Folyamatmodellezés (BPMN), adatfolyamhálók

Rendszermodellezés 2015.

Budapesti Műszaki és Gazdaságtudományi Egyetem
Hibatűrő Rendszerek Kutatócsoport

DEMÓ

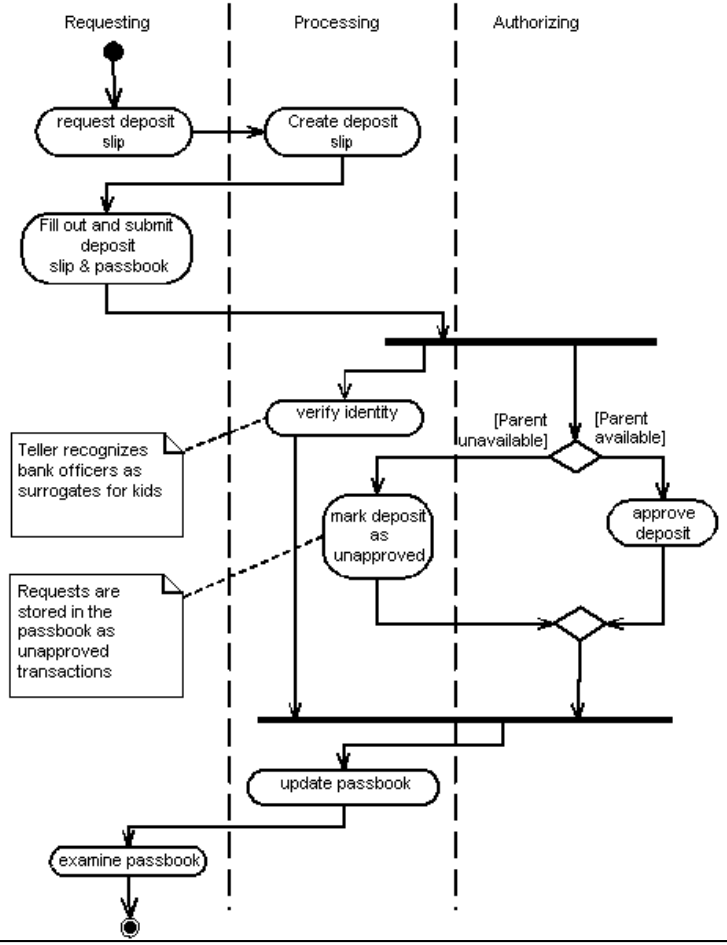
Futtatható üzleti folyamat (Bonita): alkalmazásbolt

https://inf.mit.bme.hu/edu/courses/materials/szolg%C3%A1ltat%C3%A1sintegr%C3%A1ci%C3%B3/2014-tavasz/1-gyakorlat-bpmn-bonita

ÜZLETI FOLYAMAT MODELLEK A GYAKORLATBAN

UML Activity Diagram

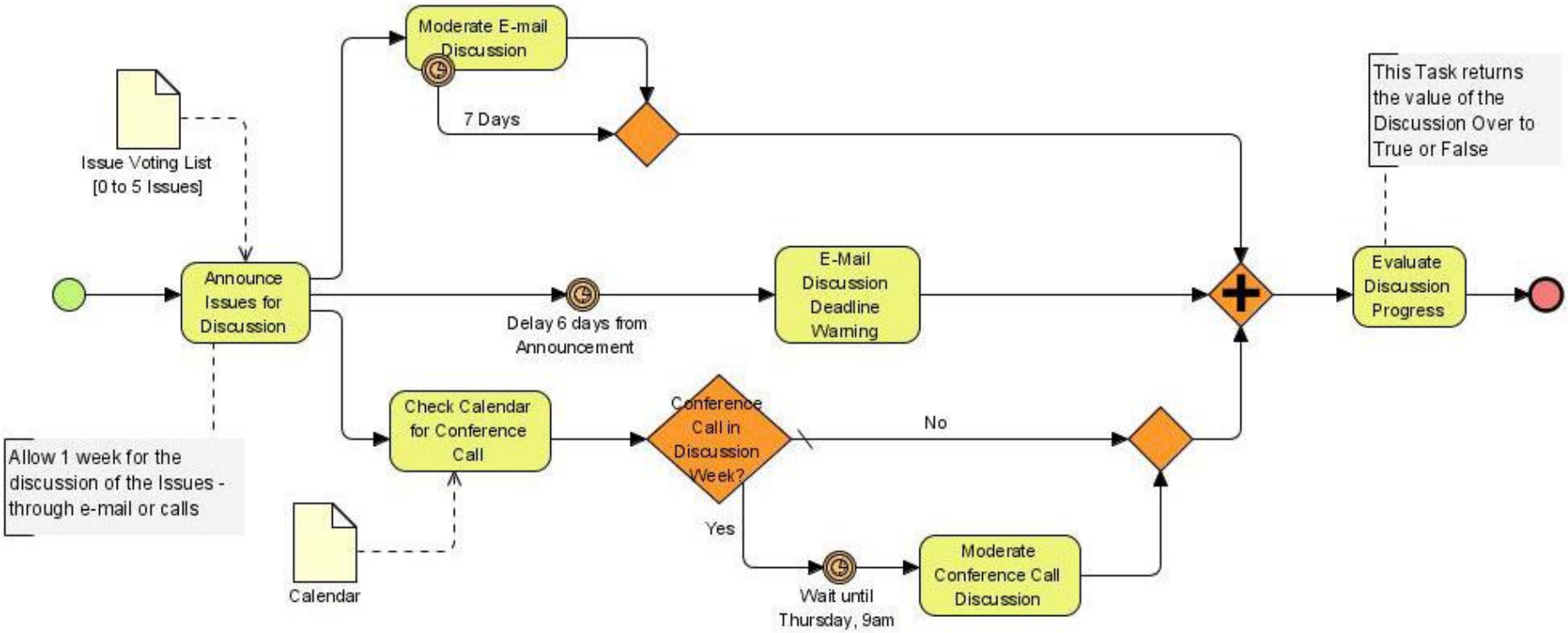
- Szabványosított jelölés, kiterjesztésekkel
  - Részletesen ld. SzoftTech, 3. félév



Business Process Modeling Notation (BPMN)

- Business Process Management Initiative (BPMI)
  - 2004. május: BPMN 1.0 specifikáció
  - 2011: végleges BPMN 2.0
- Célok
  - Közérthetőség
    - Felhasználó
  - Üzleti elemző
    - kezdeti folyamatterv
  - Műszaki fejlesztő
    - Implementáció
    - Belső modell automatikus generálás céljára
    - BPEL4WS
  - Üzleti végfelhasználó (monitorozás, menedzsment)

Példa BPMN


Moderate E-mail Discussion
7 Days
Issue Voting List
[0 to 5 Issues]
This Task returns the value of the Discussion Over to True or False
Announce Issues for Discussion
Delay 6 days from Announcement
E-Mail Discussion Deadline Warning
Evaluate Discussion Progress
Check Calendar for Conference Call
Conference Call in Discussion Week?
No
Allow 1 week for the discussion of the Issues - through e-mail or calls
Yes
Calendar
Wait until Thursday, 9am
Moderate Conference Call Discussion

Adatfolyam

Esemény

Állapotváltozás
Ok-hatás
Eseménytípusok:
Start, Intermediate, End

Tevékenység

Atomi/összetett
Taszk/alfolyamat

Átjáró

Szekvencia
konvergencia/divergencia
AND, OR, XOR, …

Összeköttetés

Szekvencia	Tevékenységek sorrendje a folyamatban	⟶
Üzenet	Két független folyamat részvevő közötti információcsere	- - - - - - →
Asszociáció	Adat, szöveg stb. hozzárendelés	..... →

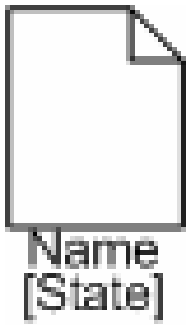
Tagolás

Pool — Résztvevő jelölése

Sáv — Tevékenységek csoportosítása

Artifact

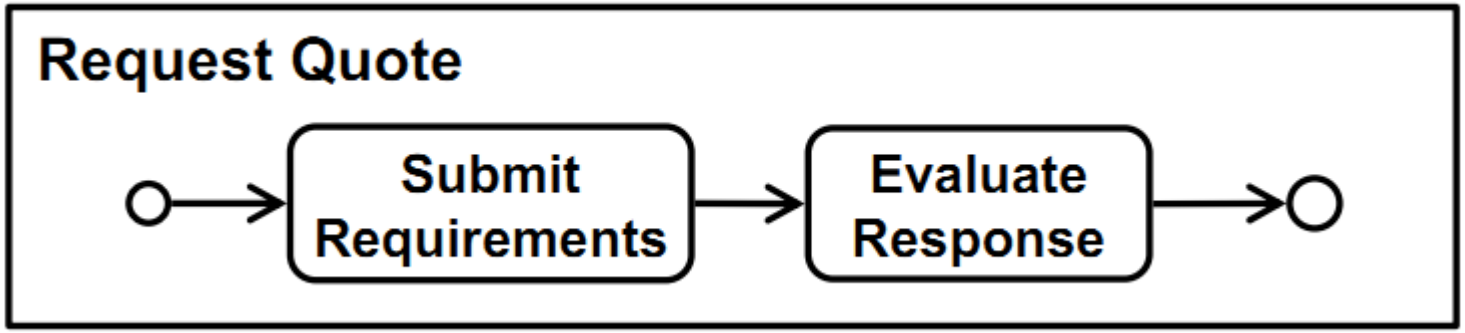
Adat objektum	Szimbolikus token
Csoport	Tevékenységek csoportosítása
Annotáció	Kiegészítő szöveges információ (komment)



