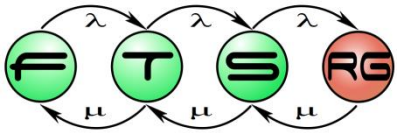
Strukturális modellezés

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

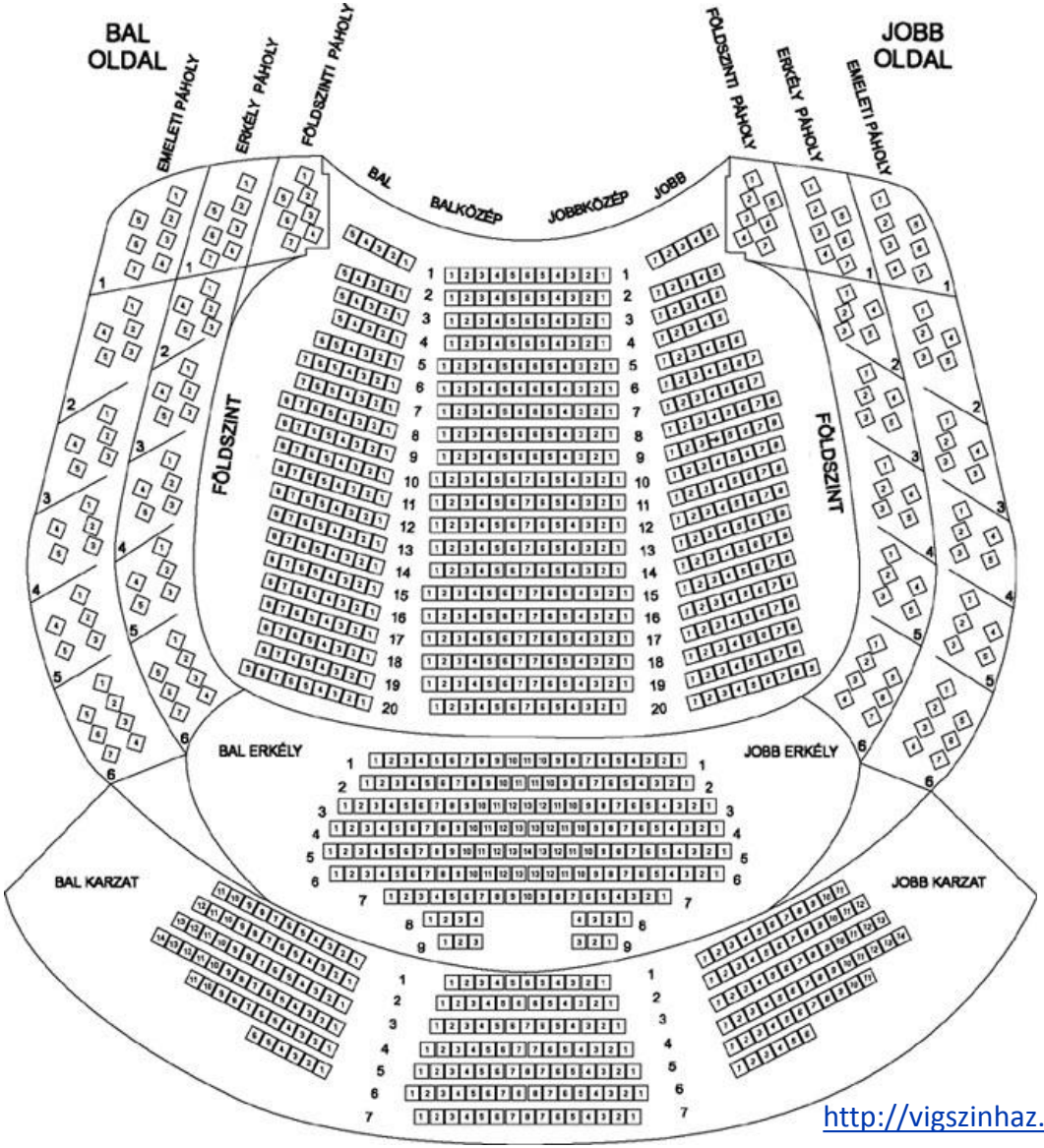
Budapesti Műszaki és Gazdaságtudományi Egyetem
Hibatűrő Rendszerek Kutatócsoport



Miről lesz szó?

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

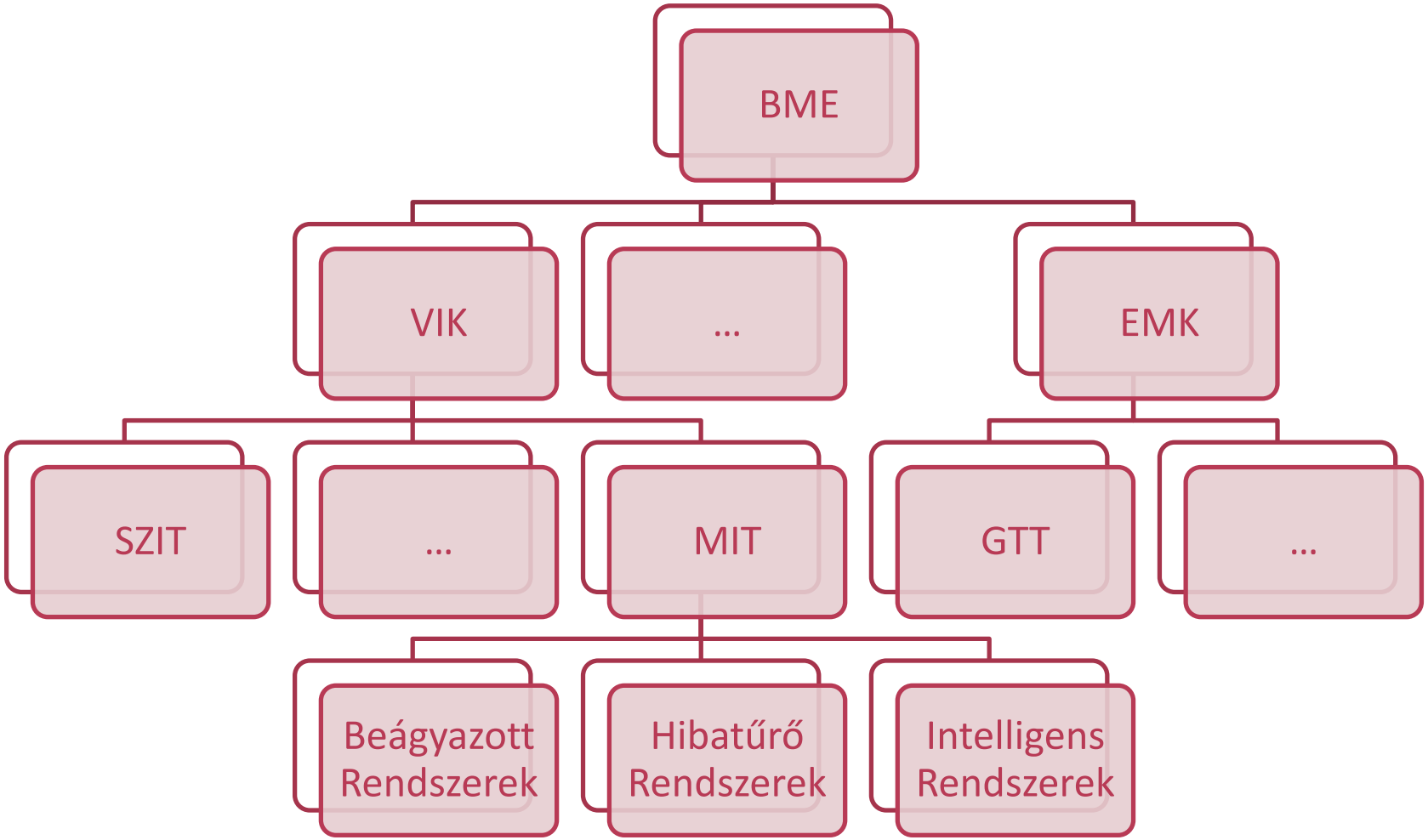
Illusztráció – Felépítési modellek



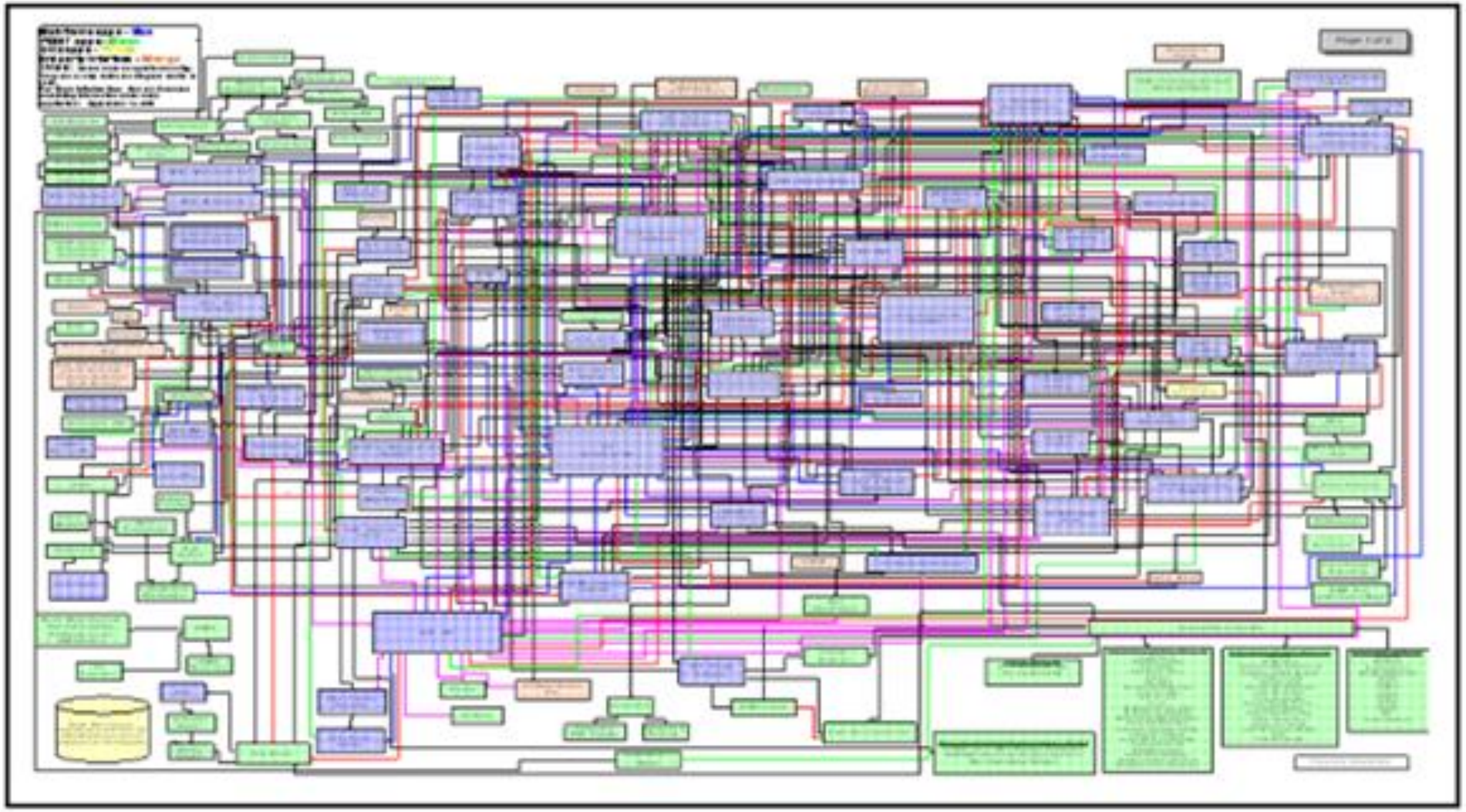
http://vigszinhaz.hu/images/upload/nezo_v1.jpg

Illusztráció – Felépítési modellek

Szervezeti felépítés (ld. tartalmazási hierarchia)



(… átlát a káoszon?)

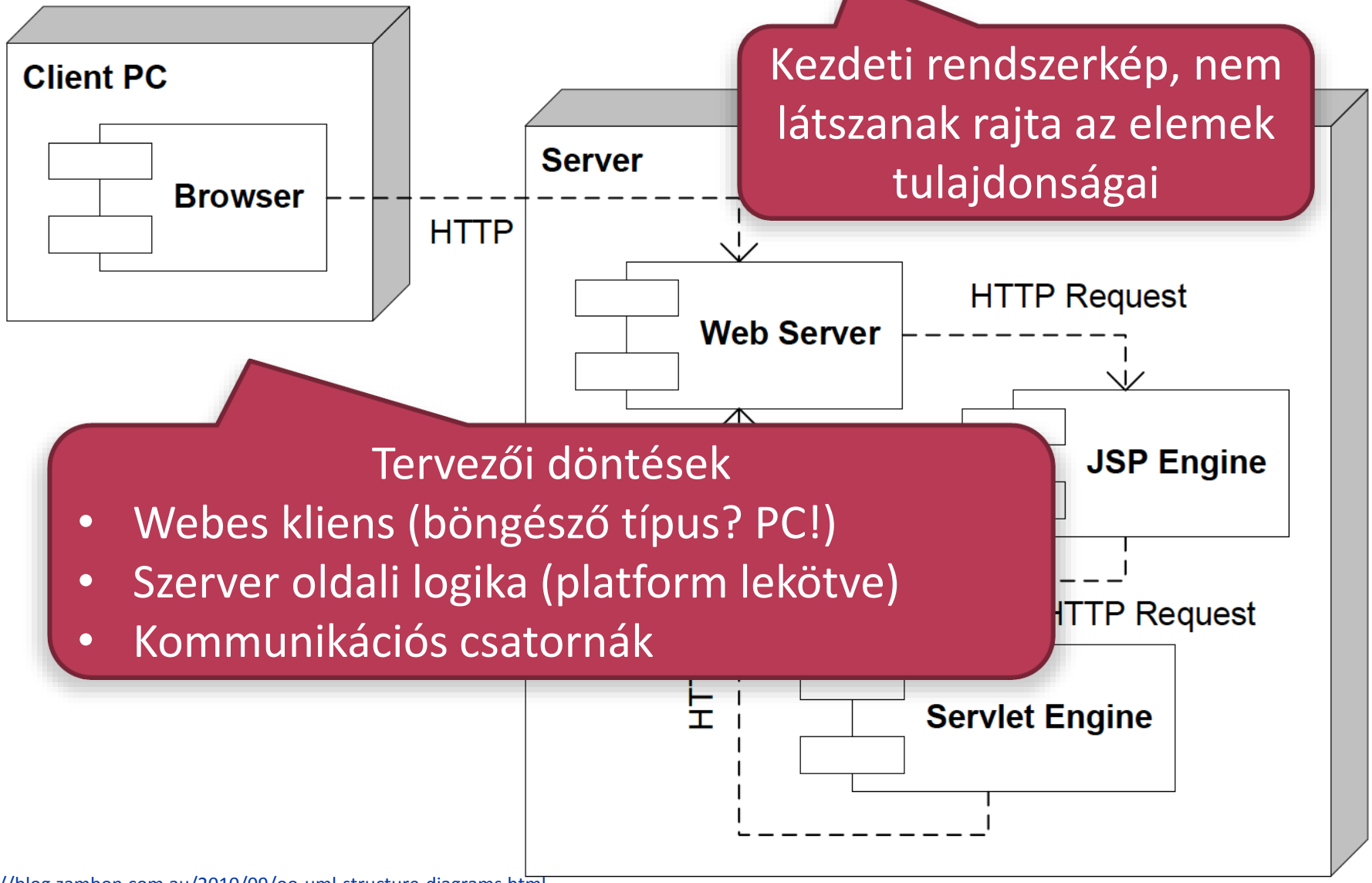


http://www.mikethearchitect.com/2012/10/challenge-the-status-quo-and-advance-business-through-cloud-computing.html

