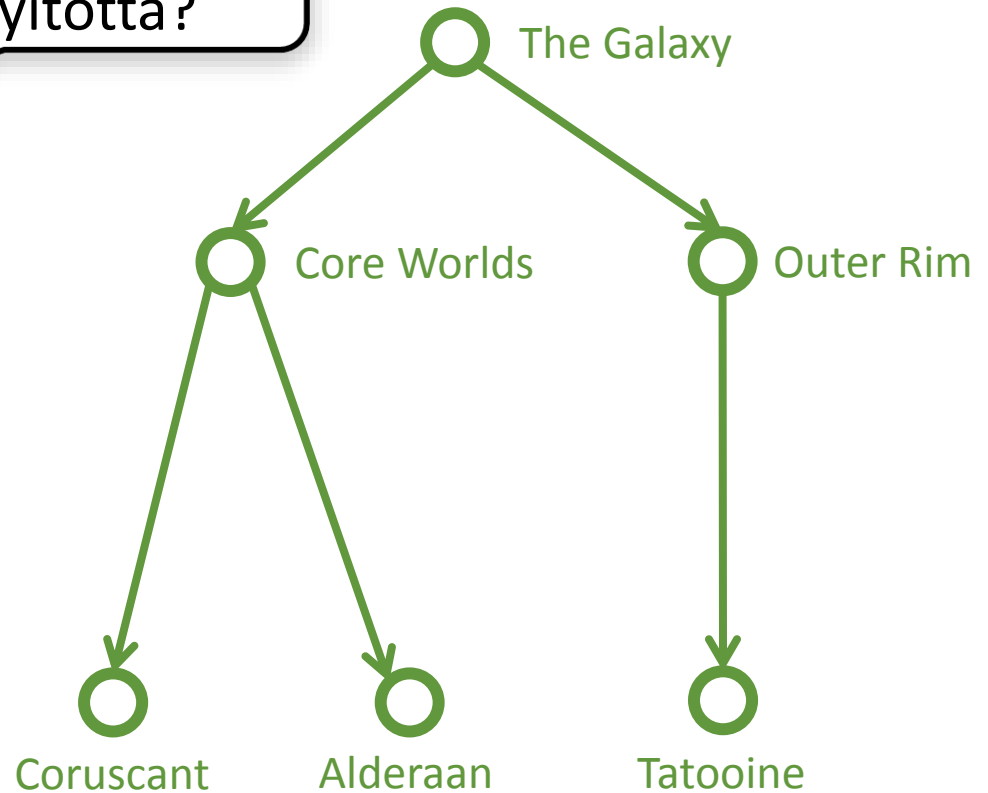
Egyszerű gráfok

kapcsolati háló:
irányítatlan gráf

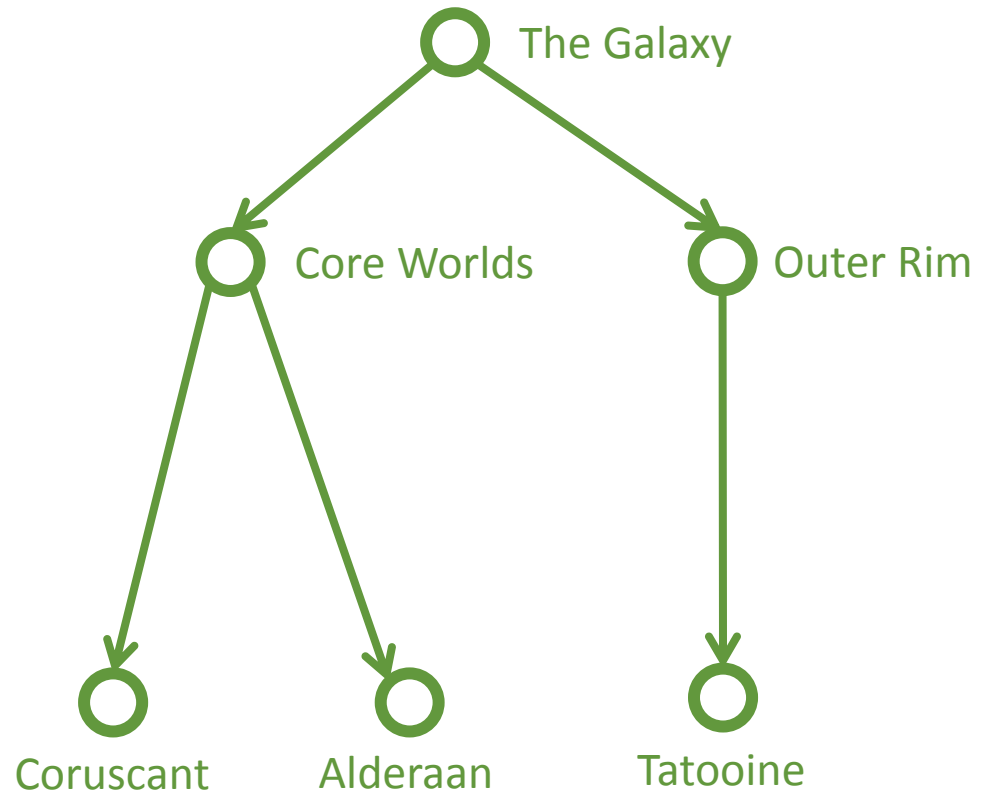
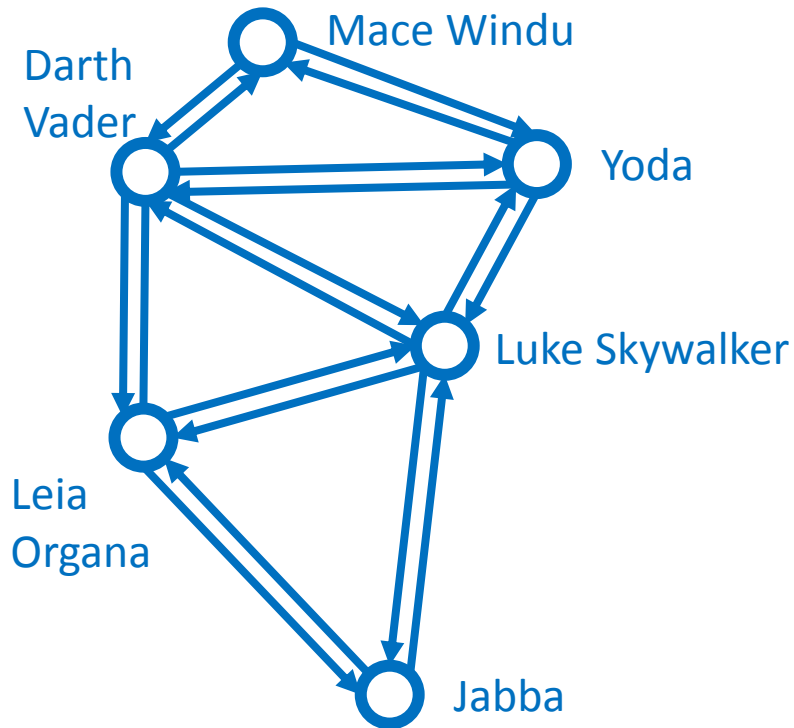


hogyan alakítható
irányítottá?