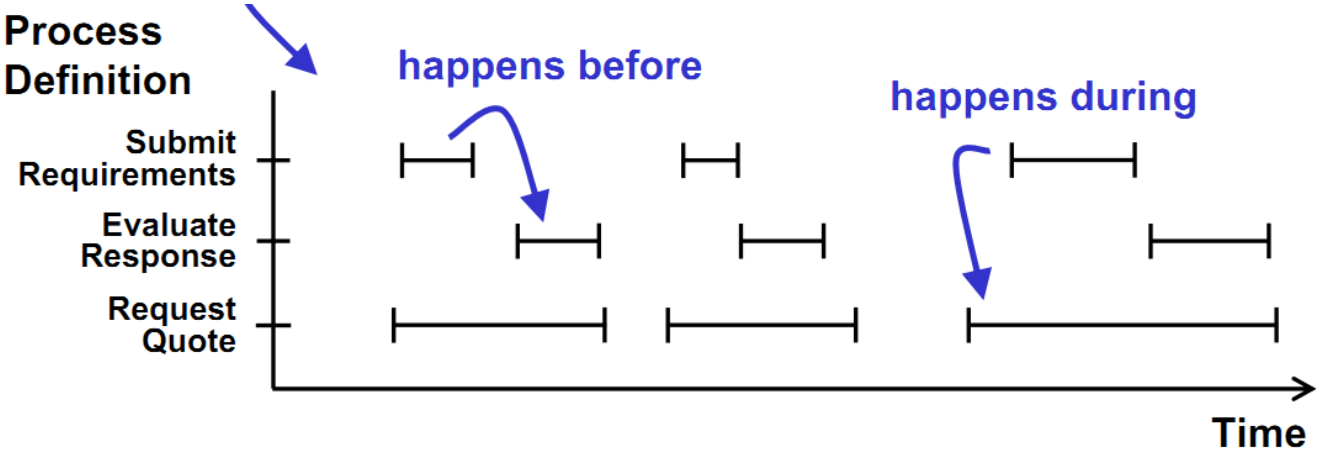
ÜZLETI FOLYAMATOK VÉGREHAJTÁSA

Folyamatok szemantikája

- Modellezés szempontjából

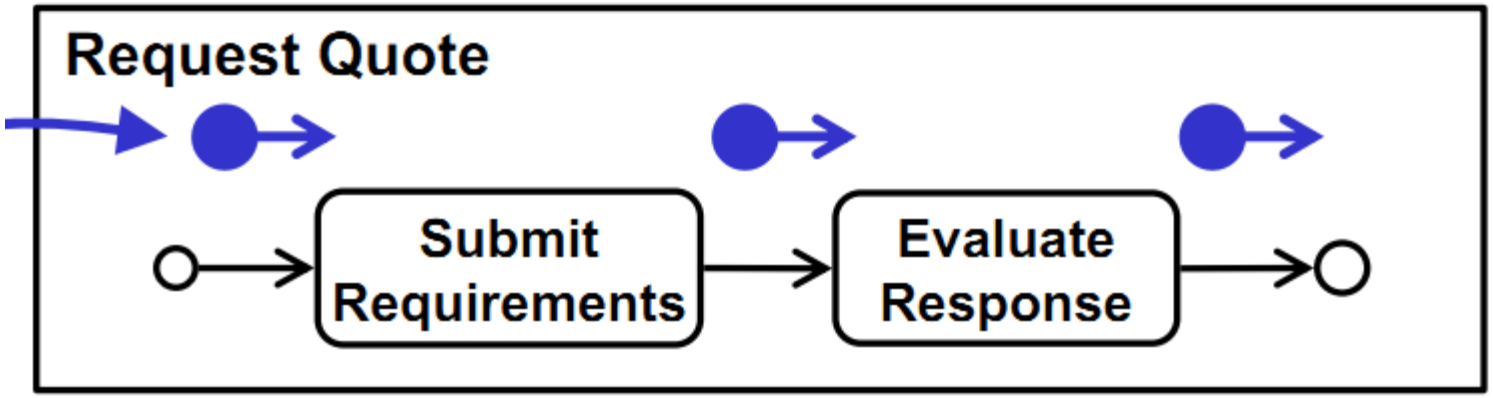


- Az elvárt működés

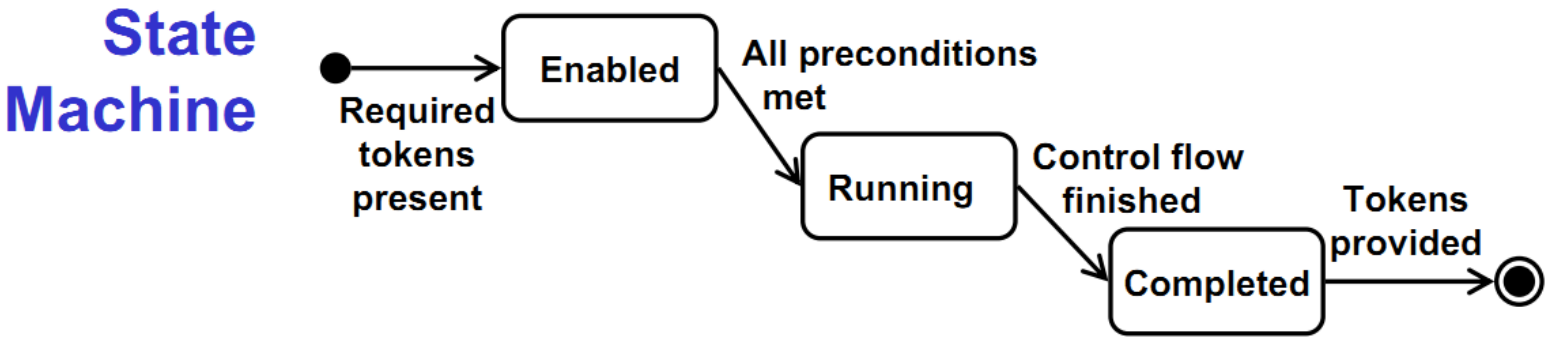


Folyamat végrehajtása

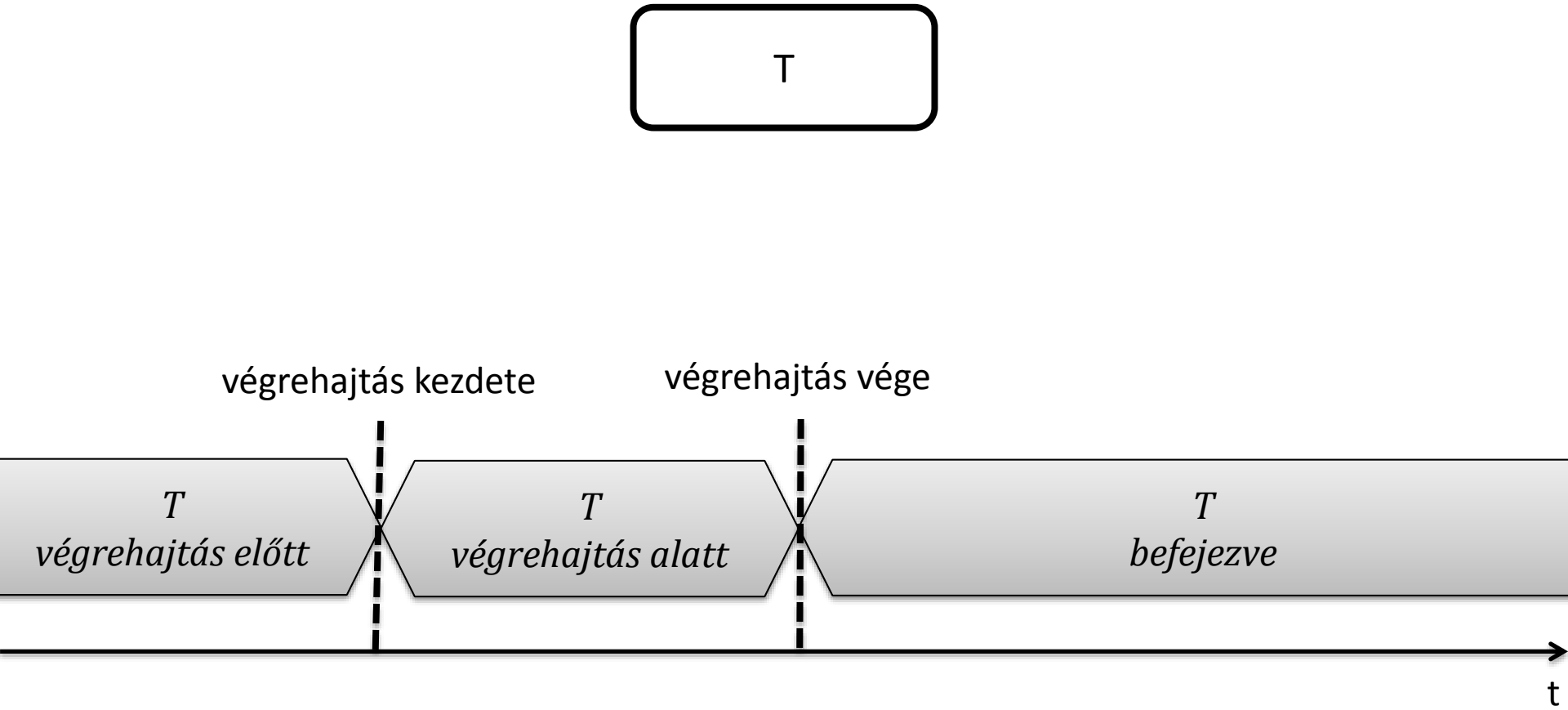
- Tokenáramlás



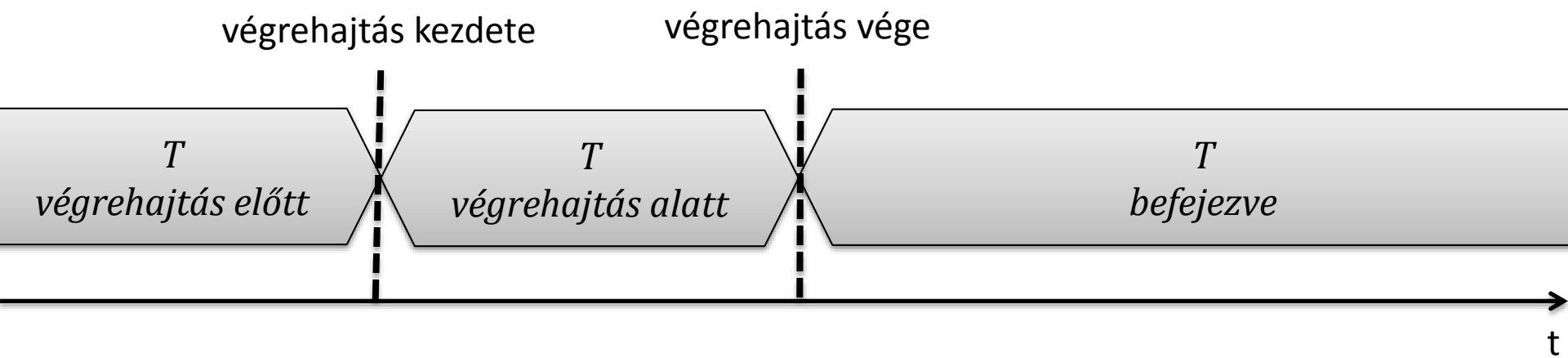
- A folyamat állapota



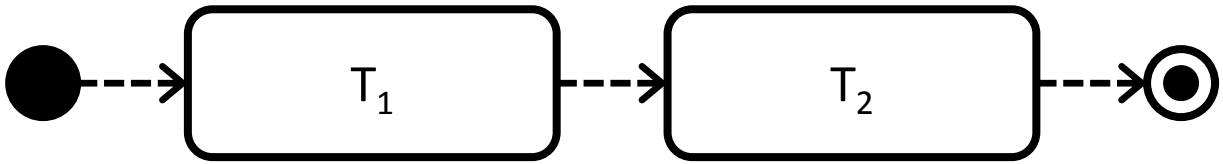
Elemi tevékenység állapotai

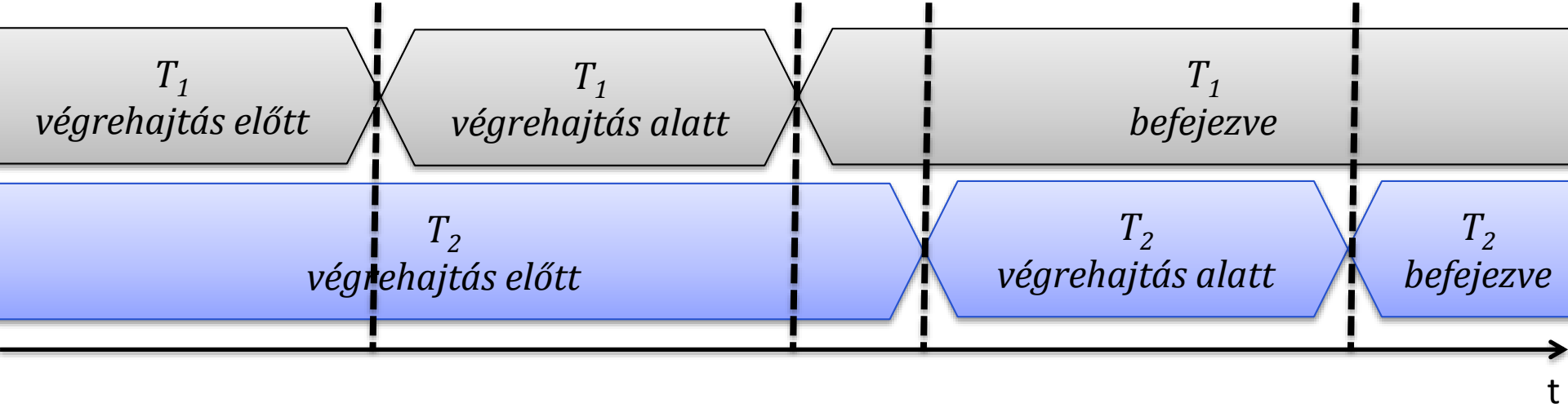


Elemi tevékenység állapotai



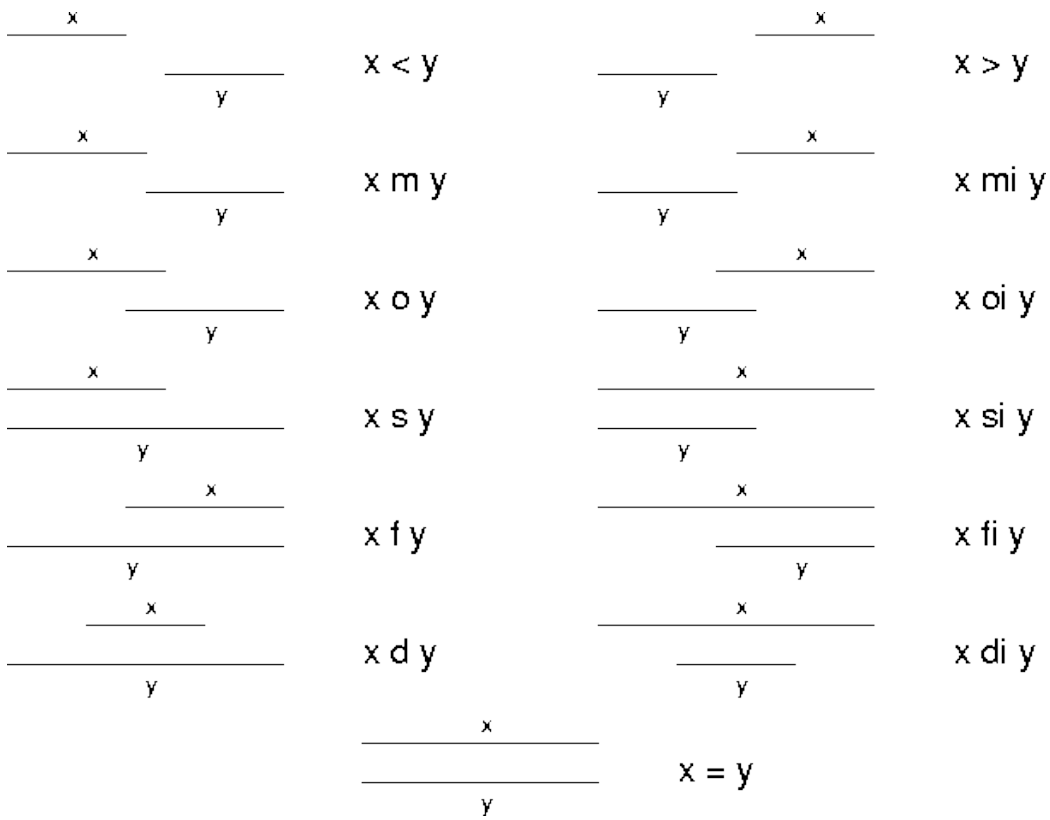
Folyamat állapotai





Háttér: matematikai modell

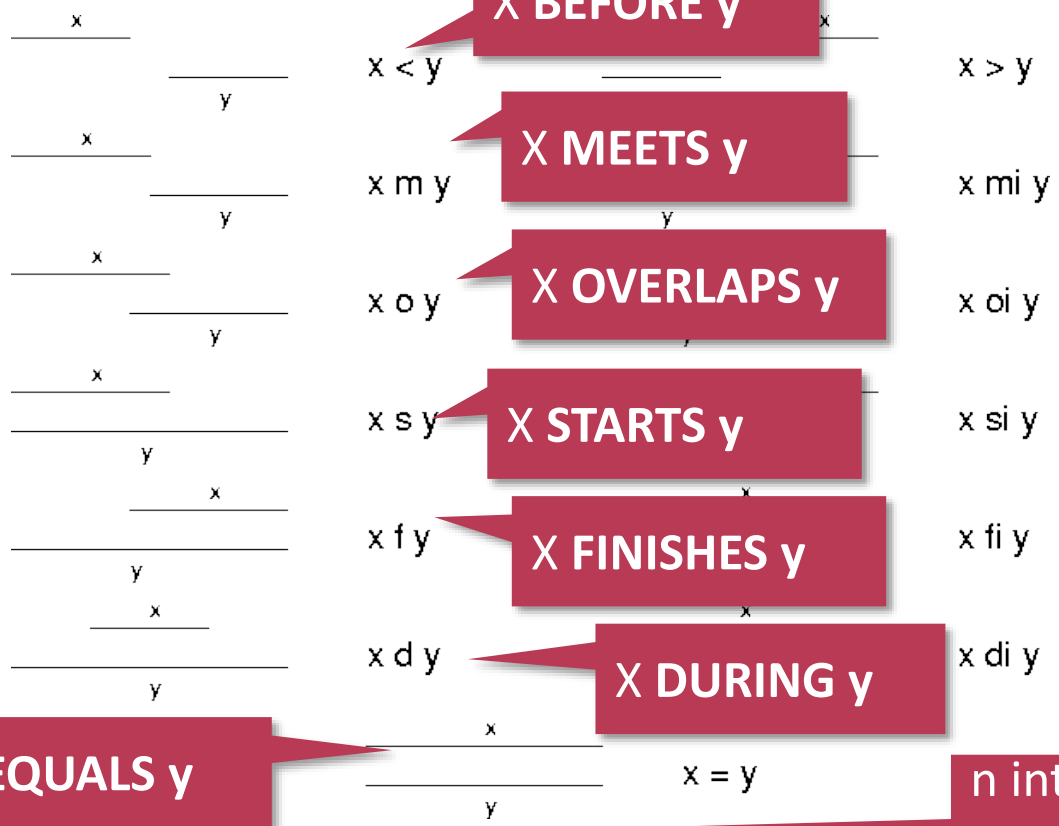
- Allen-féle intervallum logika (1983)
  - Pl. tesztelésnél használják, 13 (6 + 1 + 6) eset



James F. Allen: *Maintaining knowledge about temporal intervals*.
In: *Communications of the ACM*. 26 November 1983. ACM Press. pp. 832–843, ISSN 0001-0782

Háttér: matematikai modell

- Allen-féle intervallum logika (1983)
  - Pl. tesztelésnél ha[illegible]3 (6 + 1 + 6) eset



X BEFORE y

X MEETS y

X OVERLAPS y

X STARTS y

X FINISHES y

X DURING y

X EQUALS y

n intervallum:
1,1,13,409, 23917… eset

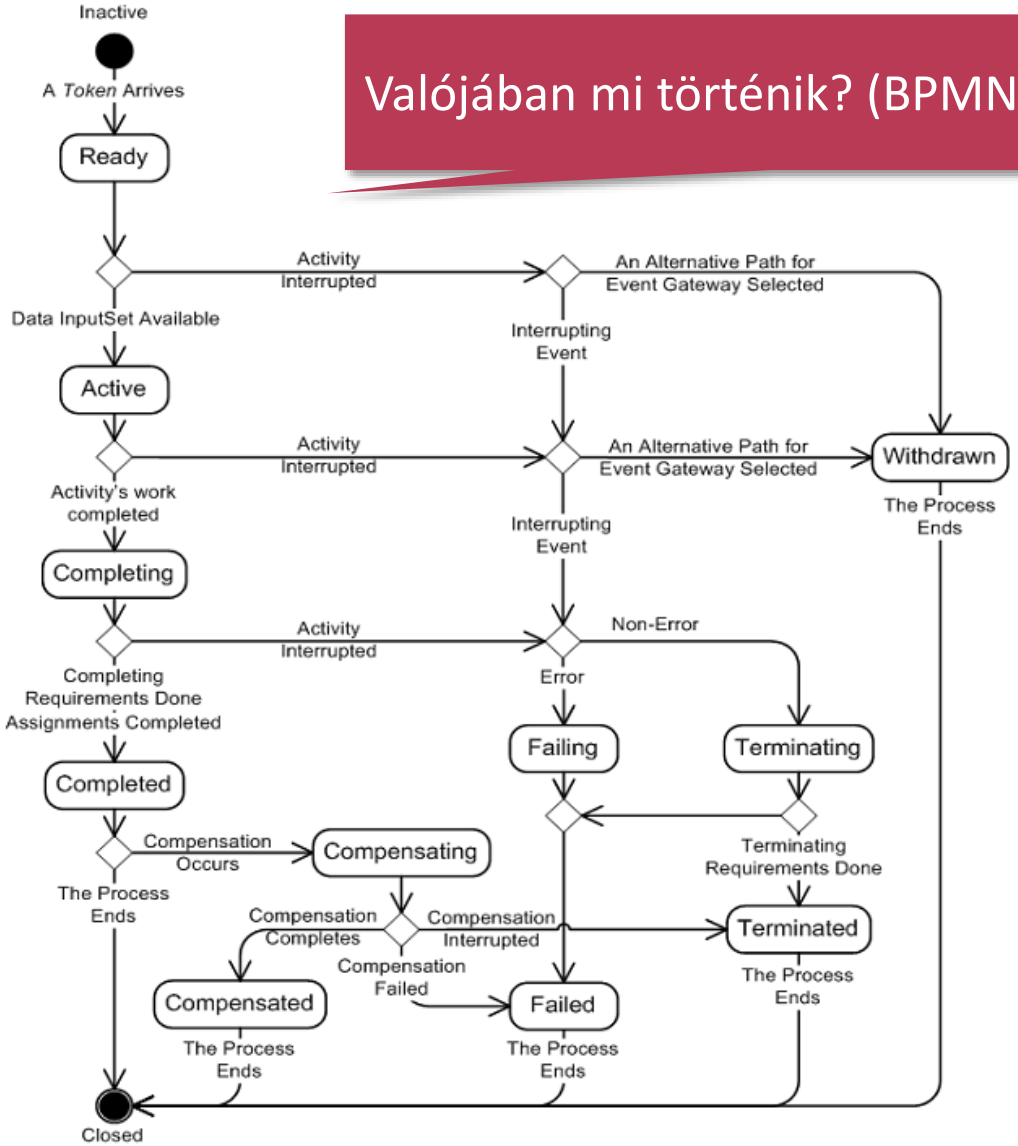
James F. Allen: *Maintaining knowledge about temporal intervals*.
In: *Communications of the ACM*. 26 November 1983. ACM Press. pp. 832–843, ISSN 0001-0782

Mit lehet ellenőrizni?

