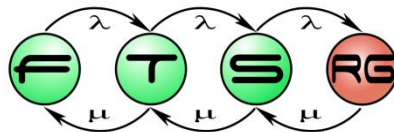


Szabályalapú üzleti logika, komplex eseményfeldolgozás

Gönczy László
Bergmann Gábor
Rendszermodellezés

Budapest University of Technology and Economics
Fault Tolerant Systems Research Group



Tartalom

- Produkciós rendszerek alapfogalmai
- Üzleti szabályrendszerek
- Komplex eseményfeldolgozás
- Esettanulmány(ok)

PRODUKCIÓS RENDSZEREK ALAPFOGALMAI

Szabály alapú működés

- Deklaratíván specifikált viselkedés
 - imperatív utasítássorozat helyett
 - „ha-akkor” szabályokkal
- Hol találkozunk szabály alapú viselkedéssel?
 - Tűzfal konfiguráció / routing tábla
 - MAKEFILE
 - Cron
 - Szakértő rendszerek (expert systems)
 - Diagnosztika, stb...
 - ...

Egy lehetséges kategorizálás

Szabály alapú (rule based) rendszerek

Következtető gépek
(inference engines)

Előre láncoló /
produkción

Tiszta logikai

Üzleti
szabálymotor

Hátra láncoló

Prolog,
stb.

Tűzfal, stb.

Szabály alapú következtető gépek

- **„Tudásbázis”** (knowledge base)
 - **„Ténybázis”** (fact base) / munkamemória (WM)
 - Változatos felépítés
 - **„Szabálybázis”** (rule base)
 - Szabályok, amelyekkel új tudást lehet kapni
 - „Ha”: feltétel rész, precondition, bal oldal (LHS)
 - „Akkor”: következmény rész, postcondition, jobb oldal (RHS)
- **Végül egy következtető mechanizmus**
 - Előre vagy hátra láncoló
 - Előre láncoló: logikai következtetés vagy üzleti szabályok

Példák

- Szakértői rendszer (pl. orvosi)
 - „Ha egy szerv gyulladt, akkor fájdalmat okozhat”
 - „Ha egy szerv gyulladt és aszpirin van a vérben, csökken a gyulladás”
 - „Fáj a lábam, mi minden okozhatja?”
 - „Ha bevennék aszpirint, mi lenne a következménye?”
- Üzleti szabályok
 - „Ha az ügyfél sokat roamingol, ajánljunk más tarifát”
 - „Ha a járat egyik buszvezetője a többihez képest kiugróan kevés jegyet értékesít, ellenőrizzük”

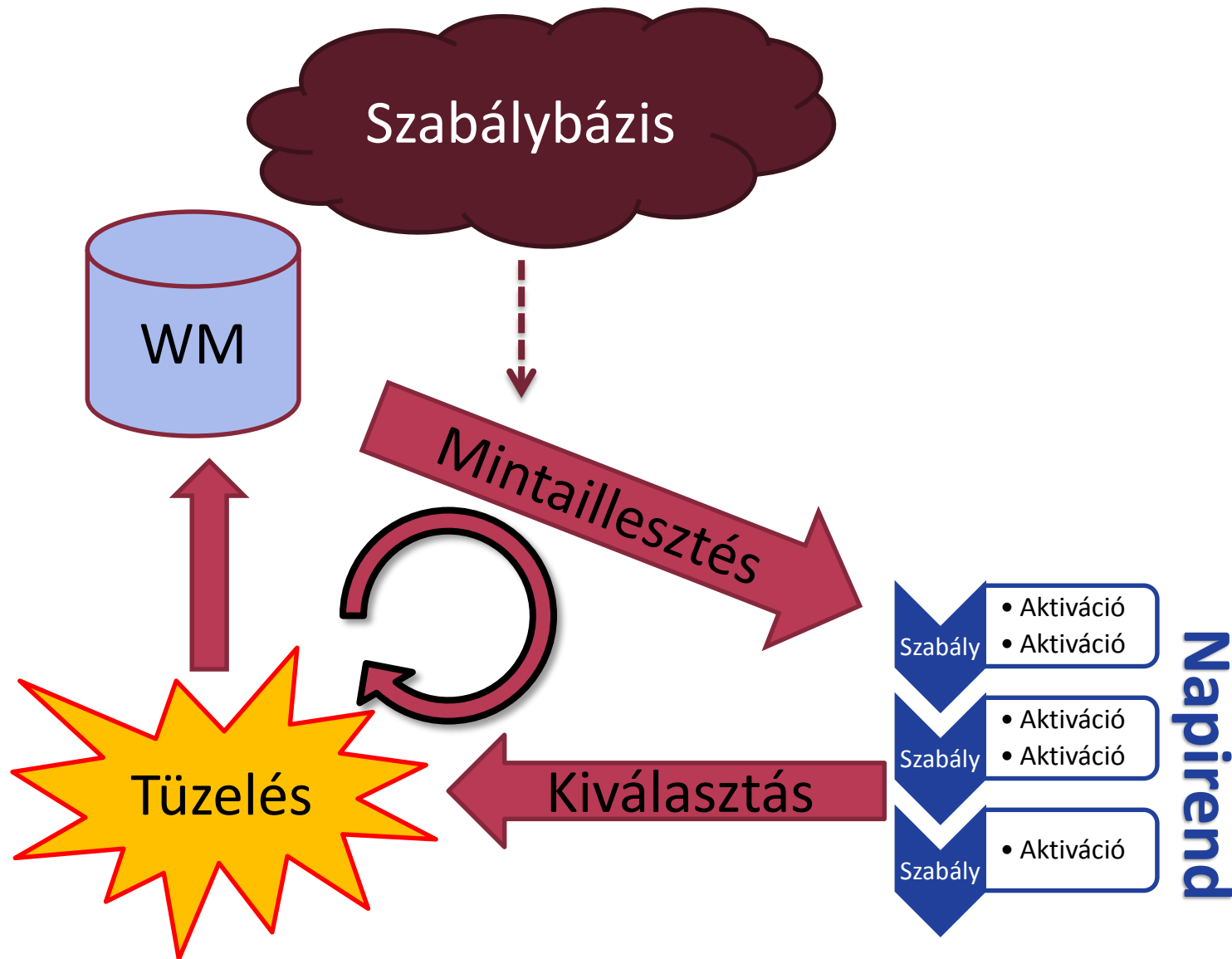
Következtetés

- **Előre** láncoló (induktív/*produkciós*, adatvezérelt)
 - A tényekből újabb tényeket képez (produkciós szabály)
 - Egy következmény teljesítheti egy szabály feltételrészét
 - Analógia: generatív nyelvtan, hatáselemzés
 - Ilyenek például a üzleti szabályrendszerek
 - Logikai következtetés (vs. üzleti szabály)
 - ha a feltétel érvénytelenné válik, a következmény is?
- **Hátra** láncoló (deduktív, igényvezérelt)
 - Egy cél-állítást próbál visszavezetni alaptényekre
 - Analógia: parser, diagnosztika
 - Ilyen például a *Prolog* és számos szakértői rendszer

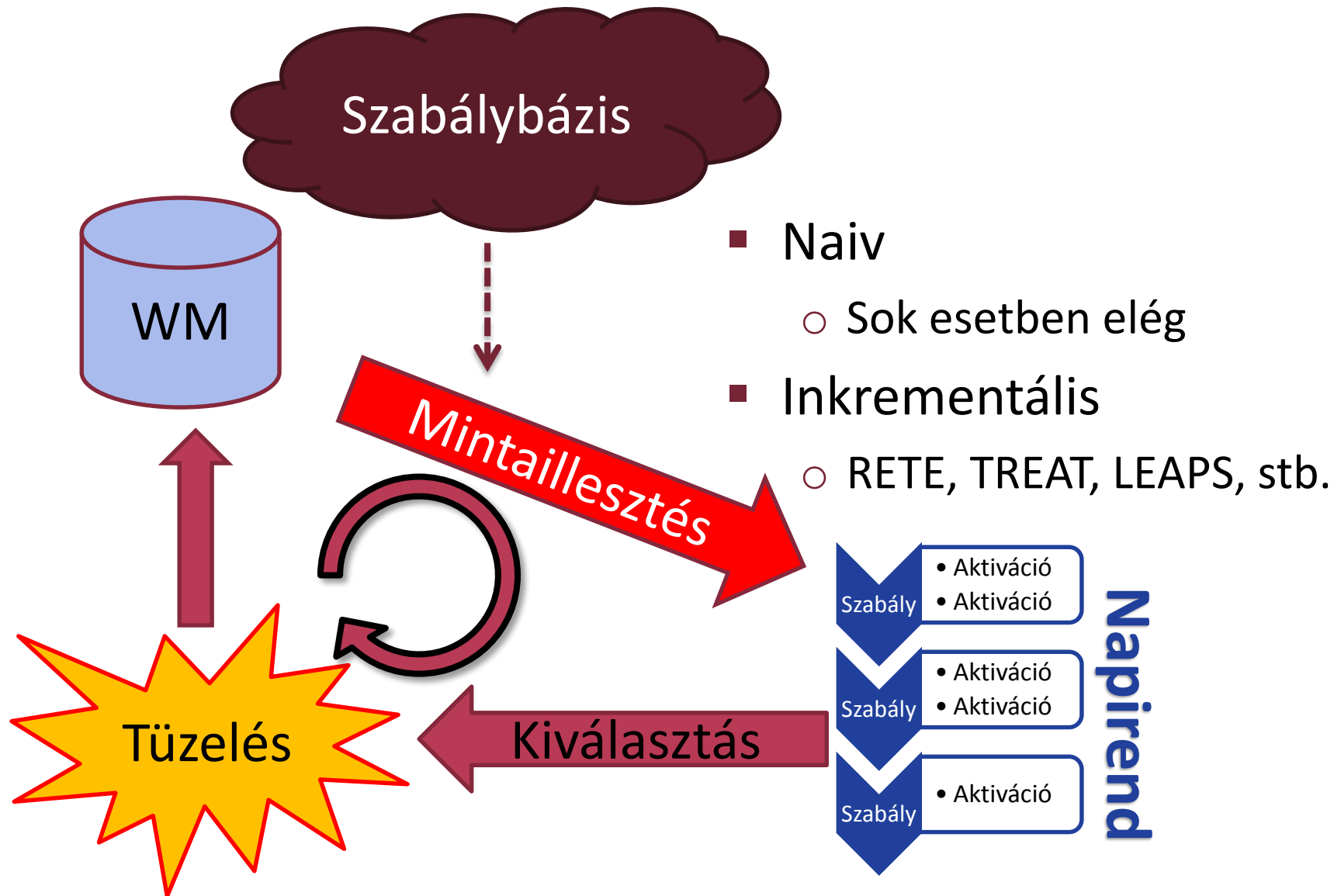
Produkción rendszer fogalomtár

- Munkamemória (working memory, **WM**)
 - Folyamatosan változó ténybázis
- **Aktivált** (activated, triggered) produkciós szabály
 - Minden feltétele ki van elégítve, tüzelhet
- **Aktiváció**
 - Szabály LHS egy konkrét kielégítő behelyettesítése
 - „n-es” (tuple), minden lekötetlen változóhoz egy érték
- **Tüzelés** (firing)
 - Szabály konkrét végrehajtása egy adott aktivációra
- **Napirend** (agenda, conflict set)
 - Összes (tüzelésre váró) aktiváció