helyszínek:
irányított gráf

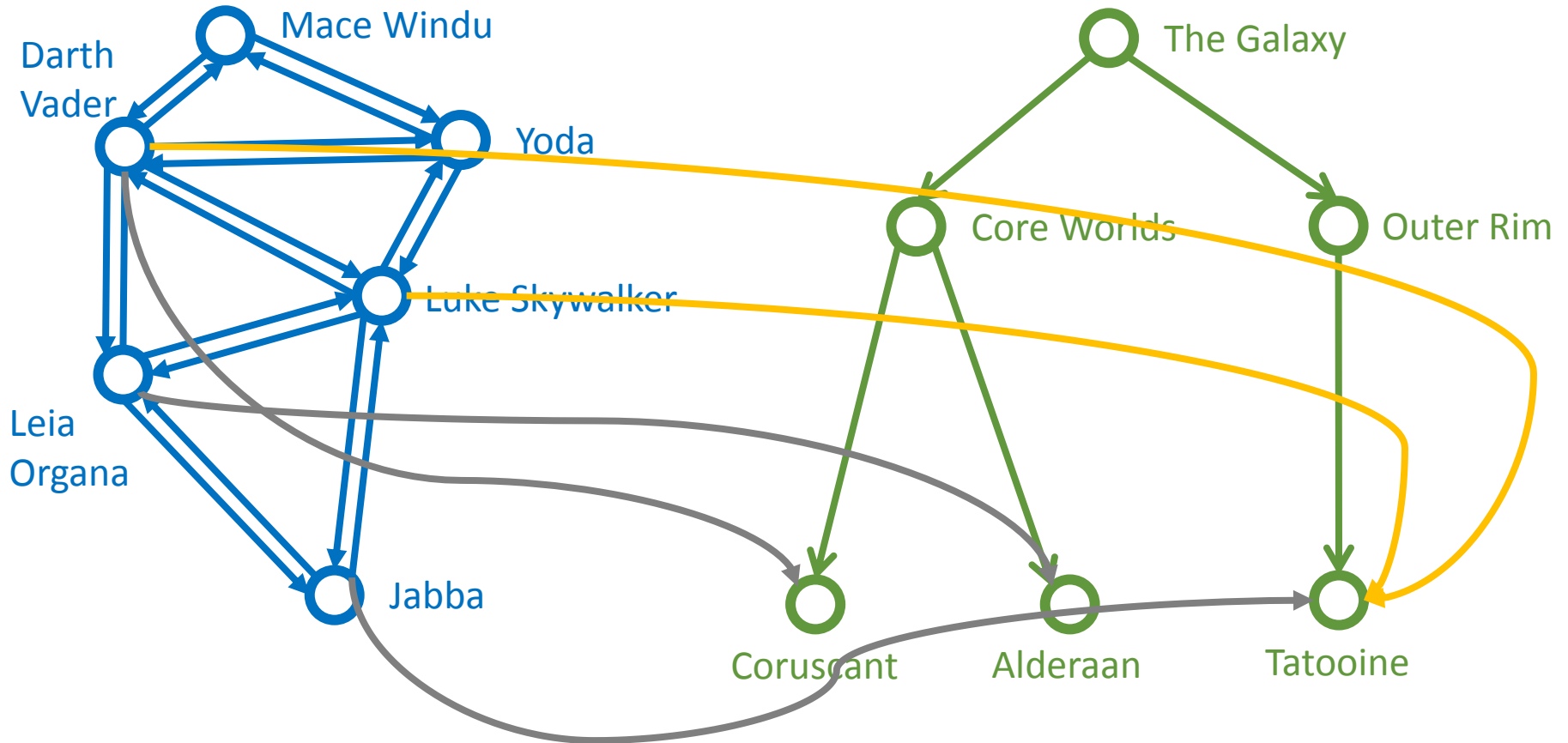


Egyszerű gráfok



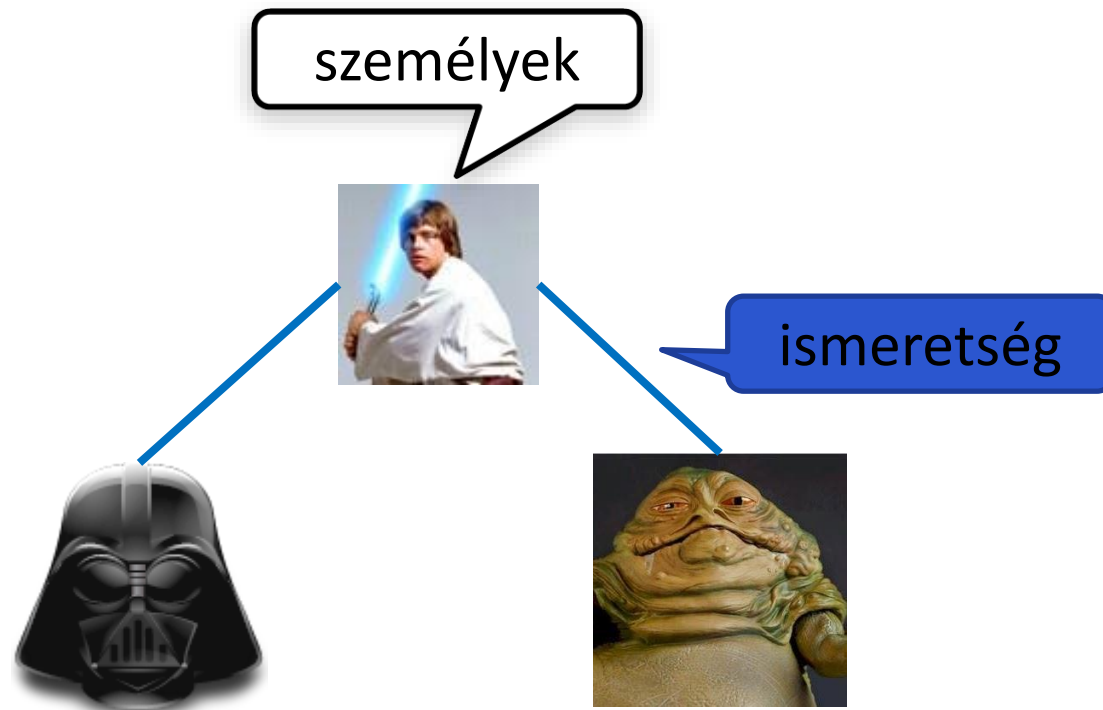
Típusos gráfok

Különböző típusú csomópontok és élek



GRÁF ALAPÚ TUDÁSREPREZENTÁCIÓ

Közösségi háló



Függőségek kezelése

GNU C compiler

függőség

Dependencies Related to gcc

● depends ◆ recommends ■ suggests • enhances

● **cpp** (>= 4:4.8.2-1ubuntu6)

GNU C preprocessor (cpp)

● **gcc-4.8** (>= 4.8.2-5~)

GNU C compiler

csomagok

◆ **libc6-dev**

Embedded GNU C Library: Development Libraries and Header Files
or **libc-dev**

virtual package provided by **libc6-dev**

■ **autoconf**

automatic configure script builder

■ **automake1.9**

A tool for generating GNU Standards-compliant Makefiles

<http://packages.ubuntu.com/trusty/gcc>

C program fordítása



app.c

include

```
1 #include "menu.h"
2 #include "io.h"
3
4 int main() {
5 // ...
```



menu.h

```
1 #include "util.h"
2
3 // ...
4
5
```

forráskódfájl



io.h

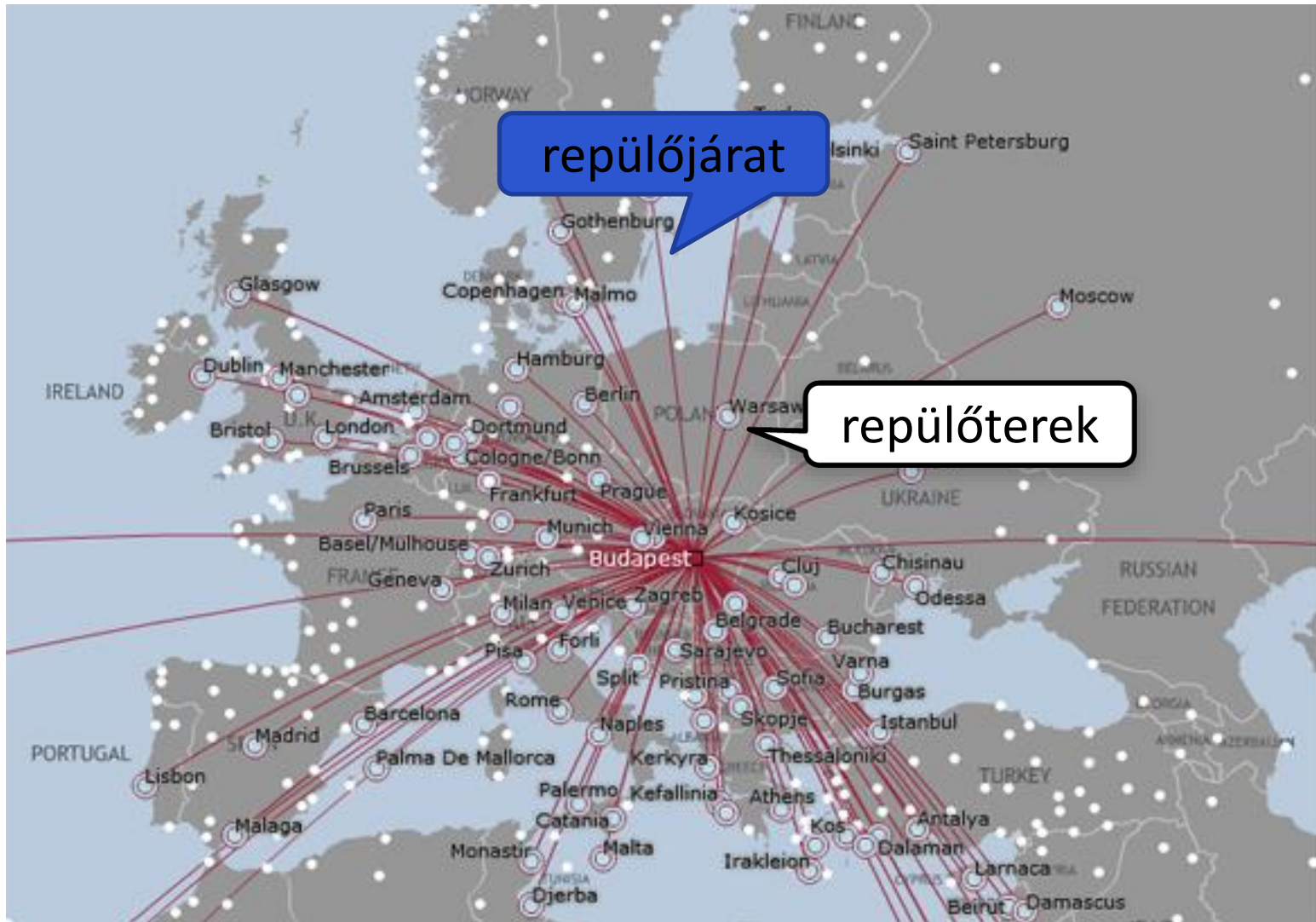
```
1 #include "util.h"
2
3 // ...
4
5
```