- Pl. a végrehajtás nem folyamat alapon történt
  - Megfelelt-e az elvárásoknak (sorrend, függetlenség)?
- Mi lehetett a “folyamat” a rendszer mögött?
  - Workflow mining
- Pl. a futtatókörnyezet megengedő
  - Lépés kihagyható
  - Ilyenkor is teljesülnek az elvárások?
- Eszköz: formális módszerek
  - Logika, Petri-hálók, modellellenőrzés, stb.

Elemi tevékenység finomított állapotgépe

Valójában mi történik? (BPMN szabvány)



Elemi tevékenység finomított állapotgépe

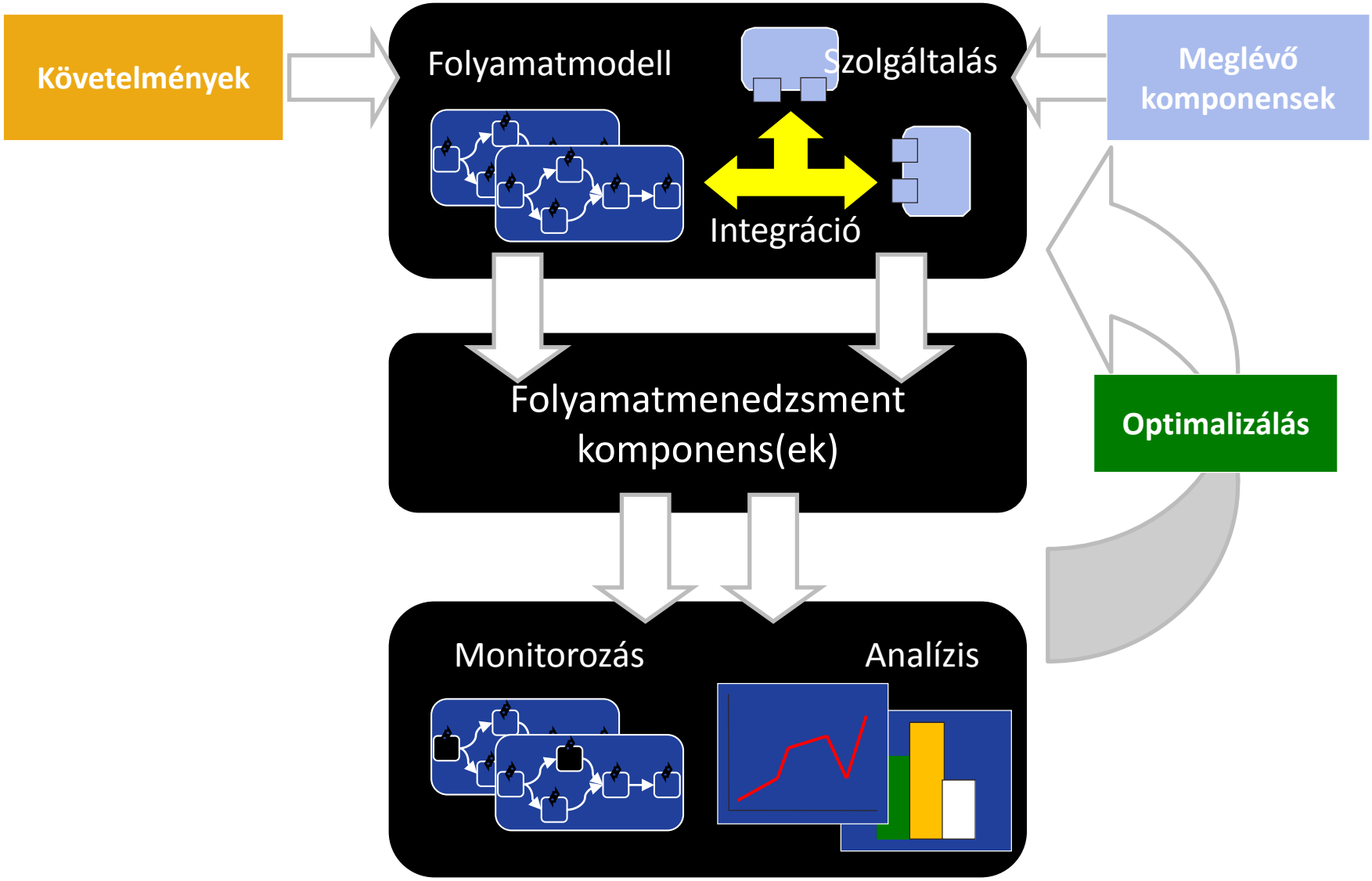
- Tevékenység megszakítható, visszavonható, hibázhat…

- Futtatókörnyezet felelőssége kezelni
- Szabvány írja le az állapotokat/átmeneteket
- Nem ugyanaz, mint a lépést végrehajtó erőforrás/alkalmazás állapota!
- Tervezési feladat: pl. mit jelent egy email esetén a “visszavonás”

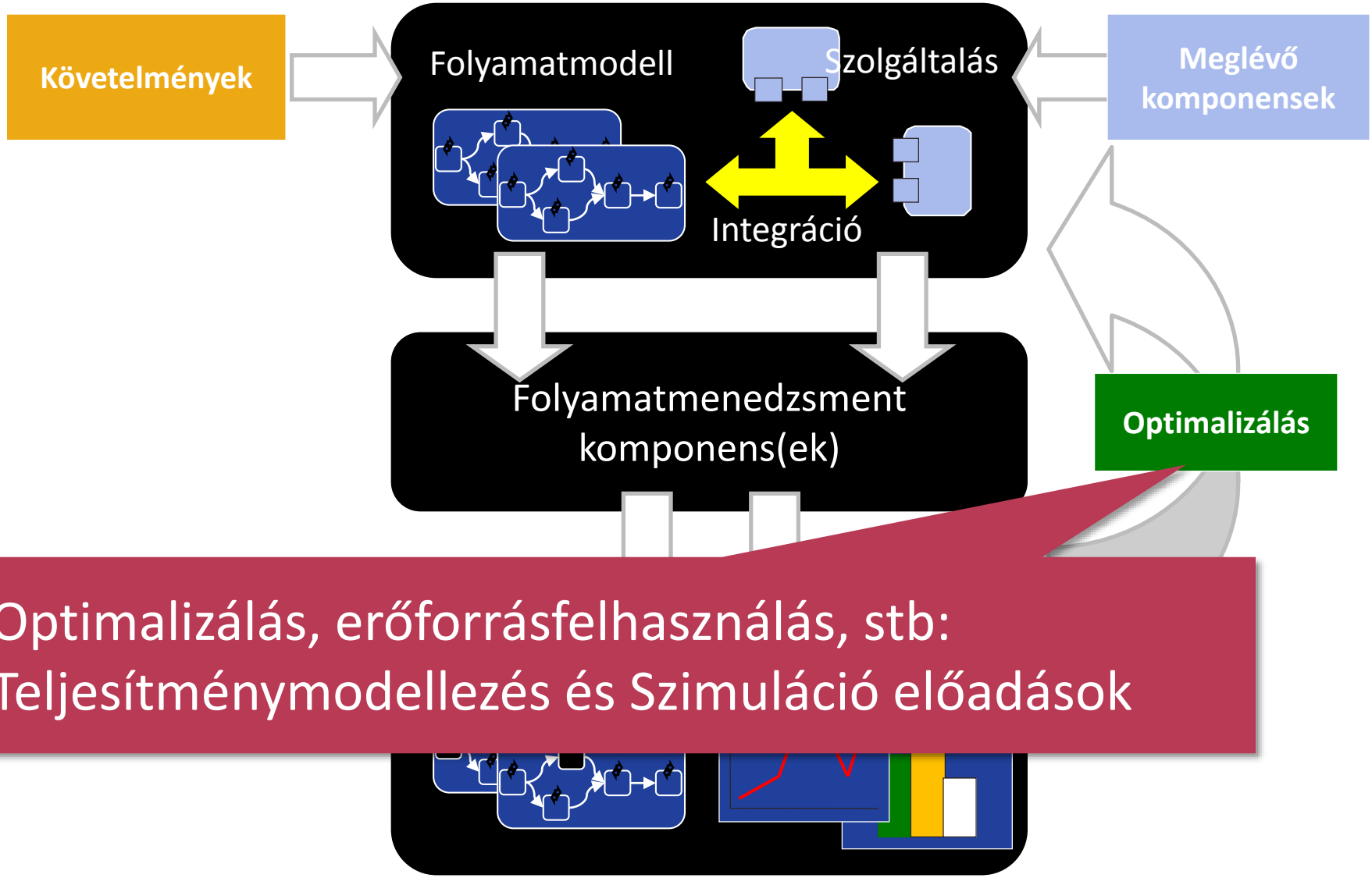
Futtatás: “workflow engine”

- Folyamatok életciklusának kezelése
  - Folyamat sablonok kezelése
  - Példányosítás, adatok kezelése
- Verziókezelés, online frissítése
- API beágyazható/csatolható elemeknek
  - REST, WS, EJB…
- Üzleti szabályok (döntések) kezelése
- Emberi lépések (human task)
  - Böngészőben megjeleníthető
  - Jogosultságok kezelése

Folyamatmenedzsment



Folyamatmenedzsment



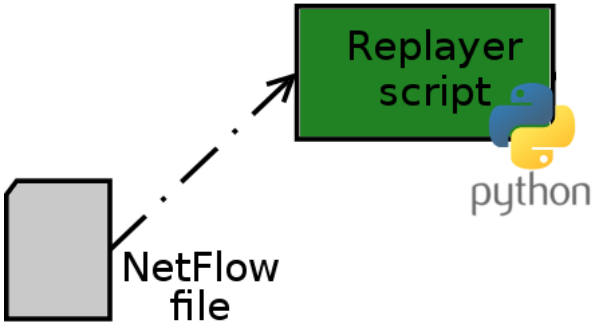
Optimalizálás, erőforrásfelhasználás, stb:
Teljesítménymodellezés és Szimuláció előadások

ESETTANULMÁNY: STORM

Adatfeldolgozás Apache Storm használatával

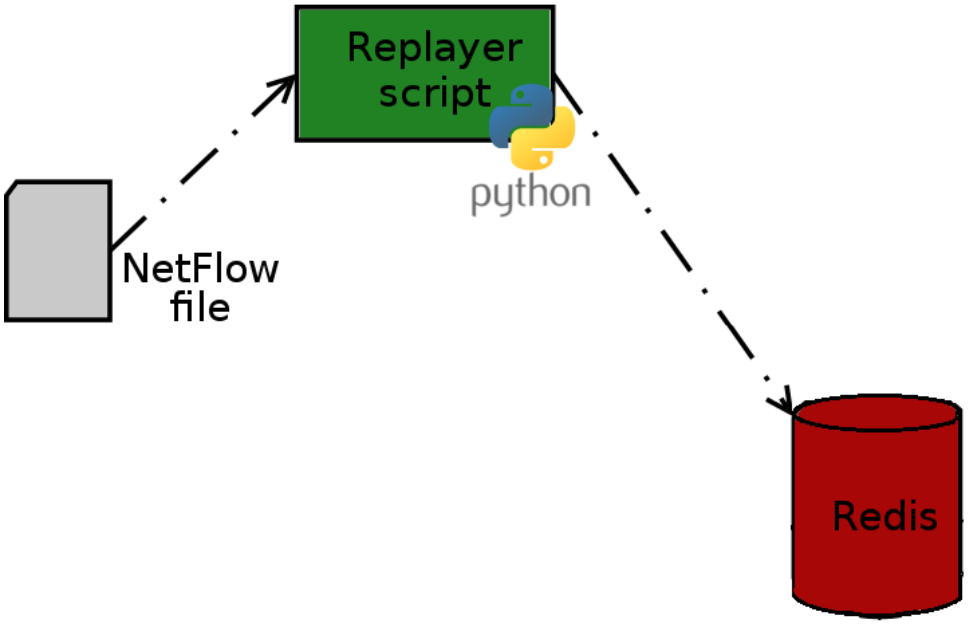
(Nádudvari Tamás: Nagyméretű adathalmazok elemzésének stream processing alapú támogatása)

Alkalmazás adatfolyam



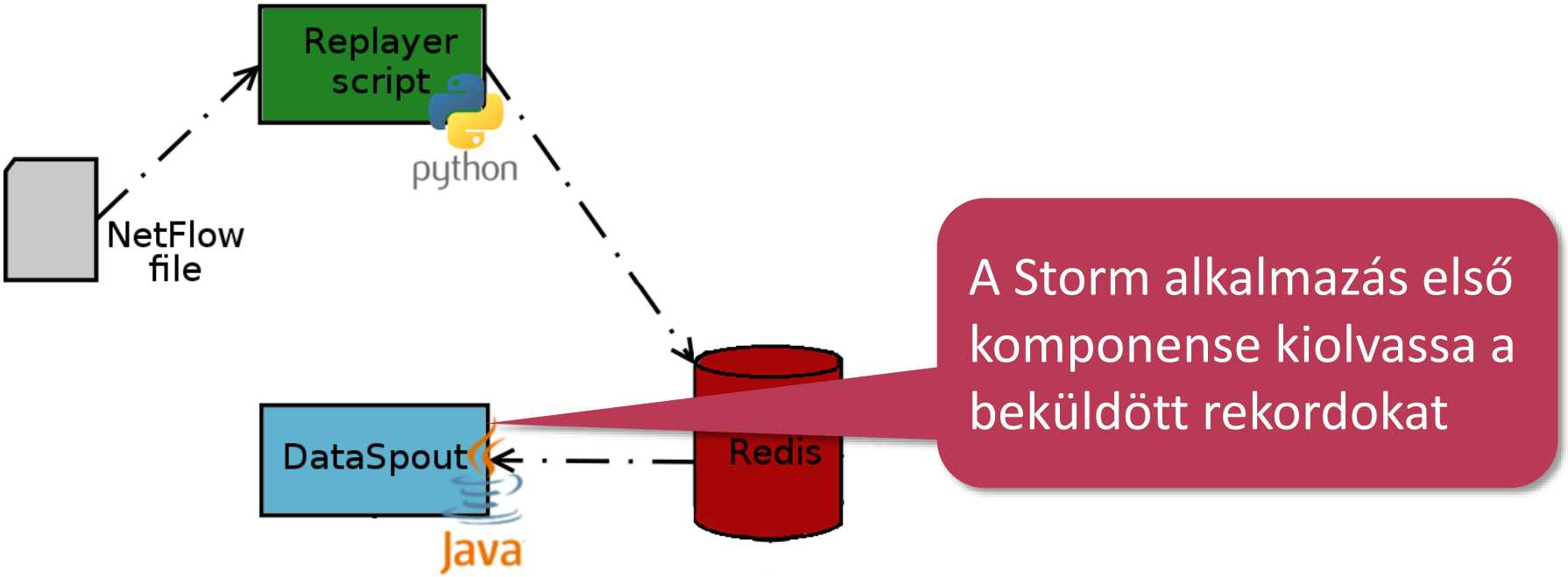
- A lementett hálózati adatokat tartalmazó rekordokat fájlból kiolvassuk
- Egy rekordban szerepel
  - a forrás és cél IP cím,
  - időpont
  - forgalmazott csomagszám
  - adatmennység