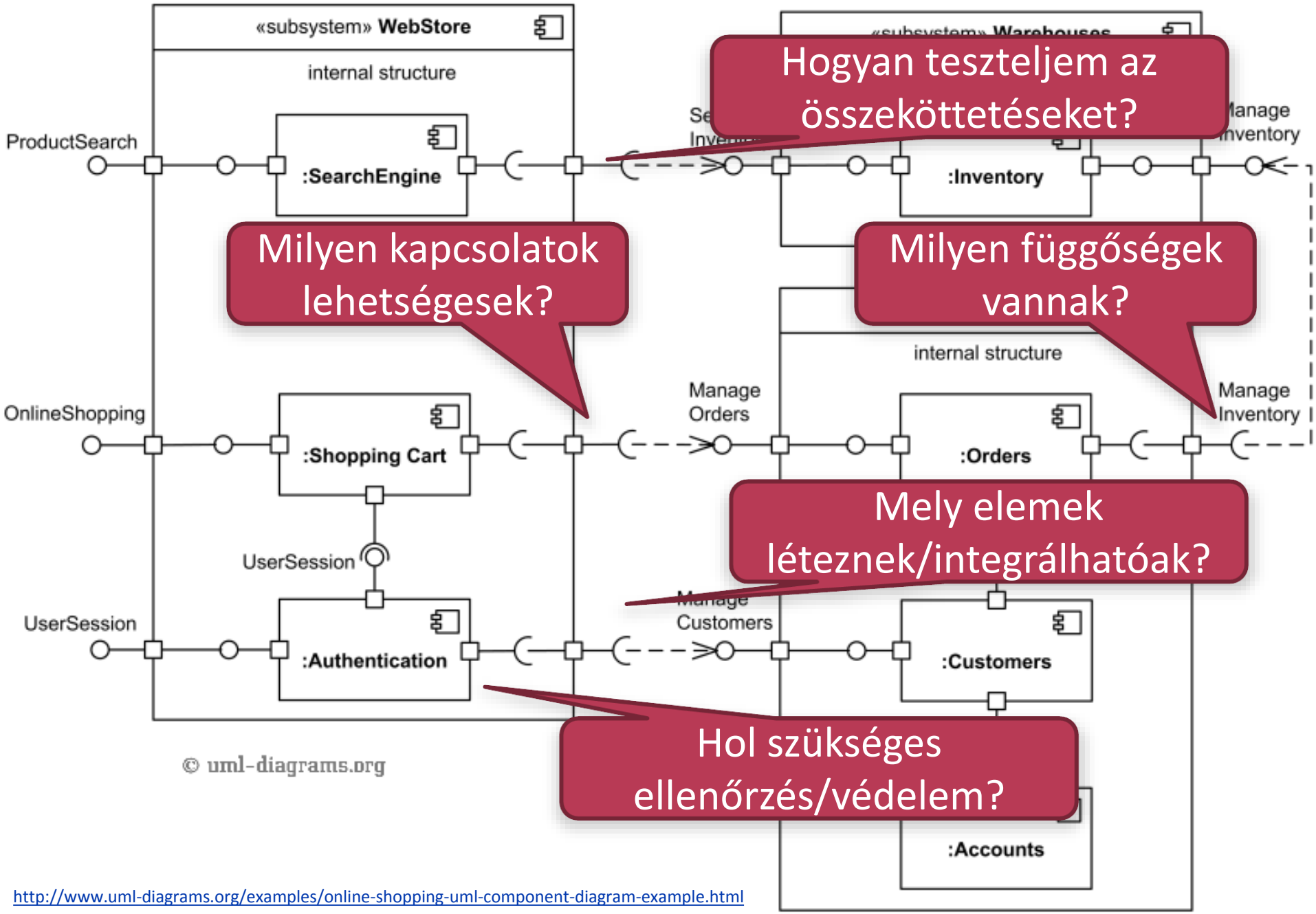
Strukturális modellezés célja

- 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://blog.zambon.com.au/2010/09/oo-uml-structure-diagrams.html



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

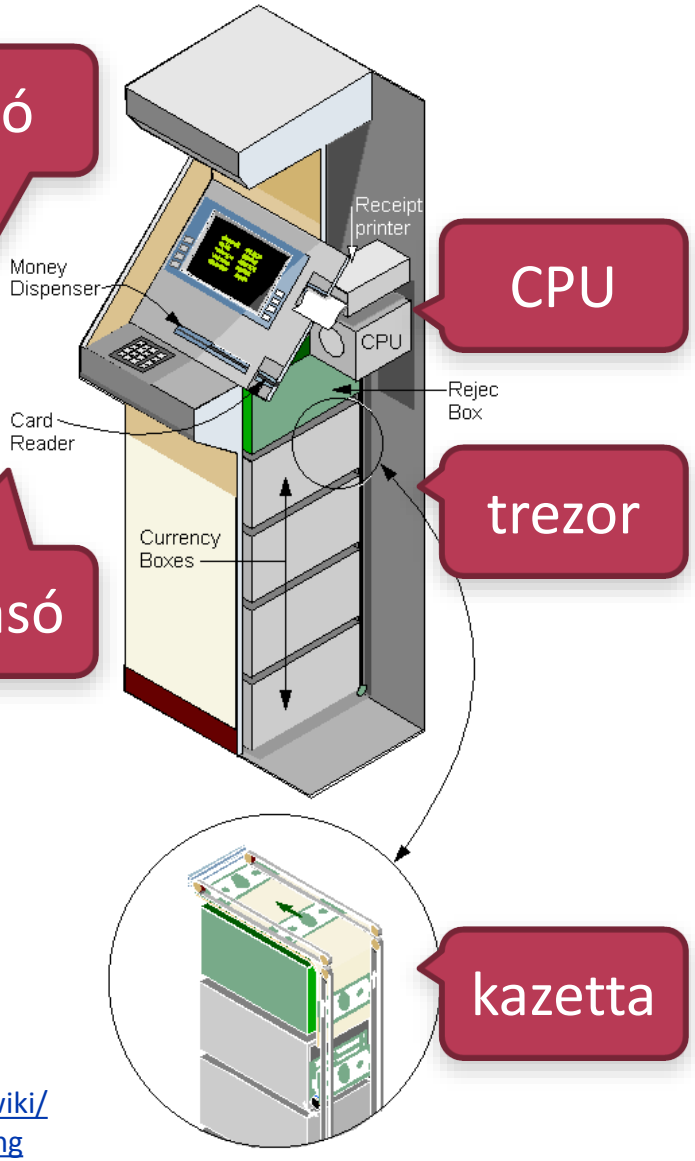
pénzkiadó

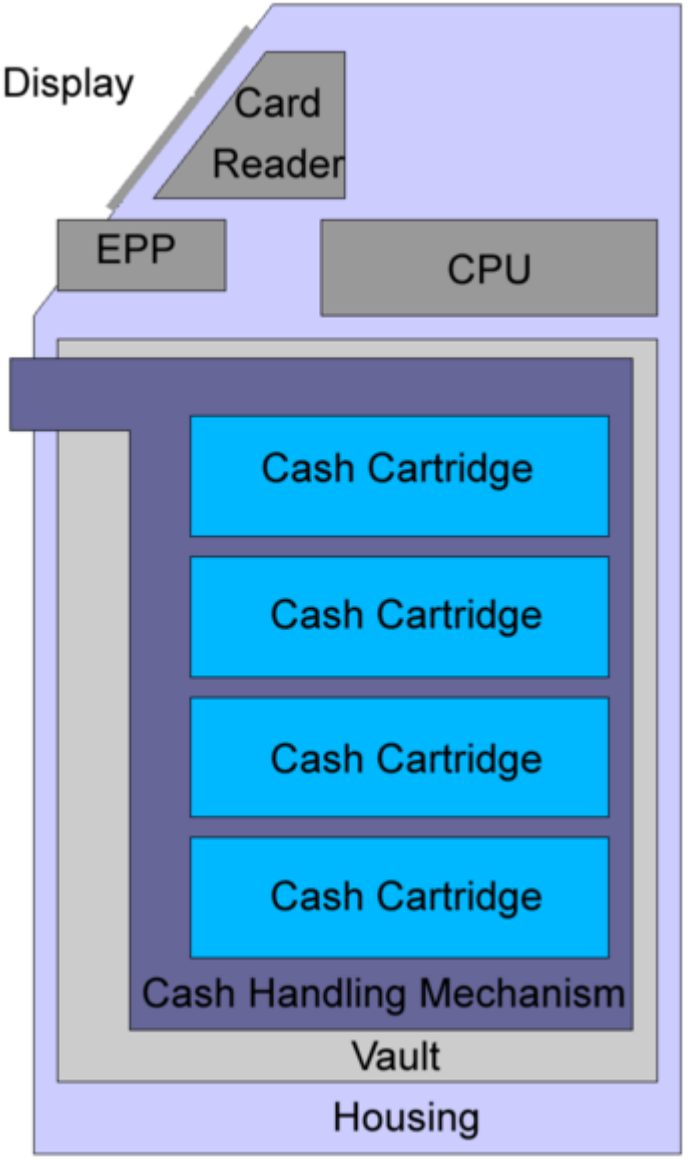
kártyaolvasó

CPU

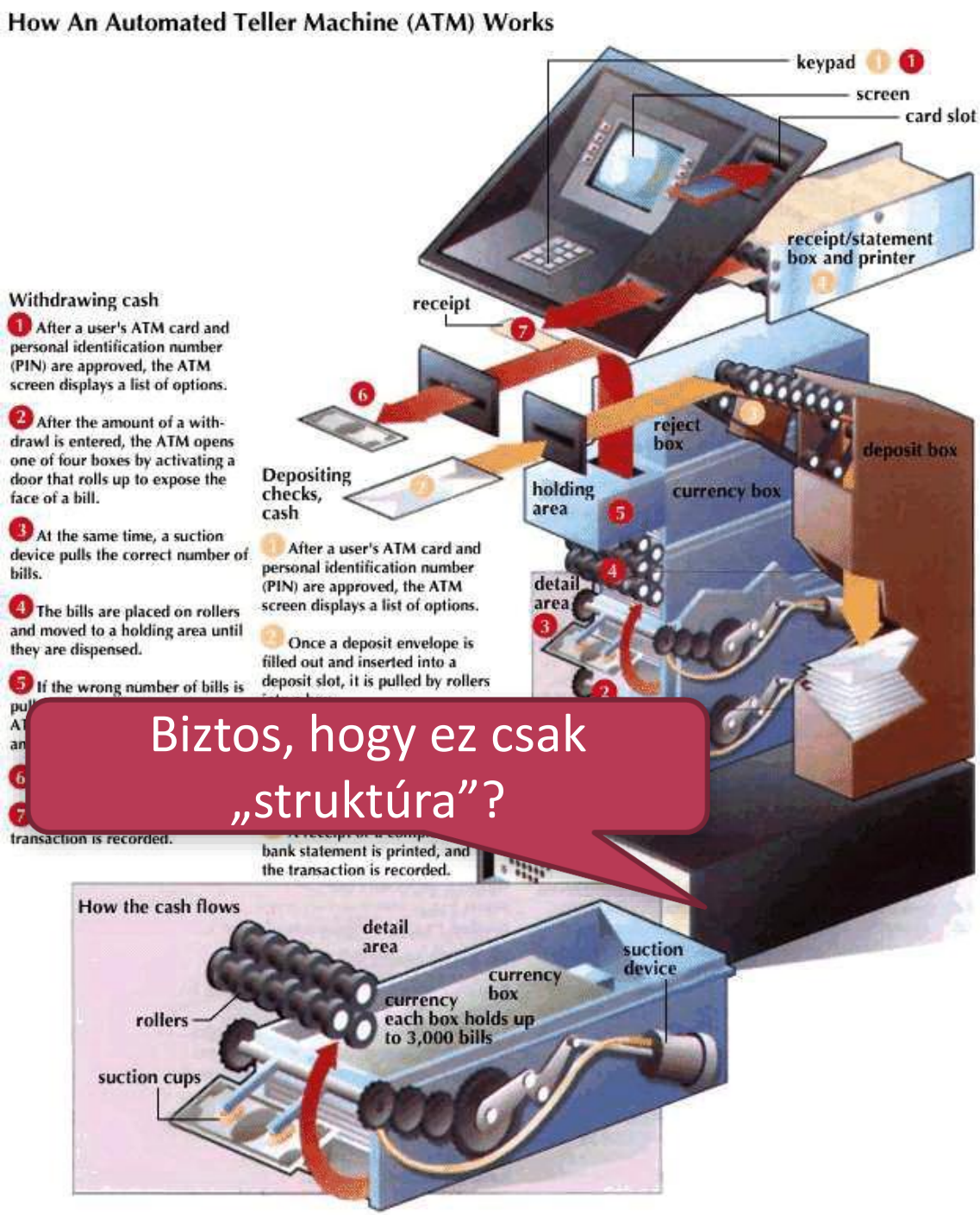
trezor

kazetta





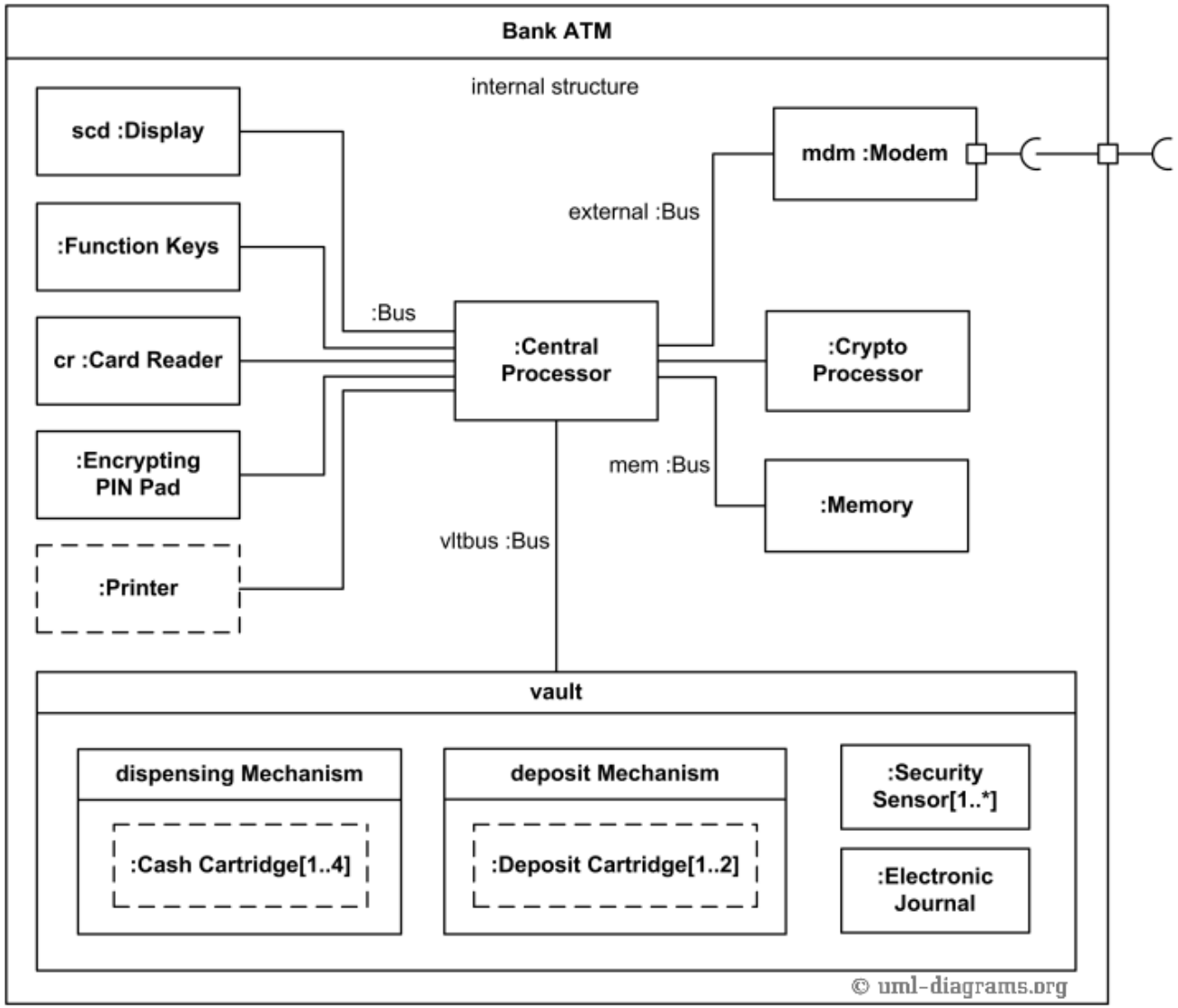
http://en.wikipedia.org/wiki/File:Atm_blockdiagram.png



Biztos, hogy ez csak „struktúra”?