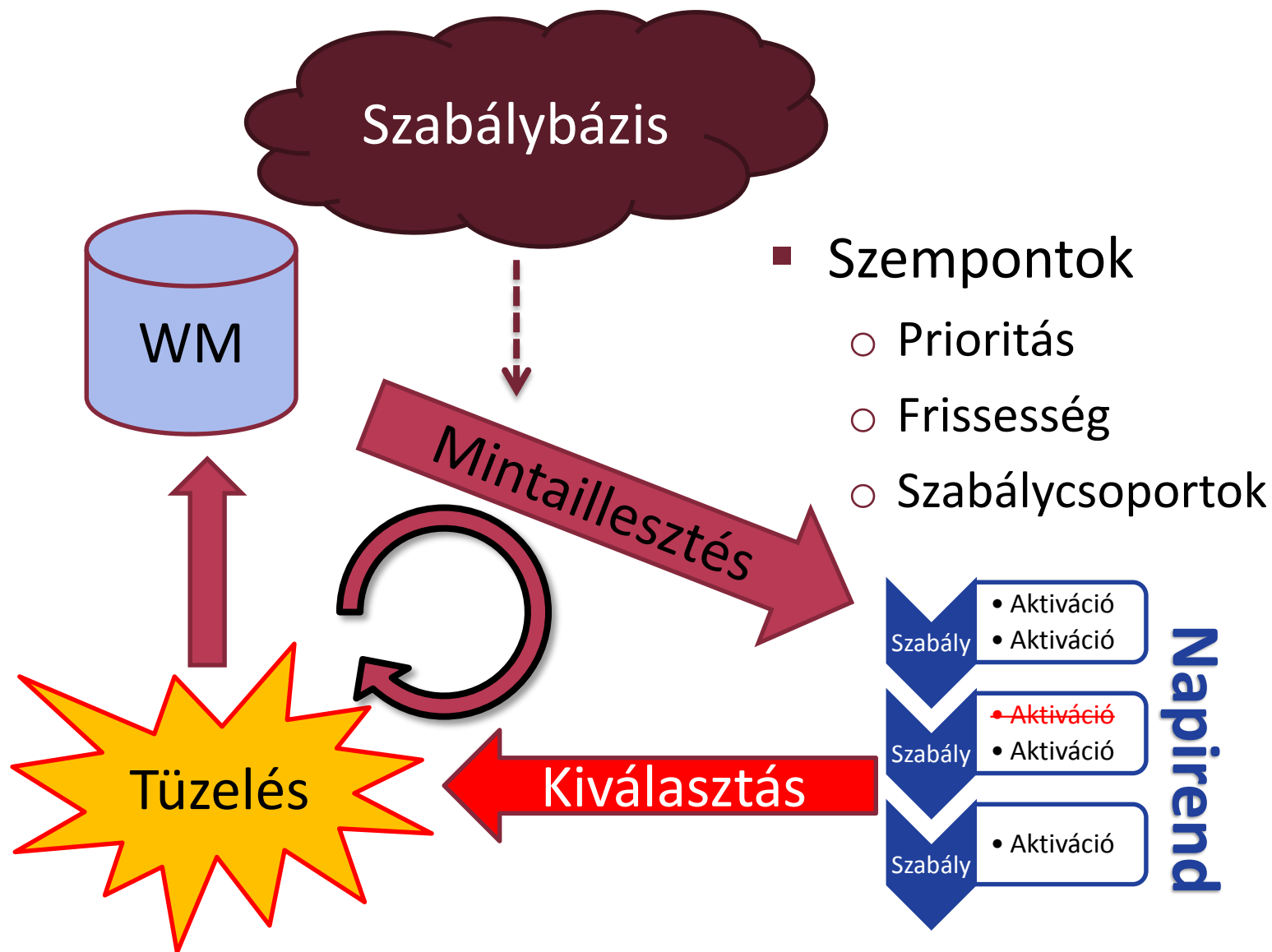
Egyszerű produkciós rendszer



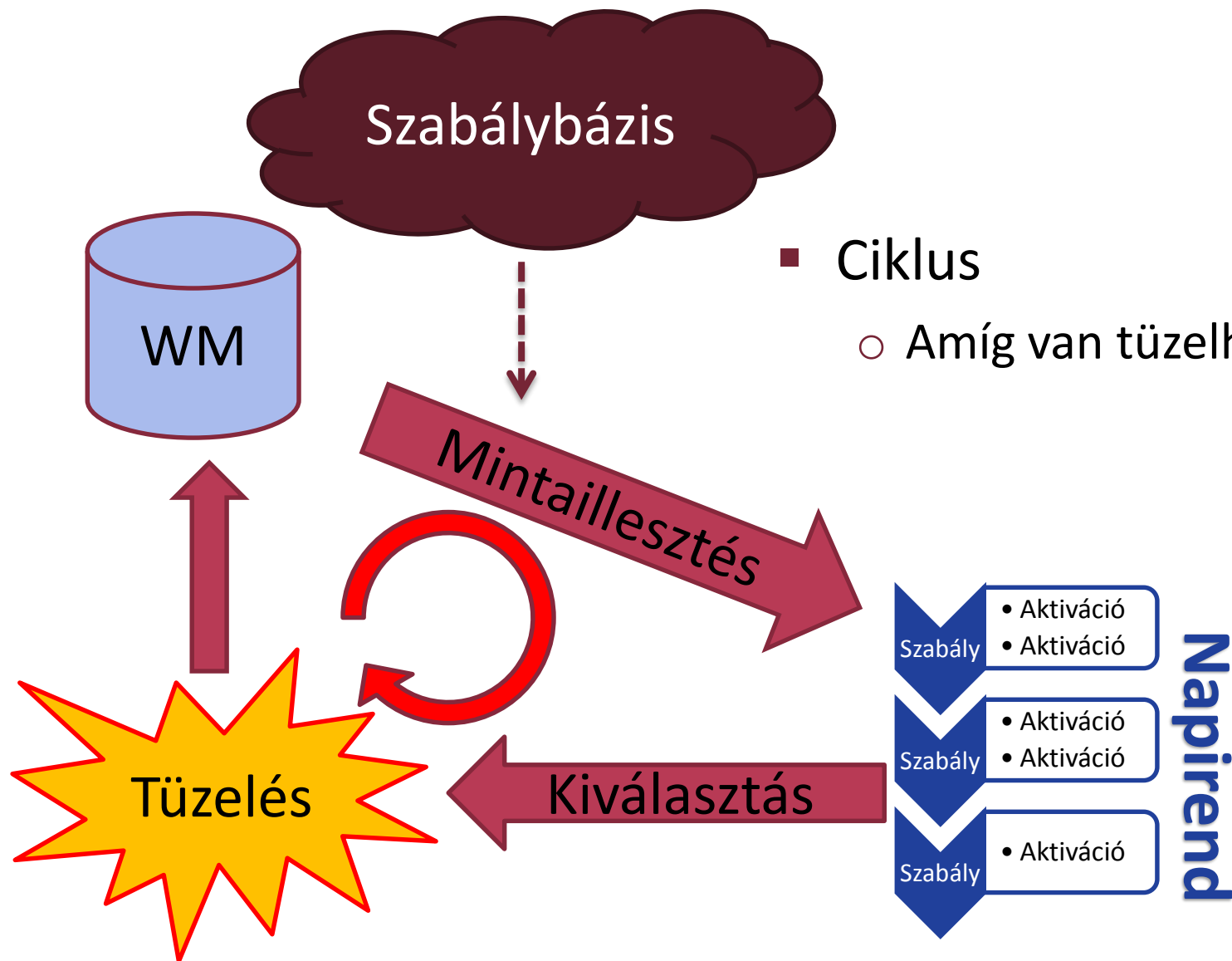
Egyszerű produkciós rendszer



Egyszerű produkciós rendszer



Egyszerű produkciós rendszer



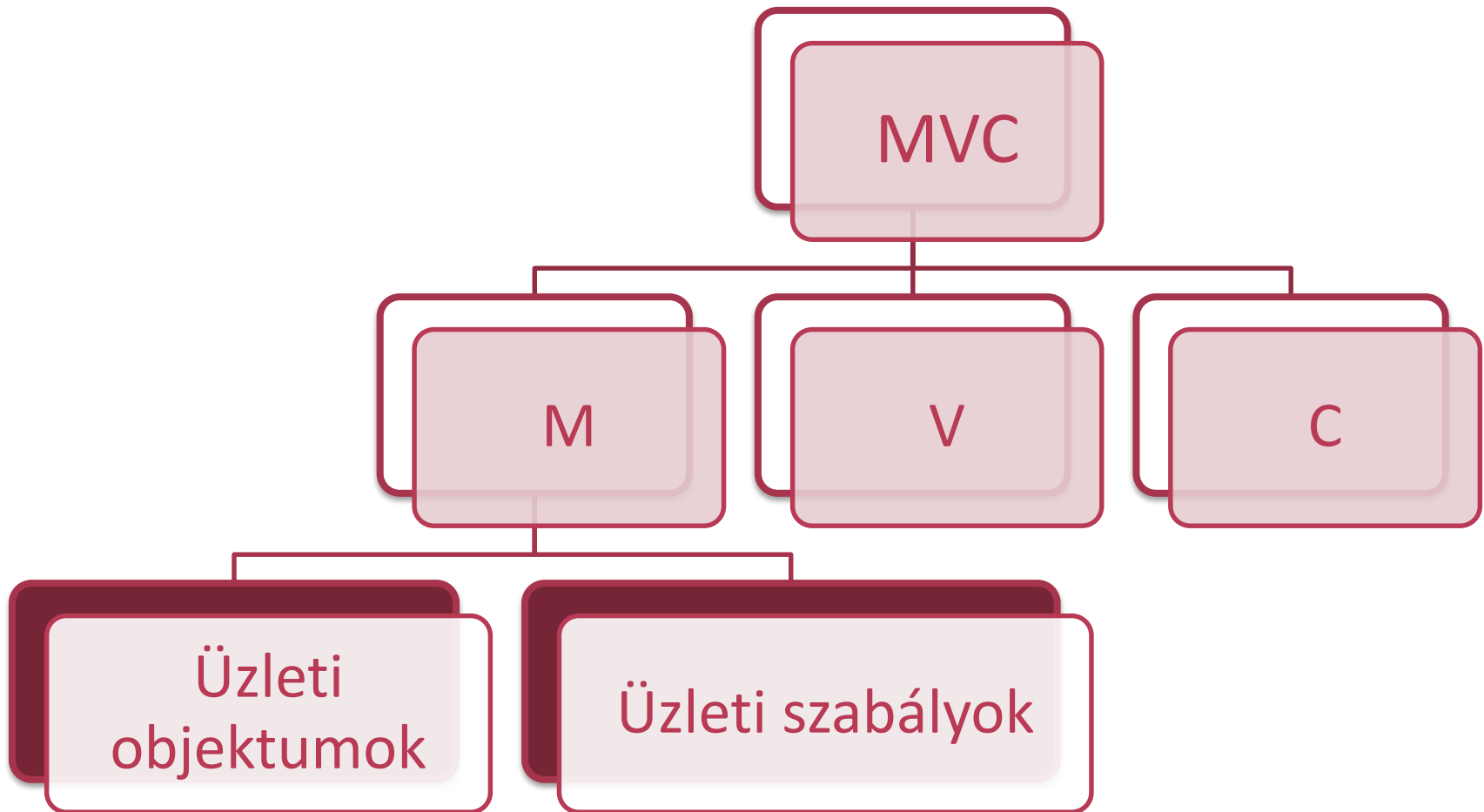
- Ciklus

- Amíg van tüzelhető szabály

ÜZLETI SZABÁLYRENDSZEREK

Business Rule Systems

- Szabály alapú üzleti logika



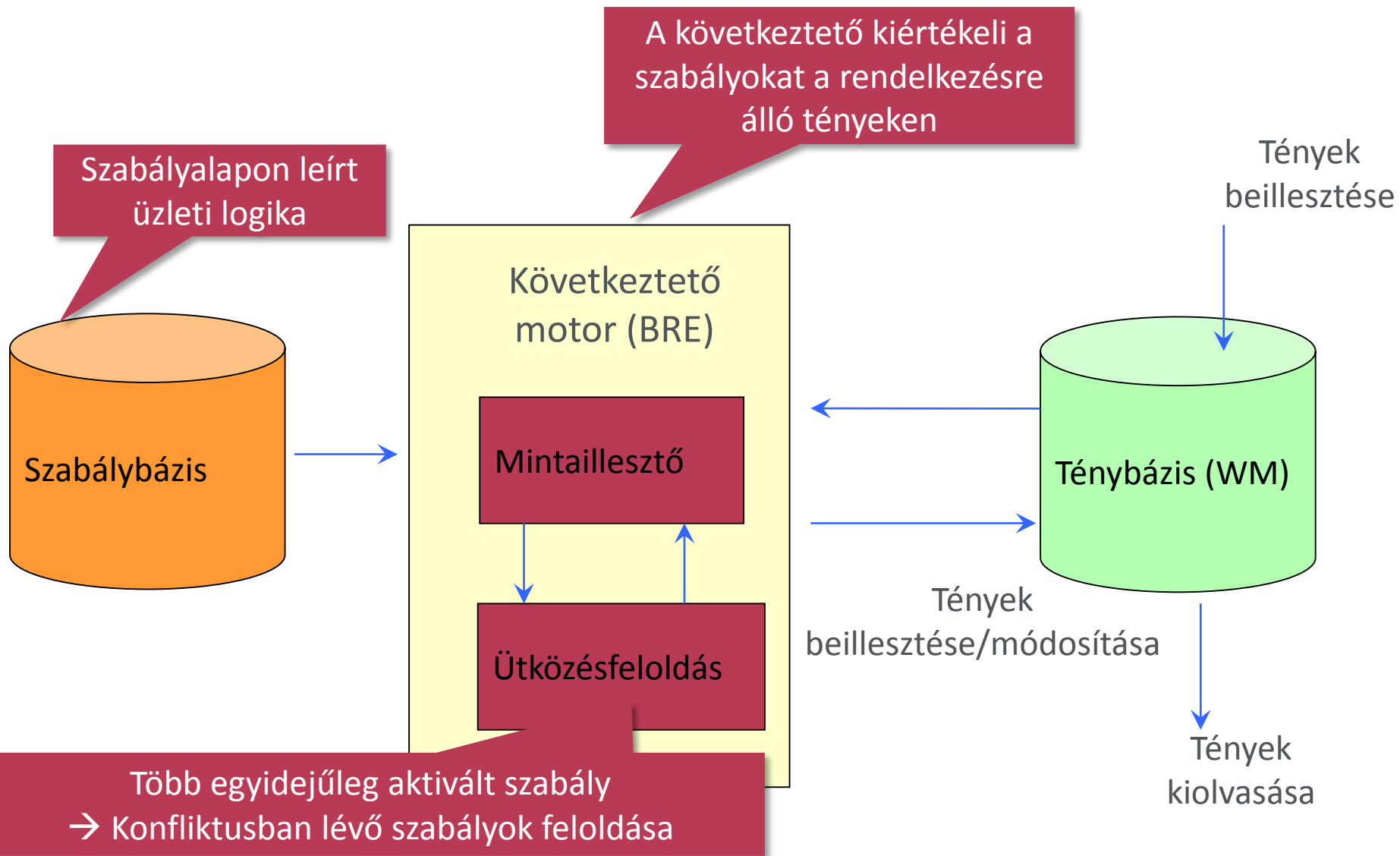
Üzleti szabályok

- **Üzleti logika** „kiszervezésére” végrehajtható modell
- **Üzleti objektumokat** figyelhet, manipulálhat
- Felépítése: ha → akkor
 - „ha az ügyfél 30 év alatti, emeljük 35%-al az ajánlatot”
 - „ha az ügyfél egyenlege 500Ft alá csökkent, értesítsük”
 - „ha más ügyfél korábban bejelentkezett már azonos lakcímre, nem adunk kedvezményt”
 - „ha a hallgatónak legalább húsz lezárt féléve van, nem szerzett aláírást diplomatervezésből és nem kapott köztársasági elnöki engedélyt, akkor megszüntetendő a jogviszonya, feltéve hogy ötéves képzésre jár és az ezt előíró jogszabály hatályba lépése óta kezdte tanulmányait”