util.h

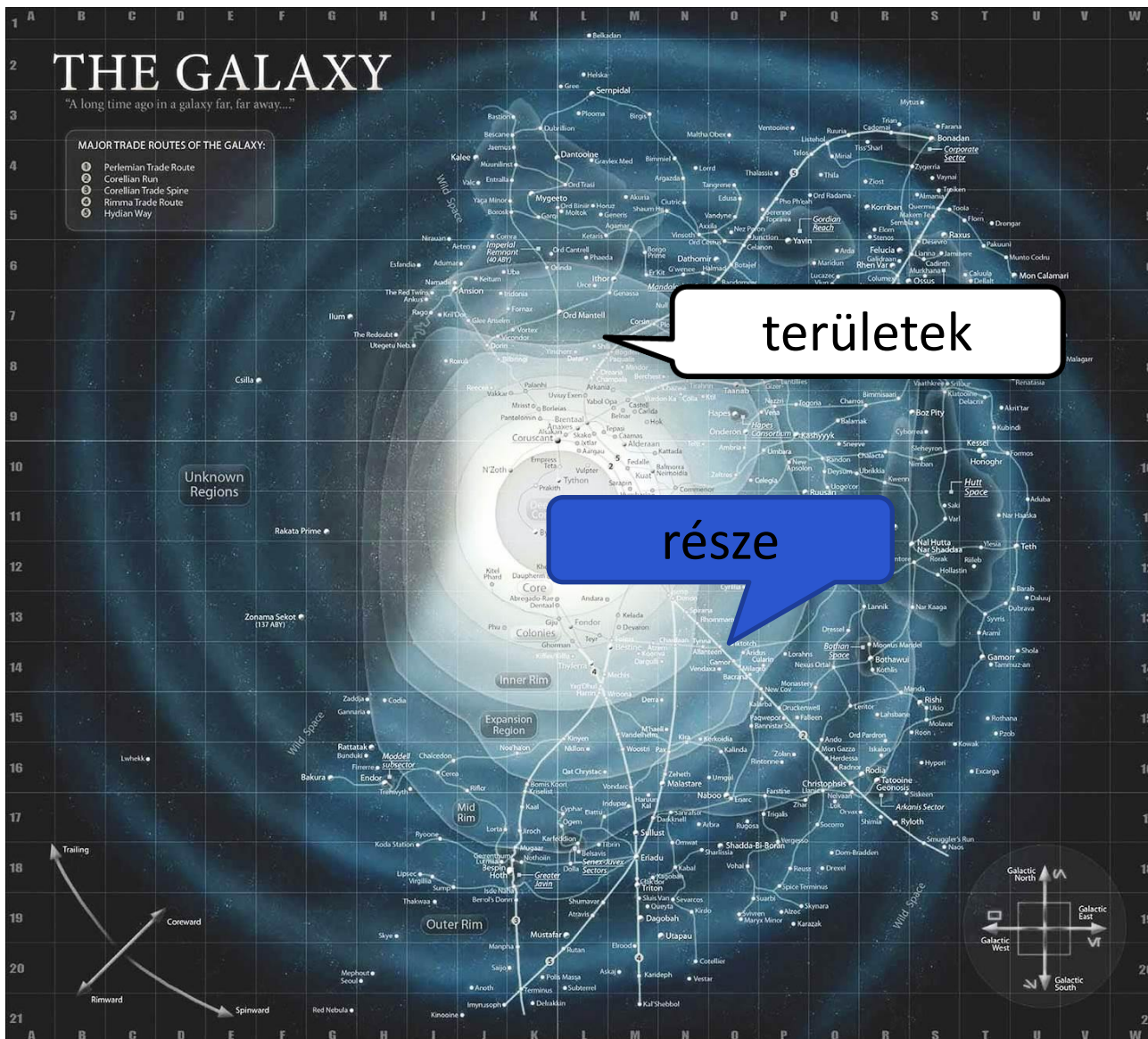
```
1 // ...
2
3
4
5
```

Repülési útvonalak



<http://budapestdentists.com/wp-content/uploads/2009/07/map.png>

A Galaxy

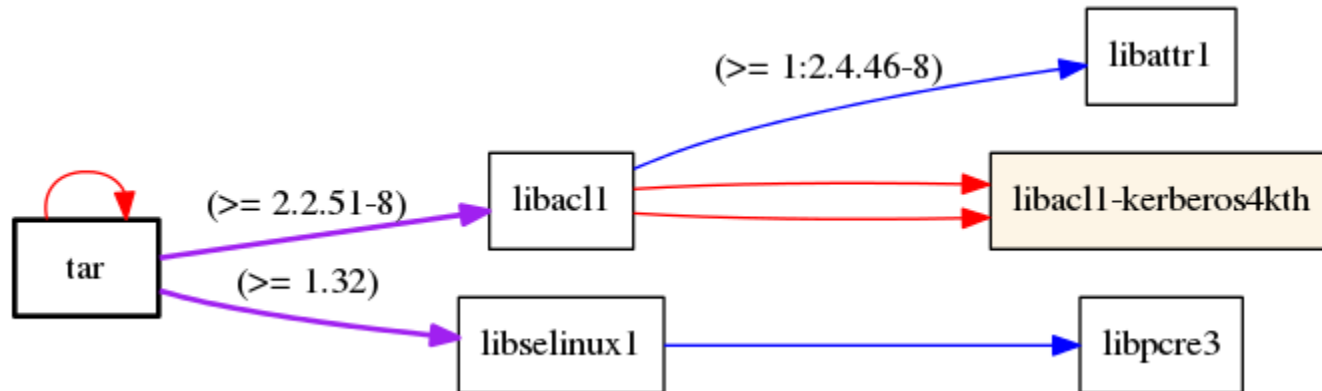


http://starwars.wikia.com/wiki/The_galaxy

Útvonal fogalma

- Szükséges-e a `libpcr3` csomag a `tar` telepítéséhez?
- Hogyan tudok eljutni a Deák Ferenc térre?

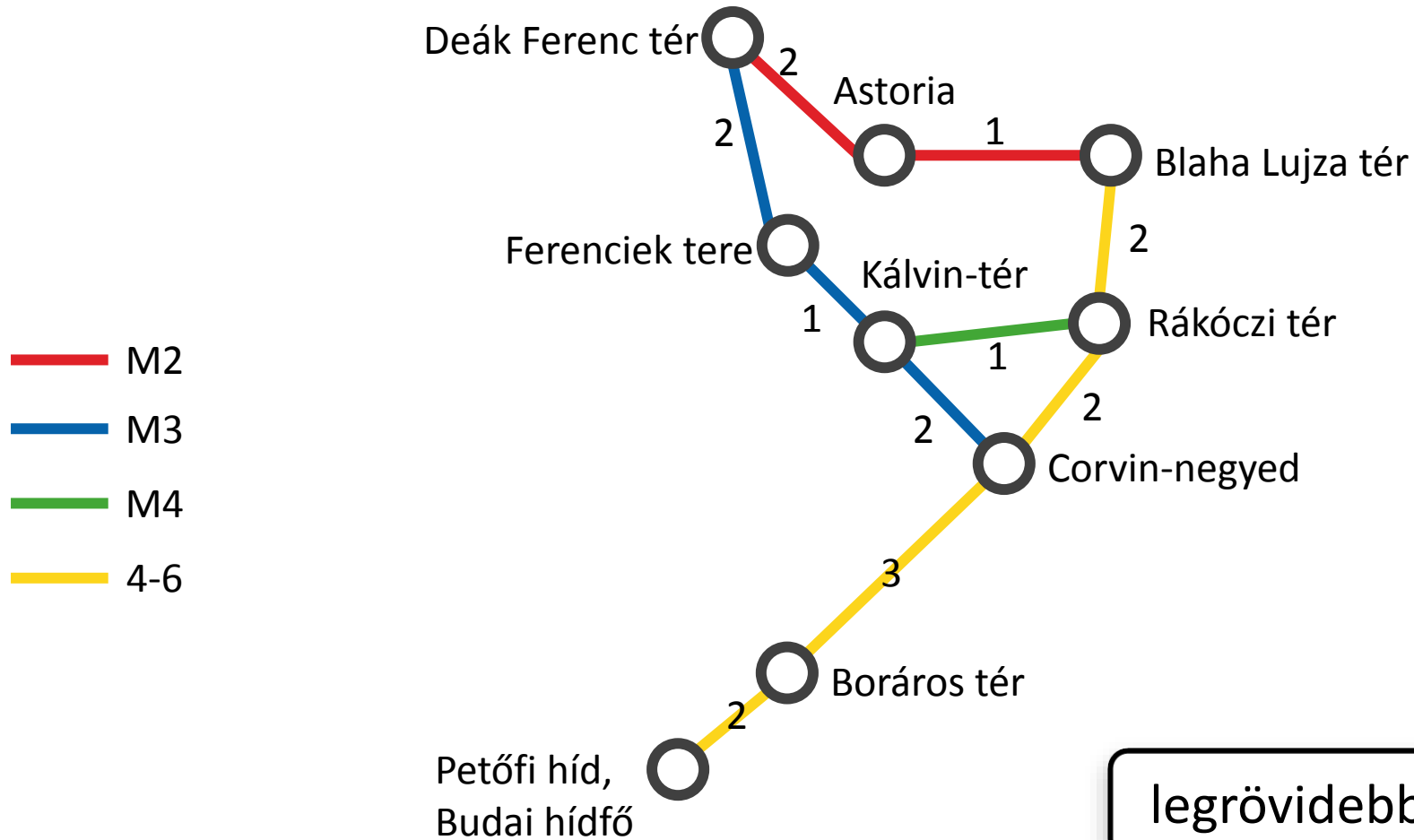
A tar csomag függőségei



Budapest – kötöttpályás közlekedés

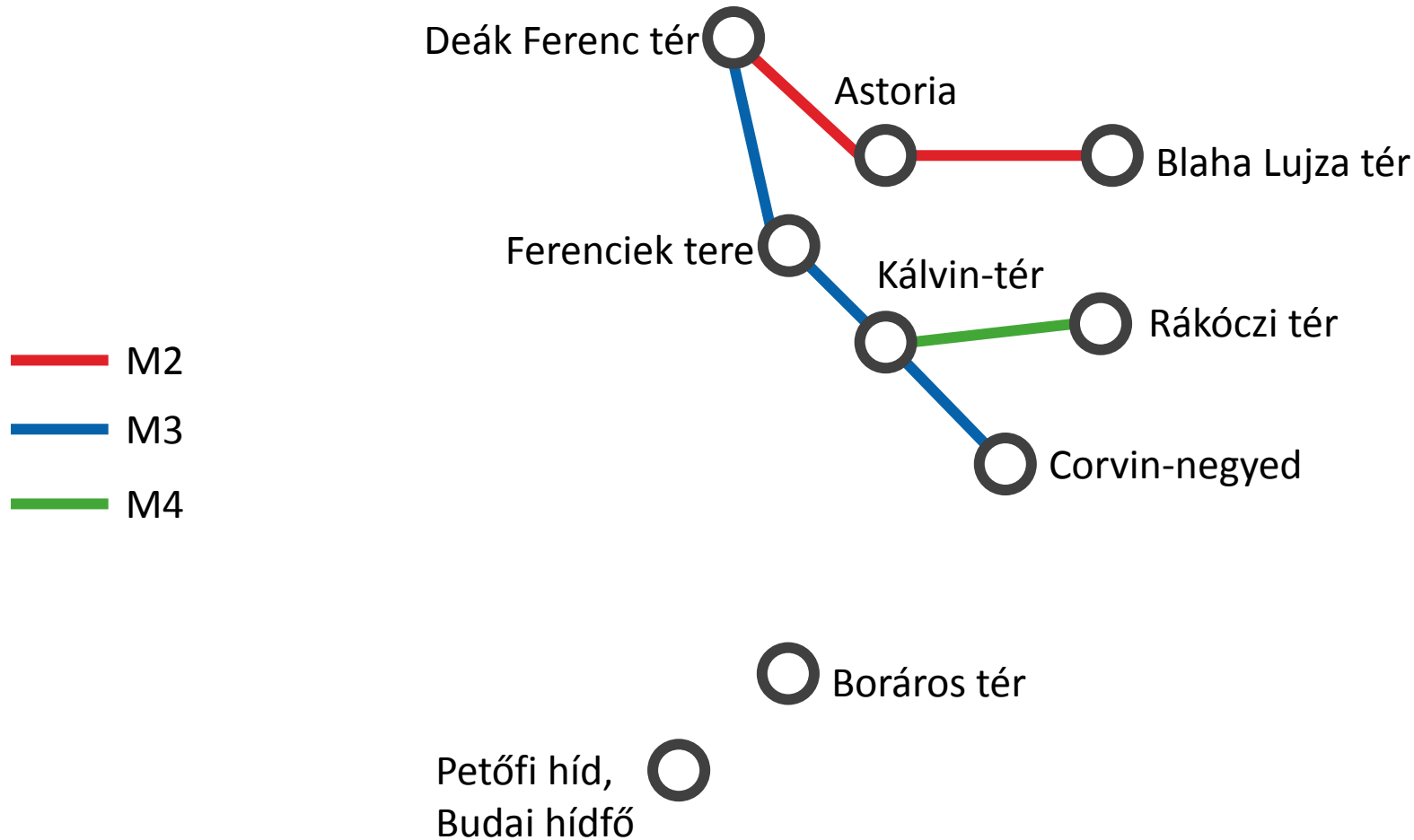


Budapest – kötöttpályás közlekedés



Szűrés: élcímke szerint (részgráf)