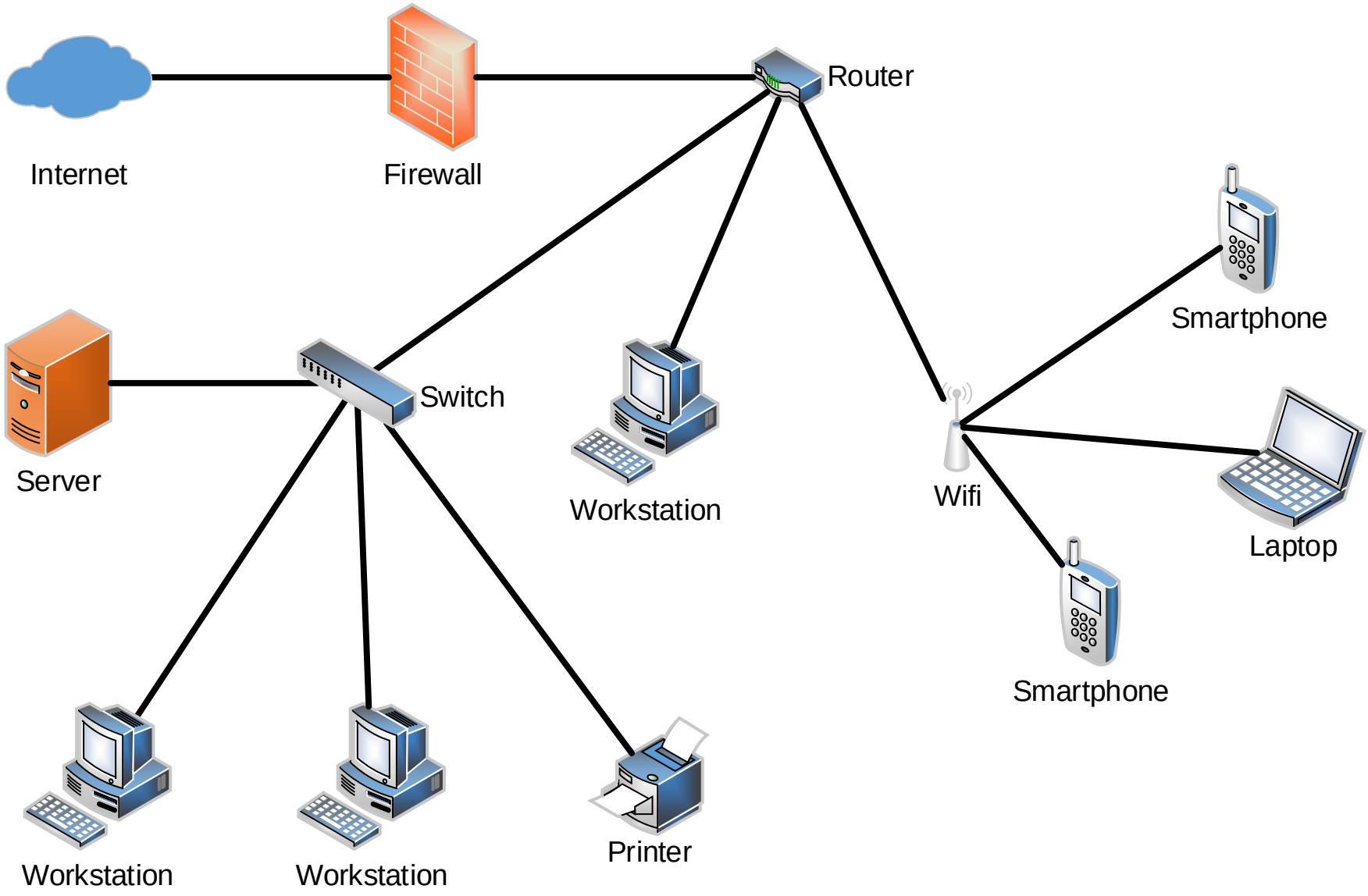
http://www.britannica.com/EBchecked/media/95649/The-components-of-an-automated-teller-machine

ATM struktúra

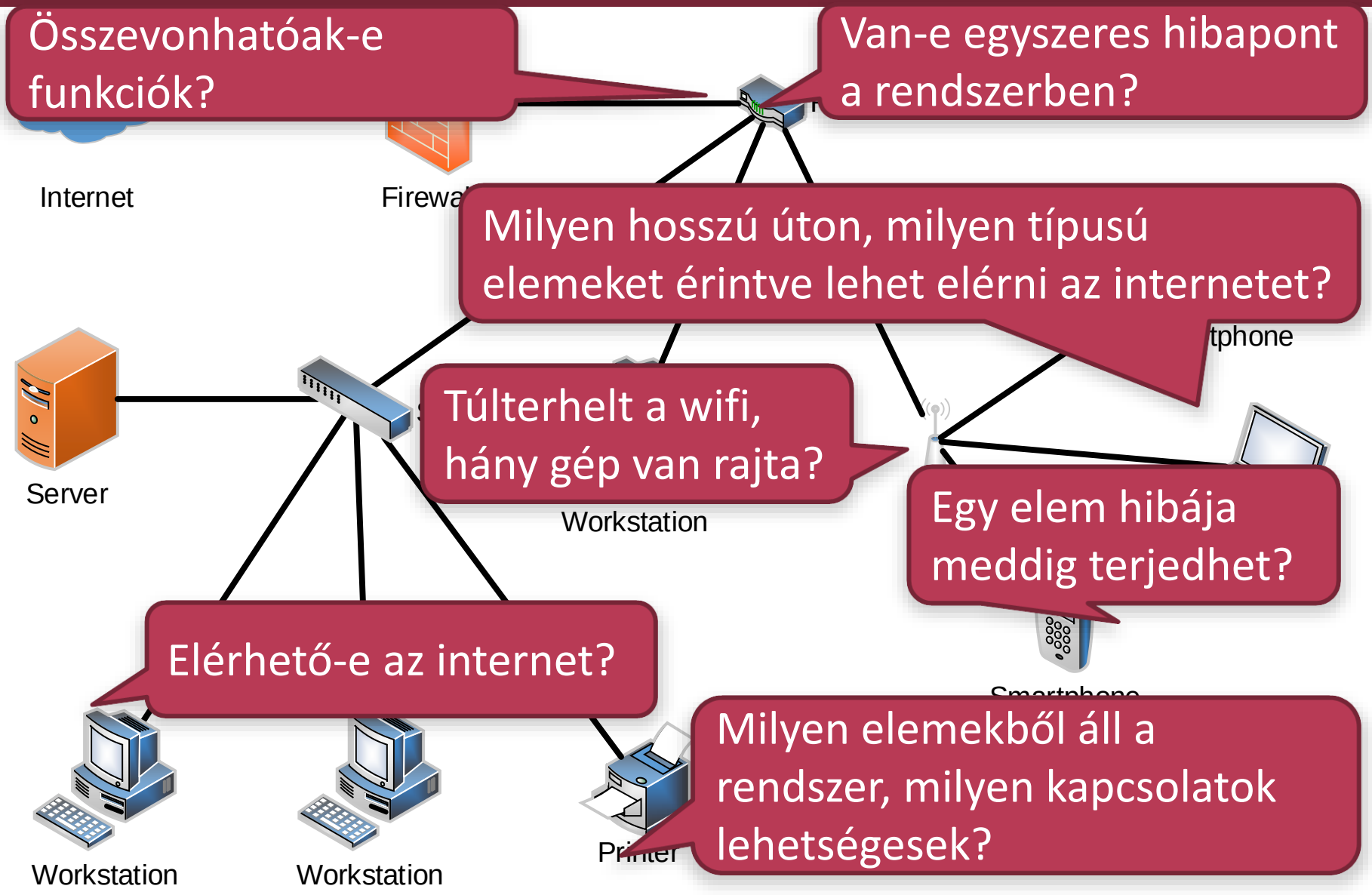


http://www.pcmag.com/encyclopedia/term/38117/atm-machine

Példa: (céges) hálózat



Példa: (céges) hálózat

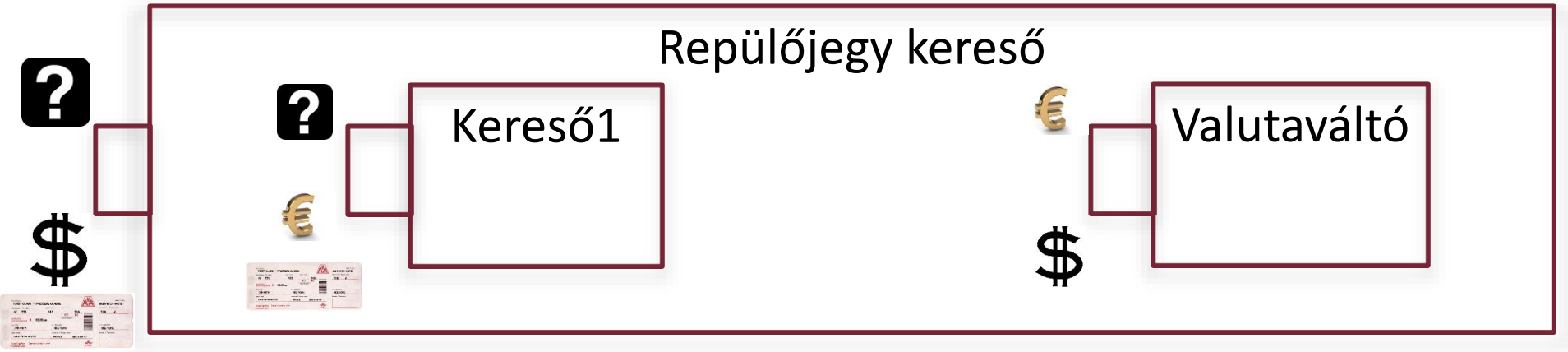


Dekompozíció

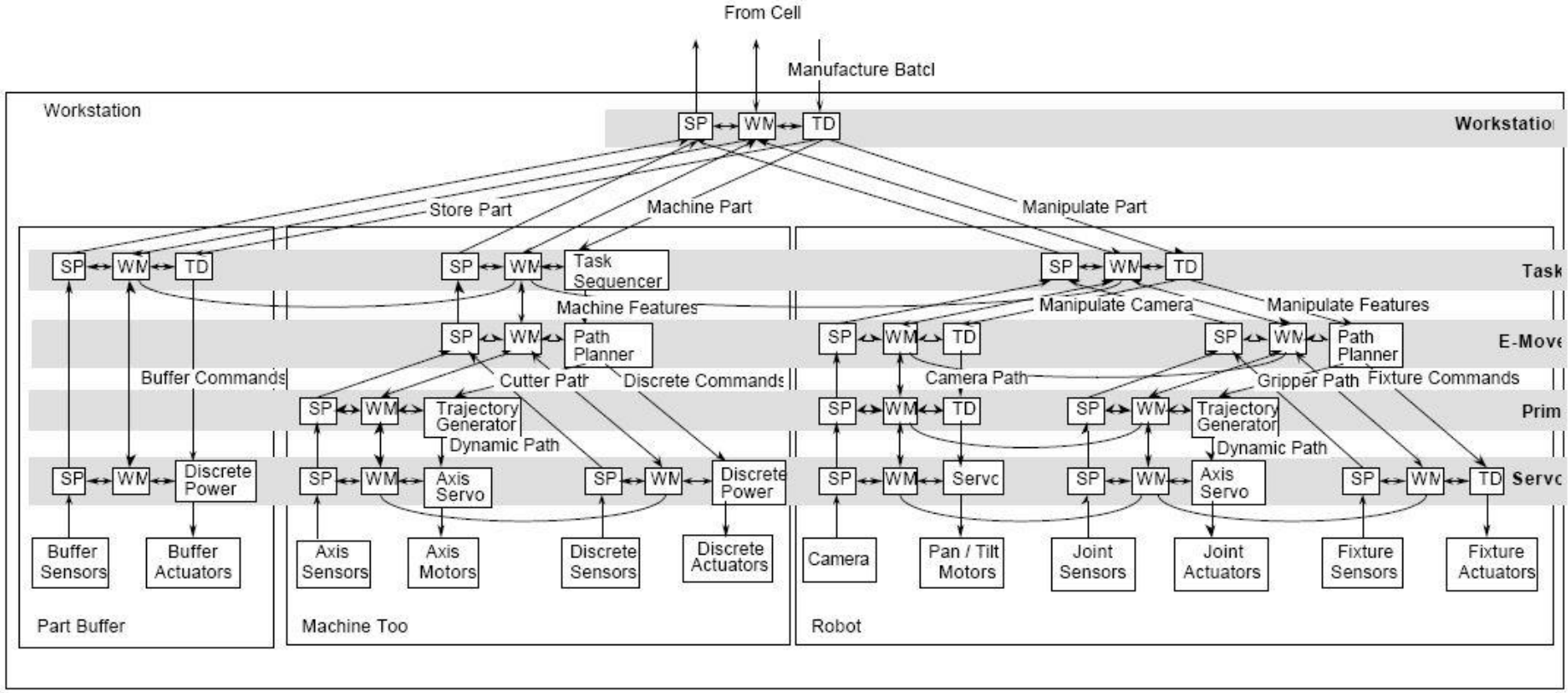
- **Decomposition** in computer science, also known as **factoring**, is breaking a complex problem or *system* into parts that are easier to conceive, understand, program, and maintain. (Wikipedia)
- A dekompozíció („faktoring”) tehát egy összetett probléma vagy *rendszer* kisebb részekre bontása, amelyek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

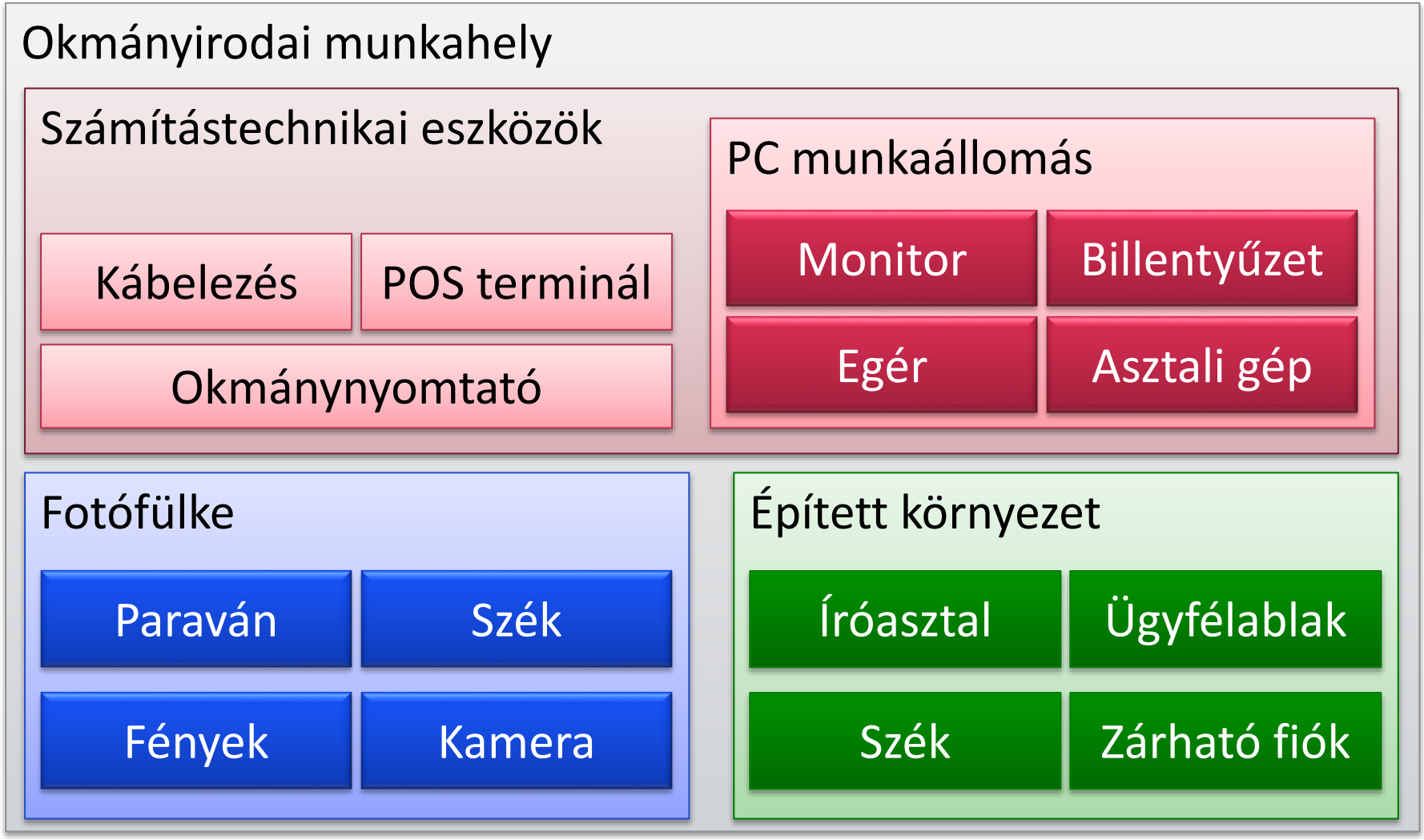


Példa: ipari vezérlő (Wikipedia)



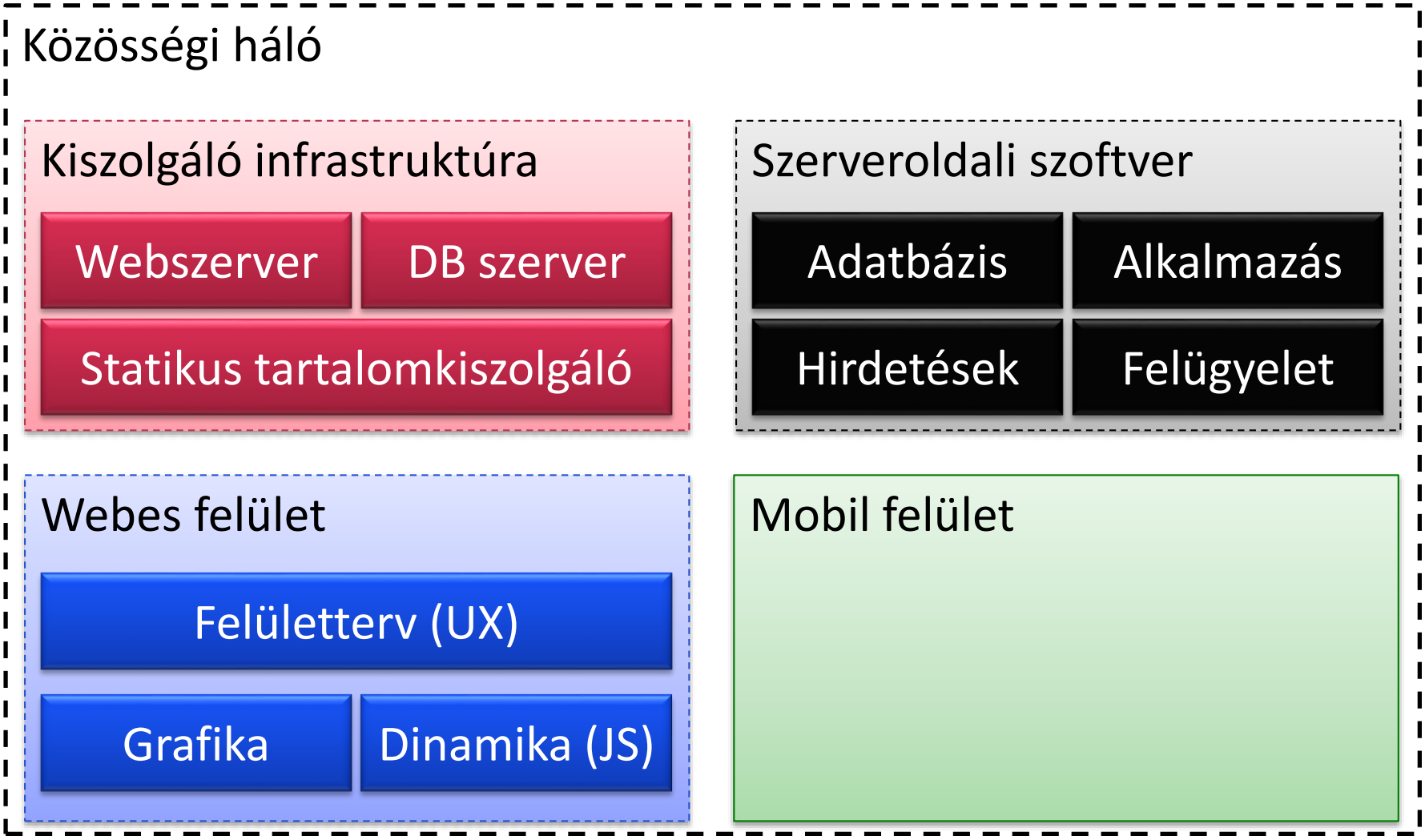
Top-down tervezés

- Alaplépés: dekompozíció



Bottom-up tervezés

- Alaplépés: kompozíció



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 készültségnél 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**