Alkalmazás adatfolyam

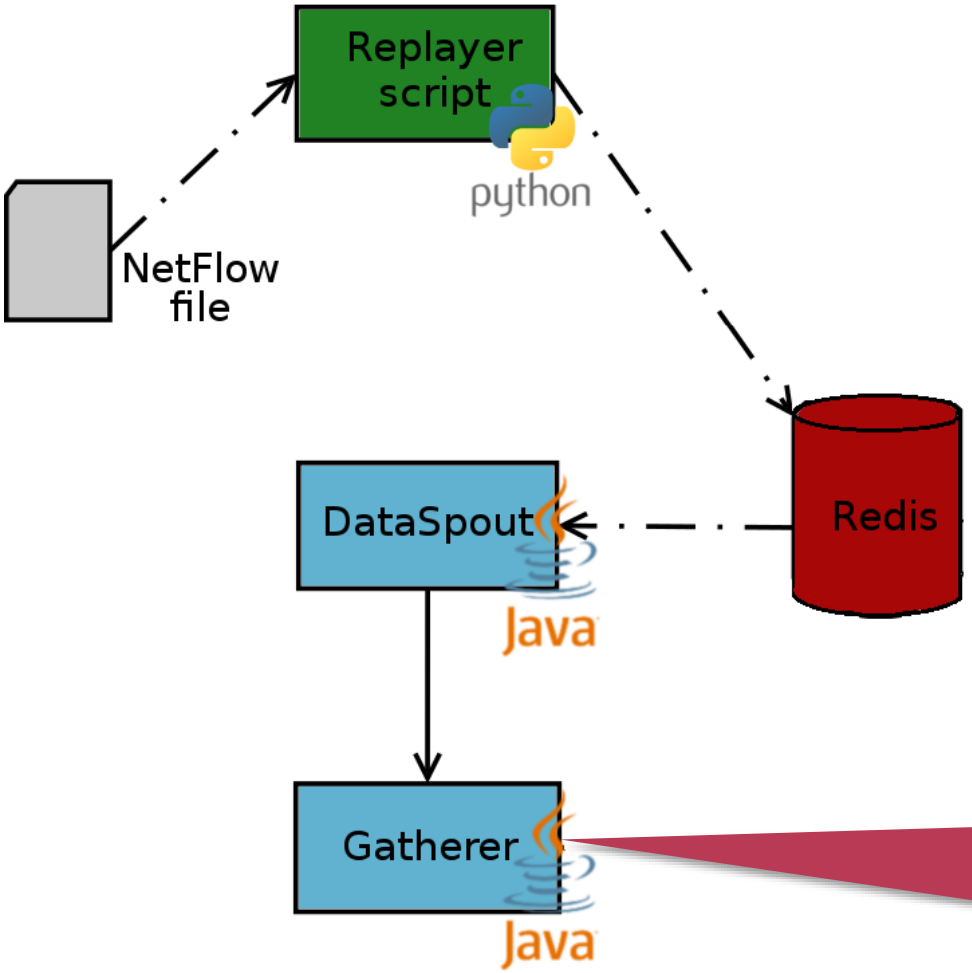


- A hálózati rekordokat egy adatbázisba küldjük

Alkalmazás adatfolyam

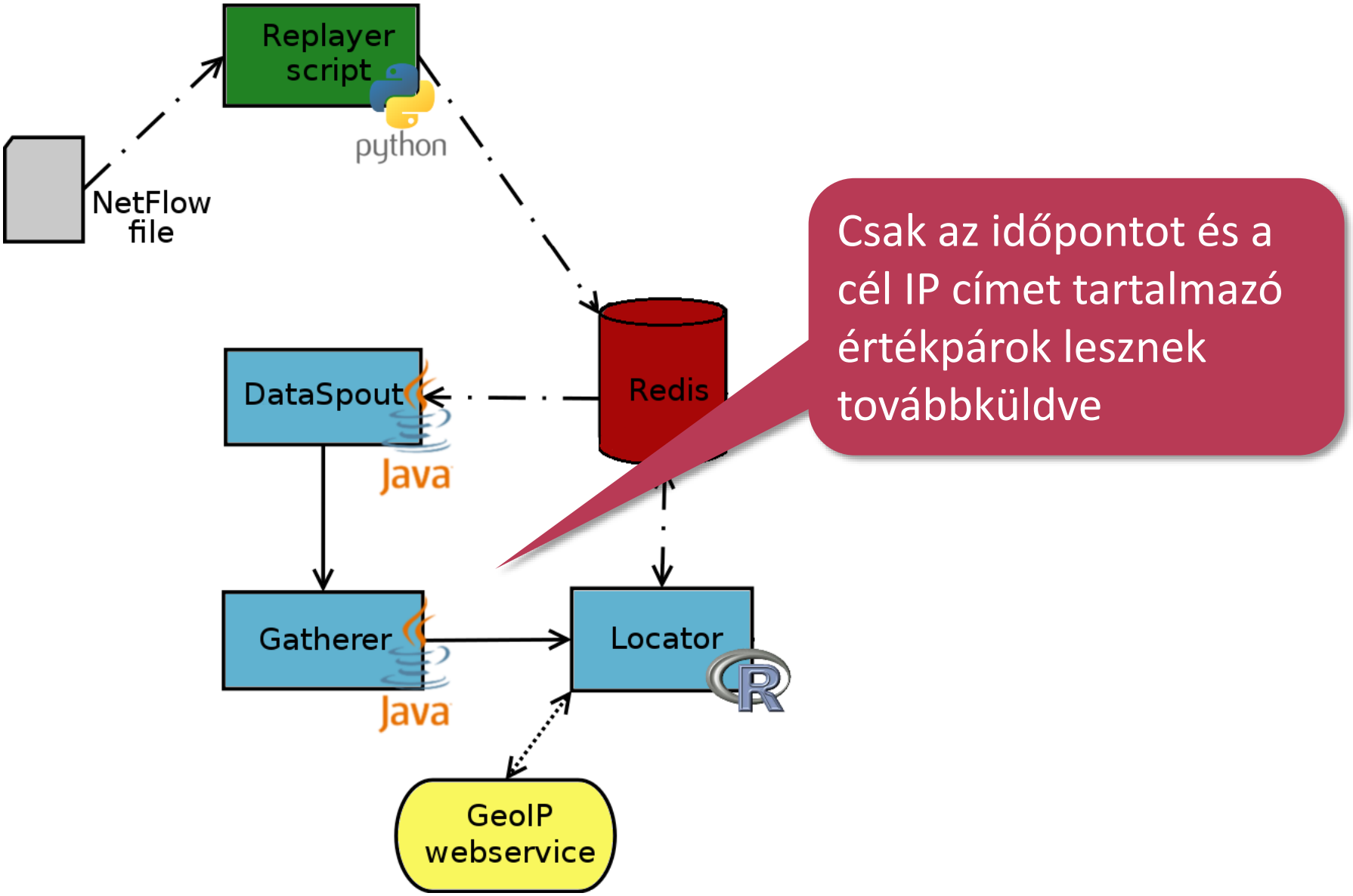


Alkalmazás adatfolyam

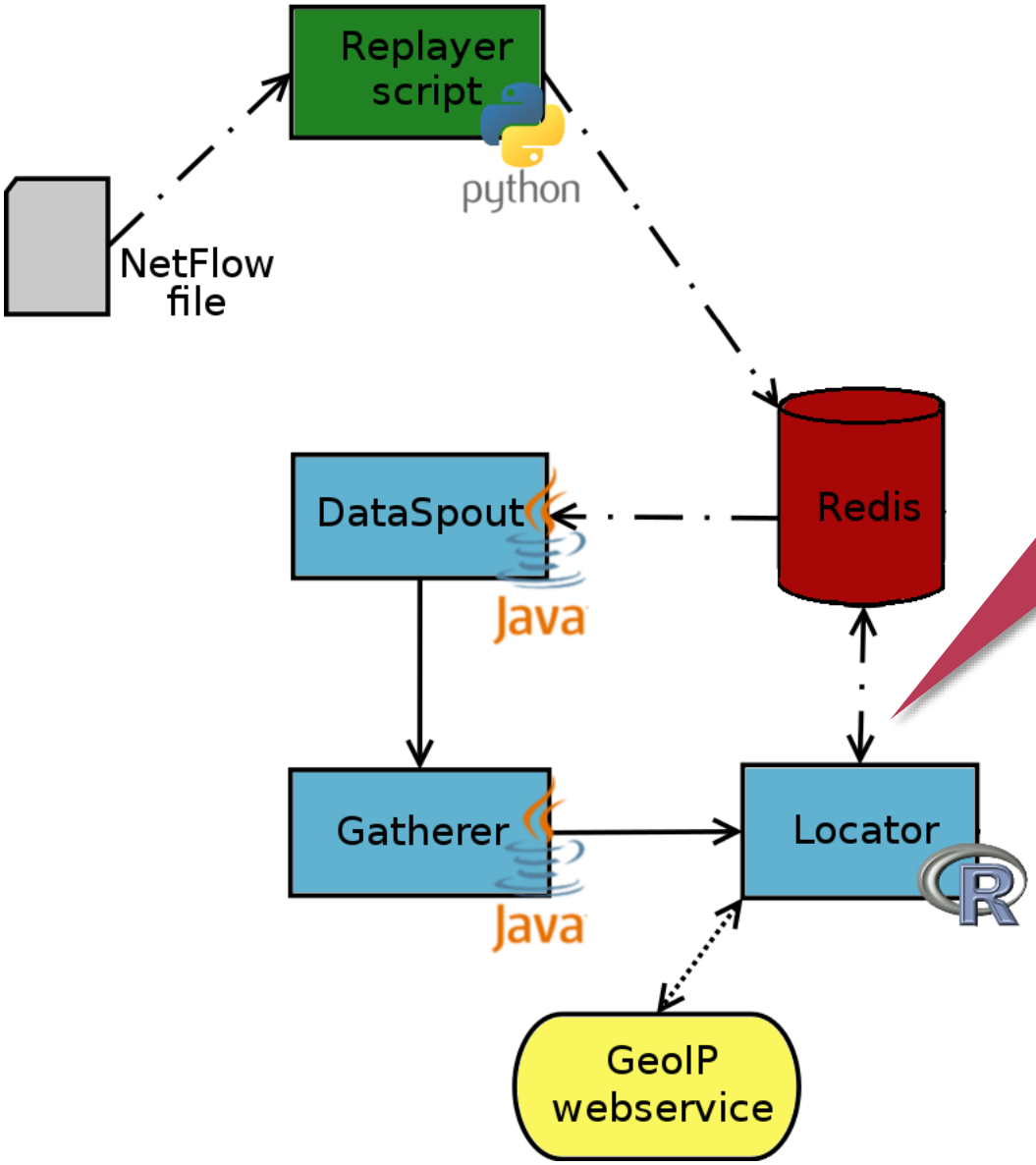


Az alkalmazás szempontjából lényegtelen adatokat levágja a rekordokból

Alkalmazás adatfolyam

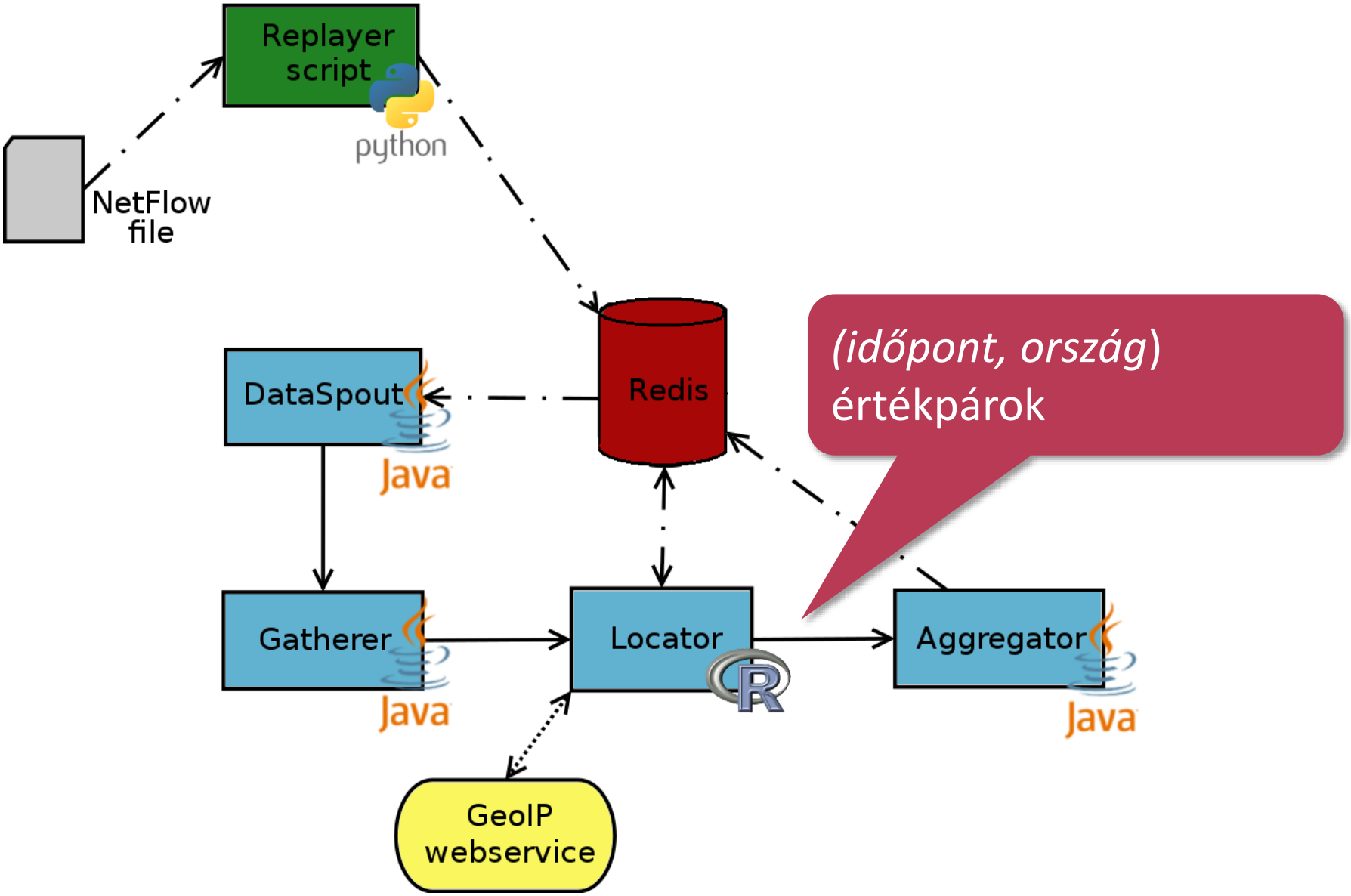


Alkalmazás adatfolyam

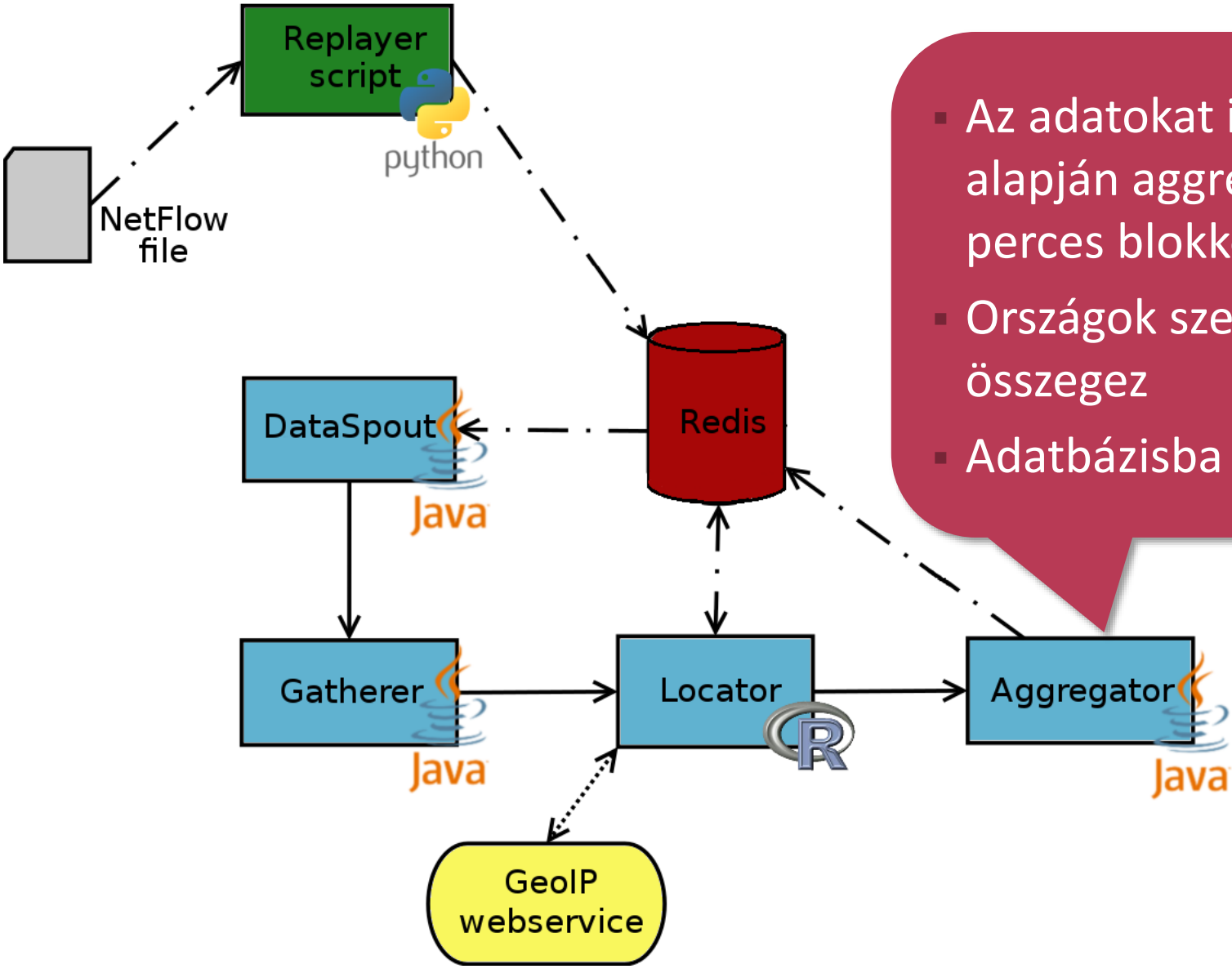