- Pl. csak a metróhálózatot szeretnénk nézni

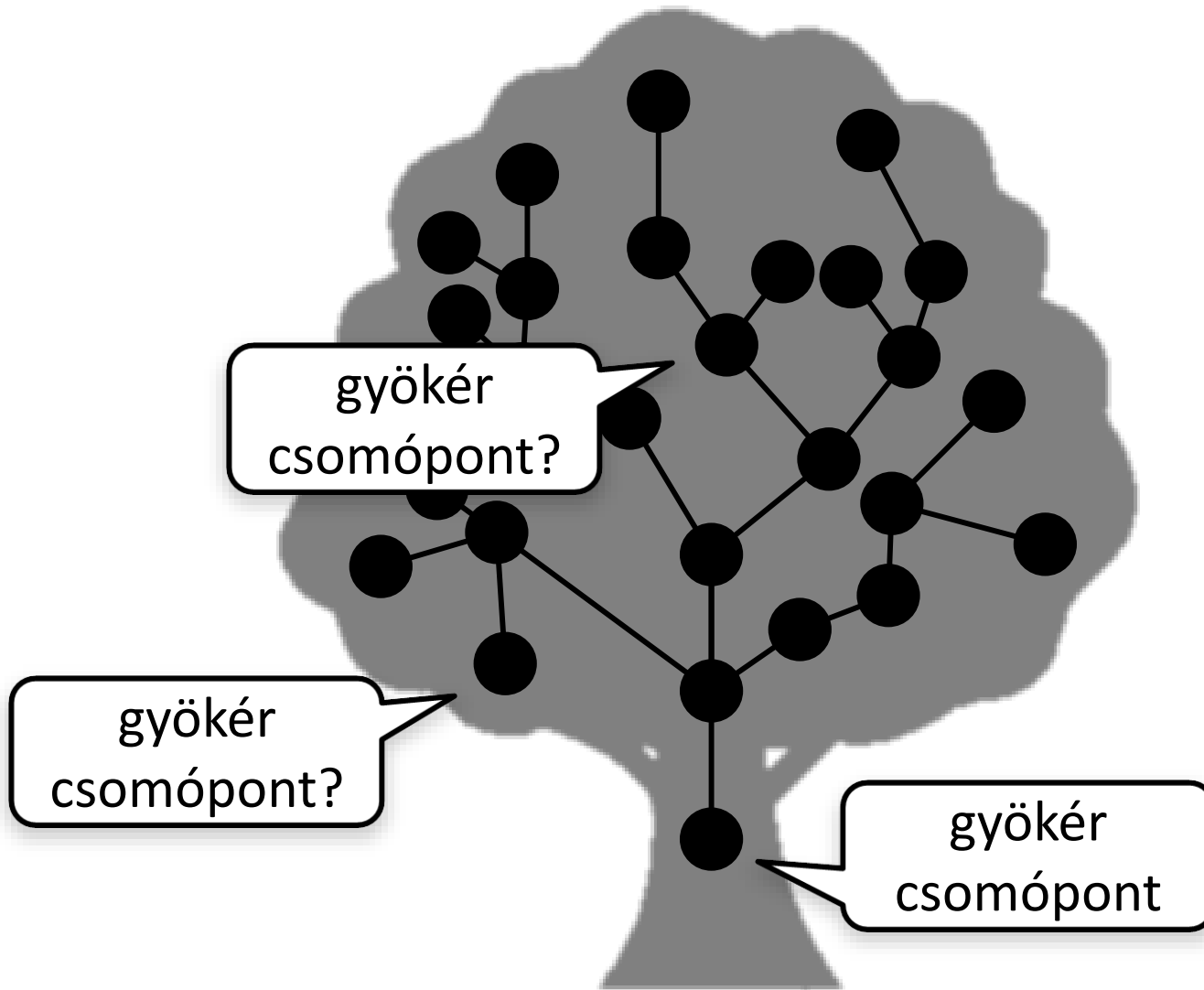


- Tranzitív lezárt: hova lehet eljutni metróval?

Hierarchia ábrázolása

- **Fa:** körmentes összefüggő gráf
- **Erdő:** körmentes gráf
- **Gyökér csomópont:** a fa egy megkülönböztetett csomópontja.
- **Gyökeres fa:** olyan fa, ami rendelkezik gyökér csomóponttal.
- **Gyökeres, szintezett fa:** a fa csomópontjaihoz hozzárendeljük a gyökértől vett távolságukat

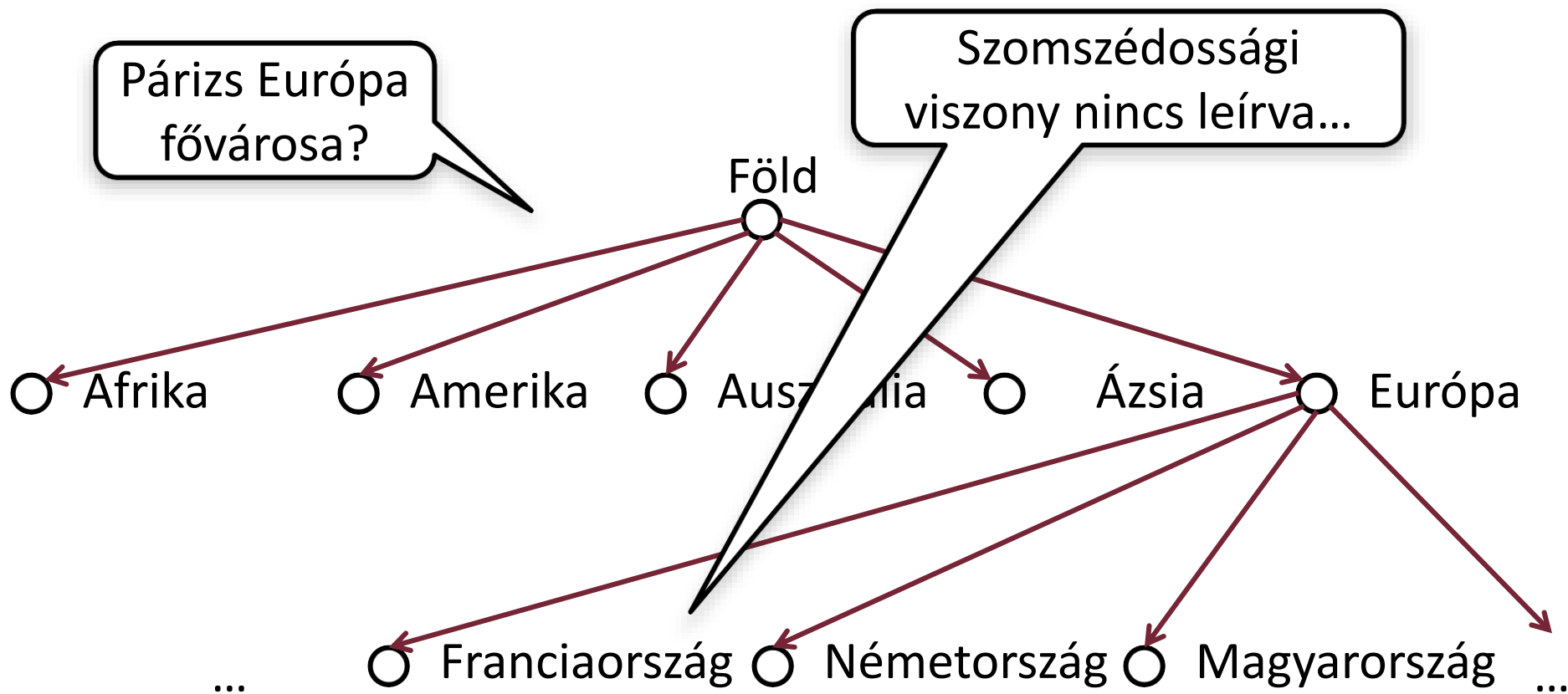
Fa



Fastruktúra tulajdonságai

- A tartalmazási részfák diszjunktak vagy alárendeltek
- Ábrázolható
 - Gráfként
 - faélek explicit módon
 - faélek implicit módon
 - Bennfoglaló ábrázolás