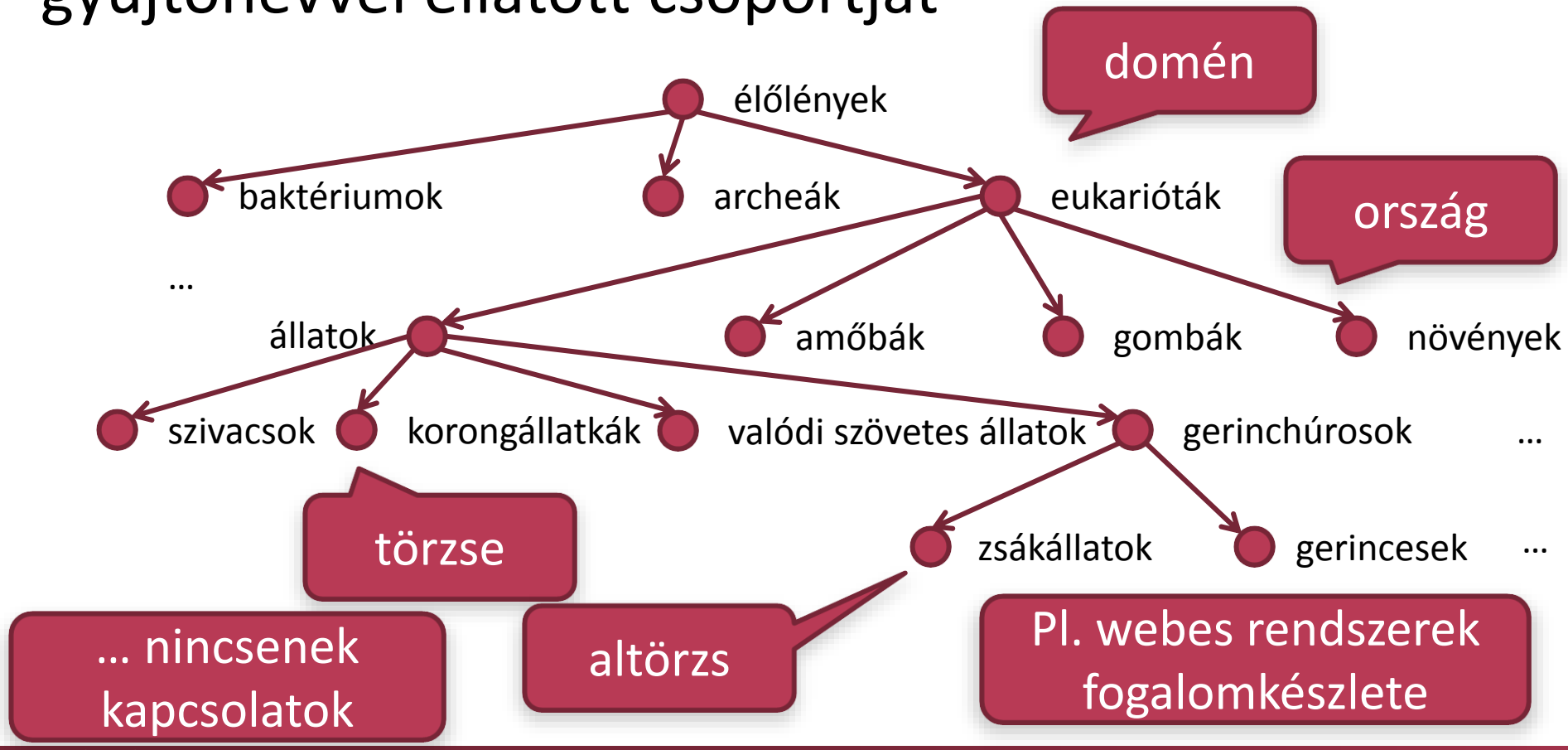
Definíciók

- Az informatikában mindennek több, gyakran ellentmondó definíciója van
  - rendszer = ?
  - modell = ?

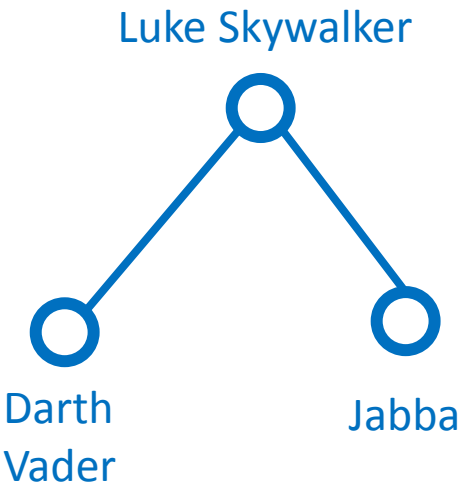
- Gyakran ugyanarra a fogalomra több nevével is hivatkozunk
  - csomópont = csúcs
    - angolban: *node, vertex, object, concept*
  - él = kapcsolat
    - angolban: *edge, link, arc, connection, relationship*

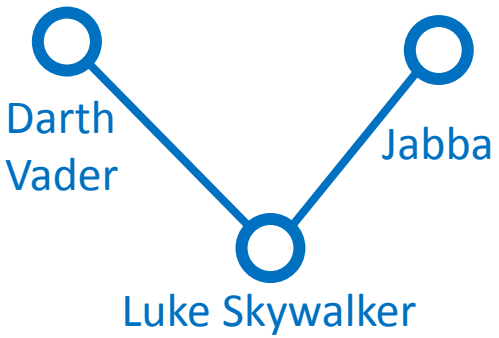
Taxonómia

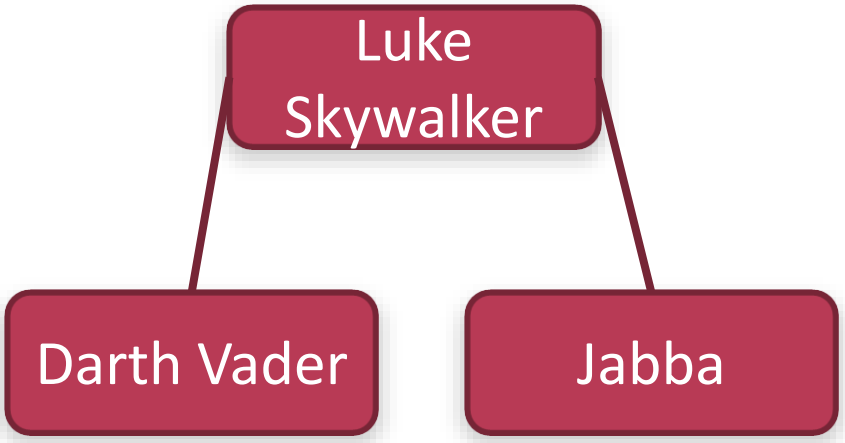
„A biológiai rendszertanban **taxonnak** nevezik az élőlények egyazon kategóriába sorolt és közös gyűjtőnévvel ellátott csoportját”