Egy külső web szolgáltatás segítségével az IP címekhez megkeresi a hozzátartozó országot

Alkalmazás adatfolyam

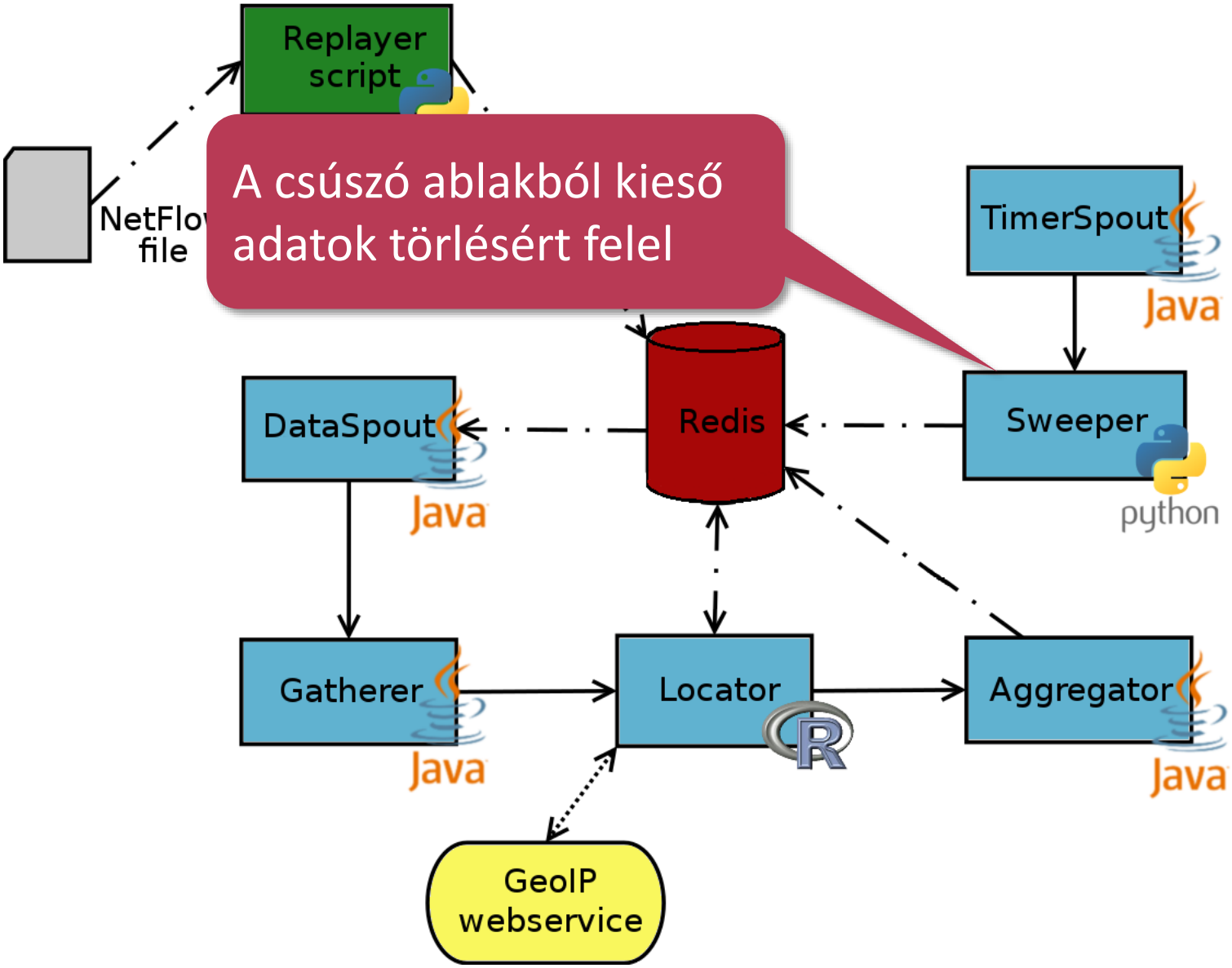


Alkalmazás adatfolyam

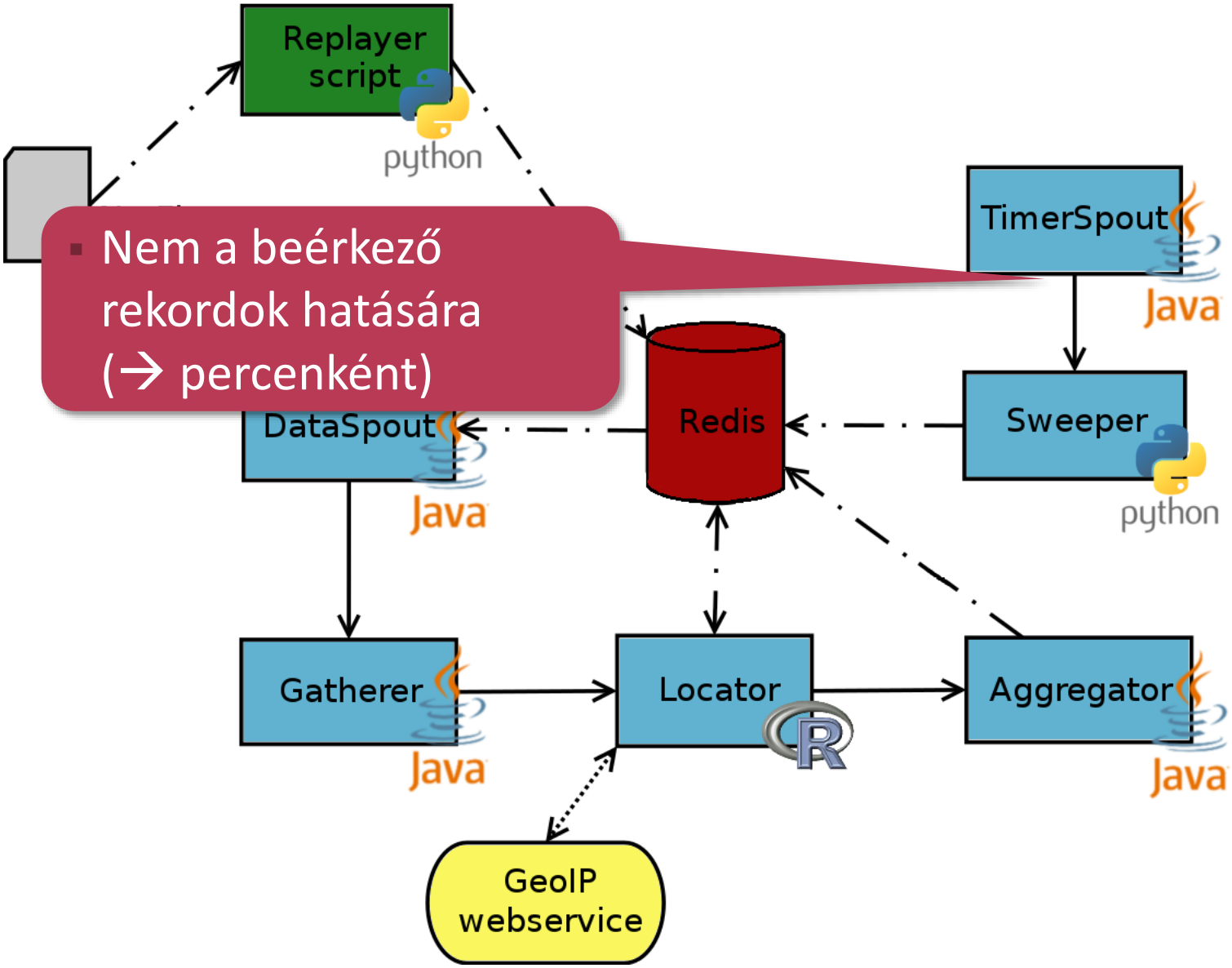


- Az adatokat idő alapján aggregálja 3 perces blokkokba
- Országok szerint összegez
- Adatbázisba ment

Alkalmazás adatfolyam

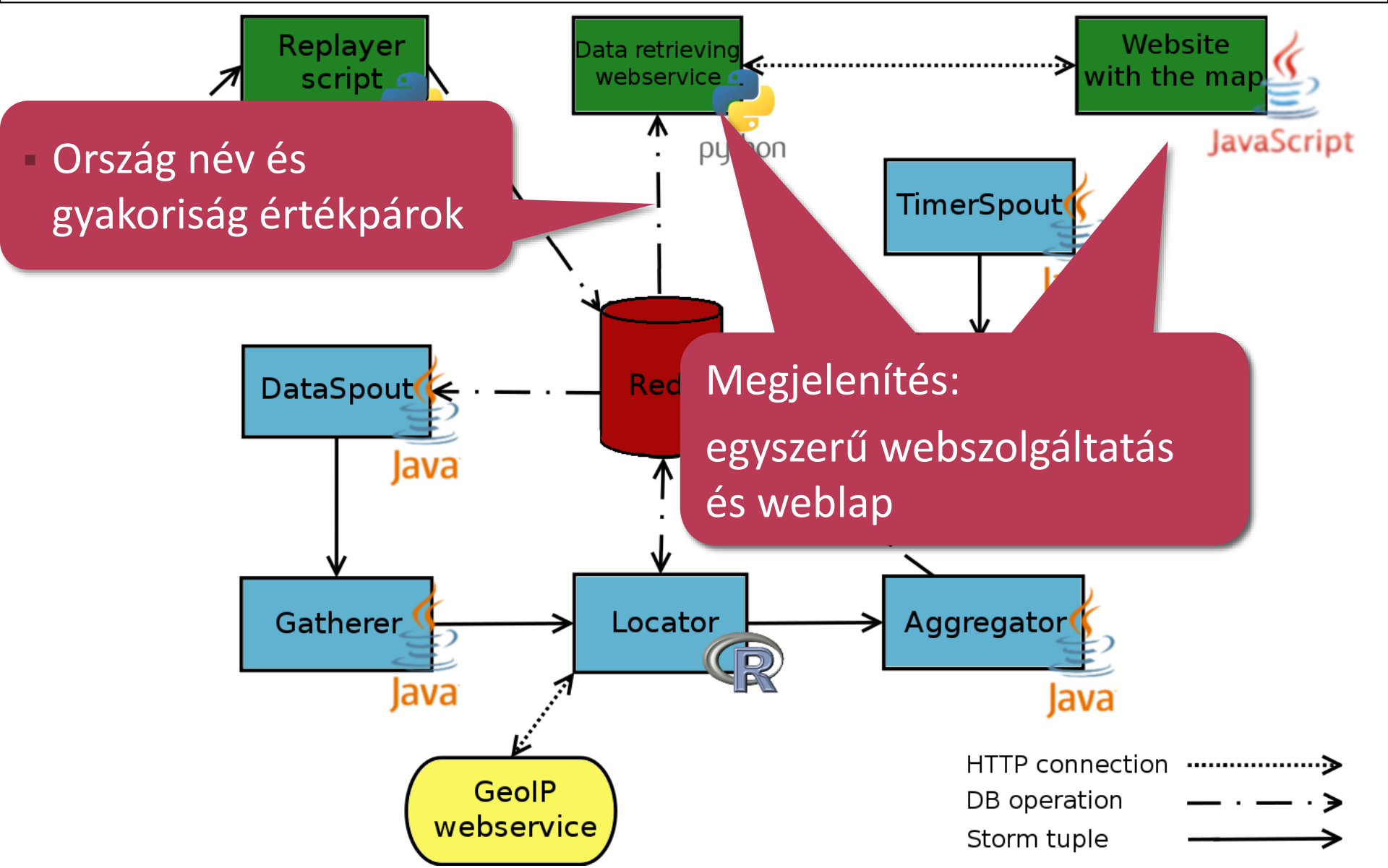


Alkalmazás adatfolyam



- Nem a beérkező rekordok hatására (→ percenként)