Üzleti szabálymotor

- Üzleti szabályok produkciós rendszer szemszögből
 - „Tények” → üzleti objektumok
 - Kvázi produkciós szabályok, de RHS tetszőleges **akció**
 - Nem (feltétlen) logikai következtetés
 - Érvénytelenné váló feltétel, akció hatása mégis megmarad
 - Egy aktiváció többször is tüzelhet (pl. addig jár a korszó...)
- Üzleti szabálymotor (Business Rules Engine, **BRE**)
 - Üzleti szabályokat végrehajtó szoftver
 - Produkciós rendszer, a matematikai háttértől elvonatkoztatva, programozási platformként
 - Kapcsolat a külvilággal: WM, vagy akciók

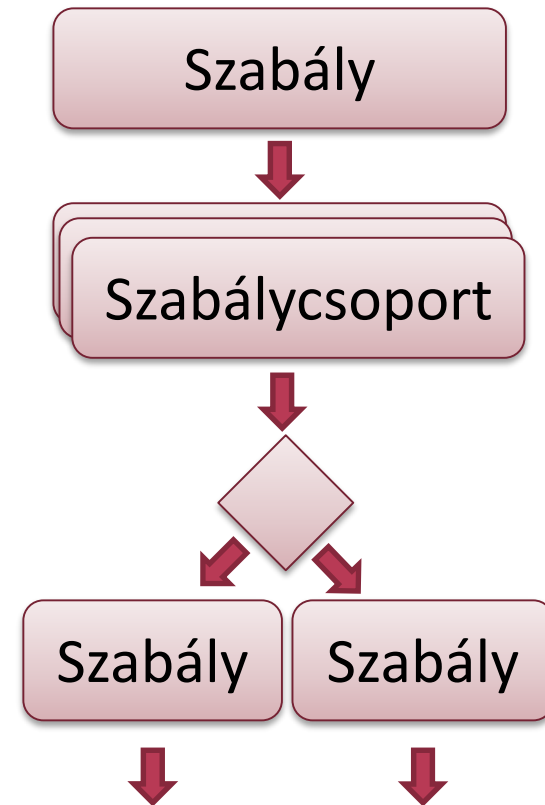
Üzleti szabálymotor működése



Több egyidejűleg aktivált szabály
→ Konfliktusban lévő szabályok feloldása

BRE vezérlése

- Alapértelmezett: tüzelési ciklus
 - Amíg van még tüzelhető szabály
 - Vagy STOP szabályig
- Komplex rendszer: vezérlési folyamat
 - Pl. jBPM workflow
 - Kiválthatja a bemutatott ciklust
- Eseményvezéreltség is elképzelhető
 - „Alvó” szabályok
 - Külön utasítás nélkül



Egyszerű Drools szabályok

```
rule "We have an honest Politician"
```

```
  salience 10
```

```
  when
```

```
    exists( Politician( honest == true ) )
```

```
  then
```

```
    insertLogical( new Hope() );
```

```
end
```

```
rule "Hope Lives"
```

```
  salience 10
```

```
  when
```

```
    exists( Hope() )
```

```
  then
```

```
    System.out.println("Hurrah!!!  
    Democracy Lives");
```

```
end
```

```
rule "Hope is Dead"
```

```
  when
```

```
    not( Hope() )
```

```
  then
```

```
    System.out.println( "We are all  
    Doomed!!! Democracy is Dead" );
```

```
end
```

```
rule "Corrupt the Honest"
```

```
  when
```

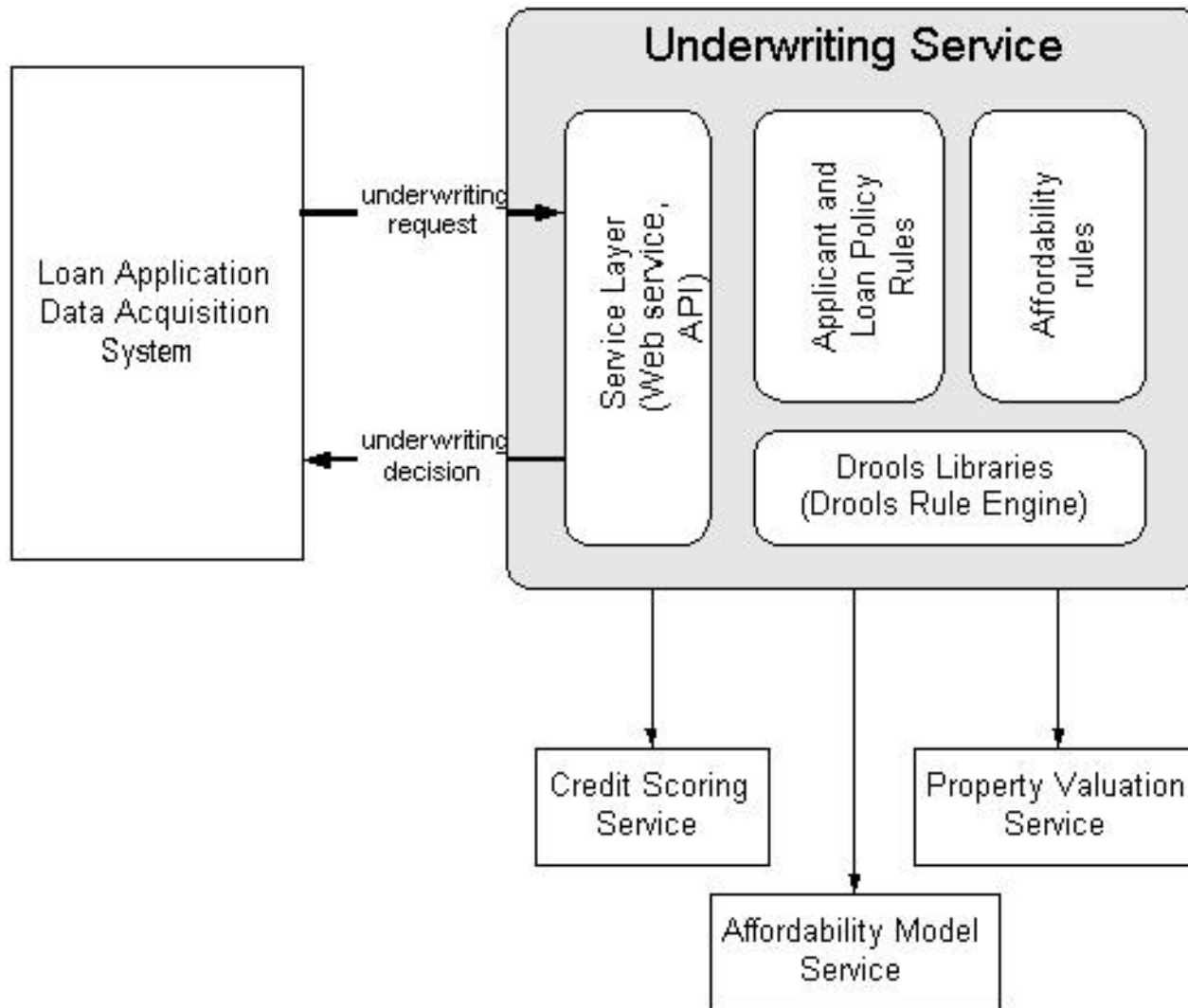
```
    politician : Politician( honest == true )  
    exists( Hope() )
```

```
  then
```

```
    System.out.println( "I'm an evil  
    corporation and I have corrupted " +  
    politician.getName() );  
    modify( politician ) {  
      setHonest( false )  
    }
```

```
end
```

Példa alkalmazás



<http://onjava.com/onjava/2007/01/17/building-enterprise-services-with-drools-rule-engine.html>

BRMS

- BRMS = **Business Rule Management System**
- BRE + kapcsolódó szolgáltatások
- Számos termék
 - *G2, JBoss Rules (**Drools**), IBM ILOG (J)Rules, Blaze Advisor, MS BRE, TIBCO iProcess, OPA stb.*

Microsoft[®]

ILOG
Changing the rules of business

Fair Isaac

gensym

Drools

TIBCO[®]
The Power of Now[®]

■ Szabálytár

- Kereshető, automatizáltan módosítható
- Verziózás

■ Végrehajtó könyvtár (BRE) → végrehajtó szerver

■ Tool support

- IDE, webes felület
- Template lehetőség, döntési tábla
- Magasabb granularitású szabályok
- Tesztelési támogatás, gyors próba
- Üzleti szótár építése meglévő adatokból

Szabály alapú üzleti logika előnyei

- Dedikált szabálytár → karbantarthatóság
 - Üzleti logika könnyebben módosítható
 - Pont ez **változhat** leggyakrabban: új rendeletek, stb.
- **Redundancia elkerülése**
 - Ugyanaz az üzleti logika sok modulban megjelenhet
- Jó esetben az **üzleti döntéshozók** is tudják olvasni
 - Sőt, akár írni is: természetes nyelvi verbalizáció, spreadsheet alapú szabálygenerálás
- Hatékony végrehajtás (inkrementális mintaillesztés)
- Cserélhető körülötte az architektúra
- Eszköztámogatás

Szabály alapú üzleti logika hátrányai

- Sorrendiség körülményesebb
 - V.ö. imperatív programnyelvekkel
 - Megoldás: integráció workflow motorral (ld. Drools) ?
- Univerzális absztrakciós nehézségek
 - Túl elvont nyelv → bizonyos feladatokra alkalmatlan
 - Nem elég elvont → nem is egyszerűbb, mint a Java
 - „Szivárgás” (law of leaky abstractions)
- Alkalmazási tapasztalatok nem mindig pozitívak
- Human factor
 - „... néha nem árt megkerülni a szabályt”

Felhasználási területek - példák

- Biztosítók, bankok
 - Kalkulációk kiemelése
 - Szabályok következetes kikényszerítése
 - Ügyek elbírálásának támogatása
- E-Kormányzat
 - Regisztráció kiértékelése
 - Adó, járulékszámítás
- Logisztika
 - Szállítmányozási döntések támogatása

Példa szabálydefinícióra: Drools döntési tábla

■ Döntési tábla – forrás: spreadsheet

○ Sok hasonló szabály

- „ha <30 éves és legalább 2 éve ügyfél, kapjon 25%-ot”
- „ha 31-49 éves és legalább 3 éve ügyfél, kapjon 17%-ot”
- ...