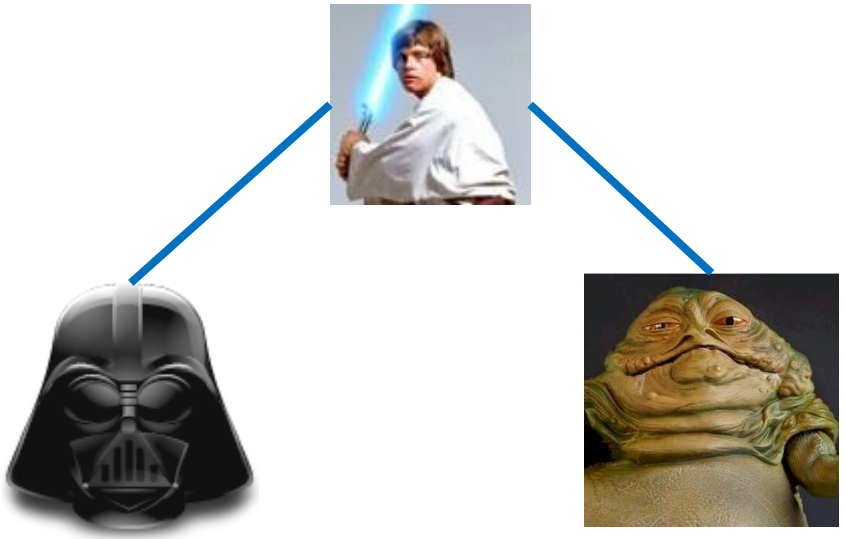


Modell (gráf) ≠ diagram

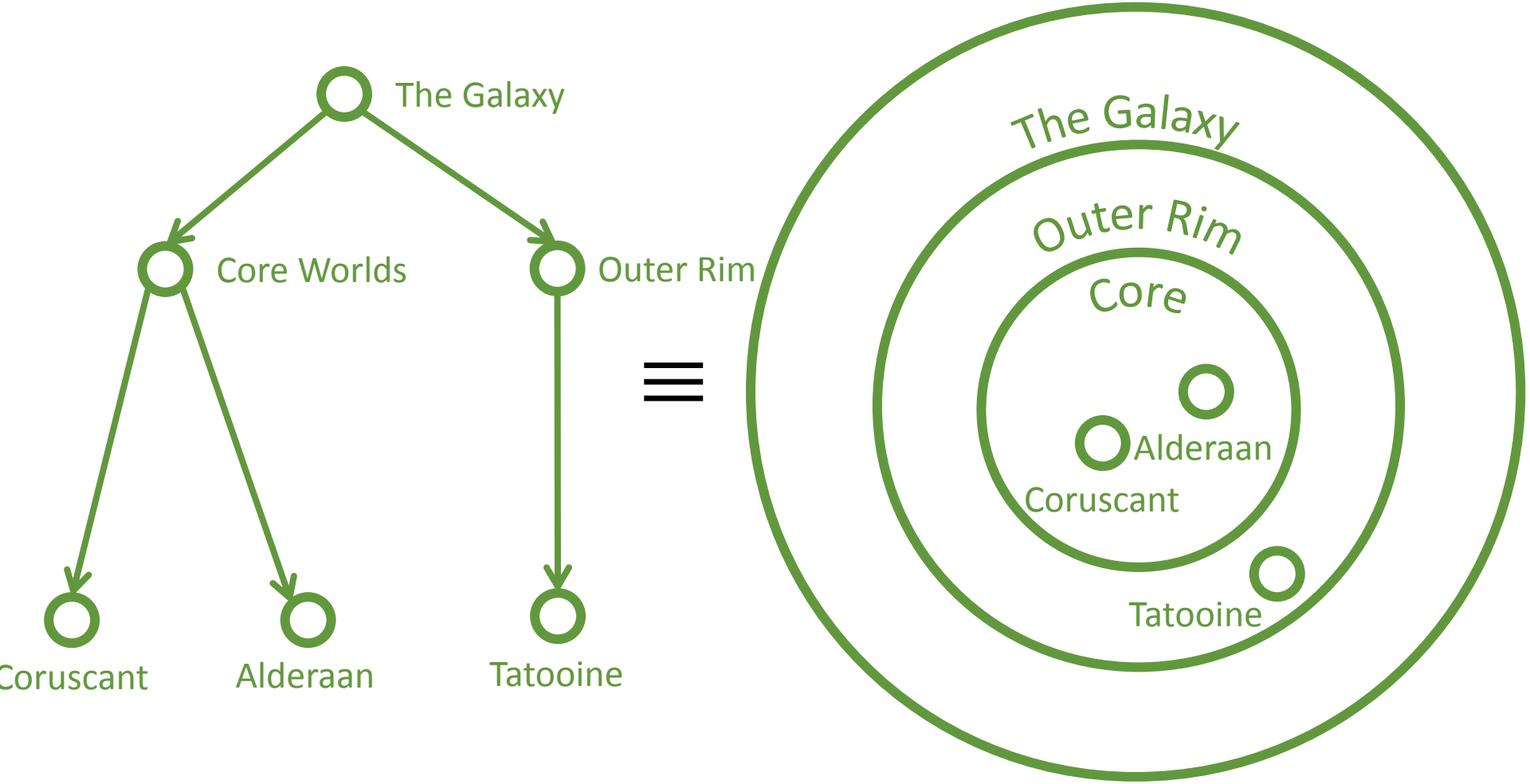








Hierarchia ábrázolása

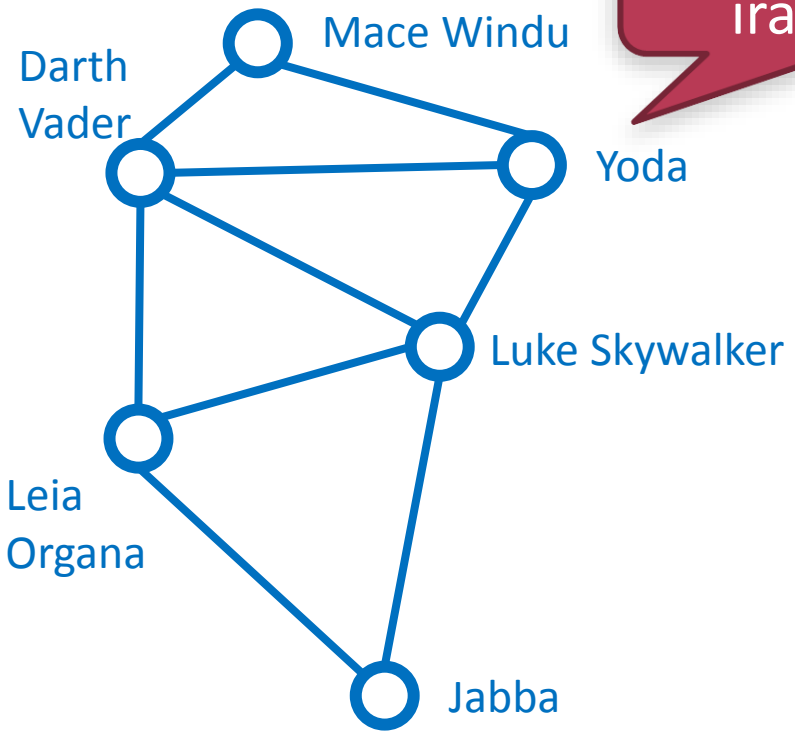


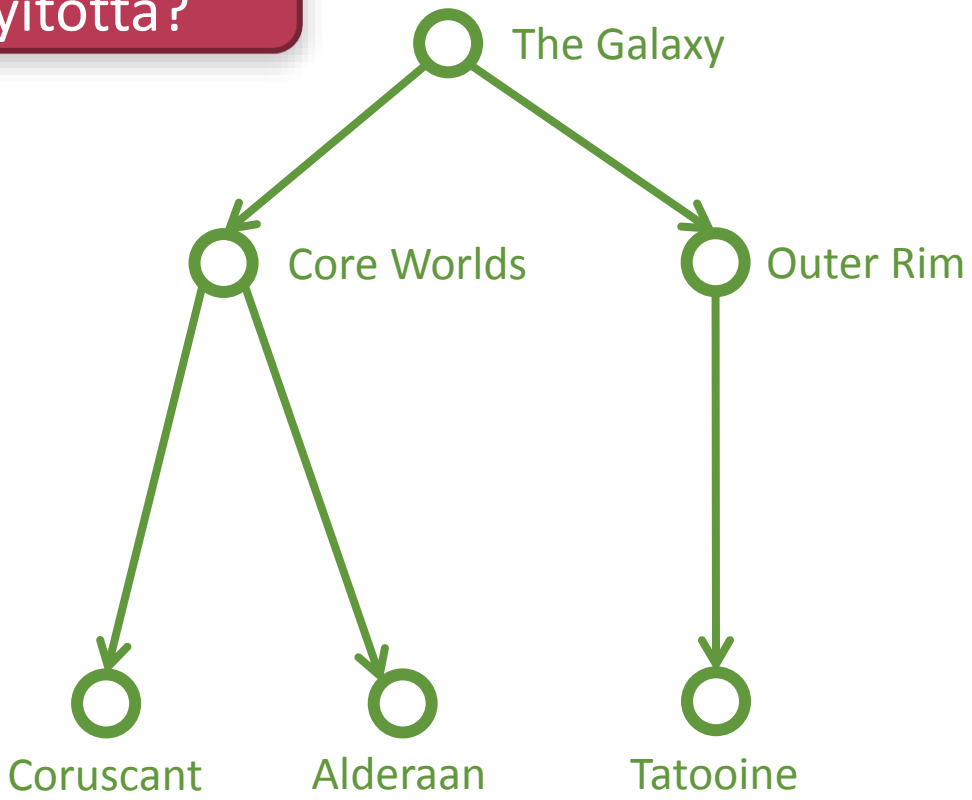
Egyszerű gráfok

kapcsolati háló:
irányítatlan gráf

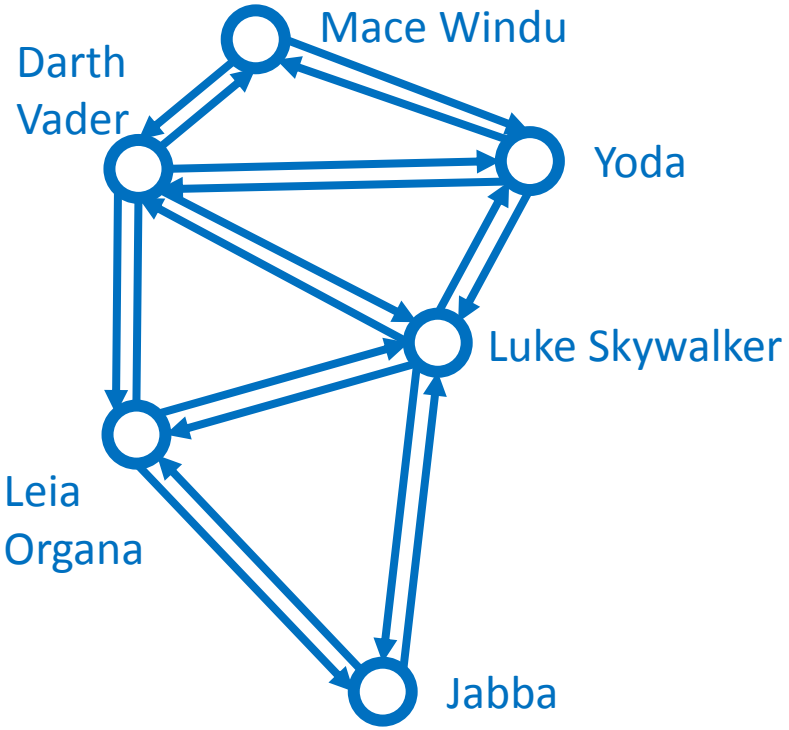
helyszínek:
irányított gráf

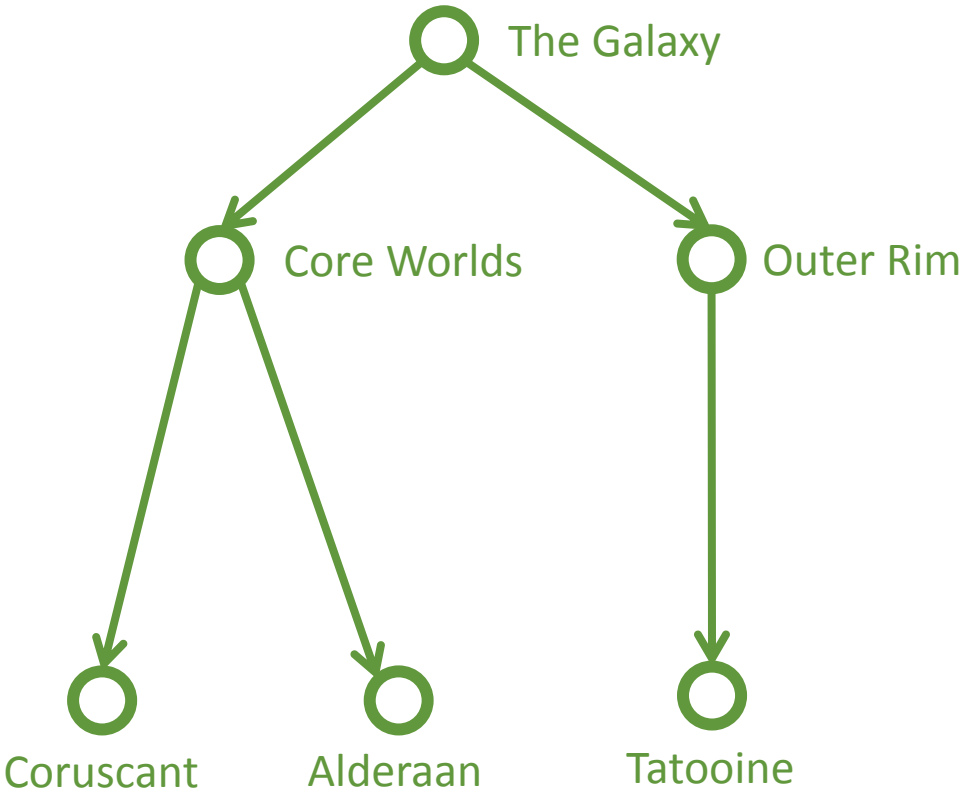
hogyan alakítható
irányítottá?