Alkalmazás adatfolyam

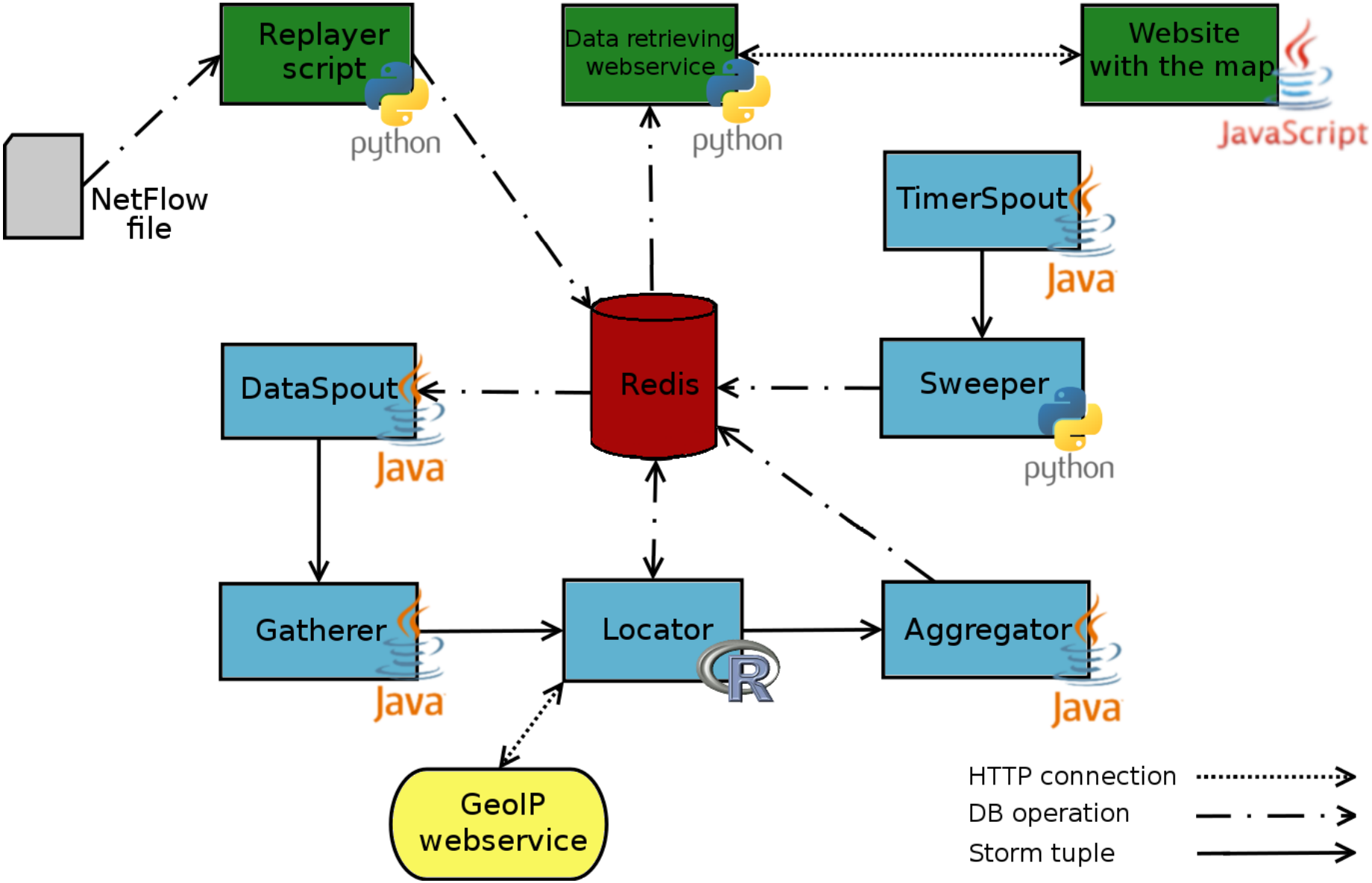


- Ország név és gyakoriság értékpárok

Megjelenítés:

egyszerű webszolgáltatás és weblap

Alkalmazás adatfolyam

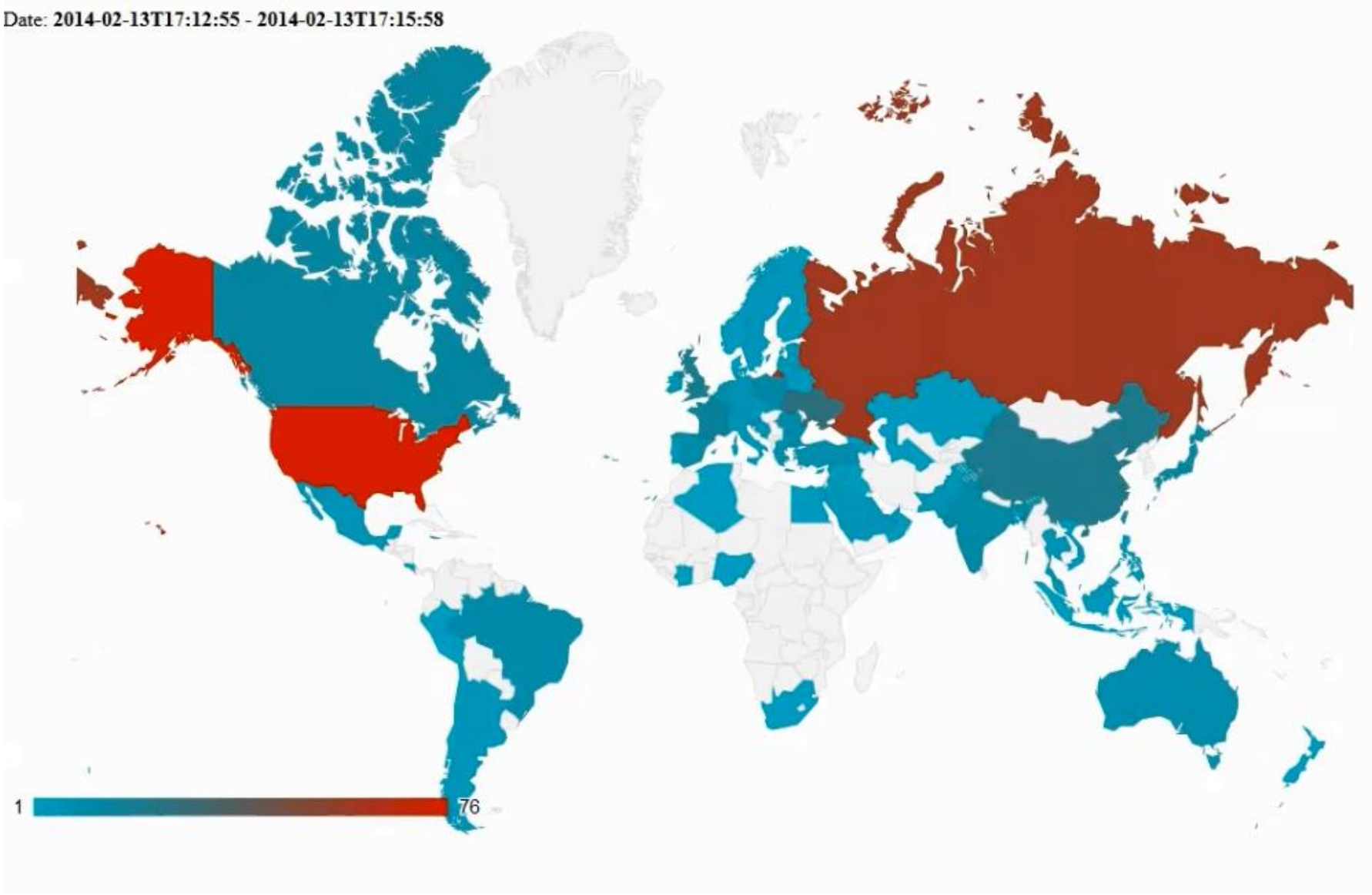


Szöveges “folyamat” (topológia)

```
TopologyBuilder builder = new TopologyBuilder();

builder.setSpout("redis_spout", new RedisSpout(), 1);
builder.setBolt("gatherer", new Gatherer(), 5)
               .shuffleGrouping("redis_spout");
builder.setBolt("locator", new GeoTagger(), 10)
                  .shuffleGrouping("gatherer");
builder.setBolt("aggregator", new Aggregator(), 10)
                  .fieldsGrouping("locator", new Fields("date"));
builder.setSpout("timer_spout", new TimerSpout(), 1);
builder.setBolt("sweeper", new Sweeper(), 5)
                  .shuffleGrouping("timer_spout");
```

Kimenet



Miért/hogyan folyamat?

- Adatáramlás explicit megjelenik
  - “Először szűr, aztán összesít”
- Implicit függőségek (DB)
- Folyamat sablon ~ topológia
  - Saját definíció, nem szabvány
- Nem általános célú
  - Kifejezetten adatfeldolgozás
  - (Eredetileg: állapotfrissítések)

ADATFOLYAMHÁLÓK

Data Flow Network, DFN

Adatfolyamhálók célja

- Csomópontok és kommunikáció modellezése
  - Pl. BPMN folyamatok leképzése (speciális eset)
- Csomópont is lehet egy modell…
  - Nem emlékezet/állapotmentes
  - Állapotgép
  - Folyamatmodell?
  - Maga is egy adatfolyamháló
- Későbbi előadásban
  - Hierarchia modellezése
  - Finomítási lépések

Komponensek kommunikációja

- Lazán csatolás → nem azonnali lépés
- Csatorna
  - FIFO vagy random access (mi alapján olvasunk belőle?)
  - Kapacitással rendelkeznek (mennyi token lehet rajta?)
  - Adatmodell rendelhető hozzá (pl. tokenhalmaz)
- Mögöttes technológia
  - Pl. üzenetsor alapú megoldások
  - MQ, JMS, MQTT, ..

Adatfolyam modellezés

Nem determinisztikus DFN formalizmus

- [Jonsson, Cannata]

- Struktúra
  - Adatfolyam gráf (DFG)
    - csomópontok
    - irányított élek (FIFO csatornák)
- Viselkedés
  - Tüzelési szabályok: <s0; in=c0; s1; out=2; π>
- Adatok
  - Tokenek

Adatfolyam modellezés

Nem determinisztikus DFN formalizmus

- [Jonsson, Cannata]

- Struktúra
  - Adatfolyam gráf (DFG)
    - csomópontok
    - irányított élek (FIFO csatornák)
- Viselkedés
  - Tüzelési szabályok: **<s0; in=c0; s1; out=c2; π>**
- Adatok
  - Tokenek

Adatfolyam modellezés

Nem determinisztikus DFN formalizmus

- [Jonsson, Cannata]

- Struktúra
  - Adatfolyam gráf (DFG)
    - csor
    - irár FIFO csatornák)

Kiinduló állapot

- Viselkedés
  - Tüzelési szabályok: **<s0; in=c0; s1; out=c2; π>**
- Adatok
  - Tokenek

Adatfolyam modellezés

Nem determinisztikus DFN formalizmus

- [Jonsson, Cannata]

- Struktúra
  - Adatfolyam gráf (DFG)
    - csor
    - irár FIFO csatornák)
- Viselkedés
  - Tüzelési szabályok: **<s0; in=c0; s1; out=c2; π>**
- Adatol
  - Toke

Kiinduló állapot

Bemeneti csatorna

Adatfolyam modellezés

Nem determinisztikus DFN formalizmus

- [Jonsson, Cannata]

- Struktúra
  - Adatfolyam gráf (
    - csor
    - irár FI
- Viselkedés
  - Tüzelési szabályok: **<s0; in=c0; s1; out=c2; π>**
- Adatol
  - Toke

Kiinduló állapot

Bemeneti csatornáról elvett token

Bemeneti csatorna

Adatfolyam modellezés

Nem determinisztikus DFN formalizmus

- [Jonsson, Cannata]

- Struktúra
  - Adatfolyam gráf (
    - csor
    - irár FI
- Viselkedés
  - Tüzelési szabályok: **<s0; in=c0; s1; out=c2; π>**
- Adatol
  - Toke

Kiinduló állapot

Bemeneti csatornáról elvett token

Bemeneti csatorna

Célállapot

Adatfolyam modellezés

Nem determinisztikus DFN formalizmus

- [Jonsson, Cannata]

- Struktúra
  - Adatfolyam gráf (
    - csor
    - irár FI
- Viselkedés
  - Tüzelési szabályok: **<s0; in=c0; s1; out=c2; π>**
- Adatol
  - Toke

Kiinduló állapot

Bemeneti csatornáról elvett token

Kimeneti csatorna

Bemeneti csatorna

Célállapot

Adatfolyam modellezés

Nem determinisztikus DFN formalizmus

- [Jonsson, Cannata]

- Struktúra
  - Adatfolyam gráf (
    - csor
    - irár FI
- Viselkedés
  - Tüzelési szabályok: **<s0; in=c0; s1; out=c2; π>**
- Adatol
  - Toke

Kiinduló állapot

Bemeneti csatornáról elvett token

Kimeneti csatorna

Bemeneti csatorna

Célállapot

Kimeneti csatornára kitett token

Adatfolyam modellezés

Nem determinisztikus DFN formalizmus

- [Jonsson, Cannata]

- Struktúra
  - Adatfolyam gráf (
    - csor
    - irár FI
- Viselkedés
  - Tüzelési szabályok: **<s0; in=c0; s1; out=c2; π>**
- Adatol
  - Toke

Kiinduló állapot

Bemeneti csatornáról elvett token

Kimeneti csatorna

Prioritás

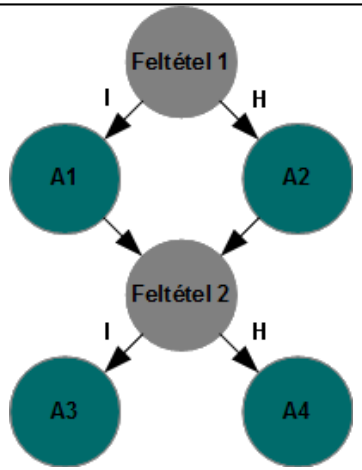
Bemeneti csatorna

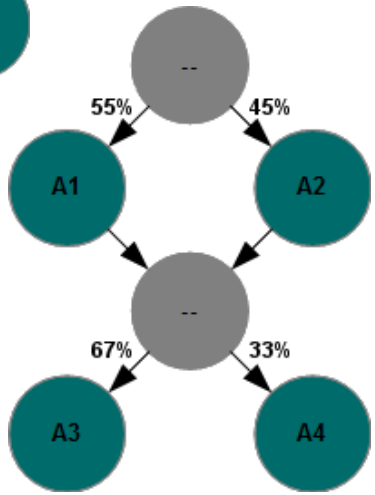
Célállapot

Kimeneti csatornára kitett token

Nem determinisztikus adatfolyam

- A rendszer determinisztikus:
  - Egy adott állapotban bekövetkező feltételek szerint hajt végre akciókat.
- A rendszer nem determinisztikus:
  - Példa1: Az eddigi feltételek helyett az akciók végrehajtásának valószínűsége adott (randomizált modell).
  - Példa2: nem tudjuk/nem modellezük a döntések belsejét (ld később: predikátum absztrakció, példa: “x<8” helyett “A”)
  - A randomizált modell nem feltétlenül „ekvivalens” a determinisztikus modellel.
  - Egymást kizáró alternatívák is lehetségesek





→ A kapott eredményt értelmezni kell

A módszer előnyei

	Alkalmas
Grafikus, moduláris, kompakt, hierarchikus	Egyszerűen áttekinthető modell
Fekete és átlátszó doboz modell	Modellezés korai fázisban
Finomítási szabályok	Többszintű modellezés
Információáramlás direkt leírása	Hibaterjedés modellezése
Elosztott modell mind finom, mind durva pontossággal	Aszinkron, konkurens események
Adatvezérelt működés	Eseményvezérelt real-time rendszerek
Hívási átlátszóság, atomi tulajdonság, információrejtés	Hibatűrő alkalmazások
Matematikai formalizmus	Formális módszerek
Transzformáció: TTPN, PA	Validáció, időbeli analízis

Adatfolyam hálózat formális leírása

- **Adatfolyam hálózat**: egy hármas (N, C, S)
  - N : csomópontok halmaza
  - C : csatornák halmaza
    - I: bemenő csatornák
    - O: kimenő csatornák