○ Eltérő paraméterek (feltételek, akció részei)

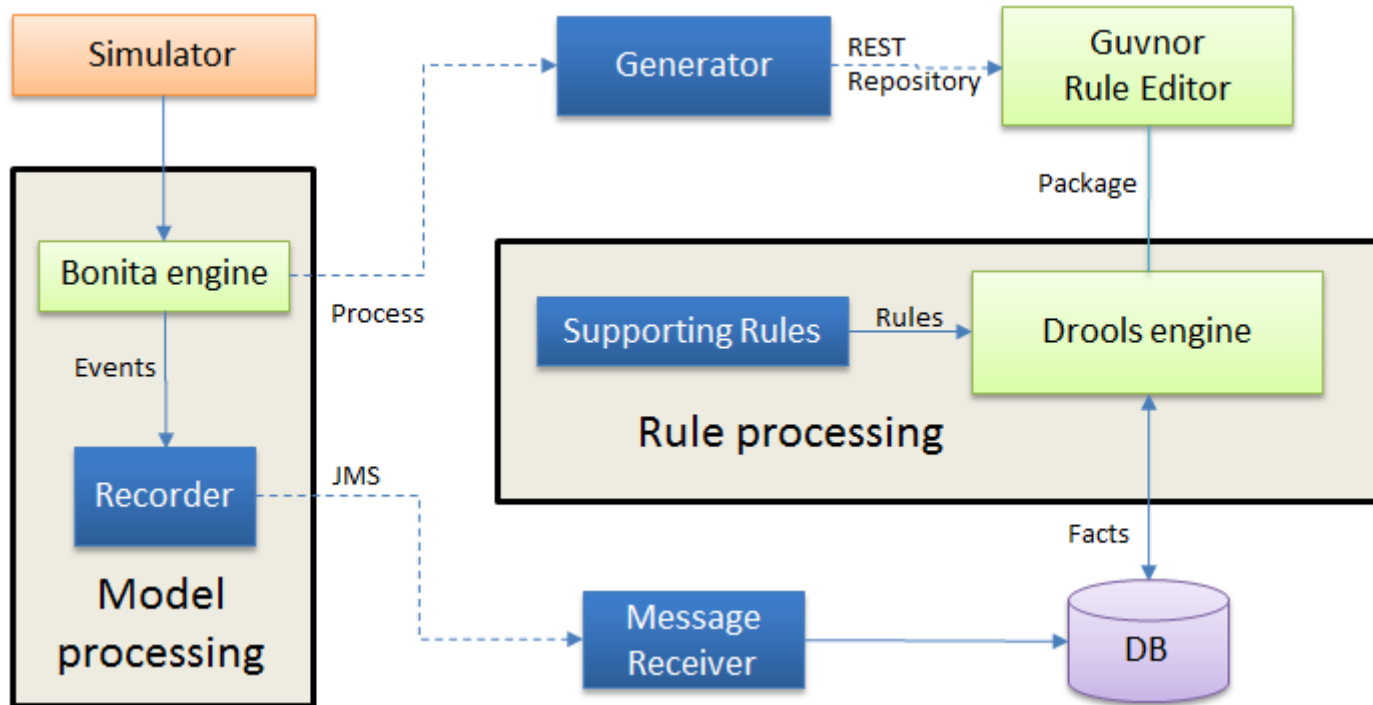
- Akár kifejezés, pl. >30

○ Üzleti döntéshozó által meghatározandó

B	C	D	E	F	G	H
1						
2	RuleSet	org.acme.insurance.base				
3	import	import org.acme.insurance.base.Approve, import org.acme.insurance.base.Driver				
4	Package	org.acme.insurance.base				
5						
6	RuleTable Old Driver					
7	CONDITION	RULEFLOW-GROUP	NO-LOOP	ACTION	ACTION	
8	\$driver: Driver					
9	licenceYears	priorClaims		insert(new Approve("\$param"));	System.out.println("Spa	
10	Persons age	Prior Claims		Inserting approval	Log	
11	d guy	30	1	risk assessment	Safe and mature	Old driver Approved
12						
13						
14						
15						
16						

Példa: üzleti folyamatok diagnosztikája

- Cél: folyamatok futásának ellenőrzése
 - Teljesítmény, biztonság, üzleti célok, ...
 - Üzleti felhasználó számára értelmezhető módon



“Szabályalapú diagnosztika üzleti folyamat vezérelt rendszerekben”,
Hartwig János, Urbán Balázs, TDK 2013.

KOMPLEX ESEMÉNYFELDOLGOZÁS (CEP)

Bergmann Gábor, Dávid István és az OptXware Kft. anyagainak felhasználásával

Tartalom

- Szabályalapú megközelítés felhasználása: komplex eseményfeldolgozó rendszerek
- CEP felhasználása
- Esettanulmány: CoMiFin
- Eseményfeldolgozás modell alapú tervezése

Kihívások

- Sok információforrás
 - „Szenzorok”
 - Felhasználói lépések szekvenciái
 - Logok
 - Külső szolgáltatások
- Sok esemény
 - Pl. ~százás nagyságrendű servermetrika, százás nagyságrendű server
- Sok „érdektelen” esemény közt néhány minta
- Párhuzamos, online adatfeldolgozás szükséges
 - Hagyományos adatbázis alapú módszerek lassúak lehetnek
 - Egyszerre nem fér el minden esemény egy feldolgozóegység memóriájában
- Feladat: események feldolgozása és korrelációja
 - Kis késleltetéssel
 - Aszinkron módon
- Kérdés: mit figyeljünk?

CEP alapelvek

- „Komplex esemény”
 - Több elemi esemény összekapcsolása
- Tulajdonságok
 - Időzítések figyelembevétele (pl. csúszóablak)
 - Aszinkron működés
 - Oksági kapcsolatok, hierarchikus események
 - Korreláció
 - „Forward chaining”
- SQL-szerű query nyelvek
 - Pl. EPL: Event Processing Language
 - Feldolgozási folyamatba láncolható lépések
 - Event-Condition-Action
- Elosztott adatforrások
 - Adatbázisok, beérkező kérések, megfigyelt események, stb.
- Skálázhatóság
 - Cloud környezet

Események szemantikája

■ Drools:

	Point-Point	Point-Interval	Interval-Interval
A before B			
A meets B			
A overlaps B			
A finishes B			
A includes B			
A starts B			
A coincides B			

Alapok:

- Allen-féle intervallum logika, 1983...

```
rule "reasoning on events over time"  
when  
  $a : A( )  
  $b : B( this after[-2,2] $a )  
  $c : C( this after[-3,4] $a )  
  $d : D( this after[1,2] $b, this after[2,3] $c )  
  not E( this after[1,10] $d )  
then  
  // do something  
end
```

<http://www.jboss.org/drools/drools-fusion.html>

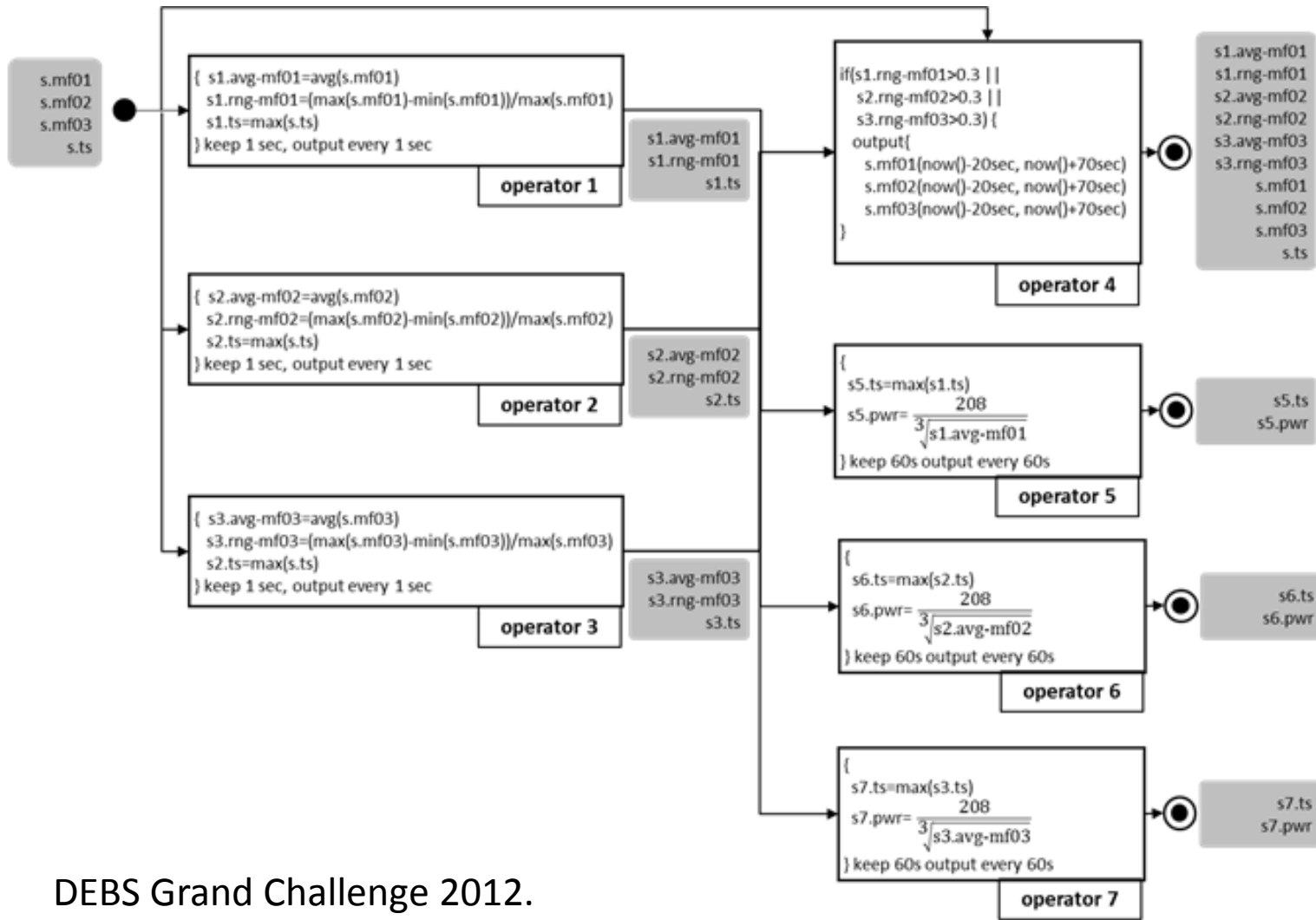
Tipikus CEP nyelvi elemek

- Esemény (típus definíció)
 - időbélyeg
- Eseményforrás
- Stream
- Időablak
- Időzítési operátorok
- Tolerancia (delta) az időbélyegekre
- Folyamatosan aktív lekérdezések

CEP alkalmazási területek

- Üzleti alkalmazások
 - Tőzsde, befektetések
 - „Treasury”
 - Kockázatkéértékelés
 - Hitelek árazása
 - Szállítmánykövetés
- „Business Activity Monitoring”
- Online visszaélések felderítése/megelőzése
 - Gyanús tranzakciók ellenőrzése
 - Fogadási adatok elemzése (pl. UEFA)
- Nagy IT rendszerek üzemeltetése
 - Komplex támadások felderítése
 - Metrika kiértékelés
- Biztonságtechnika
 - Pl. dDOS ellen
- <http://www.complexevents.com/>

Példa

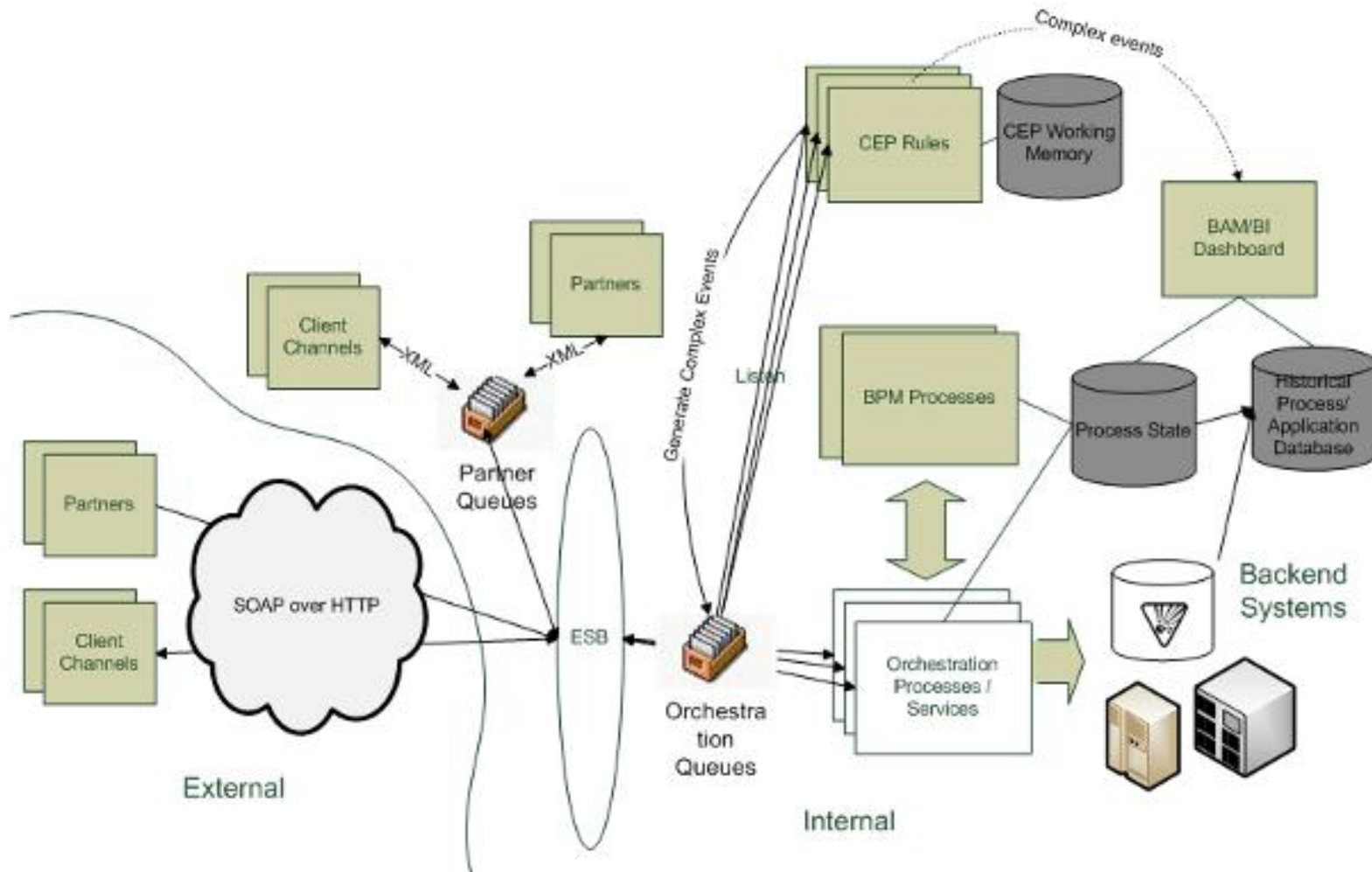


DEBS Grand Challenge 2012.