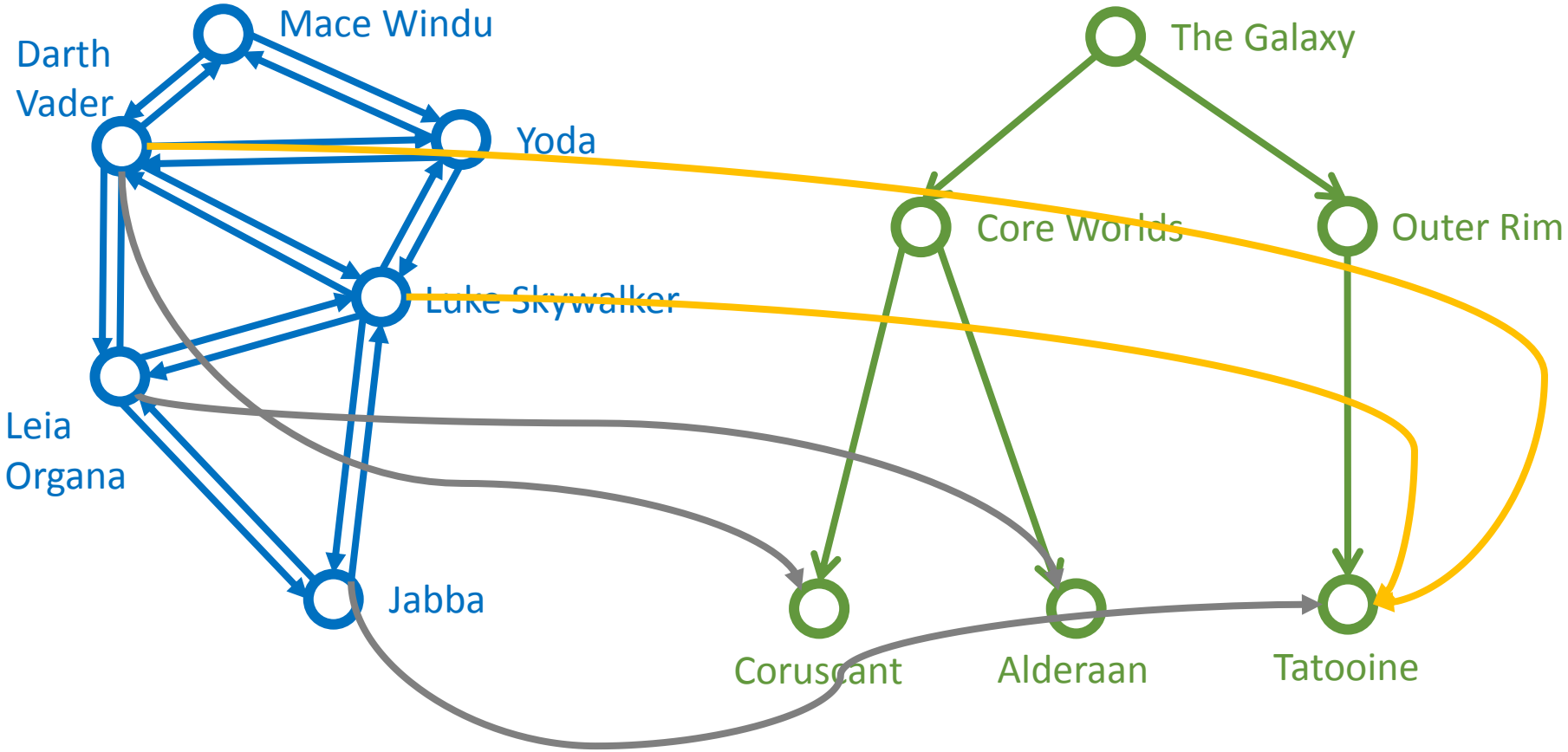
Egyszerű gráfok





Típusos gráfok

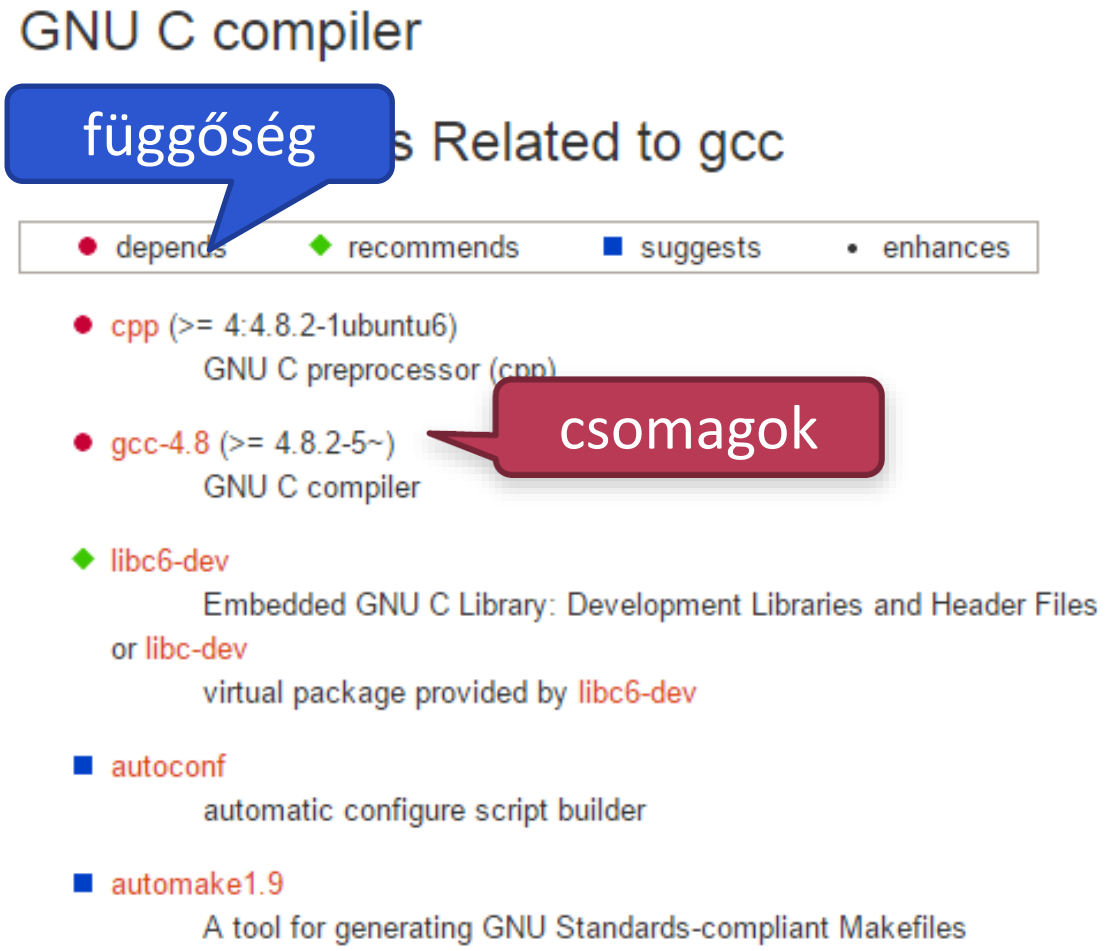
Különböző típusú csomópontok és élek



GRÁF ALAPÚ TUDÁSREPREZENTÁCIÓ

Közösségi háló

Függőségek kezelése



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

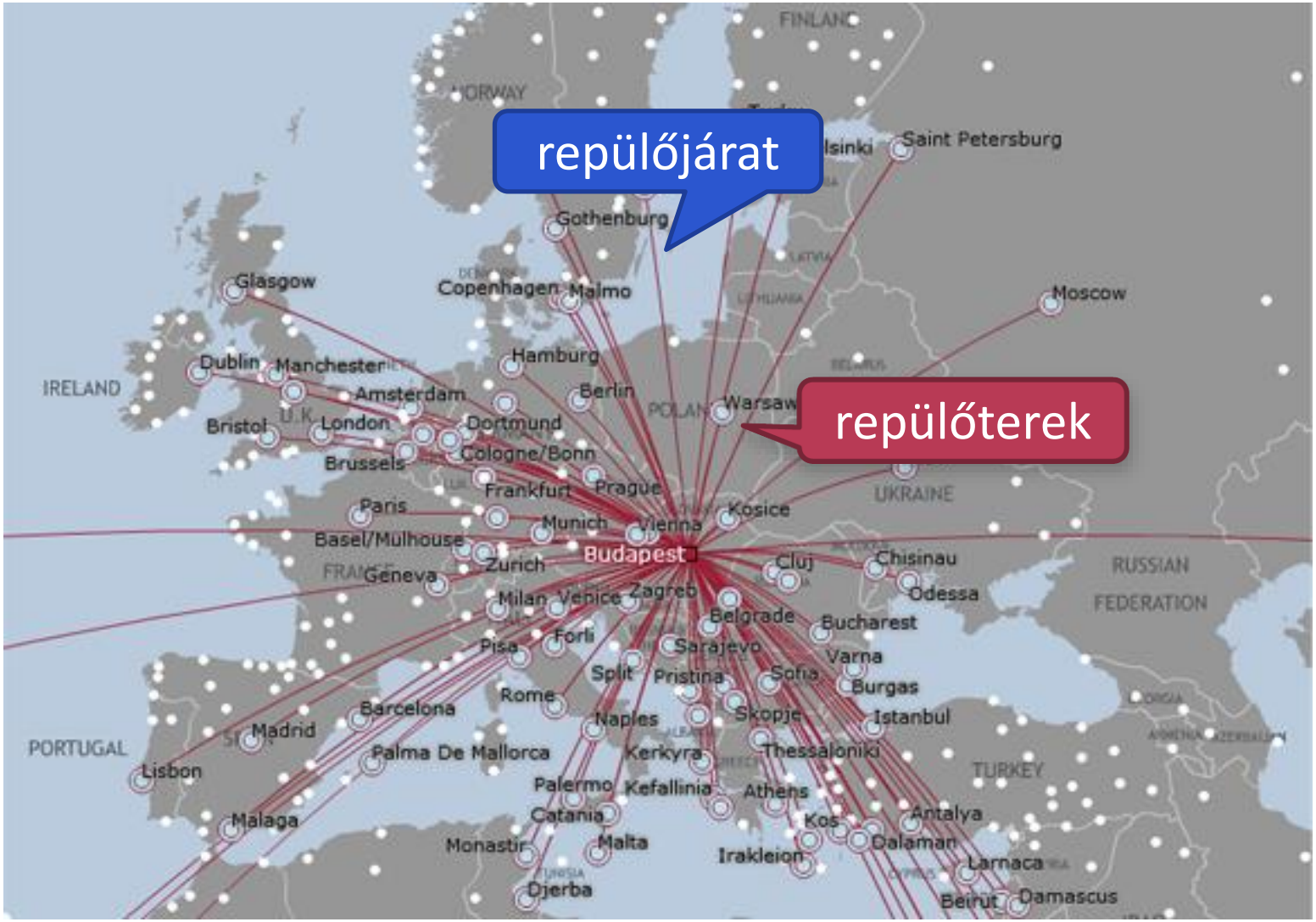
io.h

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

util.h

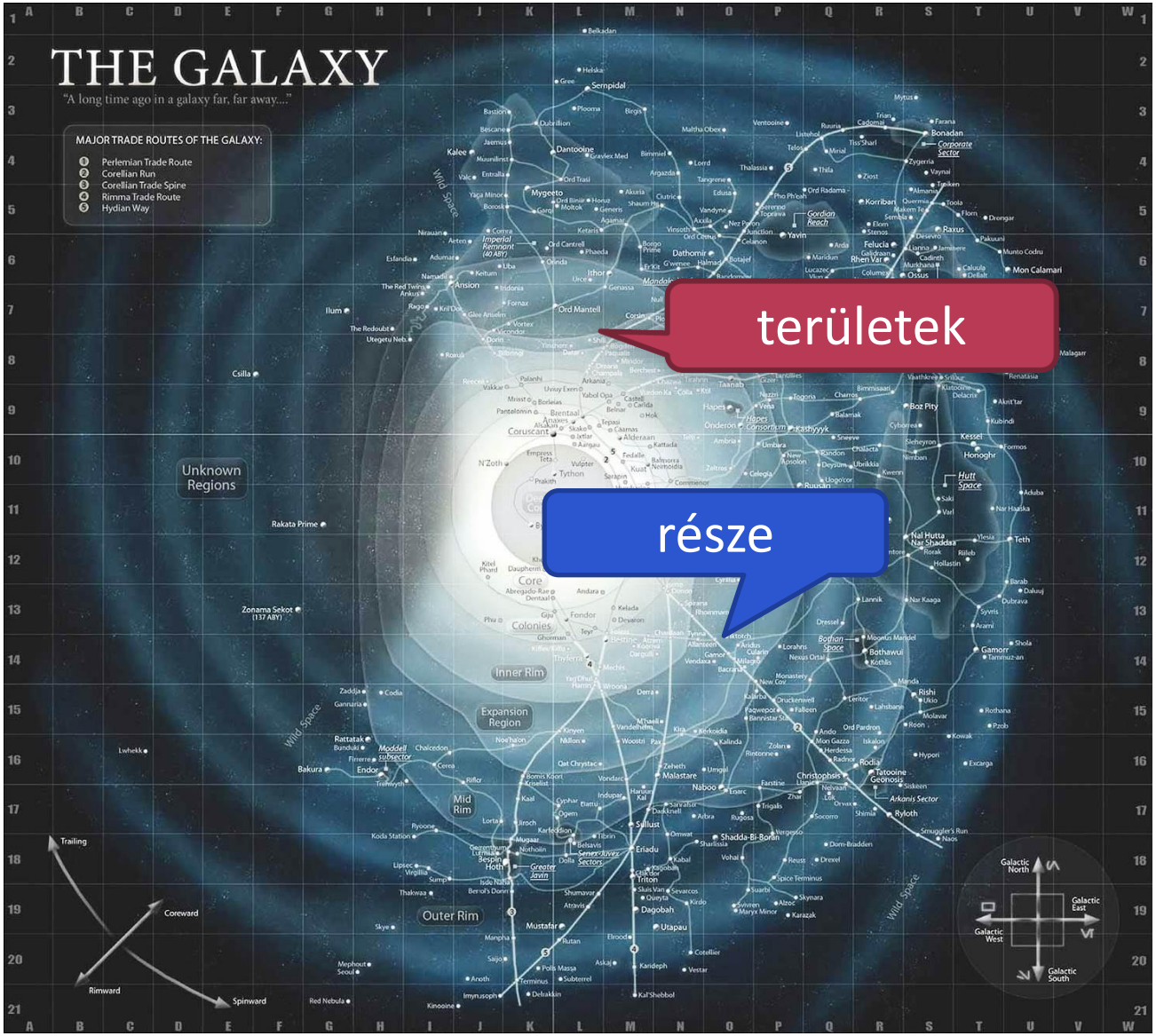
```
1 // ...
2
3
4
5
```

Repülési útvonalak



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

A Galaxis

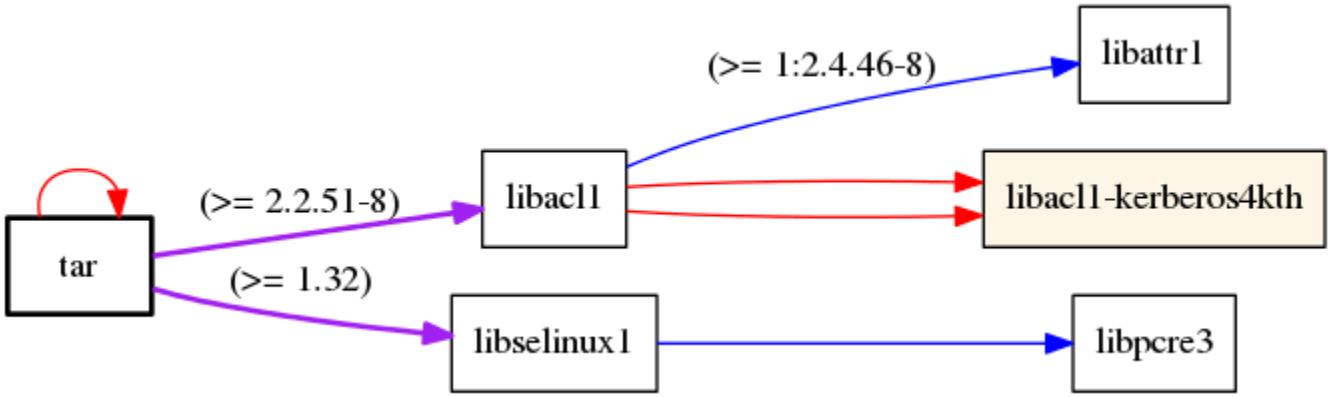


http://starwars.wikia.com/wiki/The_galaxy

Útvonal fogalma

- Szükséges-e a `libpcre3` csomag a `tar` telepítéséhez?
- Hogyan tudok eljutni a Deák Ferenc térre?

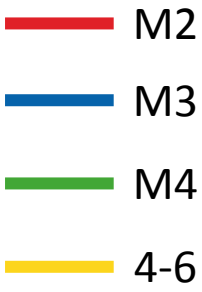
A tar csomag függőségei



Budapest – kötöttpályás közlekedés

M2
M3
M4
4-6

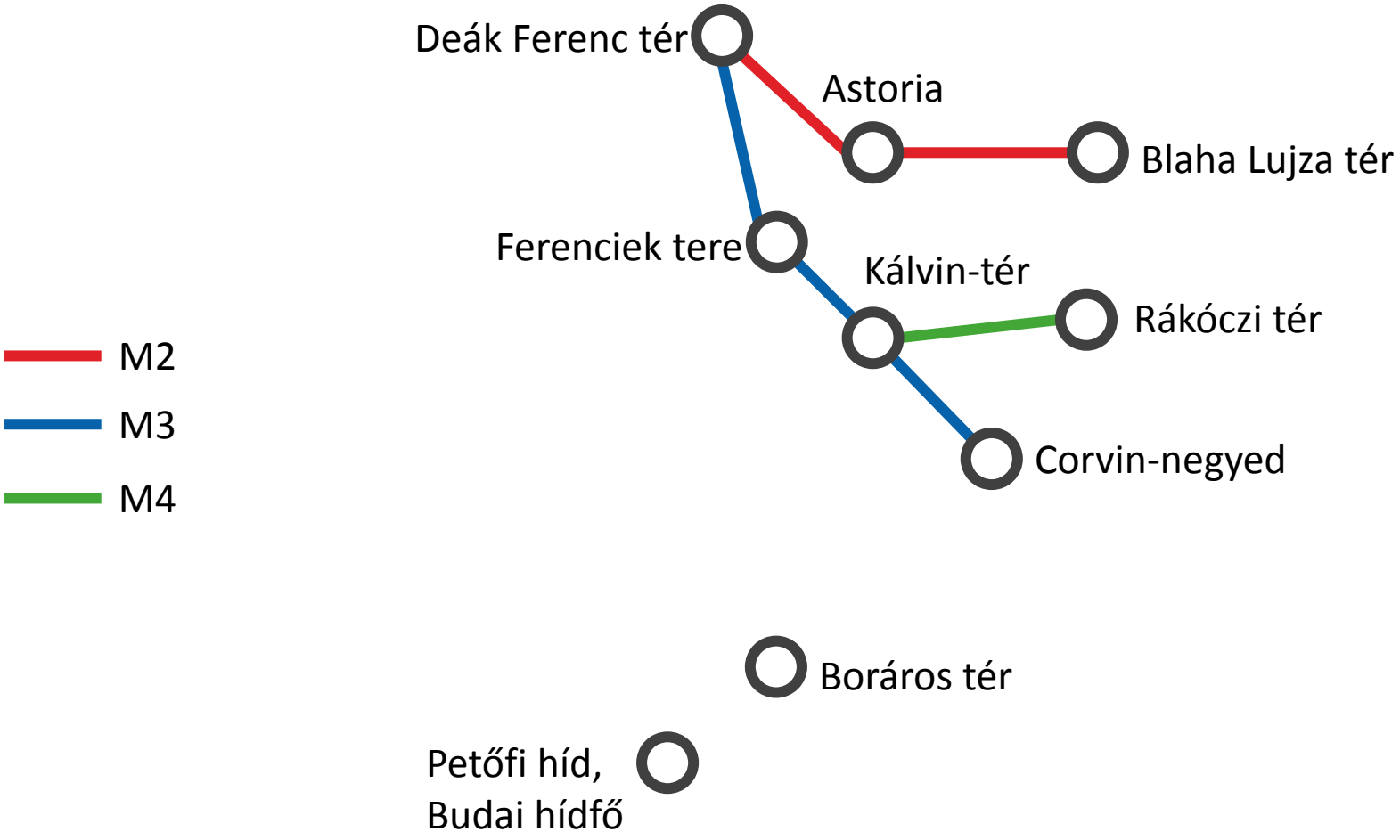
Budapest – kötöttpályás közlekedés



legrövidebb út?

Szűrés: élcímke szerint (részgráf)

- Pl. csak a metróhálózatot szeretnénk nézni

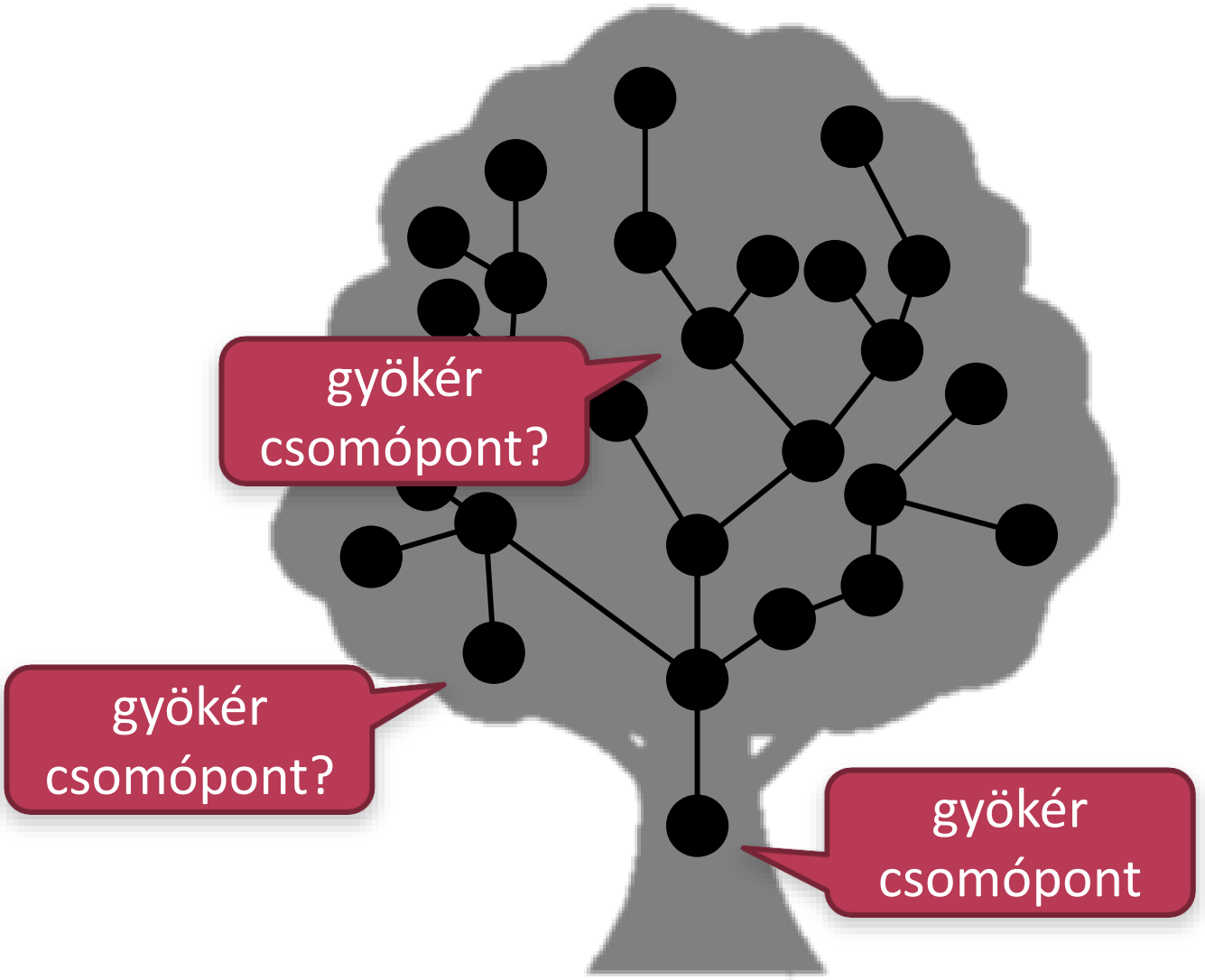


- Tranzitív lezárt: hova lehet eljutni metróval?