      } kapcsolat a külvilággal
    - IN: belső (csomópontok közötti) csatornák
  - S : állapotok halmaza
- **Adatfolyam csatorna**:
  - végtelen kapacitású FIFO csatorna,
  - egy bemeneti és egy kimeneti csomóponthoz kötve
  - állapota: $S_c = \times^{\infty} M_c$ tokenszekvencia

Adatfolyam csomópont formális leírása

Adatfolyam csomópont: $n = (I_n, O_n, S_n, s_n^0, R_n, M_n)$, ahol

I_n – bemenő csatornák halmaza

O_n – kimenő csatornák halmaza

S_n – csomópont állapotok halmaza

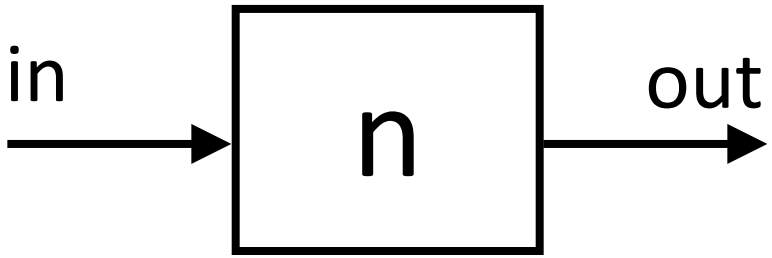
s_n^0 – csomópont kezdőállapota, $s_n^0 \in S_n$

M_n – tokenek halmaza

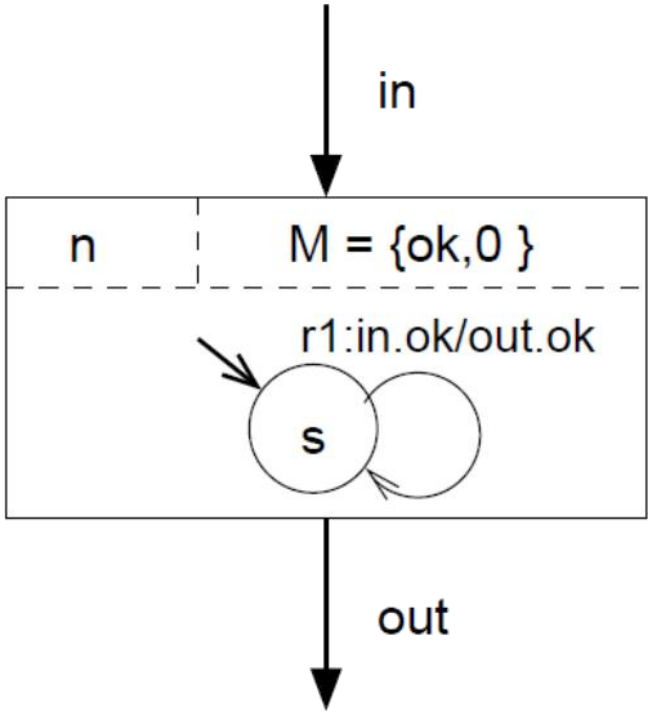
R_n – tüzelések halmaza, $r_n \in R_n$ egy ötös $(s_n, X_{in}, s'_n, X_{out}, \pi)$

- s_n – tüzelés előtti és utáni állapotok, $s'_n \in S$
- X_{in} – bemenő leképzés, $X_{in} : I_n \rightarrow M_n$
- X_{out} – kimenő leképzés, $X_{out} : O_n \rightarrow M_n$
- π – tüzelés prioritása, $\pi \in N$

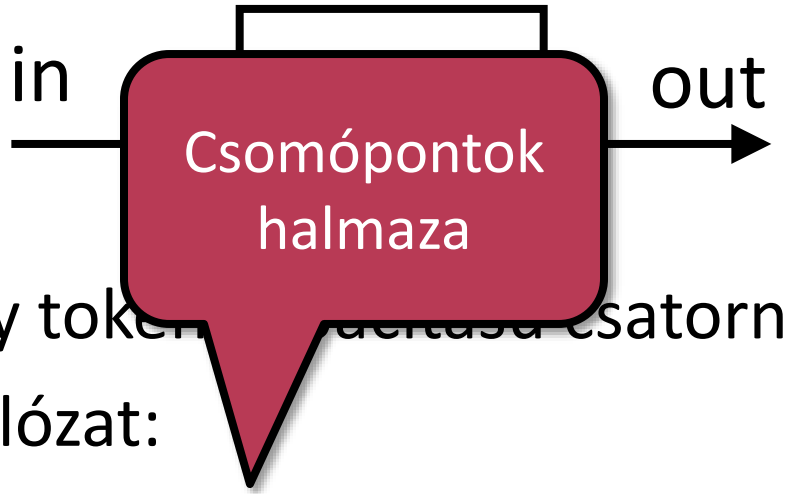
Egy példa



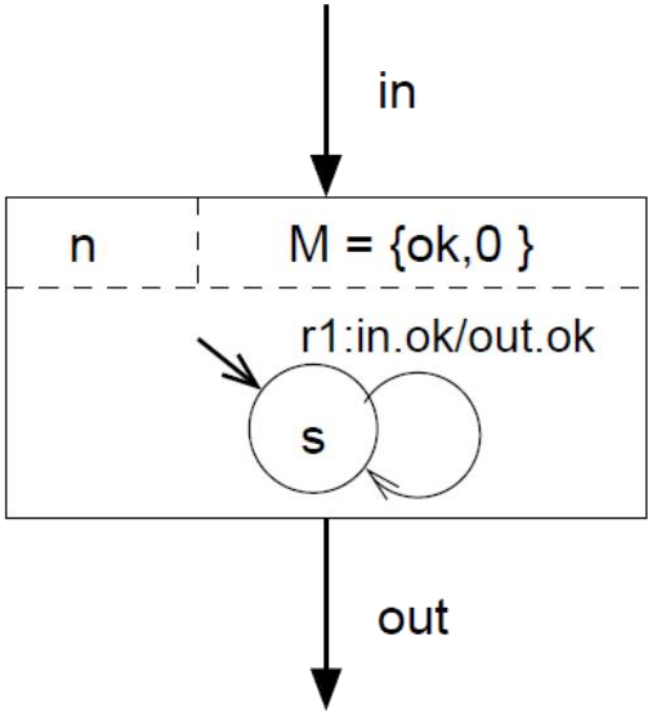
- Egy token kapacitású csatornák
- Hálózat:
  - DFN = ({n}, {in, out}, {(s,0,0), (s,ok,0), (s,0,ok), (s,ok,ok)})
- Csomópontok:
  - n = ({in}, {out}, {s}, s, {ok,0}, {r1})
- Tüzelések:
  - r1=<s; in=ok; s; out=ok >



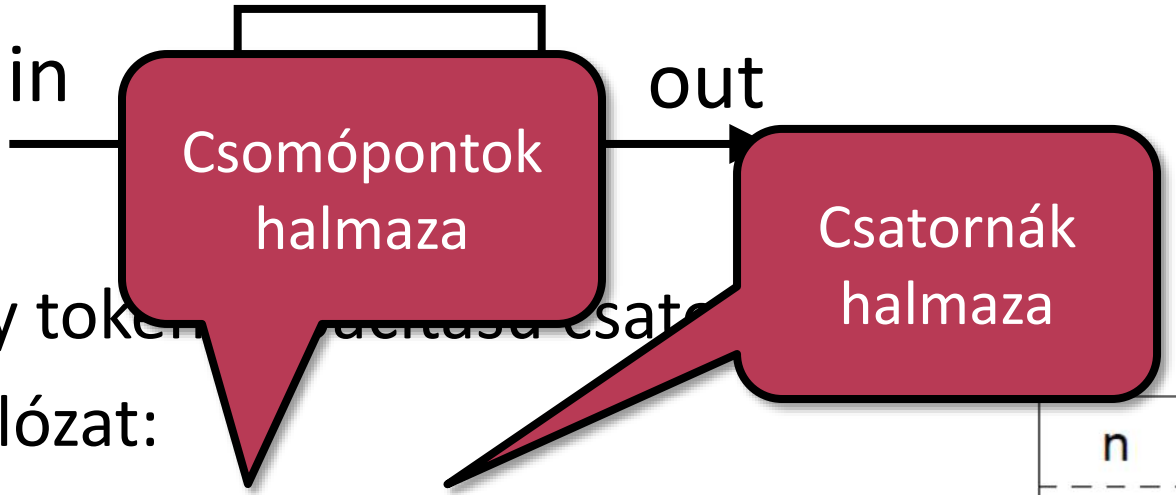
Egy példa



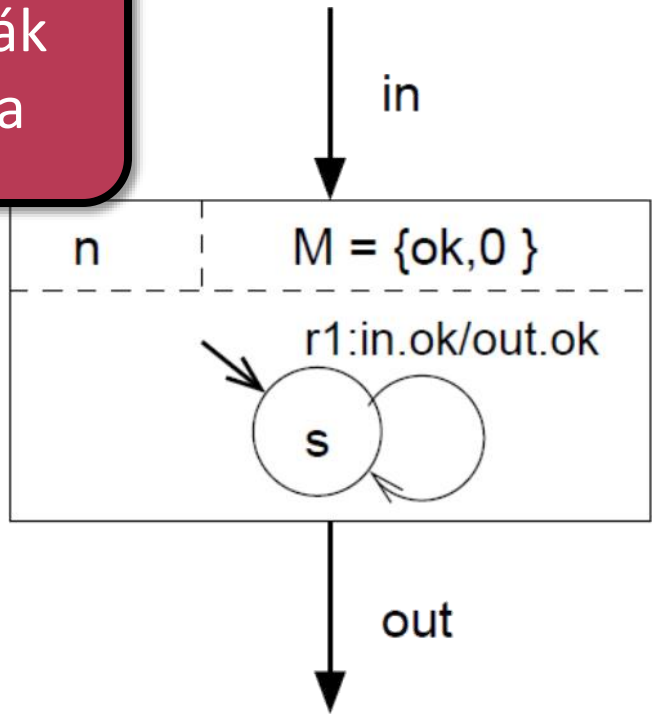
- Egy tok[illegible] csatornák
- Hálózat:
  - DFN = ({n}, {in, out}, {(s,0,0), (s,ok,0), (s,0,ok), (s,ok,ok)})
- Csomópontok:
  - n = ({in}, {out}, {s}, s, {ok,0}, {r1})
- Tüzelések:
  - r1=<s; in=ok; s; out=ok >



Egy példa



- Egy tok[illegible] csat[illegible]
- Hálózat:
  - DFN = ({n}, {in, out}, {(s,0,0), (s,ok,0), (s,0,ok), (s,ok,ok)})
- Csomópontok:
  - n = ({in}, {out}, {s}, s, {ok,0}, {r1})
- Tüzelések:
  - r1=<s; in=ok; s; out=ok >



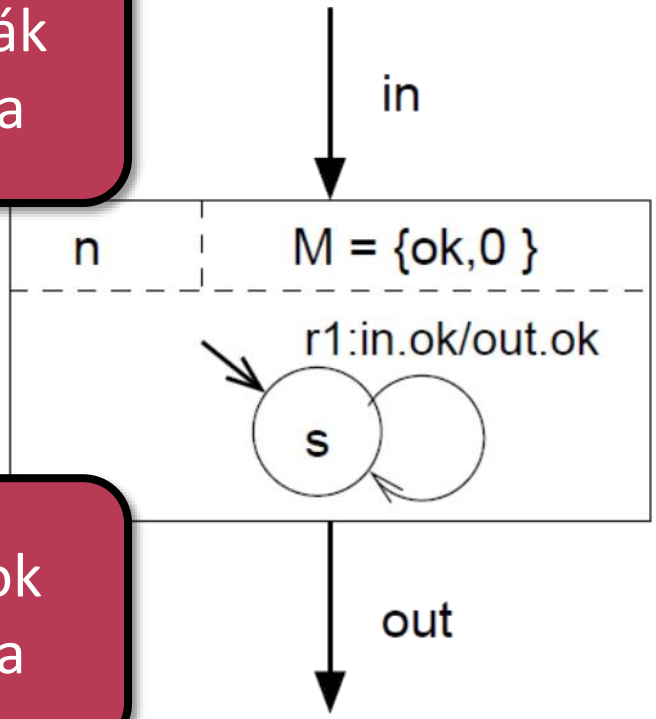
Egy példa

Csomópontok halmaza

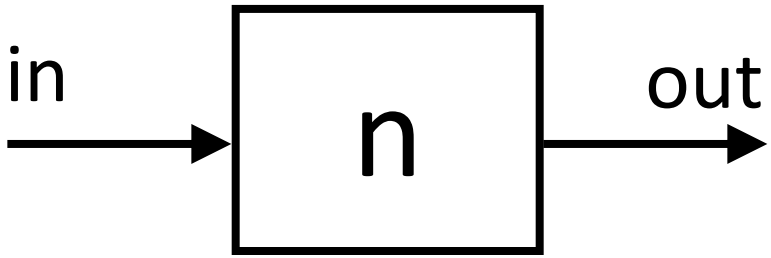
Csatornák halmaza

- Egy tok[illegible] csat[illegible]
- Hálózat:
  - DFN = ({n}, {in, out}, {(s,0,0), (s,ok,0), (s,0,ok), (s,ok,ok)})
- Csomópontok:
  - n = ({in}, {out}, {s}, s, {ok,0}, {[illegible]
- Tüzelések:
  - r1=<s; in=ok; s; out=ok >

Állapotok halmaza

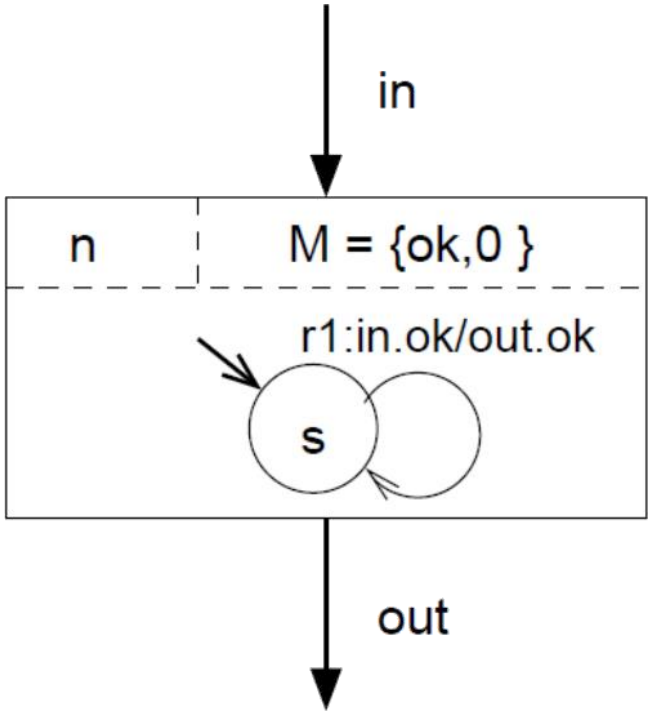


Egy példa

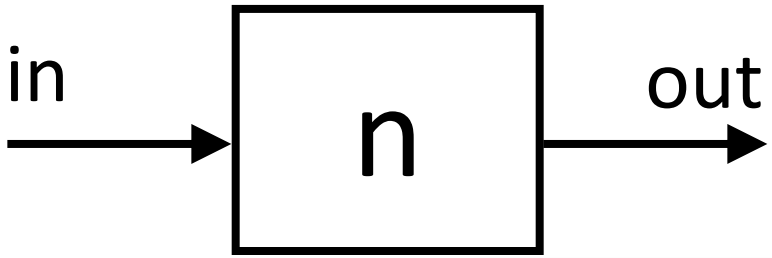


- citású csatornák
- out}, {(s,0,0), (s,ok,0), (s,0,ok), ok,ok)})
- Csomópontok:
  - n = ({in}, {out}, {s}, s, {ok,0}, {r1})
- Tüzelések:
  - r1=<s; in=ok; s; out=ok >