Fastruktúra ábrázolása – világtérkép

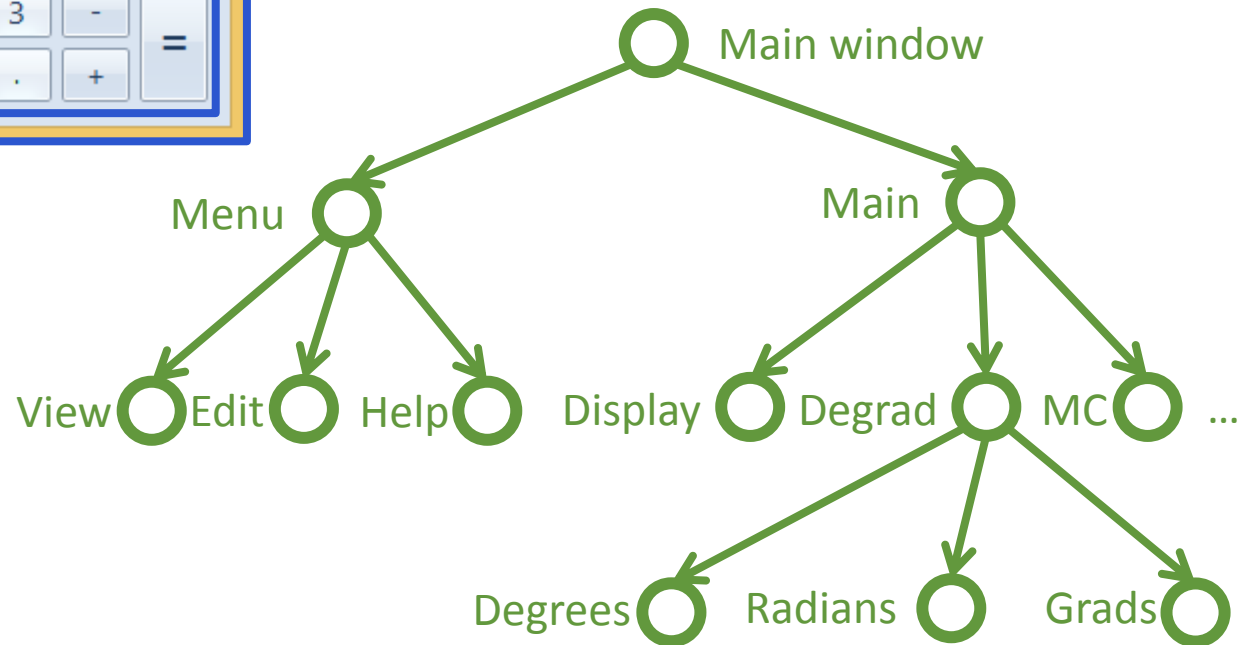
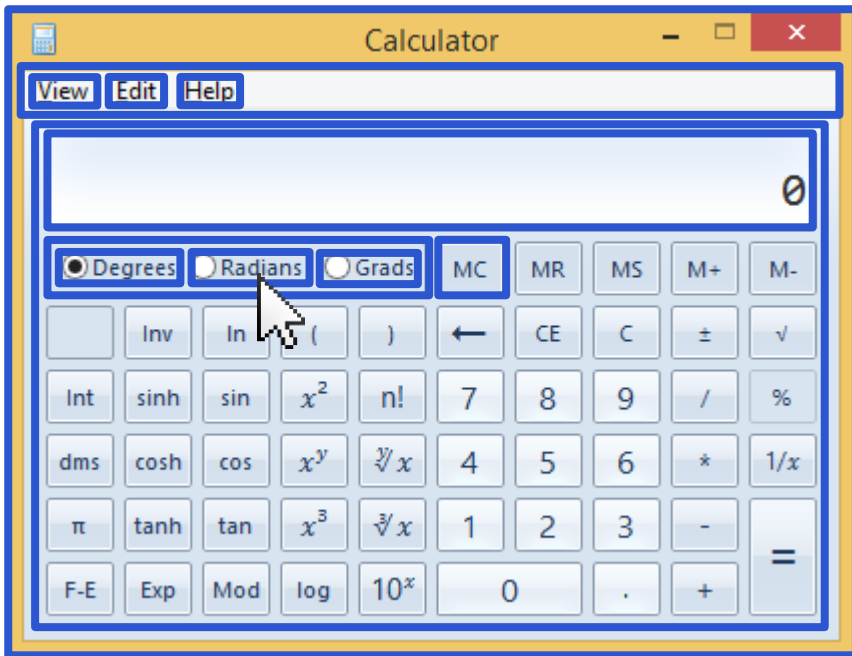


Fastruktúra – bennfoglaló ábrázolás



<http://www.worldatlasbook.com/europe/europe-political-map.html>

Grafikus felhasználó felület



Fastruktúra ábrázolása – fájlrendszer

C:

\Program Files

 \Common Files

\Windows

 \Fonts

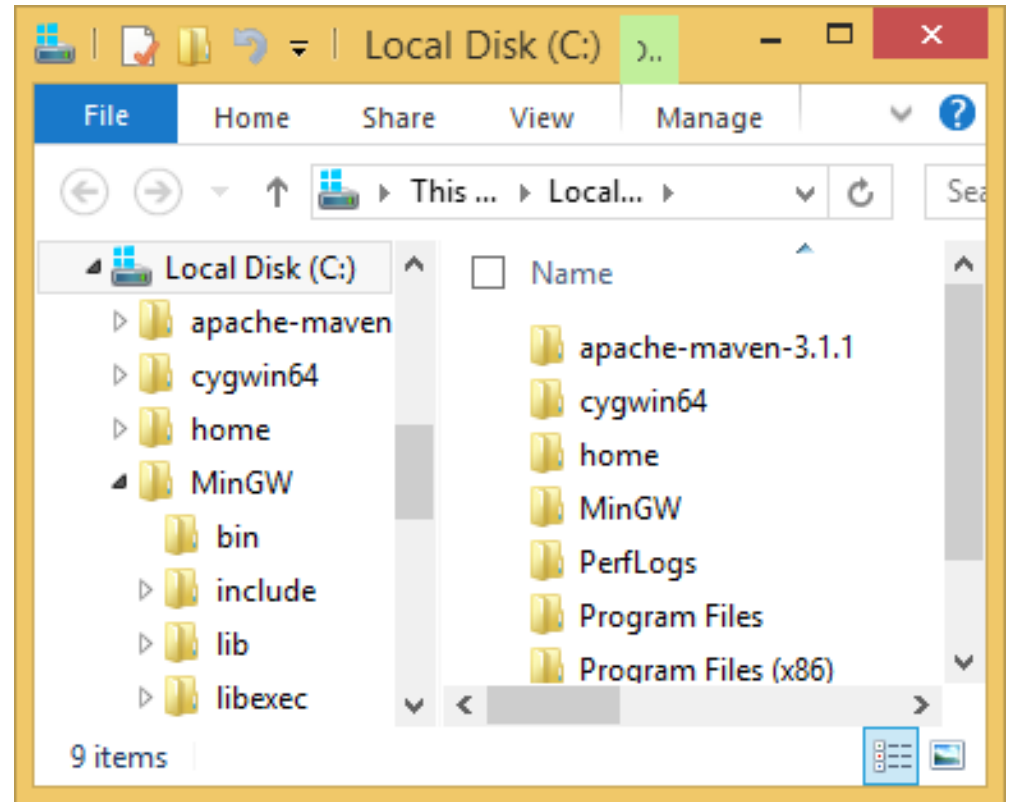
 \system32

 \drivers

 \etc

\Users

...



Fastruktúra – bennfoglaló ábrázolás

- KDE Konqueror

The image shows a KDE Konqueror window displaying a file browser view of the directory `/home/cjr/`. The interface is divided into several panes: a main file list pane, a sidebar with various system locations, and a right-hand pane showing details for the selected item. A white callout box with a black border is overlaid on the main file list, containing the text "Plusz információ: méret".

Callout Box Text: Plusz információ: méret

Main File List (Partial View):

File Name	Size
Pirates of the Caribbean- Dead Man's Chest_BBC_20_28_04.mpg	4.72 GB
Trust the Man_BBC ONE_23_02_2009_22_33_04.mpg	3.43 GB
Mary Poppins_ITV1_26_12_2008_15_23_04.mpg	3.39 GB
Gangs of New York_More 4_19_12_2008_21_58_04.mpg	3.27 GB
Stuart Little_BBC_+1_1_09_20_05.mpg	53 GB
Babe_Pig in the City_ITV2_01_01_2009_1..._21_23...	768 MB
Toy Story_2_...	...
Stuart_Litt...	...
Childrens/	1.63 GB
Aardman Animations/	1.54 GB
Downloads/	15.36 GB
VirtualBoxOld/	12.12 GB

Right-hand Pane (Selected Item Details):

Item Name	Size
{b07343d0-fdc6-403e-9124-c2492f4a3578}.vdi	6.86 GB
{e70bd910-f6ca-4327-8397-d359f12de1b3}.vdi	3.72 GB
{b69108b4-b857-4949-6d8a-2a002b03b4e5}.vdi	3.38 GB
{9f7a62e0-375e-4388-8f0c-c45d7}.vdi	2.87 GB
{2d8b5e42-f2e2-4d3b-9b99-4c4d9ab73042}.vdi	2.88 GB
{dd09f1cd-85f0-4175-ecb2-968b...}.vdi	1.44 GB
{ce5b7cal3dca4e4...}.vdi	1.44 GB

TULAJDONSÁGOK MODELLEZÉSE

Táblázatos ábrázolás

- **Sor** = modellelem
- **Oszlop** = jellemző (gráf elemeinek tulajdonsága)

név	fénykard színe	nem	holdak száma	keringési idő
Alderaan			1	364
Coruscant			4	368
Darth Vader	piros	férfi		
Jabba		hímnős		
Leia Organa		nő		
Luke Skywalker	zöld	férfi		
Mace Windu	lila	férfi		
Tatooine			3	304
Yoda	zöld	férfi		

parciális függvény:
NULL / NA
attribútumok

Alapműveletek

■ Szűrt nézet

- Csak azon sorok, amelyek egy bizonyos oszlopban bizonyos értéket vesznek fel
- fénykard színe = "zöld"

név	fénykard színe	nem	holdak száma	keringési idő
Luke Skywalker	zöld	férfi		
Yoda	zöld	férfi		

- holdak száma > 2

név	fénykard színe	nem	holdak száma	keringési idő
Coruscant			4	368
Tatooine			3	304

Alapműveletek

■ Vetített nézet

- Csak bizonyos oszlopok megtartása
- {holdak száma, keringési idő} ○ {név, fénykard színe}

Biztos, hogy csak ide tartozik?

holdak száma	keringési idő	név	fénykard színe
1	364	Alderaan	
4	368	Coruscant	
3	304	Darth Vader	piros
		Jabba	
		Leia Organa	
		Luke Skywalker	zöld
		Mace Windu	lila
		Tatooine	
		Yoda	zöld