Hasonló technológiák

- Szabálymotorok
 - CEP esetén tipikusan nem tartjuk meg az adatot, nem módosítjuk (hierarchia, riasztások)
 - Nincs: explicit időzítés, eseményforrás, operátorok
- RDBM
 - Lekérdezőnyelvek hasonlóak
 - Nincs: időablak karbantartás, triggerek nem jól skálázódnak, válaszidő nehezen garantálható
- „Döntéstámogató” rendszerek
 - Nem realtime

Példa „architektúra”



<http://www.packtpub.com/article/cep-complex-event-processing-soa-service-oriented-architecture>

CEP eszközök

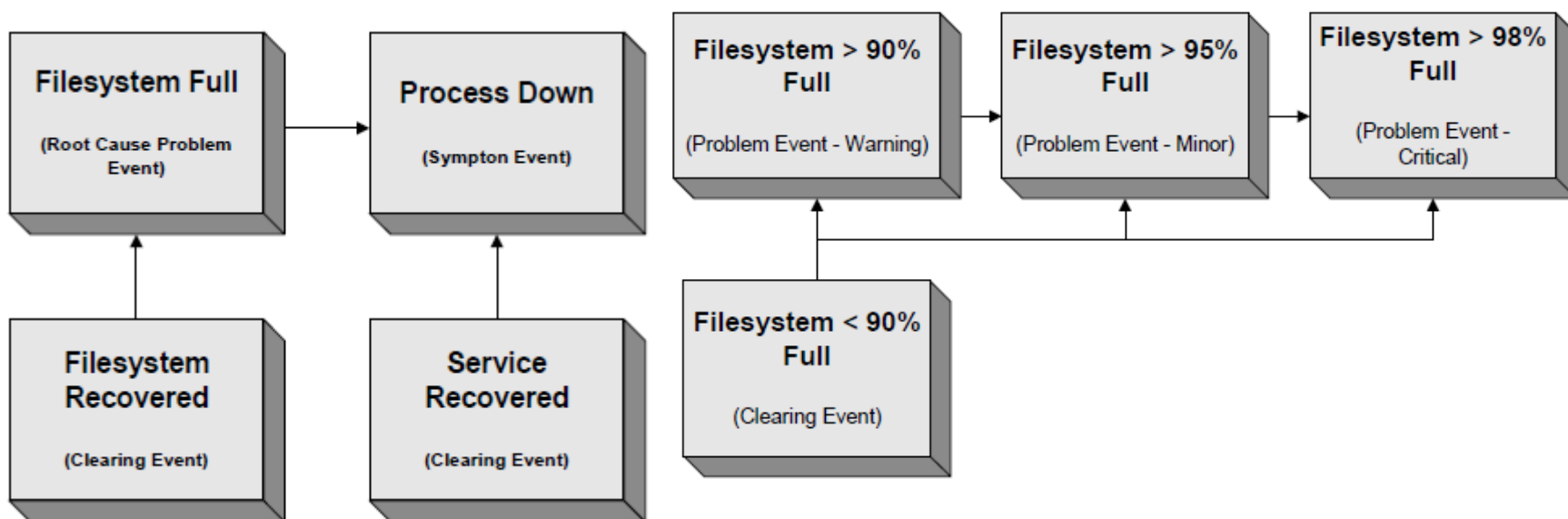
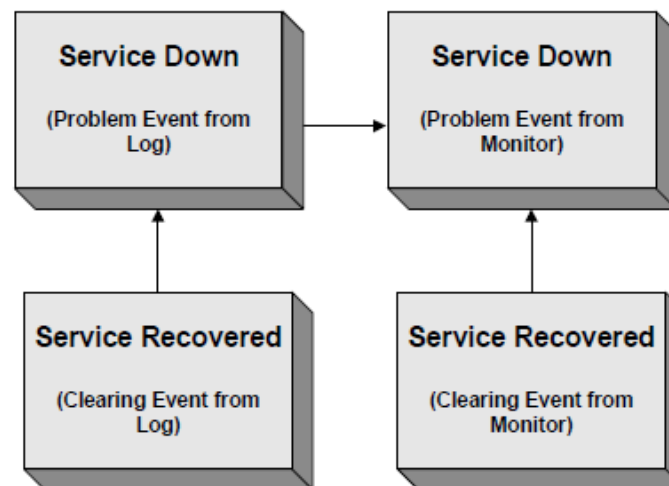
- Számptalan megoldás
 - Esper
 - Drools Fusion
 - IBM InfoSphereStreams (System S), WebSphere Decision Server
 - OpenESB - Intelligent Event Processor
 - Apache Hadoop + ráépülő projektek
 - TIBCO CEP
 - Microsoft StreamInsight
- Döntési szempontok
 - Eseményfeldolgozási logika
 - Áteresztőképesség
 - Elvárt válaszidő („low latency”)

Eseményfeldolgozás lépései

- **Előkészítés**
 - Események azonosítása („Mi honnan jön?”)
 - Események kiválasztása/szűrése
 - Események kiegészítése
 - Aggregálás
- **Elemzés**
 - Események „osztályozása” (rating, scoring, classification)
 - Elemzési minták (pl. elnyomás, topológia alapú függőségek figyelembevétele)
 - Események → komponens állapot
- **Feldolgozás**
 - Továbbítás
 - Előrejelzés
 - Esemény alapú tanulás

Mit kezdünk az eseményekkel?

- Korreláció
 - Szolgáltatás leáll- újraindul
- Eszkaláció
- Ok-hatás analízis

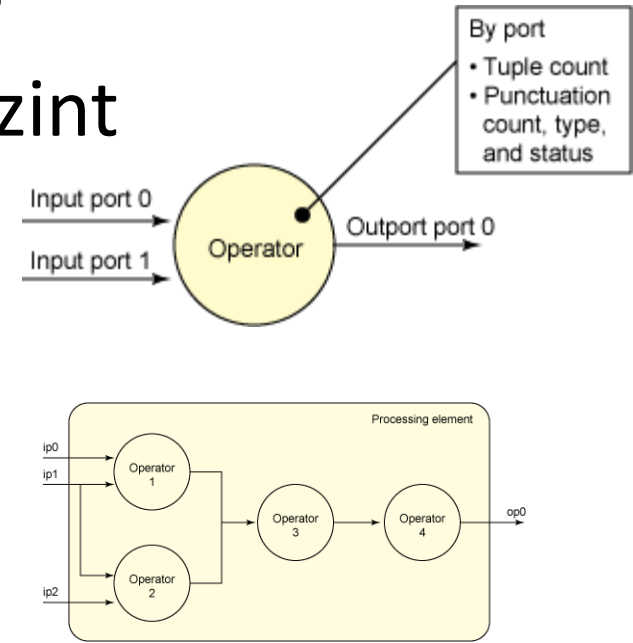


CEP Benchmarking

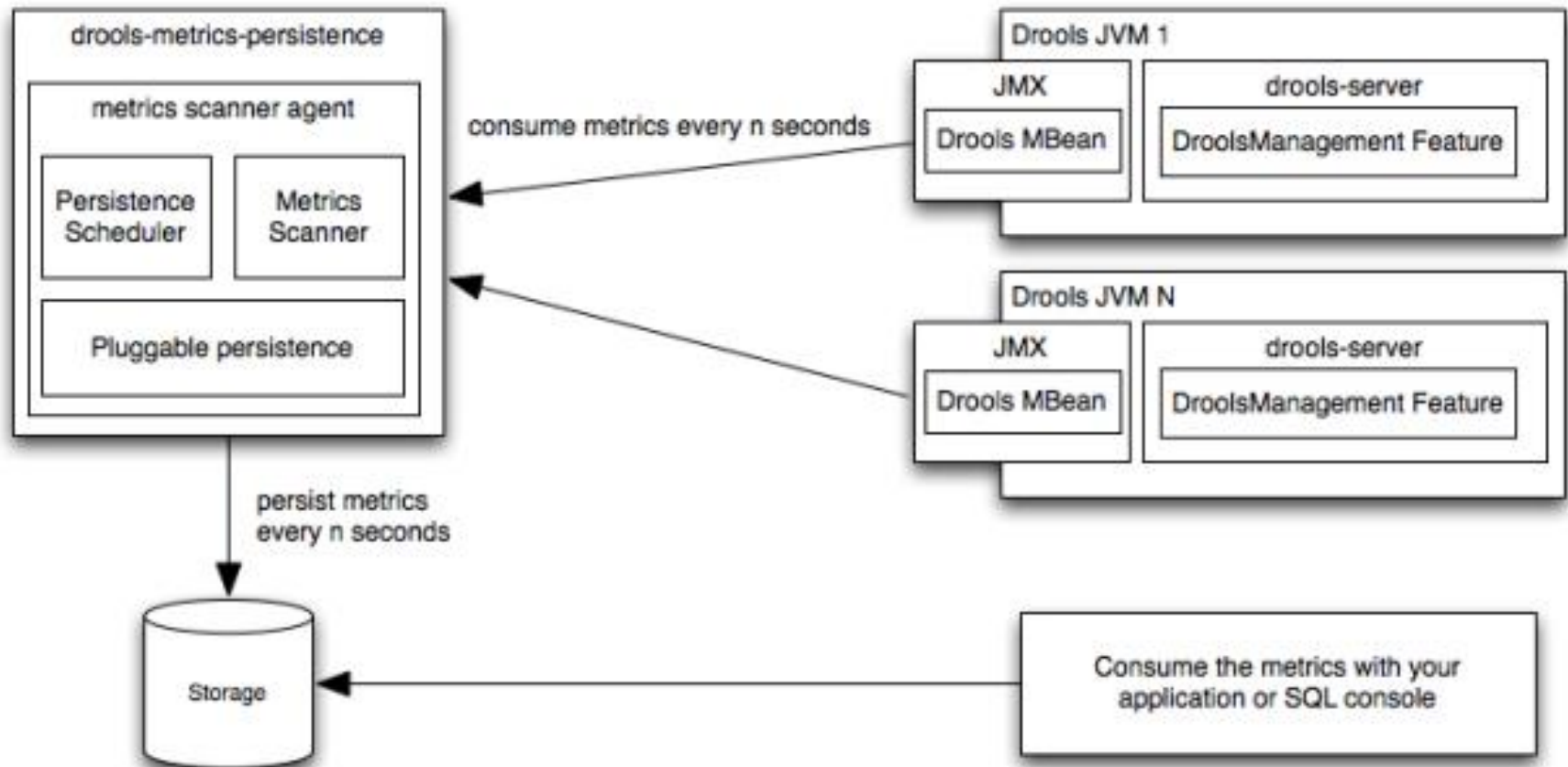
- Kihívások
 - Nagy tömegű eseménynél ritka illeszkedés
 - Feldolgozás/adattovábbítás overhead
 - Kapcsolat a feldolgozott események és az eredmény közt
 - Terhelésgenerálás is lehet szűk keresztmetszet
- Mit mérünk?
 - Áteresztőképesség (bejövő/kimenő események)
 - Válaszidő
 - Skálázhatóság
 - ... elvárt pontosság mellett
 - Mögötte: query plan sharing, átfedő lekérdezések kiszűrése
 - http://bicep.dei.uc.pt/images/3/34/BiCEP_wosp2010.pdf

Hogyan mérjük az eseményfeldolgozást?