Bemenő csatornák halmaza



Egy példa

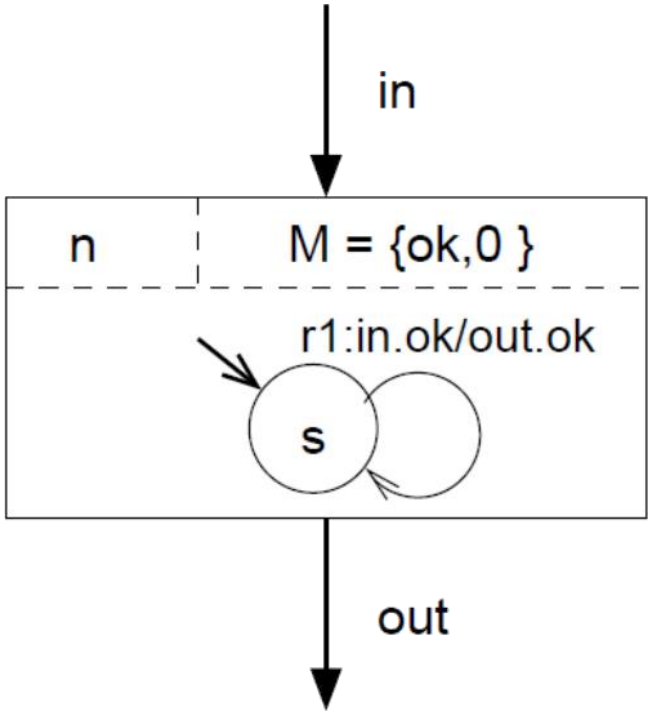


- [illegible]cit
- [illegible] out, [illegible]),
  (s,0,ok) [illegible],ok,ok)}

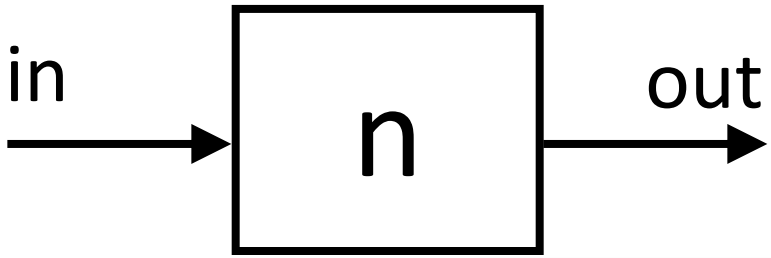
Bemenő csatornák halmaza

Kimenő csatornák halmaza

- Csomópontok:
  - n = ({in}, {out}, {s}, s, {ok,0}, {r1})
- Tüzelések:
  - r1=<s; in=ok; s; out=ok >



Egy példa

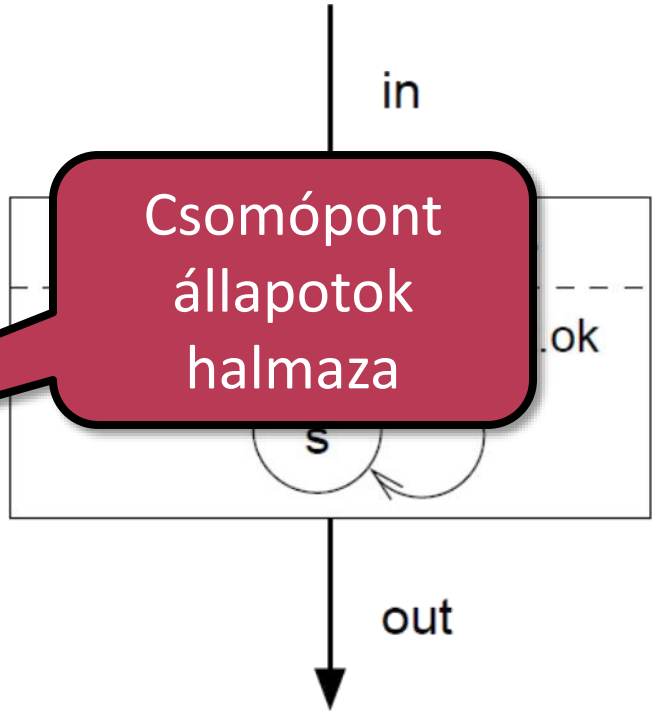


- [illegible]cit
- [illegible] out, [illegible]),
  (s,0,ok) [illegible],ok,ok)}
- Csomópontok:
  - n = ({in}, {out}, {s}, s, {ok,0}, {r1})
- Tüzelések:
  - r1=<s; in=ok; s; out=ok >

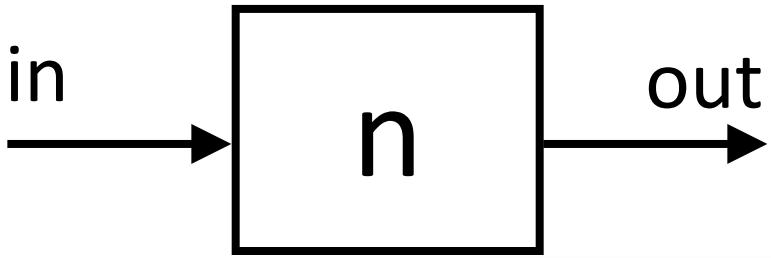
Bemenő csatornák halmaza

Kimenő csatornák halmaza

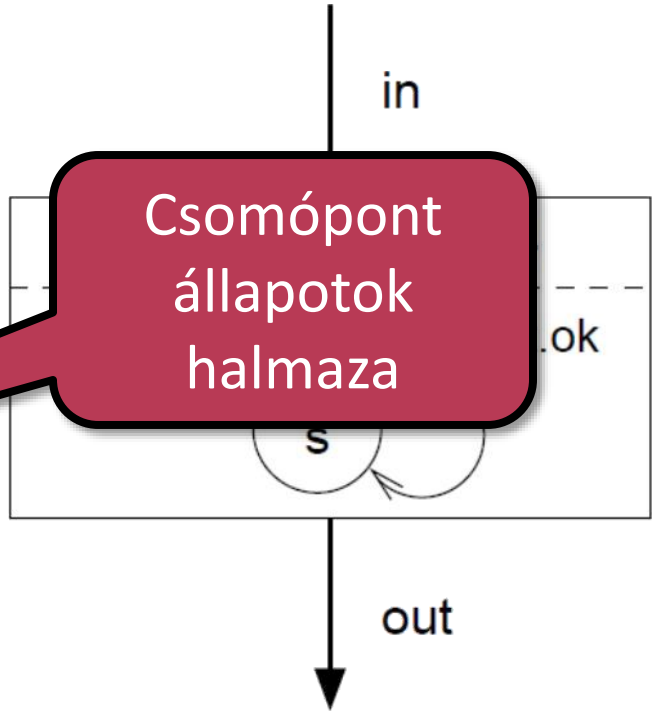
Csomópont állapotok halmaza



Egy példa



- cit
- out,), (0,ok,0),
  (s,0,ok) ,ok,ok)}
- Csomópontok:
  - n = ({in}, {out}, {s}, s, {ok,0}, {r1})
- out=ok >



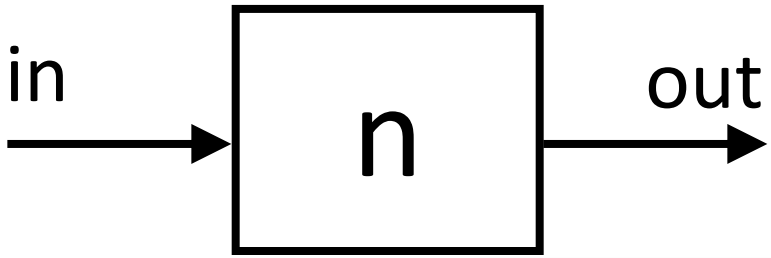
Bemenő csatornák halmaza

Kimenő csatornák halmaza

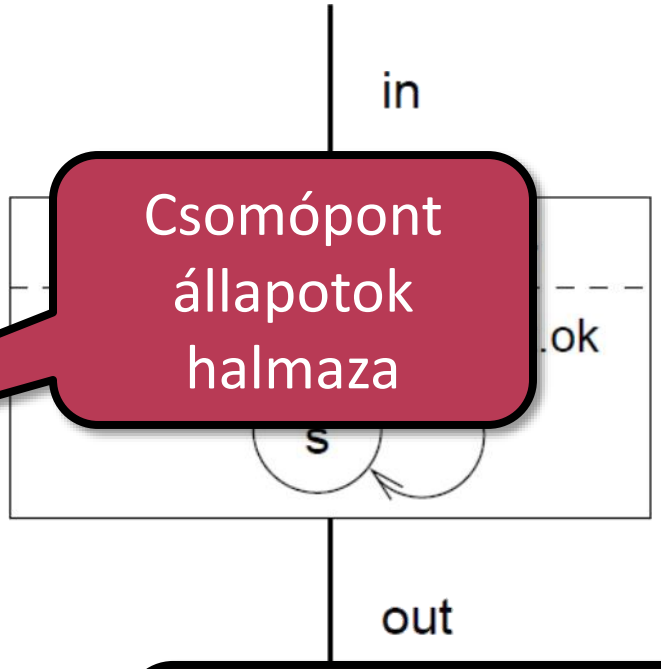
Csomópont állapotok halmaza

Tokenek halmaza

Egy példa



- [illegible]cit
- [illegible] out, [illegible]),
  (s,0,ok) [illegible],ok,ok)}
- Csomópontok:
  - n = ({in}, {out}, {s}, s, {ok,0}, {r1})
- [illegible] out=ok >



Bemenő csatornák halmaza

Kimenő csatornák halmaza

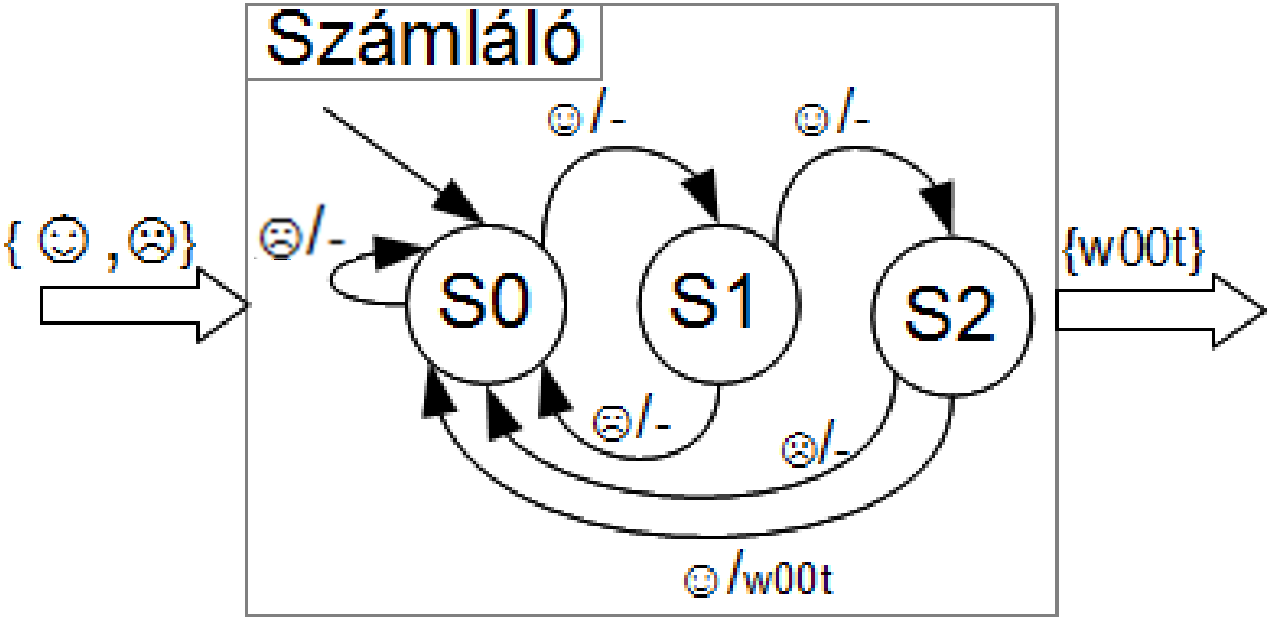
Csomópont állapotok halmaza

Tokenek halmaza

Tüzelések halmaza

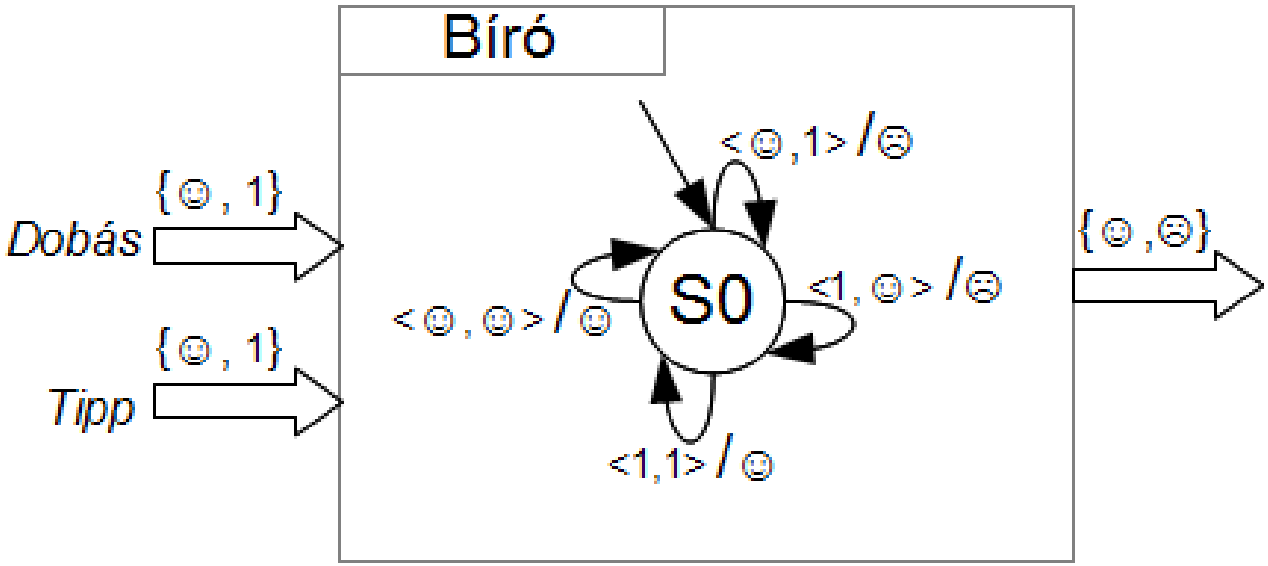
Példa - Számláló

- Készítsük el egy adatfolyam „Számláló” csomópontját, amely számláló bemenetén ☺ és ☹ tokeneket kap, majd a kimenetén a *w00t* token jelenik meg, amennyiben egymás után 3 db ☺ jelet olvas a bementről.



Példa - Bíró

- Készítsük el egy adatfolyam „Bíró” csomópontját. A csomópont két bemenetéről egyszerre olvassa be egy érme feldobásának eredményét és a játékos tippjét. Ha a dobás és a tipp megegyezik a kimeneten a ☺ jelet, egyébként a ☹ jelet adja ki.



{Tipp.1,Dobás.☺} := <1,☺>

Adatfolyam modellek kiértékelése

+ Interaktív szimuláció
- Validáció, helyességbizonyítás (direkt/indirekt)
  - Dinamikus tulajdonságok: elérhetőség, holtpontmentesség
+ Időbeli analízis (indirekt)
  - Tüzelési szabályokban végrehajtási idő, mint valószínűségi változó
+ Hibaszimuláció (direkt, diszkrét esemény szimuláció)
  - Működési modell kiegészítése hibamodellel, hibahatások elemzése
+ Teszttervezés (indirekt)
  - Tesztgenerálás, tesztelhetőségi analízis, tesztkészlet optimalizálás
- Hibahatás analízis (direkt)
  - FMEA: hibamód és hatás analízis, hibafa és eseményfa generálás
- (Megbízhatósági analízis) (indirekt)
  - Klasszikus mértékek: megbízhatóság, rendelkezésre állás, MTBF, …