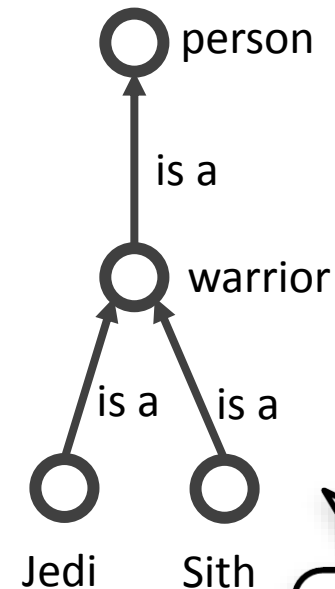
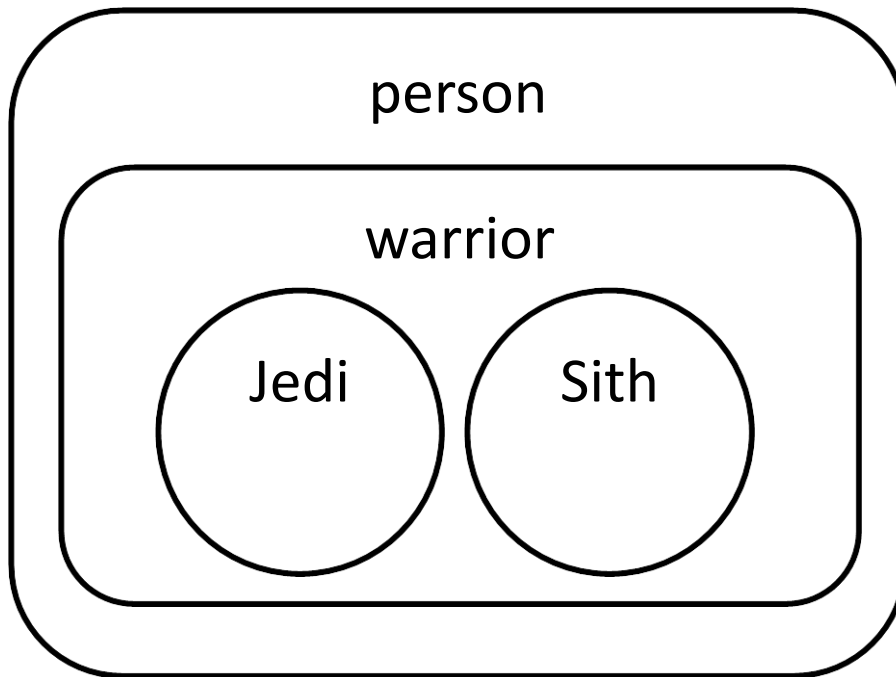
TÍPUSOK

Típus és tulajdonságok

- A típus a címkében kifejezett tudás része
 - Tehát a *típus* egy kitüntetett attribútum
 - A többi jellemző: a *tulajdonságok*
- Gyakori konvenció:
 - tulajdonságok akár változhatnak is
 - típus egy elemre tipikusan állandó
- Módosítás: Luke fénykardja: kék → zöld
 - Típus nem változik

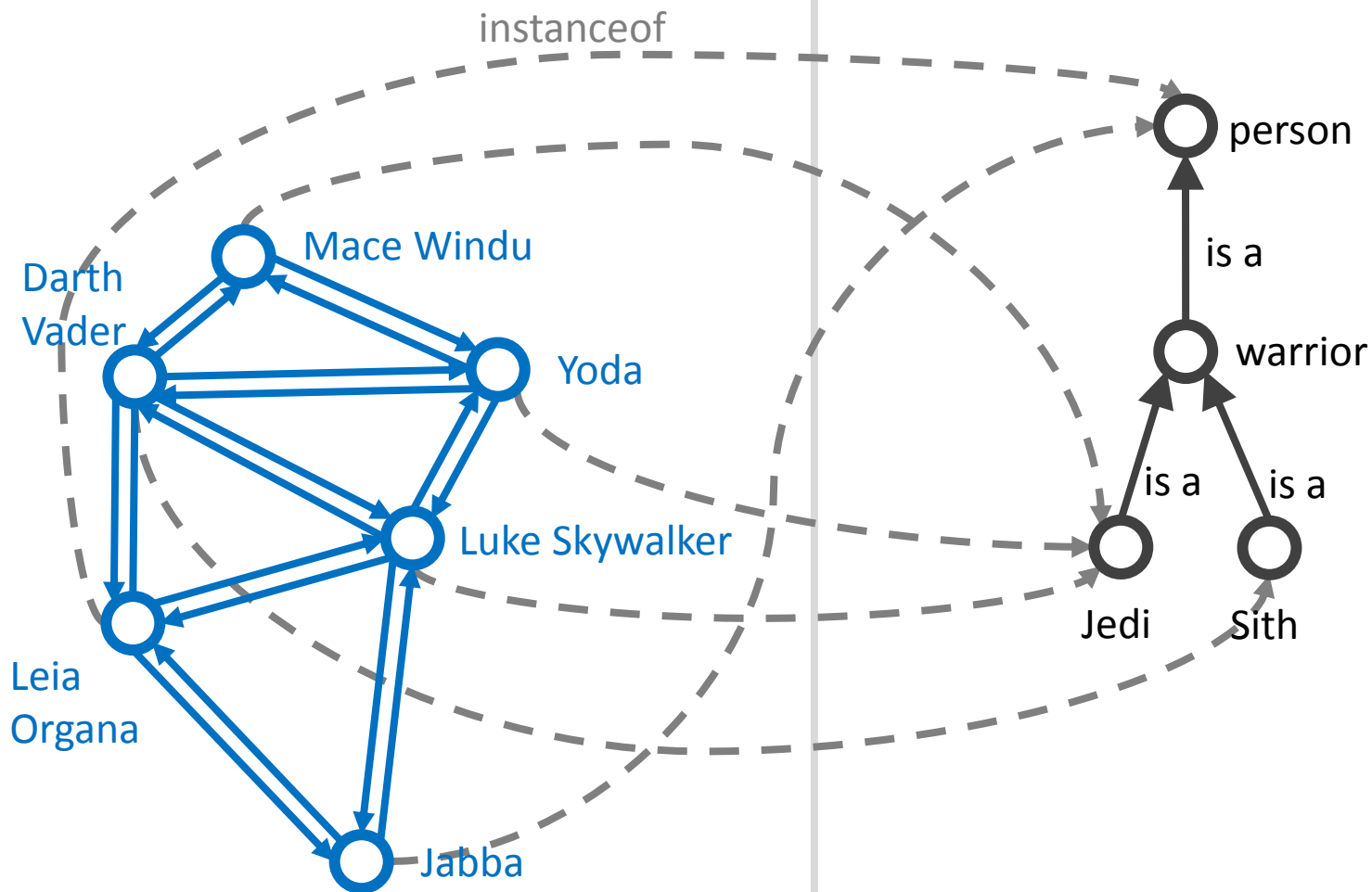
Típusgráf

- Minden csomóponttípushoz egy típuscsomópont
- Minden éltípushoz egy típusél

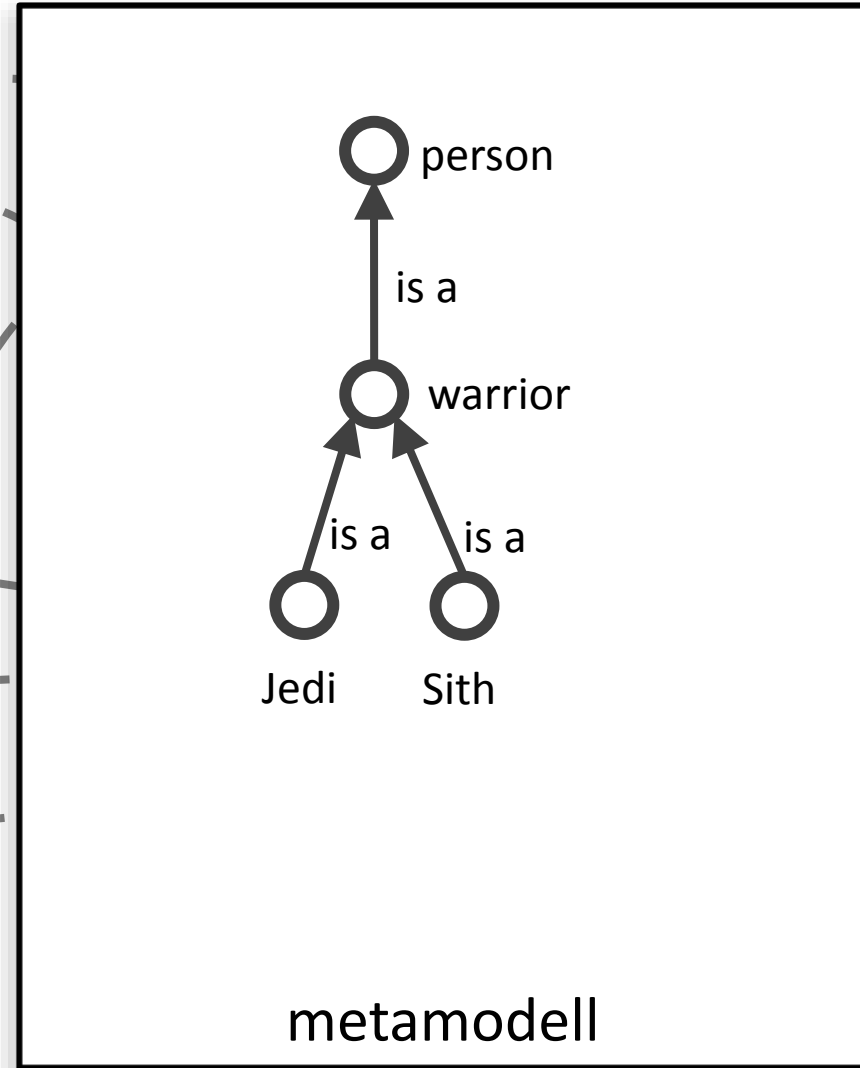
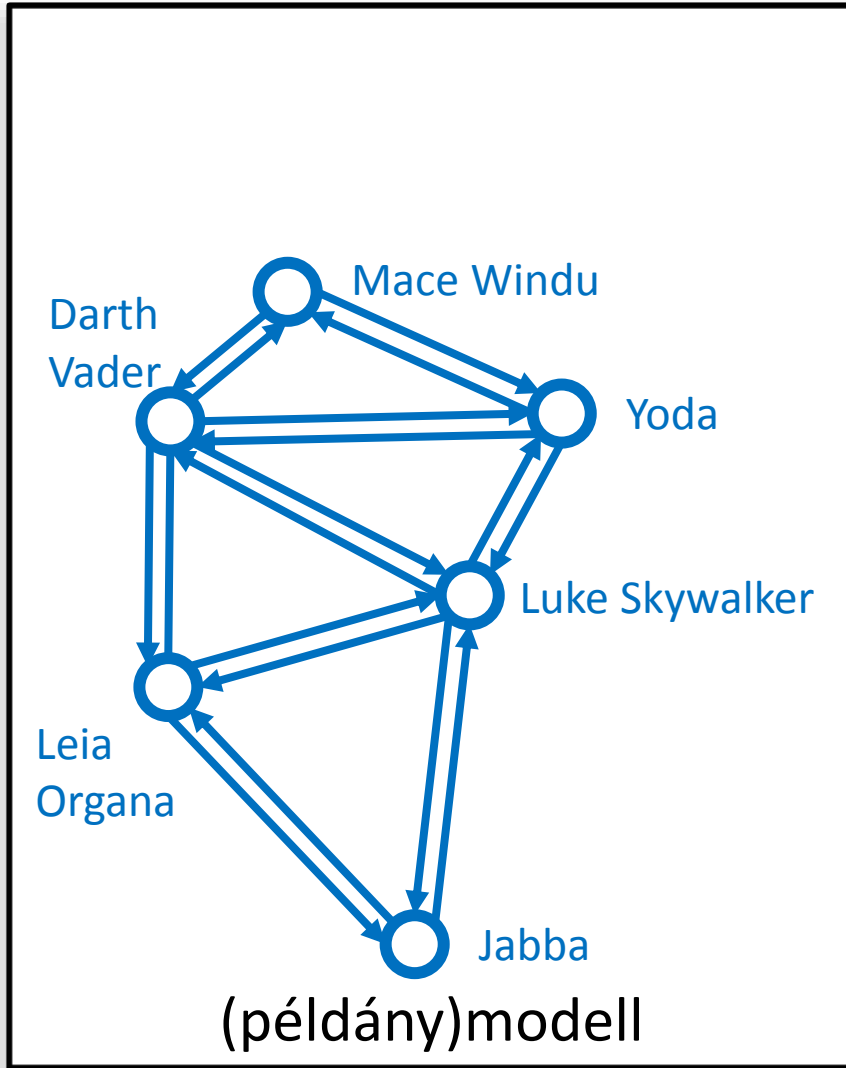