- Példa: IBM InfoSphere Streams
- Operátor/feldolgozási egység szint
 - Feldolgozott/eldobott adatok
 - Továbbított adatok
 - Sorhossz
- Feldolgozási egység szintje

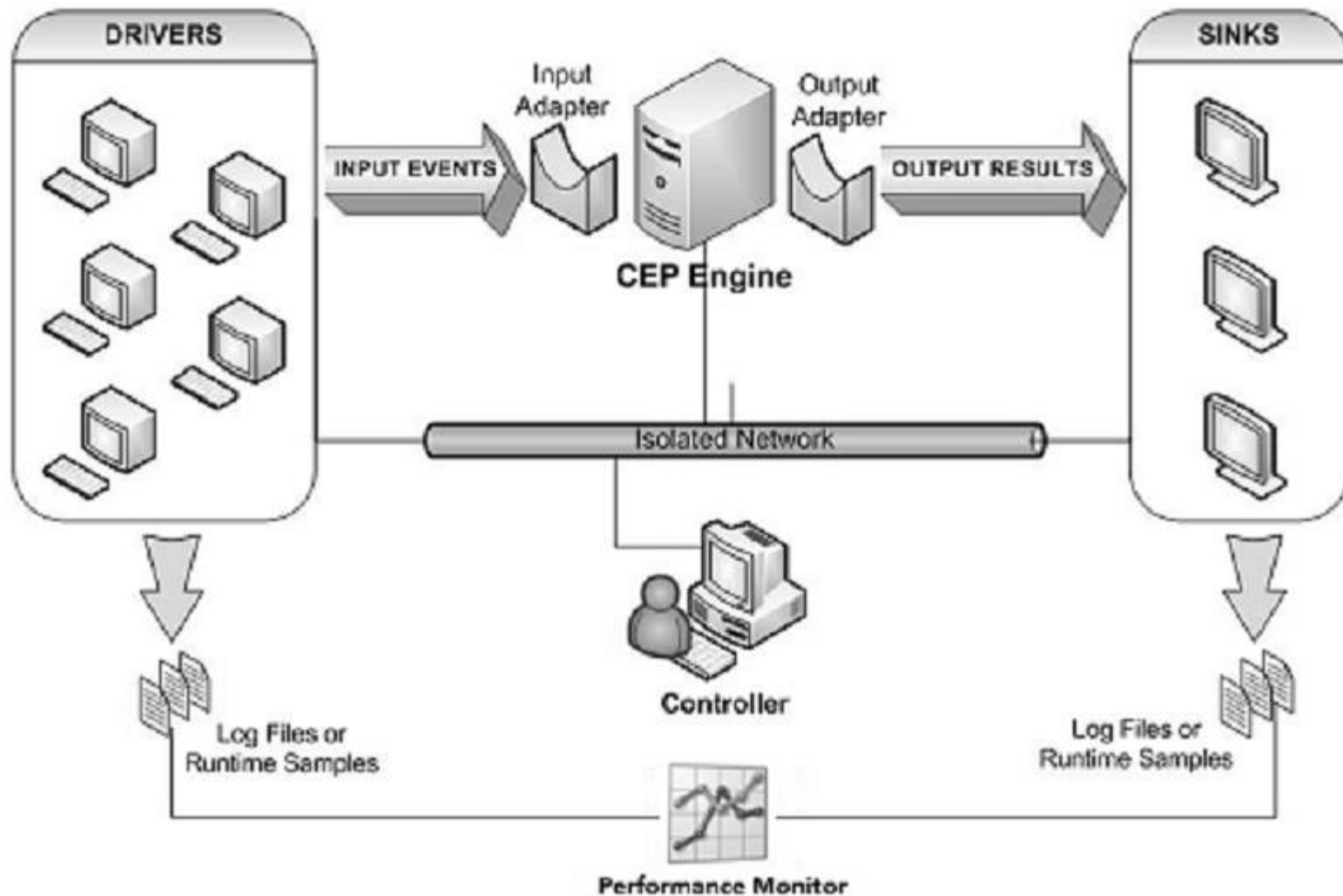


Példa: Drools metrika gyűjtés



<http://lucزامador.wordpress.com/2011/01/07/drools-metrics-persistence/>

Példa: FinCOS

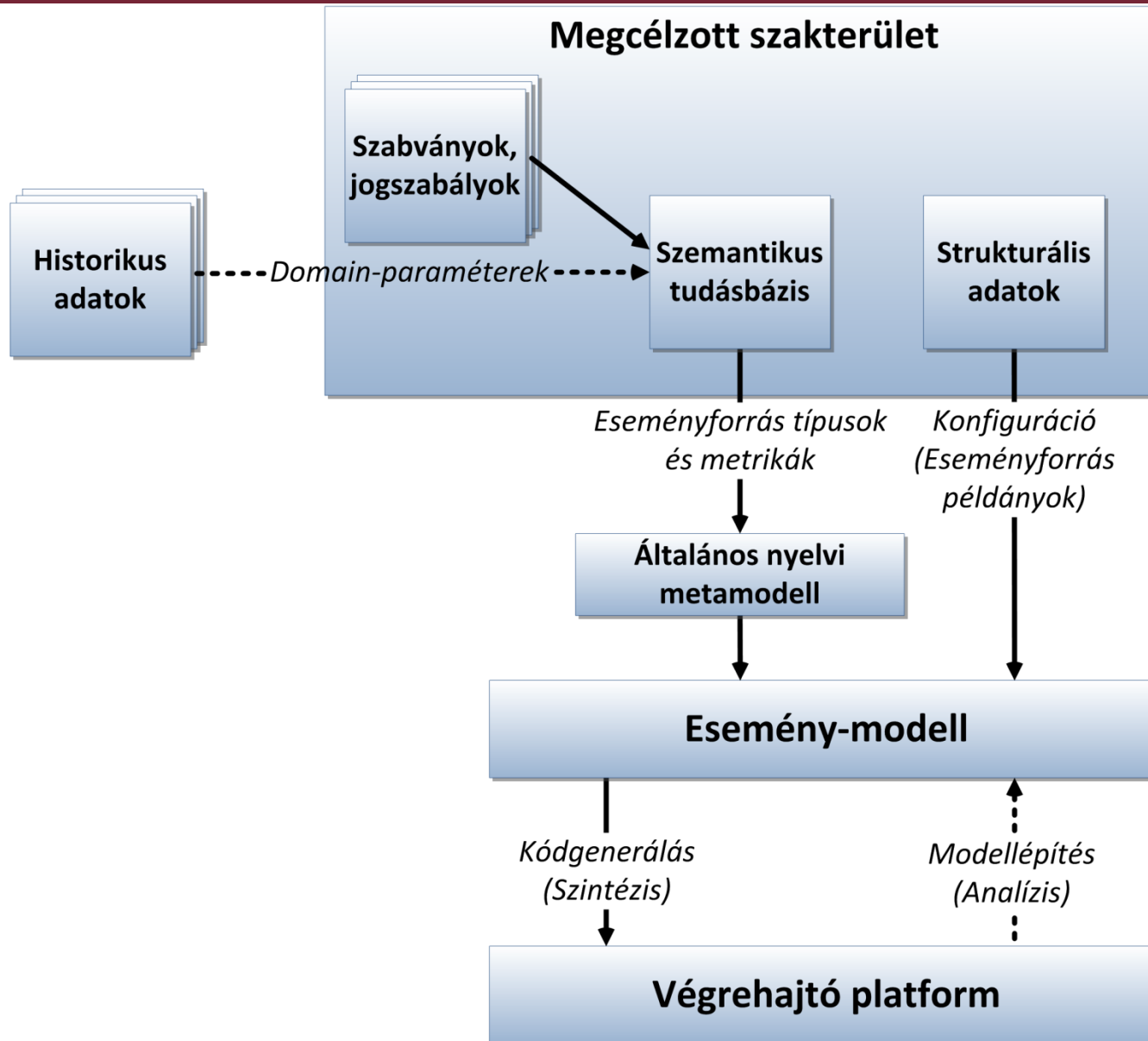


<https://code.google.com/p/fincos/>

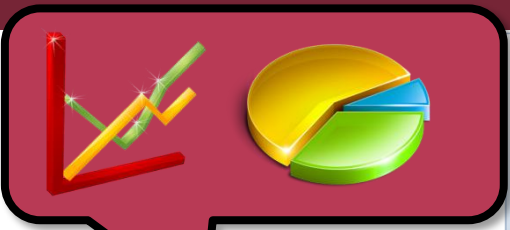
Kihívások

- Események szemantikája
 - Mit jelent? Melyiket figyeljük? (~100 eseményforrás)
 - Milyen kapcsolata van a rendszer dinamikus működésével? (folyamatok)
- Minta alapú tanulás
 - Pl. küszöbértékek hangolása
- Ritka események hatékony azonosítása
- Teljesség? Helyesség?
- Eseményleírás definíciója
 - Nincs egységes szabvány (BEMN, RuleML,)

Példa: modell alapú feldolgozás

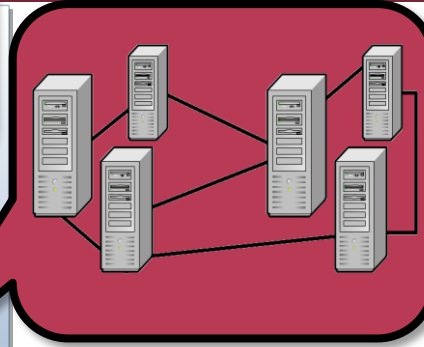


Példa: model-driven analysis



$L_1 = \{$
 „webszerver”,
 „szolgáltatás”
 ...}

$L_2 = \{$
 „kritikus”,
 „bizalmas”
 ...}



Historikus adatok

Szabványok, jogszabályok

Szemantikus tudásbázis

Strukturális adatok

Domain-metaméterek

Eseményforrás típusok és metrikák

Konfiguráció (Eseményforrás példányok)

Általános nyelvi metamodel

$L = \{$
 Event,
 ComplexEvent,
 Source
 ...}

COBIT PO6:
 Gondoskodni arról, hogy a kritikus és bizalmas információkhoz ne lehessen jogosulatlanul hozzáférni.

Esemény-modell

```
select fraud.  
accountNumber as  
aNm, fraud.warning  
as warn  
from  
FraudEvent.win:  
time(30 min) as fd  
...
```

```
Event WebServerCritical {  
source WebServer1  
LiteralMeasurement  
Authenticated [User]  
}
```

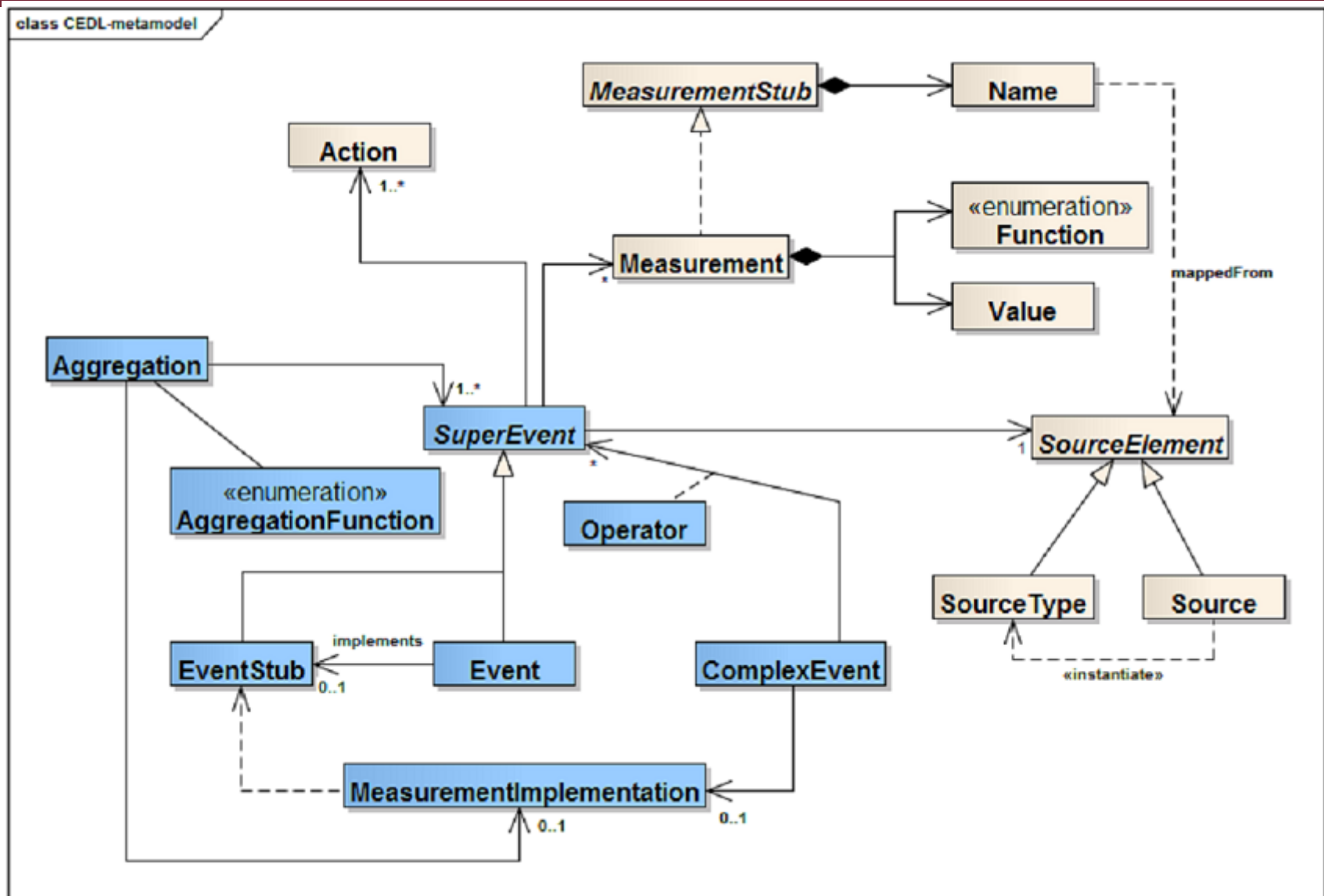
A

Kódgenerálás (Szintézis)