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

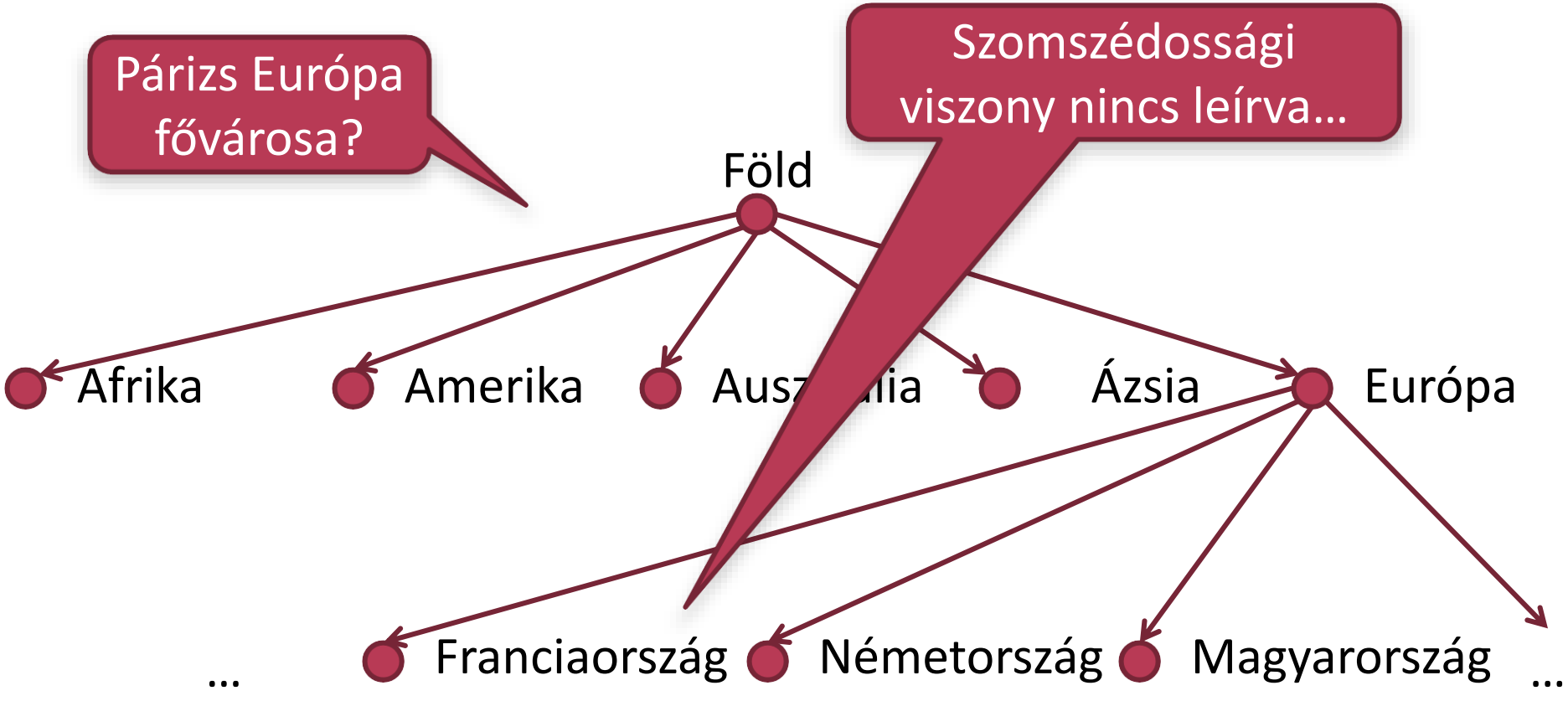
Fa



Fastruktúra tulajdonságai

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

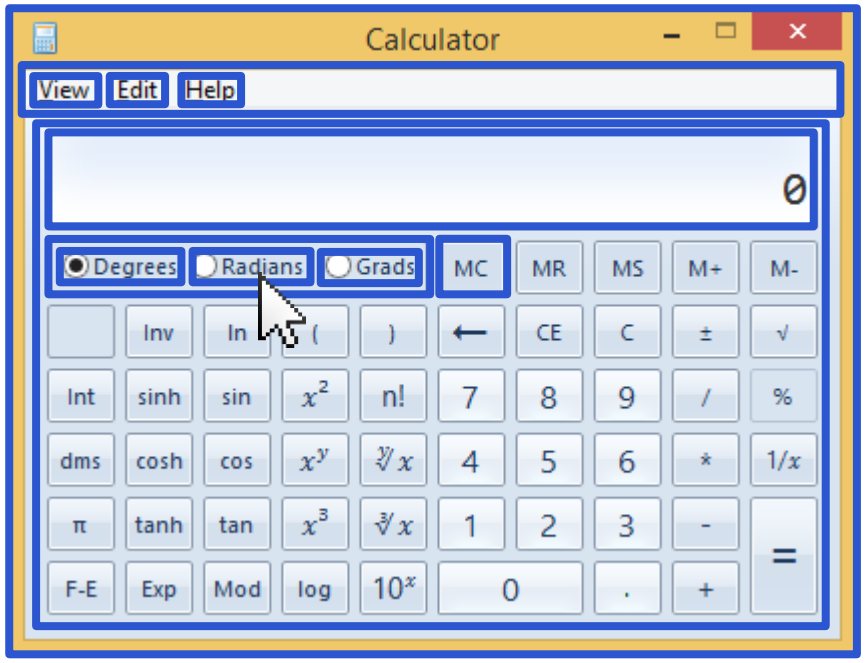
Fastruktúra ábrázolása – világtérkép

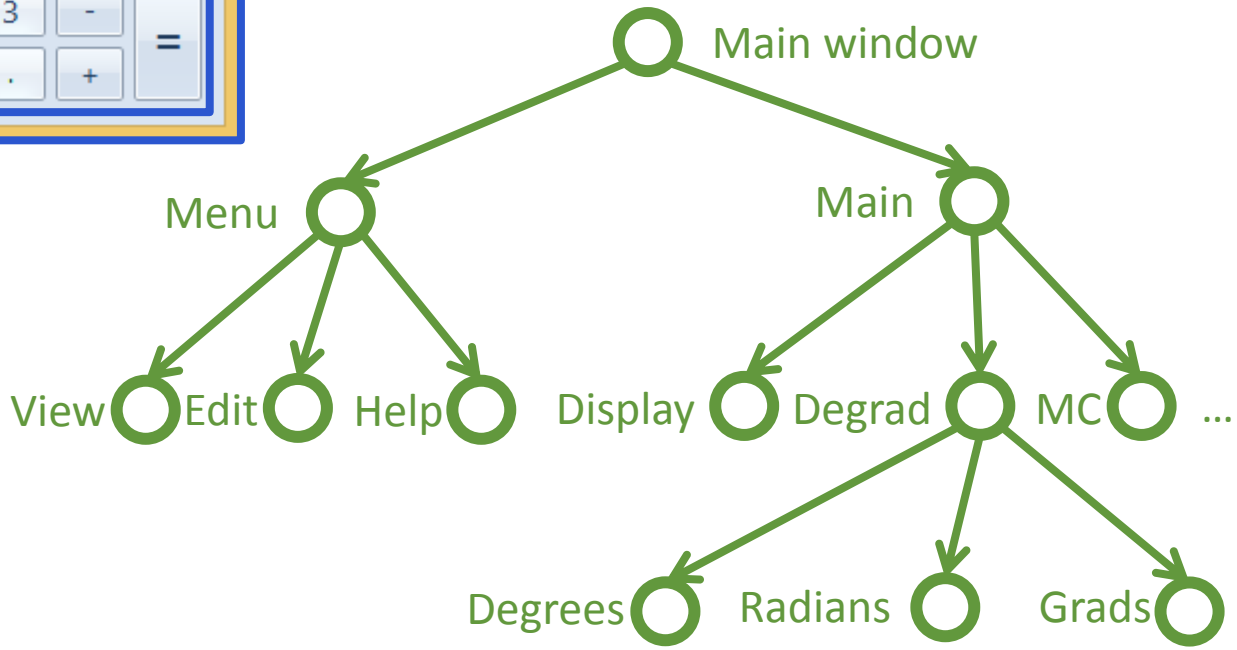


Fastruktúra – bennfoglaló ábrázolás

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

Grafikus felhasználó felület





Fastruktúra ábrázolása – fájlrendszer

C:

\Program Files

\Common Files

\Windows

\Fonts

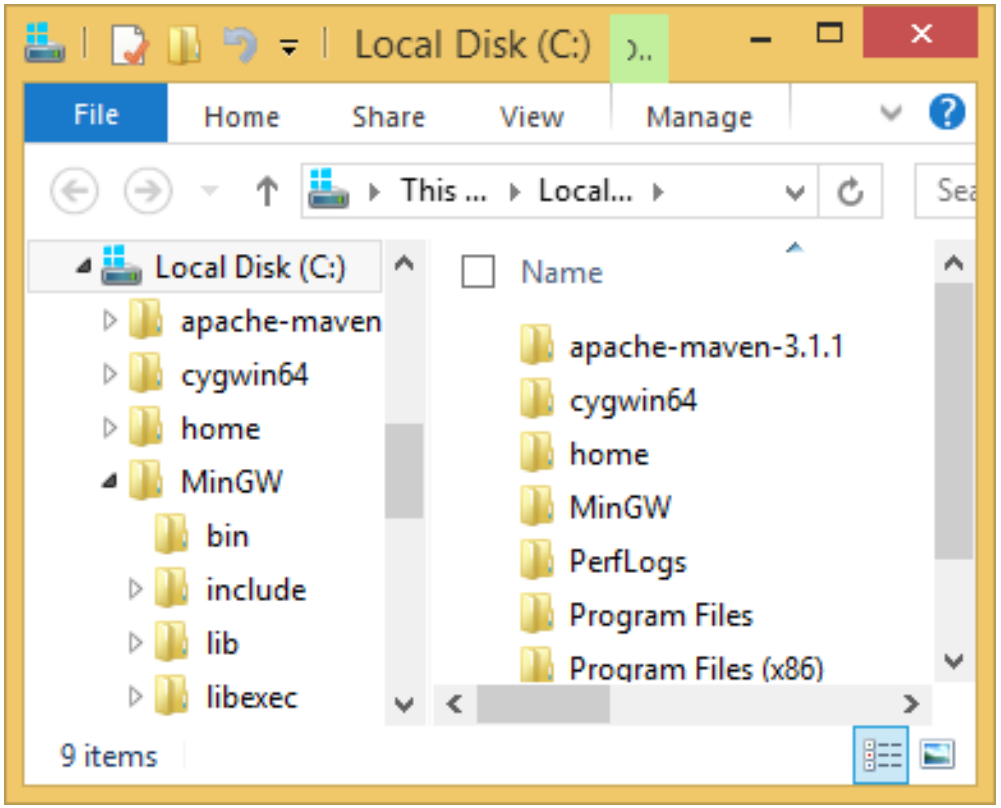
\system32

\drivers

\etc

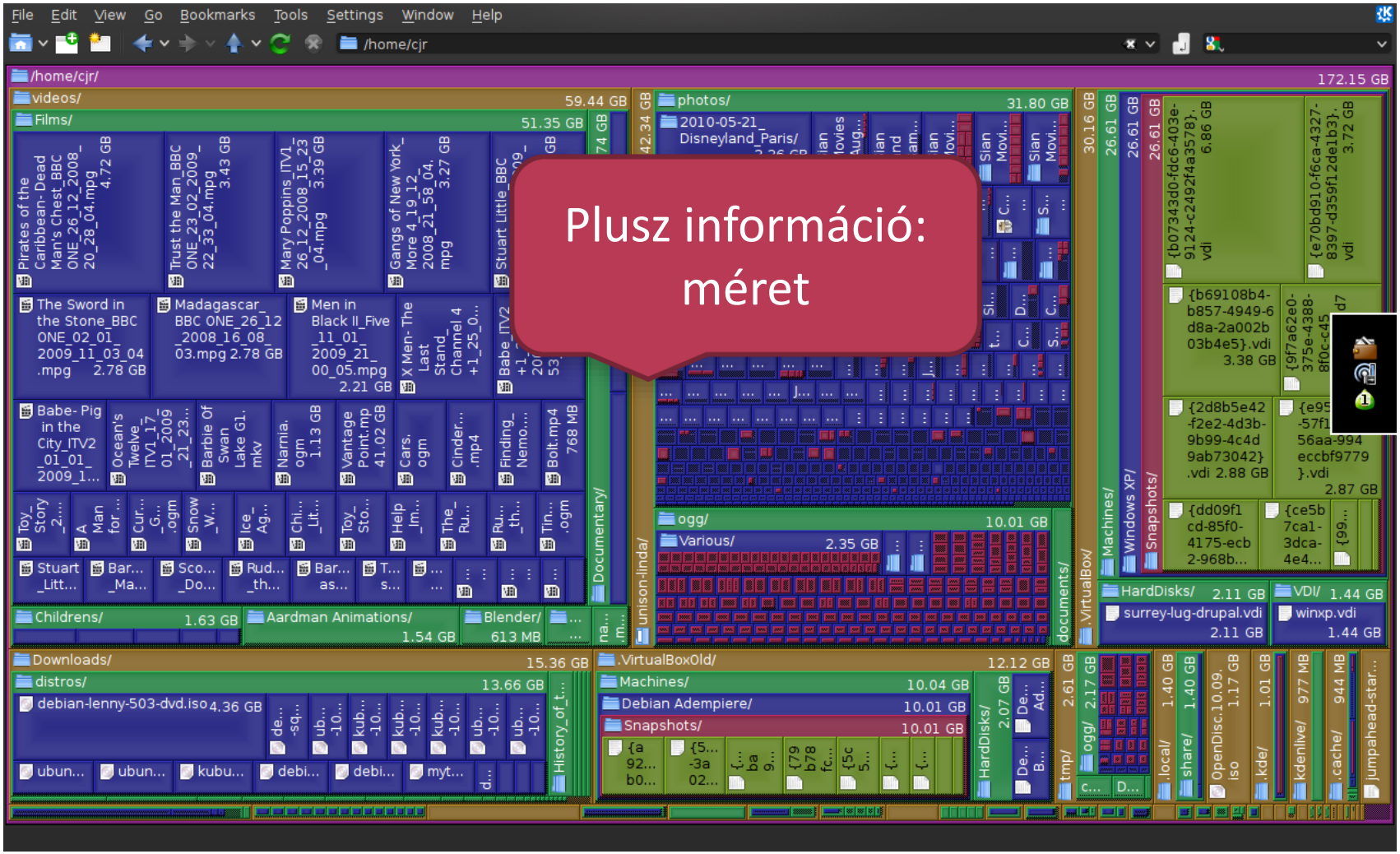
\Users

…



Fastruktúra – bennfoglaló ábrázolás

- KDE Konqueror



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
1	364
4	368
3	304

név	fénykard színe
Alderaan	
Coruscant	
Darth Vader	piros
Jabba	
Leia Organa	
Luke Skywalker	zöld
Mace Windu	lila
Tatooine	
Yoda	zöld

TÍPUSOK

Típus és tulajdonságok

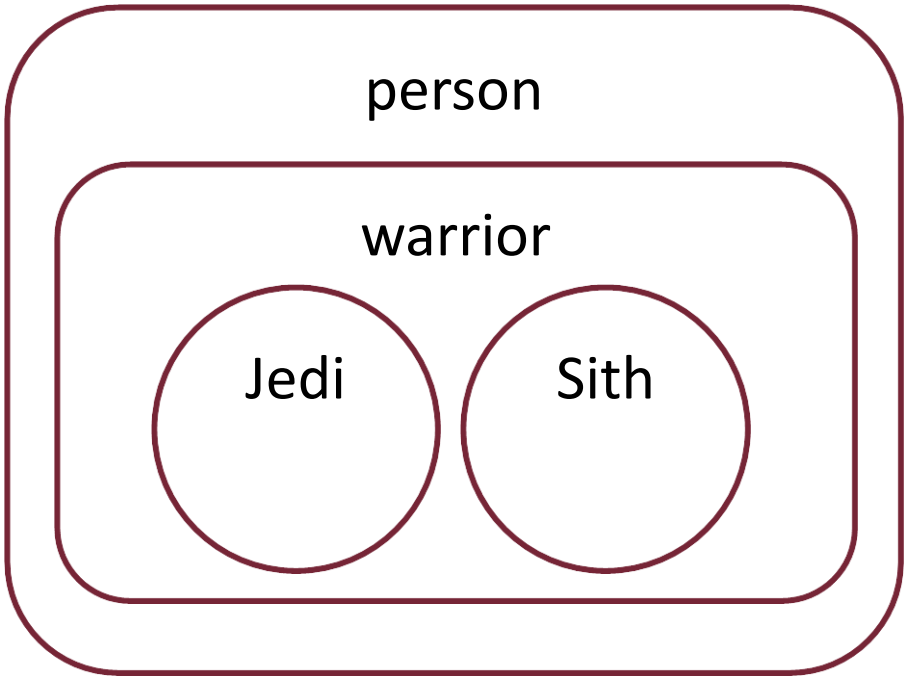
- A típus a címkében kifejezett tudás része
  - Tehát a *típus* egy kitüntetett attribútum
  - A többi jellemző: a *tulajdonságok*

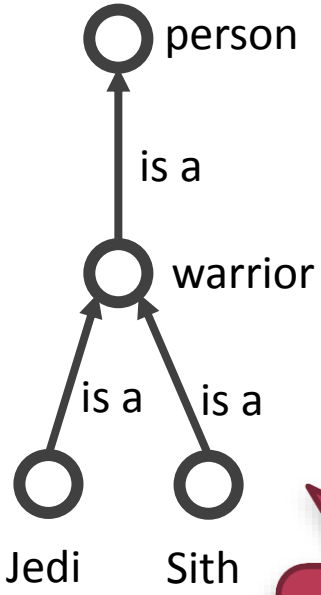
- Gyakori konvenció:
  - tulajdonságok akár változhatnak is
  - típus egy elemre tipikusan állandó

- Módosítás: Luke fénykardja: kék → zöld
  - Típus nem változik

Típusgráf

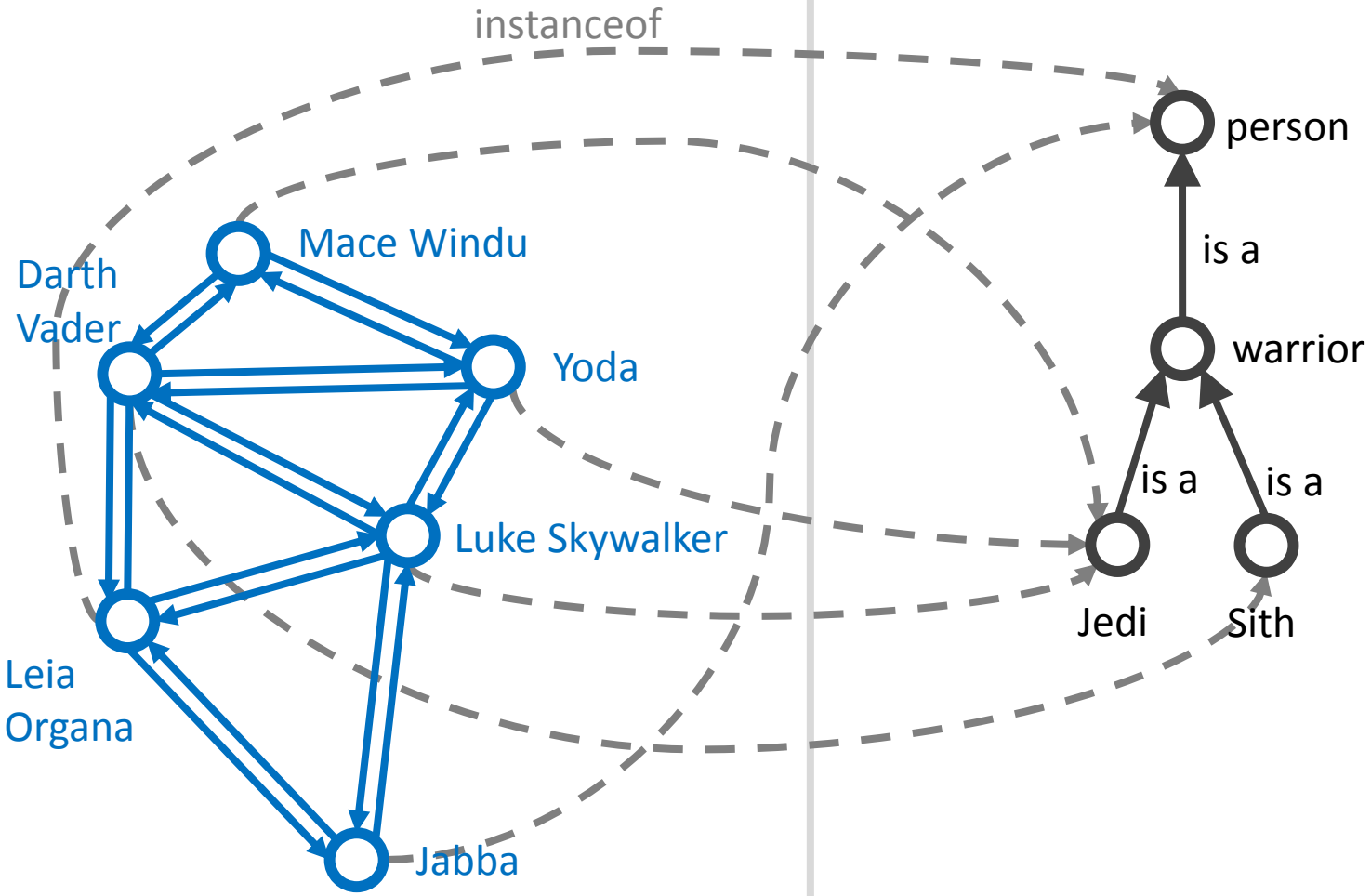
- Minden csomóponttípushoz egy típuscsomópont
- Minden éltípushoz egy típusél