hierarchikus
dekompozíció

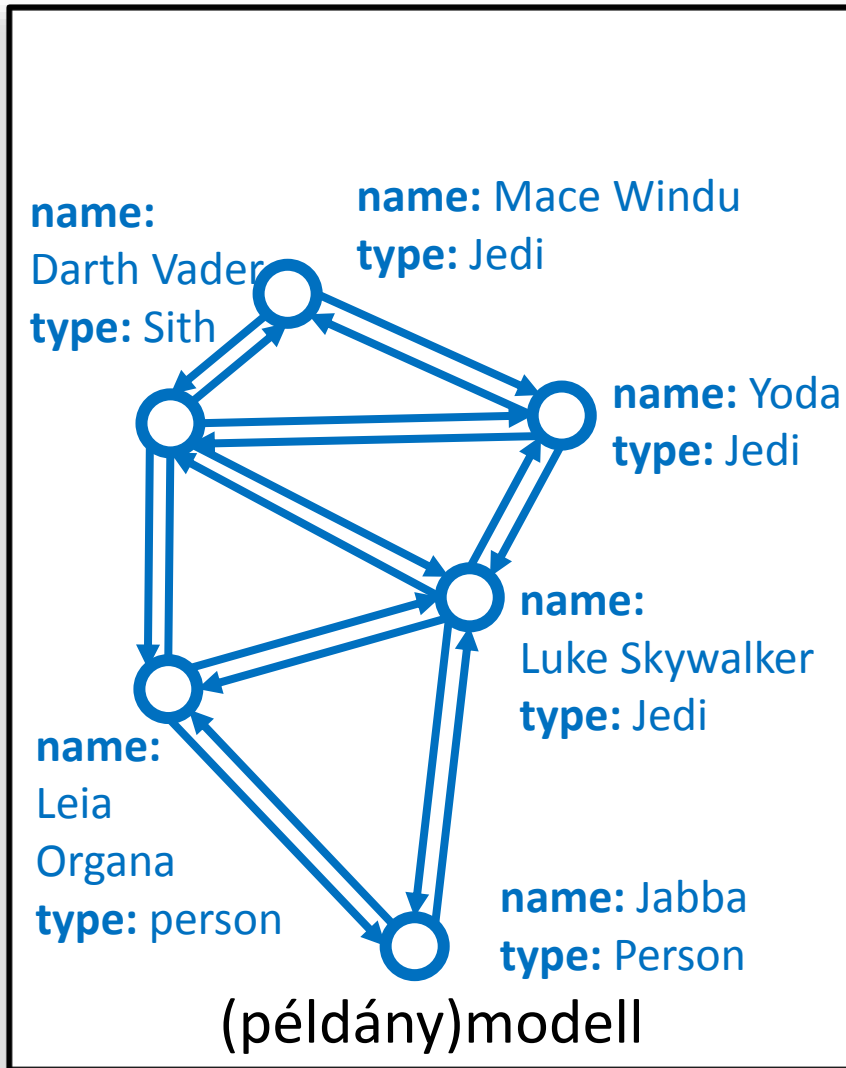
Típus-példány viszonyok ábrázolása



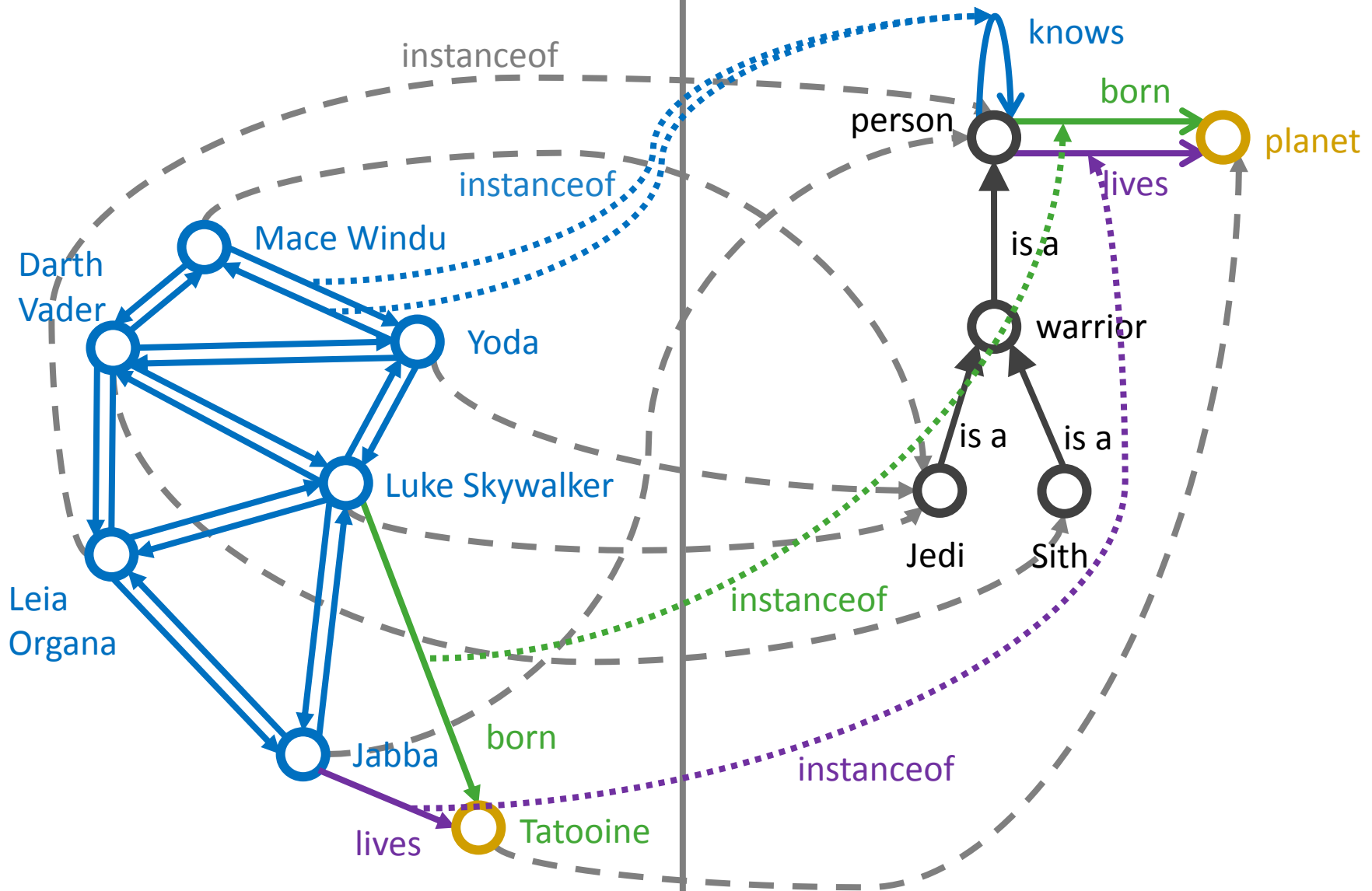
Típus-példány viszonyok ábrázolása



Típusok ábrázolása jellemzőként



Különböző éltípusok



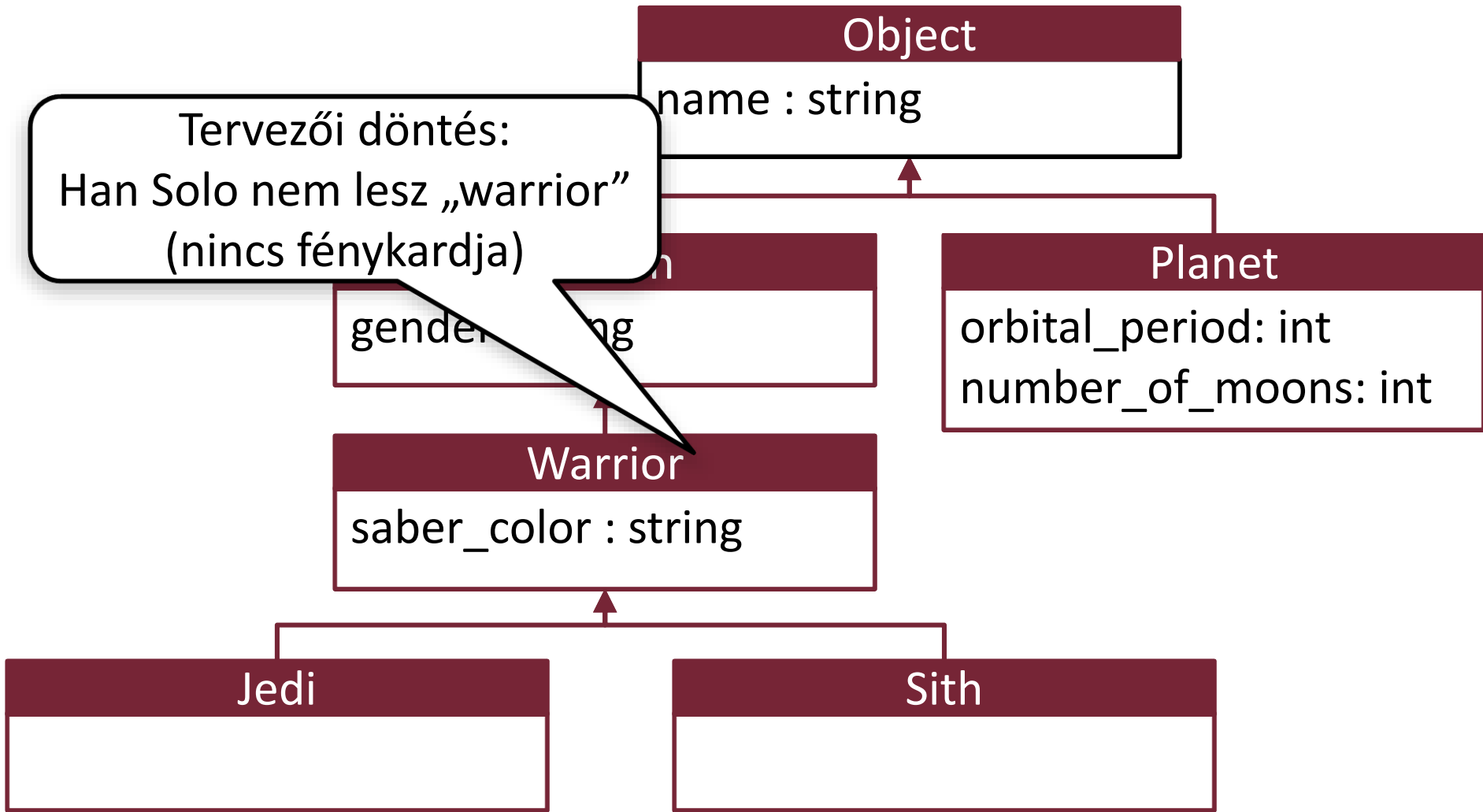
STRUKTURÁLIS MODELL REPREZENTÁLÁSA PROGRAMBAN

Programozási paradigmák

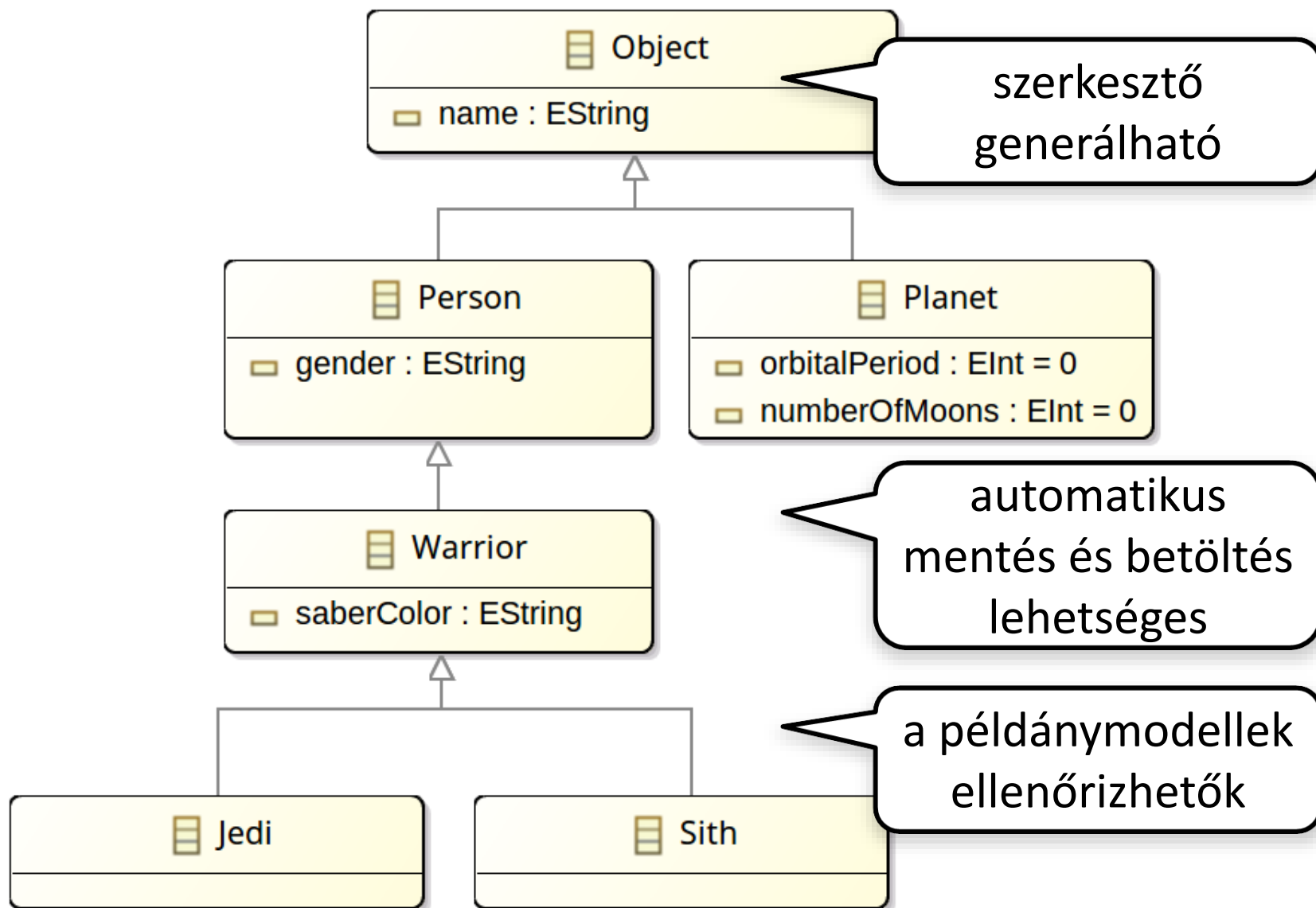
- **Programozási paradigma:** egy programozási nyelv elméleti modellje
- **Strukturált programozás (C)**
 - változók struktúrába szervezése: struct
- **Objektum-orientált programozás, OOP (C++, Java)**
 - típus: *osztály*
 - példány: *objektum*
 - attribútum: *mező*
 - műveletek: *metódusok*
 - attribútumok/metódusok láthatósága, egységbe zárás stb. – bővebben ld. a Programozás alapjai 2. tárgyban

OOP: öröklés

Tervezői döntés:
Han Solo nem lesz „warrior”
(nincs fénykardja)



Eclipse Modeling Framework – Ecore diagram



Szöveges reprezentációk

- XML (Extensible Markup Language)
 - szabványos, általános célú nyelv leírónyelvek definiálására
 - ember számára (elvileg...) olvasható
- JSON (JavaScript Object Notation)
 - szöveg alapú szabvány ember által olvasható adatszerére

XML példa: időjárás webszolgtatás

```
▼<current>
  ▼<city id="3054643" name="Budapest">
    <coord lon="19.04" lat="47.5"/>
    <country>HU</country>
    <sun rise="2015-02-17T05:45:24" set="2015-02-17T16:10:12"/>
  </city>
  <temperature value="268.061" min="268.061" max="268.061" unit="kelvin"/>
  <humidity value="83" unit="%"/>
  <pressure value="1034.42" unit="hPa"/>
  ▼<wind>
    <speed value="2.12" name="Light breeze"/>