Modellépítés (Analízis)

Végrehajtó platform

Eseményleíró nyelv metamodellje



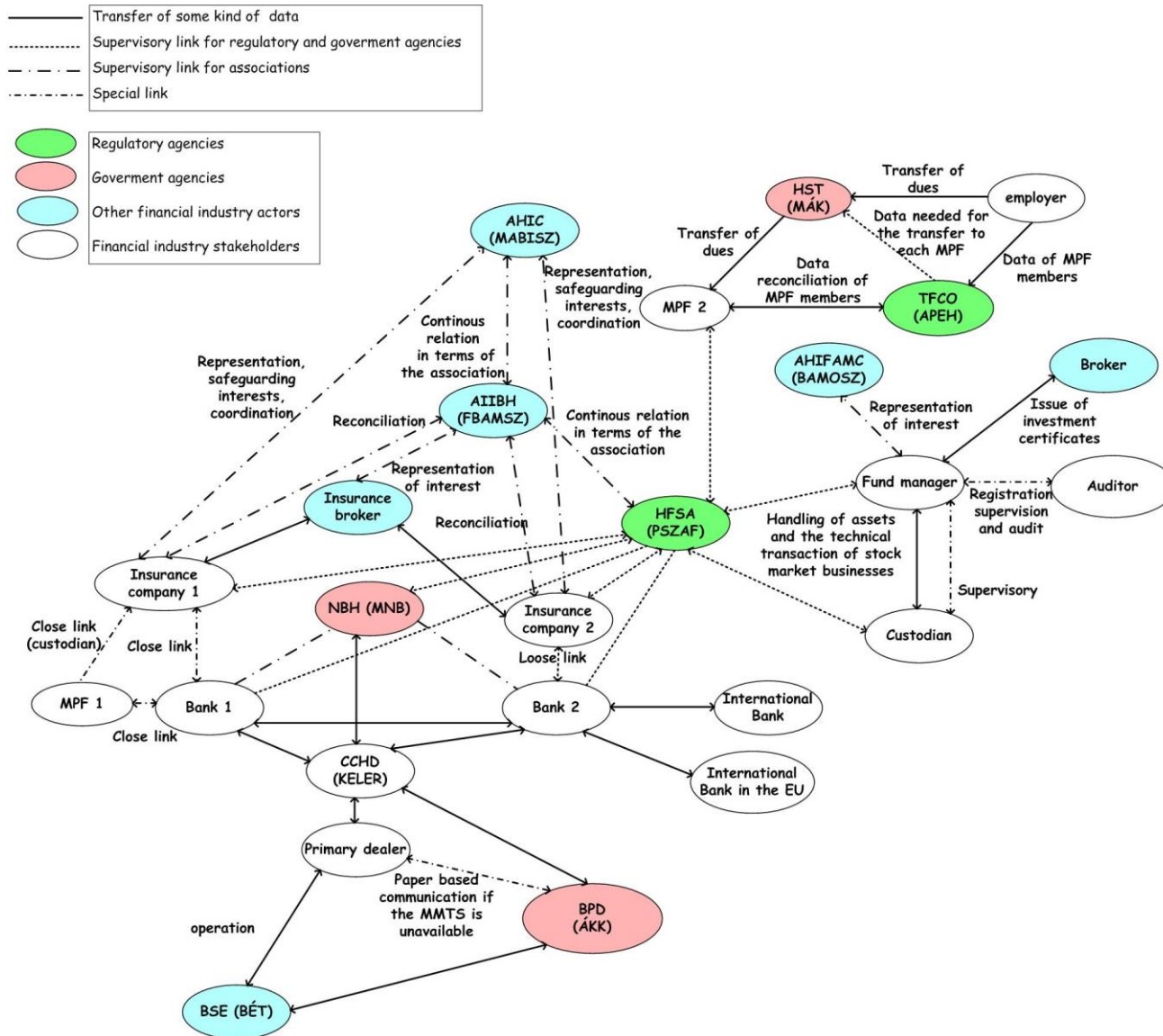
ESETTANULMÁNY: COMIFIN

Szolgáltatásalapú rendszerek, modellvezérelt fejlesztés, komplex eseményfeldolgozás,...

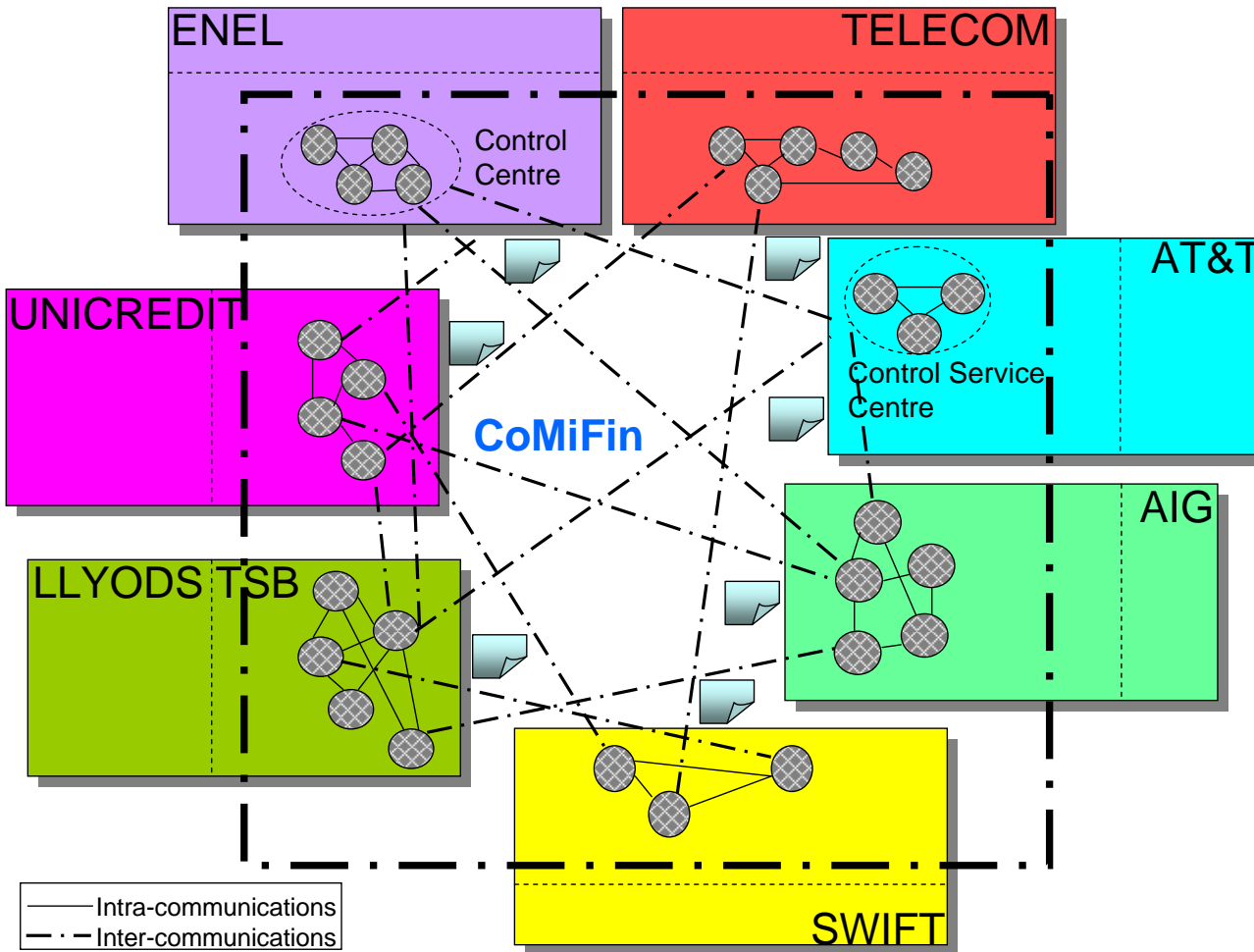
Esettanulmány: CoMiFin

- „Communication Middleware for Financial Infrastructures”
- Motiváció
 - Banki rendszerek egyre erősebben függenek külső szolgáltatóktól
 - Támadások egyre kifinomultabbak
 - Kritikus infrastruktúrák (pl. mobilhálózat, áramellátás, Internet) elleni komplex támadások kivédése
 - Hagyományos kommunikáció lassú (példa: 8 nap egy eset lezárása)
- Cél
 - Scheme to set up and manage a secure environment (software, hardware, monitoring tools, etc.) for information exchange and analysis
- Tanszéki spin-off (OptXware) vezette a demonstrátor fejlesztését