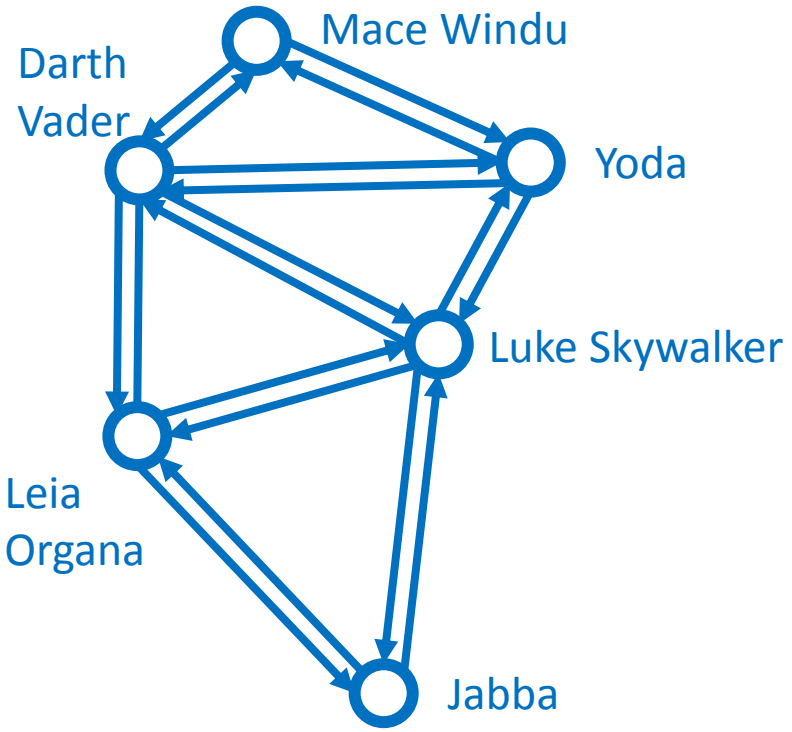


hierarchikus dekompozíció

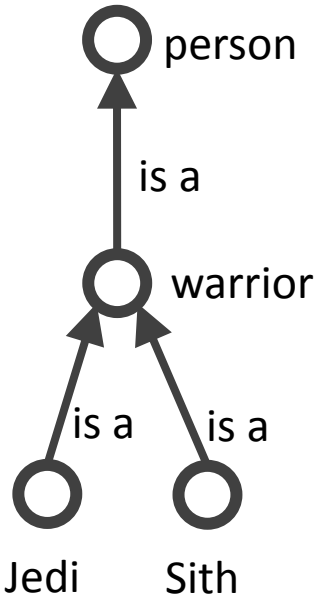
Típus-példány viszonyok ábrázolása



Típus-példány viszonyok ábrázolása

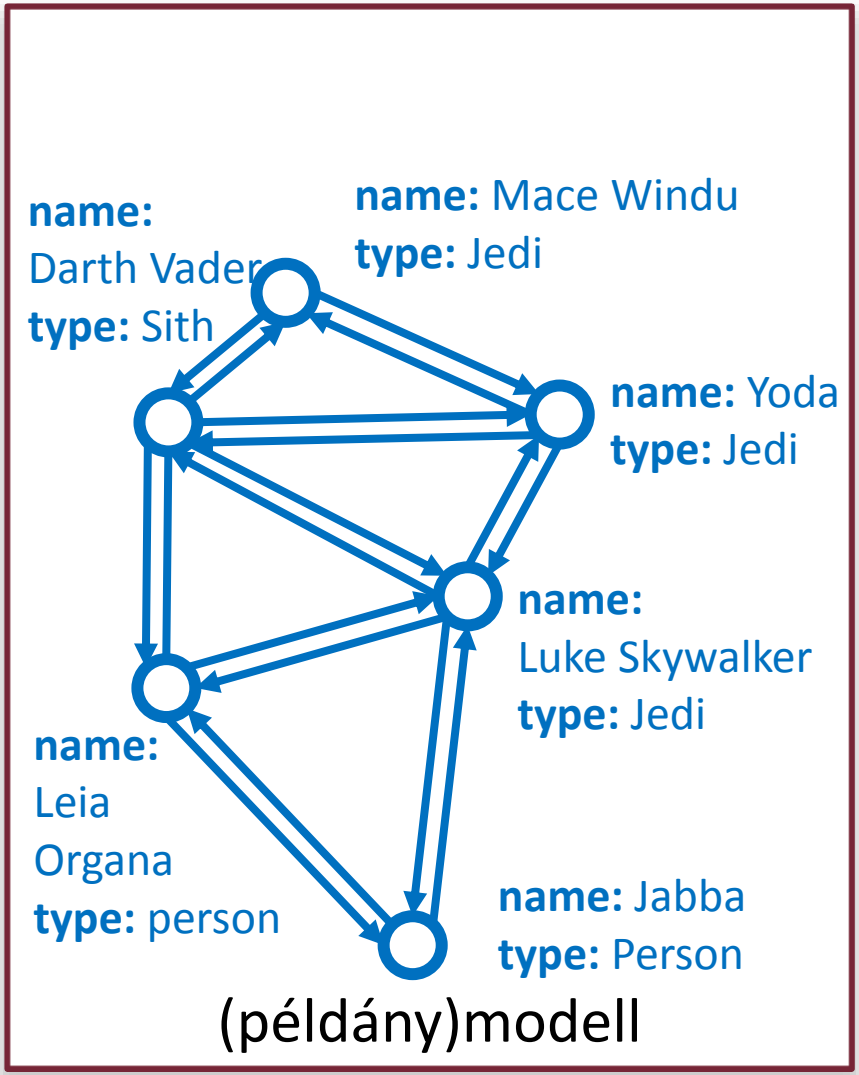


(példány)modell

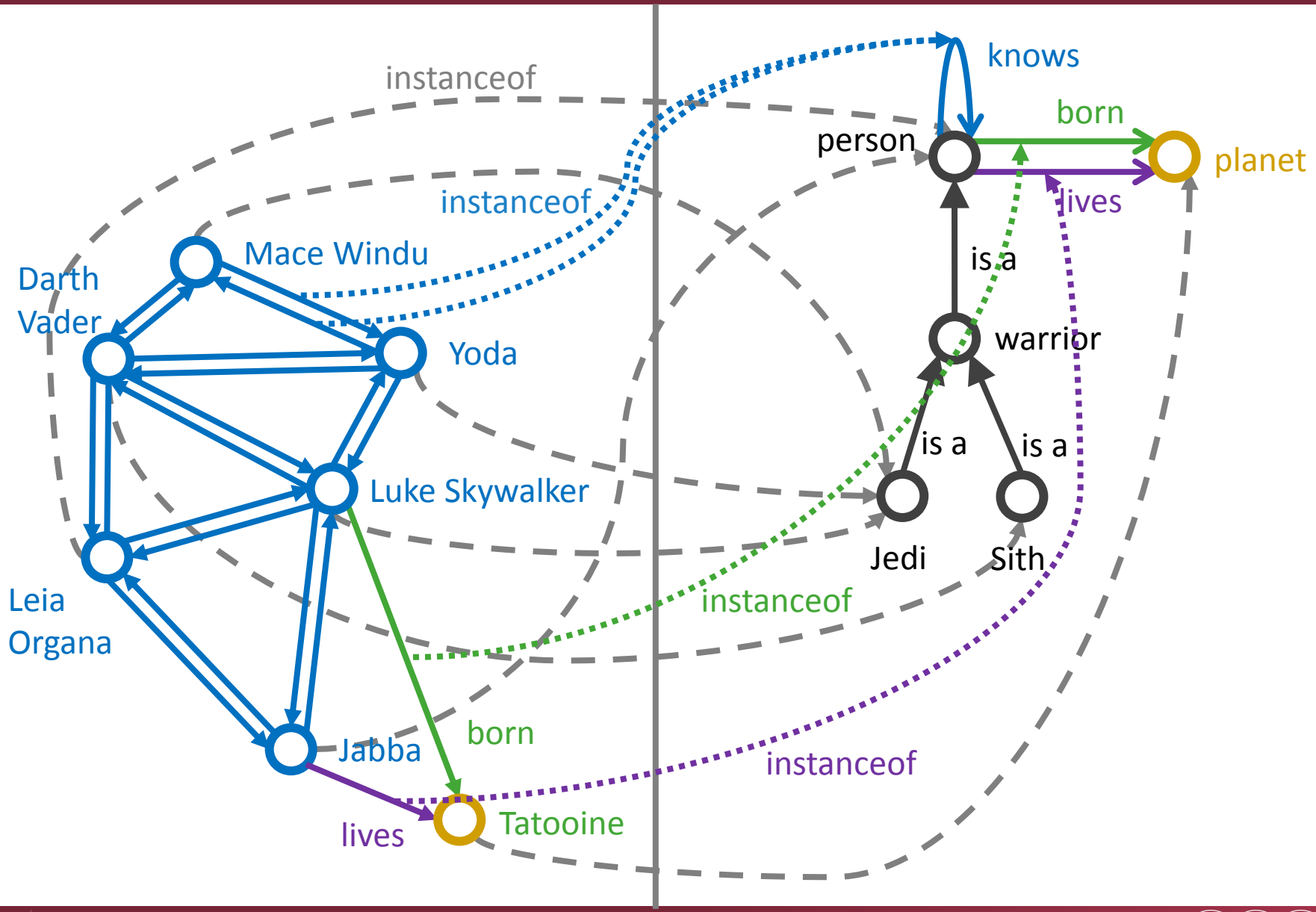


metamodell

Típusok ábrázolása jellemzőként



Különböző éltípusok

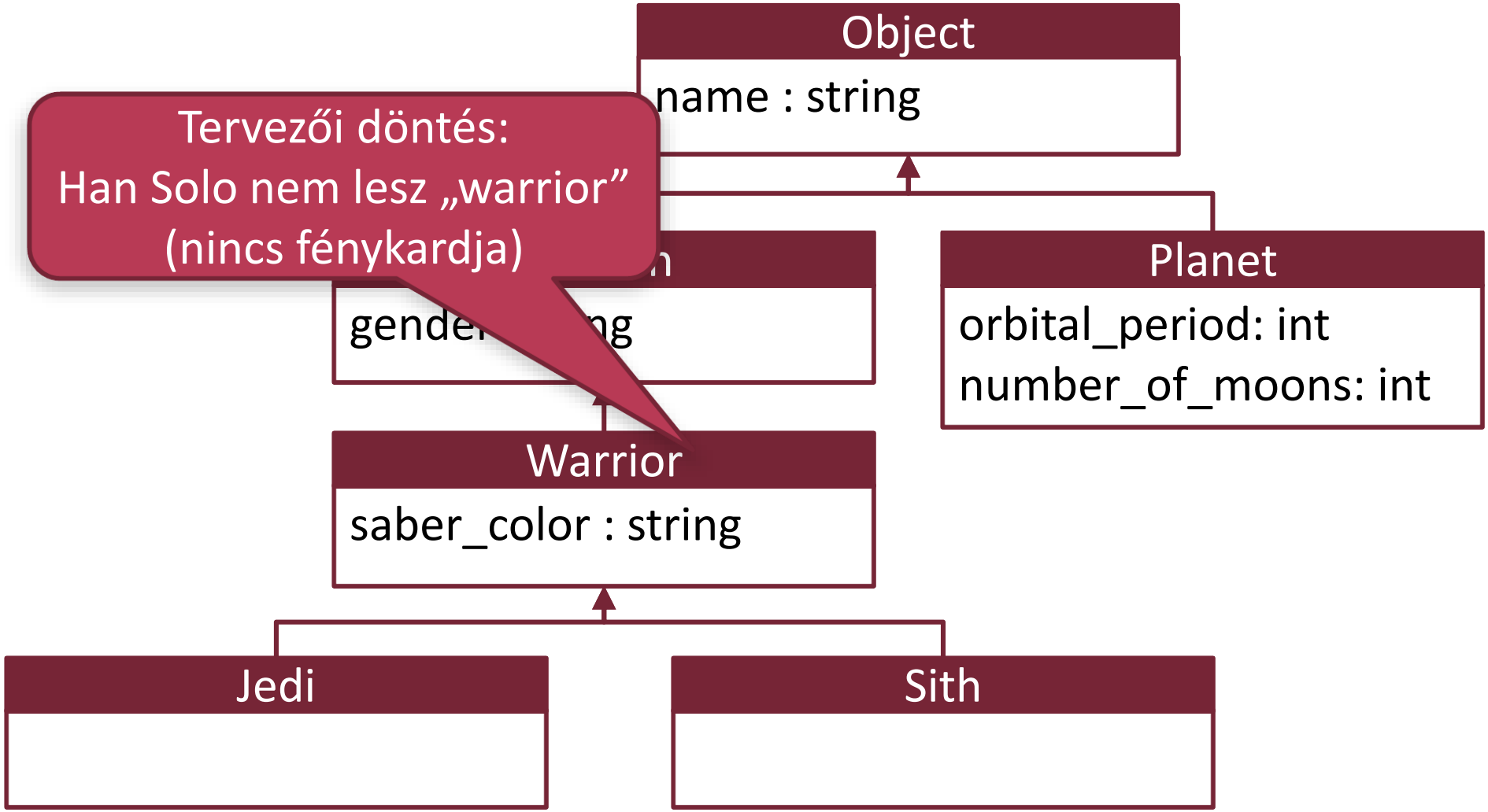


STRUKTURÁLIS MODELL REPREZENTÁLÁSA PROGRAMBAN

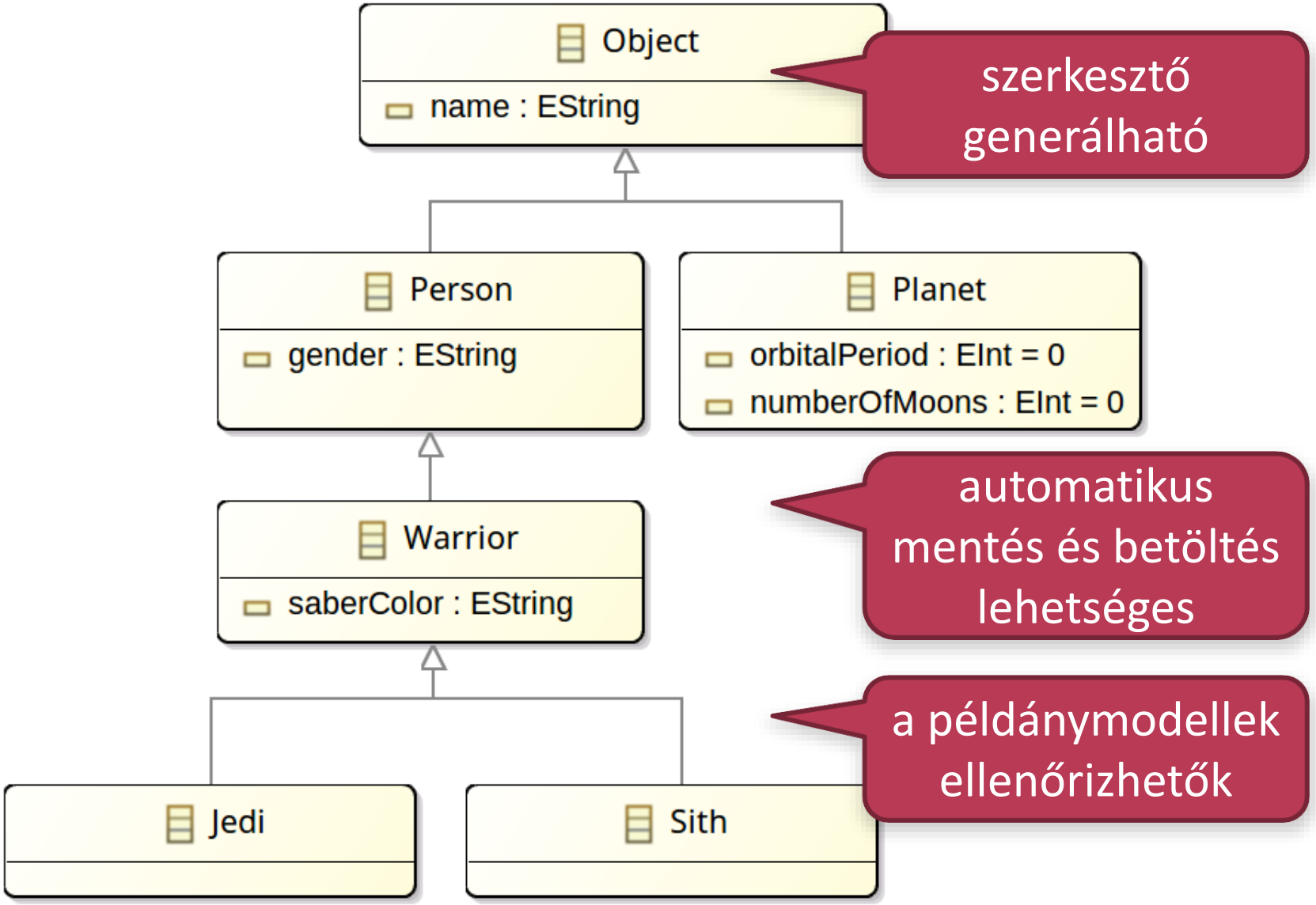
Programozási paradigmák

- **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


Object
name : string
Tervezői döntés:
Han Solo nem lesz „warrior”
(nincs fénykardja)
Planet
orbital_period: int
number_of_moons: int
Warrior
saber_color : string
Jedi
Sith

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 adatcserére

XML példa: időjárás webszolgáltatás

```
▼<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”