    <direction value="52.0001" code="NE" name="NorthEast"/>
  </wind>
  <clouds value="0" name="clear sky"/>
  <visibility/>
  <precipitation mode="no"/>
  <weather number="800" value="Sky is Clear" icon="01n"/>
  <lastupdate value="2015-02-17T20:11:20"/>
</current>
```

JSON példa: Google Maps API

```
{
  "results": [
    {
      "address_components": [
      ],
      "formatted_address": "1600 Amphitheatre Pkwy, Mountain View, CA 94043, USA",
      "geometry": {
        "location": {
          "lat": 37.42291810,
          "lng": -122.08542120
        },
        "location_type": "ROOFTOP",
        "viewport": [
        ]
      },
      "types": [
      ]
    }
  ],
  "status": "OK"
}
```

Modellezési nyelvek

- UML (Unified Modeling Language)
 - általános célú modellezési nyelv
- AADL (Architecture Analysis & Design Language)
 - architektúraleíró nyelv
- SysML (Systems Modeling Language)
 - UML-alapú általános modellezési nyelv rendszertervezési célokra
- EMF (Eclipse Modeling Framework, Ecore)
 - modellezési nyelv készítésére alkalmas
- (Adatközpontú modellezés, szakterületi modellek)

Kitekintés: refaktoring

- **Code refactoring is the process of restructuring existing computer code – changing the factoring – without changing its external behavior.**
Refactoring improves nonfunctional attributes of the software. Advantages include
 - improved code readability and
 - reduced complexity to improve source code maintainability, and
 - create a more expressive internal architecture or object model to improve extensibility.
- **Bővebben: felsőbbéves szoftveres tantárgyak**

Refaktoring példa: „pull up”