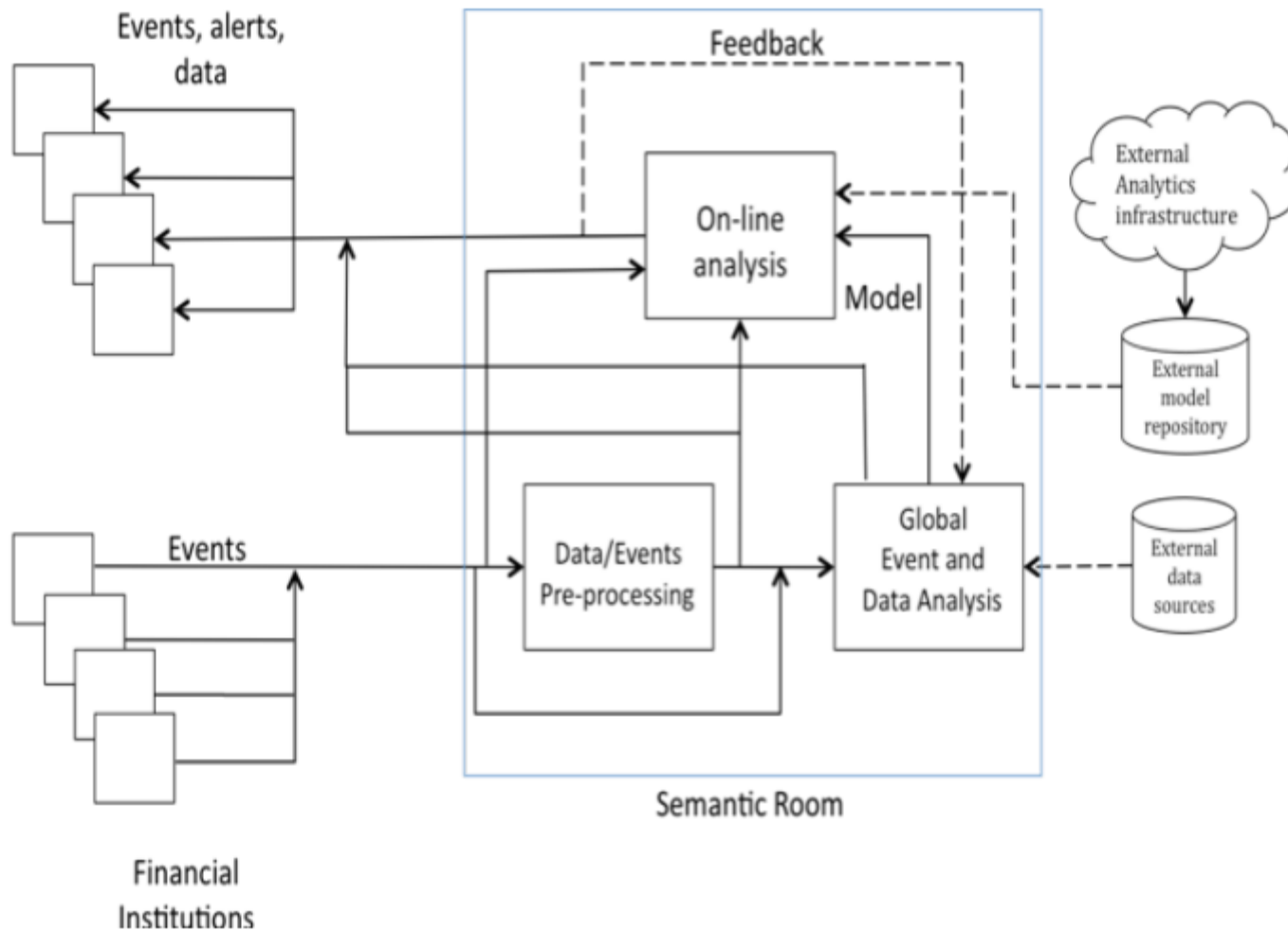
Példa: magyar infrastruktúra



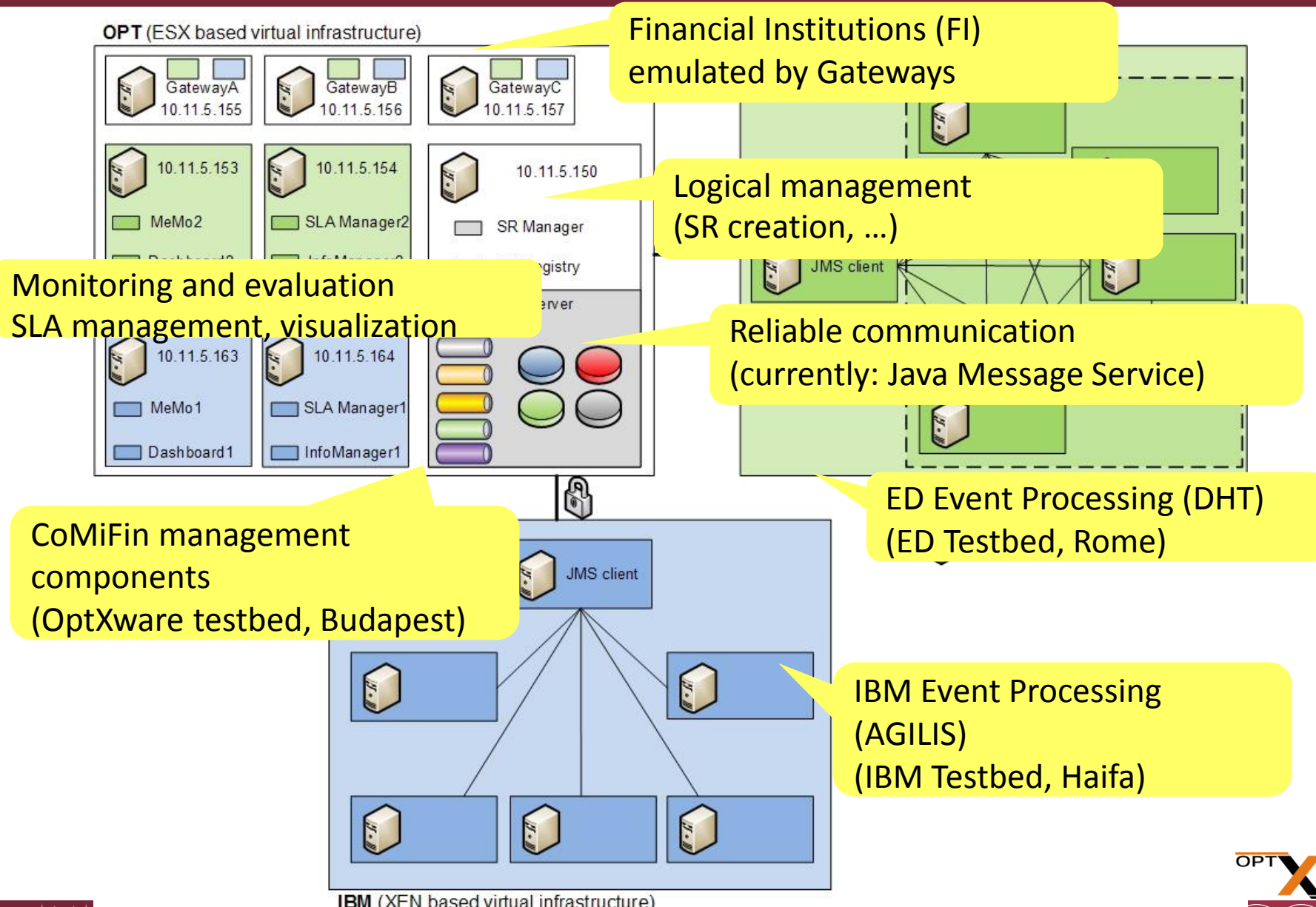
Logikai architektúra



Online adatfeldolgozás (CEP)



Architektúra



Financial Institutions (FI) emulated by Gateways

Logical management (SR creation, ...)

Monitoring and evaluation
SLA management, visualization

Reliable communication (currently: Java Message Service)

ED Event Processing (DHT) (ED Testbed, Rome)

CoMiFin management components (OptXware testbed, Budapest)

IBM Event Processing (AGILIS) (IBM Testbed, Haifa)

Eredmények megjelenítése

The screenshot shows the JBoss Portal 2.7.2-GA interface in Mozilla Firefox. The browser address bar shows the URL: http://10.11.5.153:8080/portal/auth/portal/Dashboard/6_AlertList. The page title is "AlertList". The user is logged in as "asi_sr_manager". The table below displays a list of alerts with the following columns: Date, Origin, Participating FIs, Type, Description, Affected Services, Suspicious FIs, and Priority.

Date	Origin	Participating FIs	Type	Description	Affected Services	Suspicious FIs	Priority
2010-07-01 10:07:43.0	DHT Analytics	Bank of Vanyar	ALERTMitM	Statistical anomaly detected	Service15	1X.16X.XX.X89	-0.1799294
2010-07-01 10:07:43.0	DHT Analytics	Bank of Vanyar	ALERTMitM	Statistical anomaly detected	Service15	13X.X5X.XX.X5	-0.181737
2010-07-01 10:07:37.0	DHT Analytics	Bank of Vanyar	ALERTMitM	Statistical anomaly detected	Service12	6X.8X.X0X.X	-0.1778012
2010-07-01 10:07:37.0	DHT Analytics	Bank of Vanyar	ALERTMitM	Statistical anomaly detected	Service4	17X.XX.X3X.X29	-0.1718082
2010-07-01 10:07:32.0	DHT Analytics	Bank of Vanyar	ALERTMitM	Statistical anomaly detected	Service8	20X.XX.X4X.X5	-0.1802342
2010-07-01 10:07:31.0	DHT Analytics	Bank of Vanyar	ALERTMitM	Statistical anomaly detected	Service8	21X.XX.XX.X79	-0.175243
2010-07-01 10:07:27.0	DHT Analytics	Bank of Vanyar	ALERTMitM	Statistical anomaly detected	Service11	19X.X1X.X0X.X05	-0.183489
2010-07-01 10:07:27.0	DHT Analytics	Bank of Vanyar	ALERTMitM	Statistical anomaly detected	Service11	10X.X2X.X1X.X02	-0.1774248
2010-07-01 10:07:26.0	DHT Analytics	Bank of Vanyar	ALERTMitM	Statistical anomaly detected	Service10	6X.13X.X2X.X73	-0.1799292
2010-07-01 10:07:26.0	DHT Analytics	Bank of Vanyar	ALERTMitM	Statistical anomaly detected	Service10	9X.16X.X5X.X24	-0.182237
2010-07-01 10:07:09.0	DHT Analytics	Bank of Noldor	ALERTMitM	Statistical anomaly detected	Service11	19X.X1X.X0X.X05	-0.18592
2010-07-01 10:07:08.0	DHT Analytics	Bank of Noldor	ALERTMitM	Statistical anomaly detected	Service11	19X.X1X.X0X.X05	-0.1923068
2010-07-01 10:07:08.0	DHT Analytics	Bank of Noldor	ALERTMitM	Statistical anomaly detected	Service10	6X.13X.X2X.X73	-0.1886134
2010-07-01 10:07:08.0	DHT Analytics	Bank of Noldor	ALERTMitM	Statistical anomaly detected	Service10	9X.16X.X5X.X24	-0.1909274
2010-07-01 10:07:06.0	DHT Analytics	Bank of Vanyar	ALERTMitM	Statistical anomaly detected	Service8	19X.XX.XX.X79	-0.1778006
2010-07-01 10:07:05.0	DHT Analytics	Bank of Vanyar	ALERTMitM	Statistical anomaly detected	Service14	8X.20X.X4X.X6	-0.1798036

Alert details (time, source, target, etc.)

Service effected

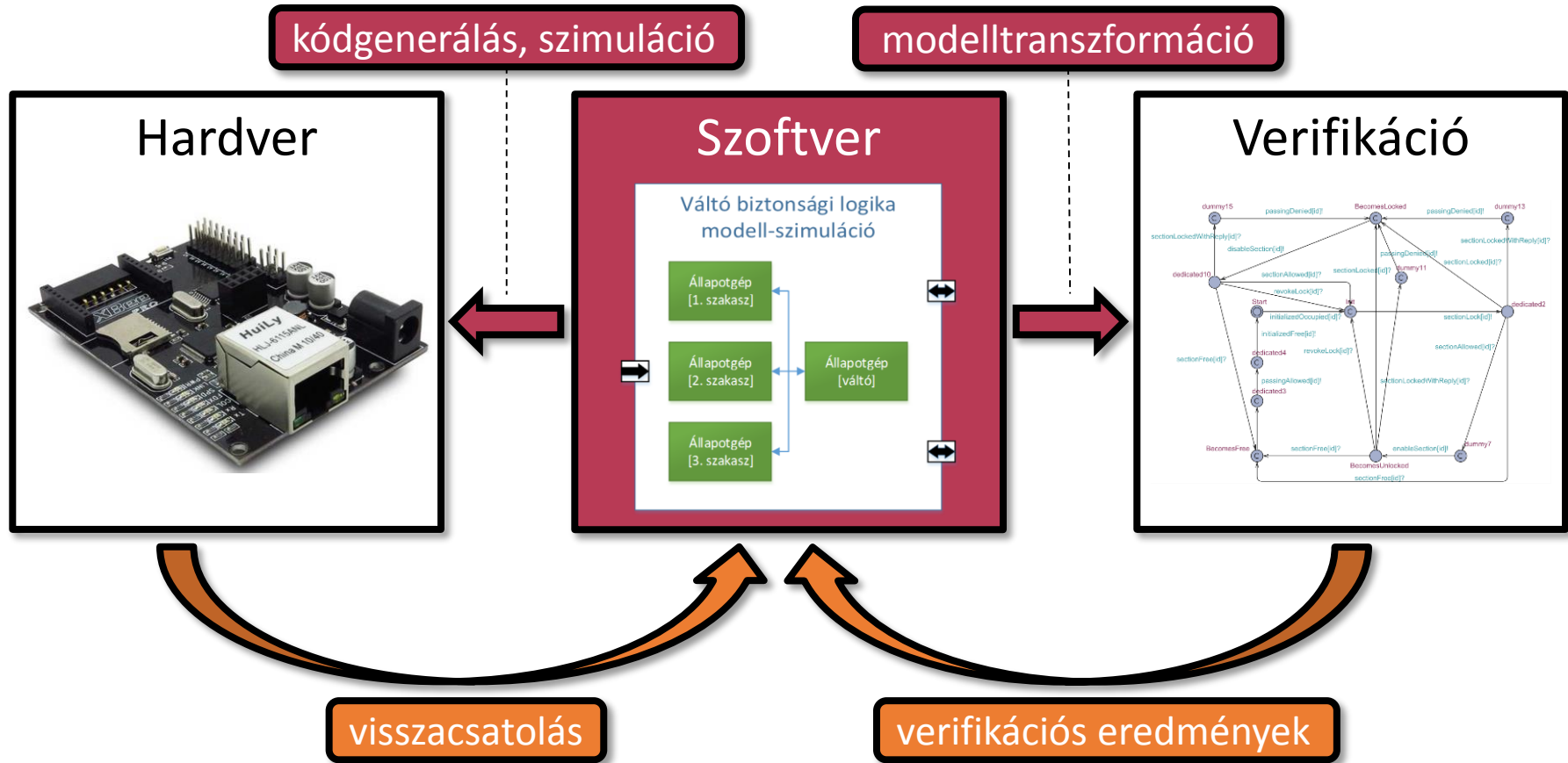
Score on the alert

Collaborative Financial Infrastructure Protection: Tools, Abstractions, and Middleware
Springer, 2012.



ESETTANULMÁNY

Modellvezérelt szoftverfejlesztés



Horváth Benedek, Konnerth Raimund-Andreas, Mázló Zsolt
Elosztott biztonságkritikus rendszerek modellvezérelt fejlesztése, TDK 2014

Hardver architektúra

TDK 2015:

Értékeljük ki a rendszer működését a szenzorjelek alapján (CEP...)

- Hol járnak a vonatok (kamera)?*
- Lesz-e veszélyhelyzet?*

Hierarchikus futásidejű verifikáció kritikus kiberfizikai rendszerekhez, TDK2015

Hegy Bálint Bence, Balogh László Márk, Deé Flórián