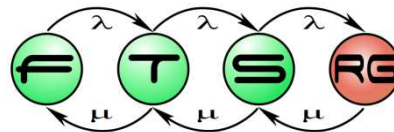


Szabályalapú üzleti logika, eseményfeldolgozás

Gönczy László
Bergmann Gábor és az OptXware Kft. anyagainak
felhasználásával



Tartalom

- Produkciós rendszerek alapfogalmai
- Üzleti szabályrendszerek
- Komplex események feldolgozása (CEP)
- Esettanulmány

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 /
produkciós

Tiszta logikai

Üzleti
szabálmotor

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, küldjünk rá ellenőrt”

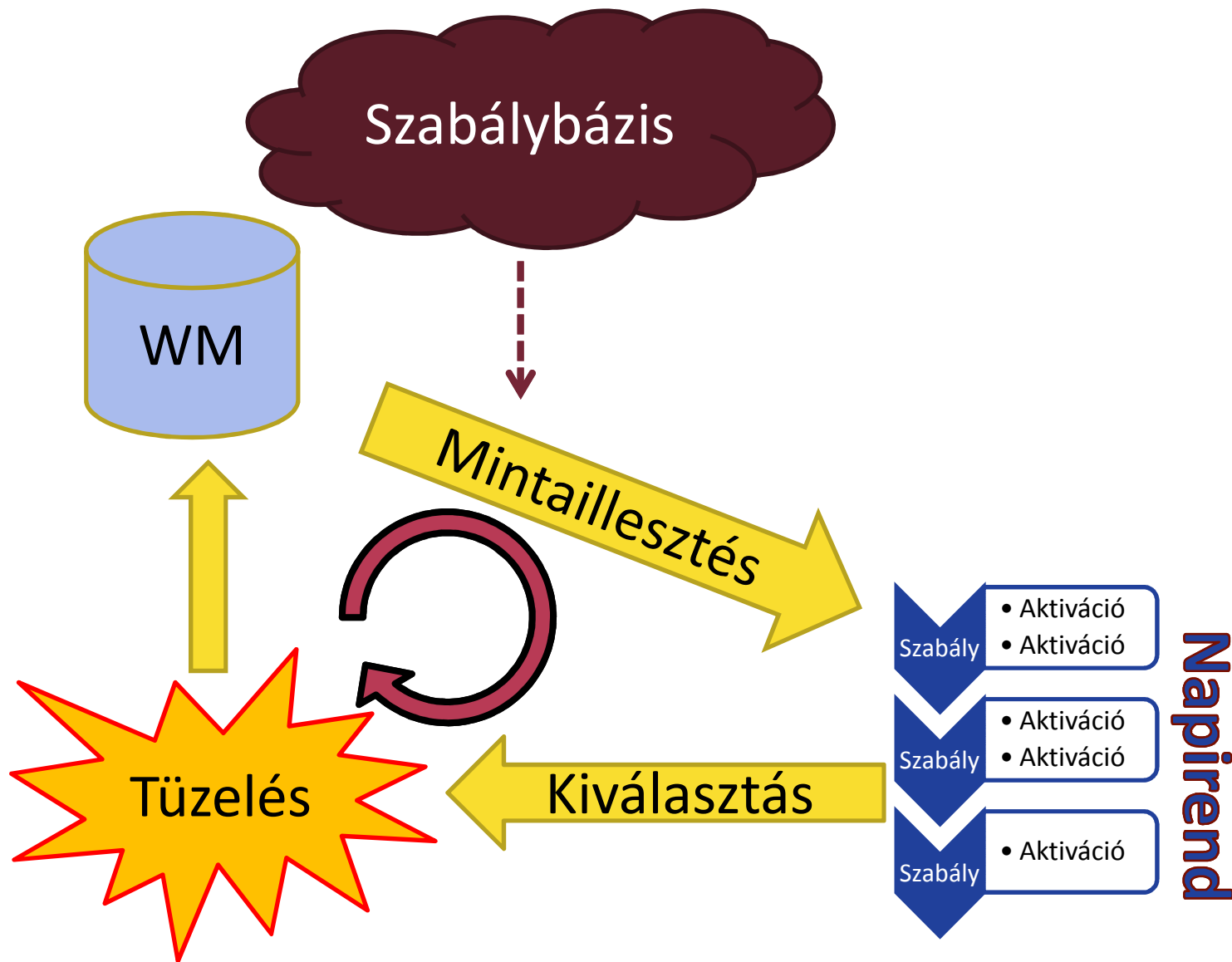
Következtetés

- **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

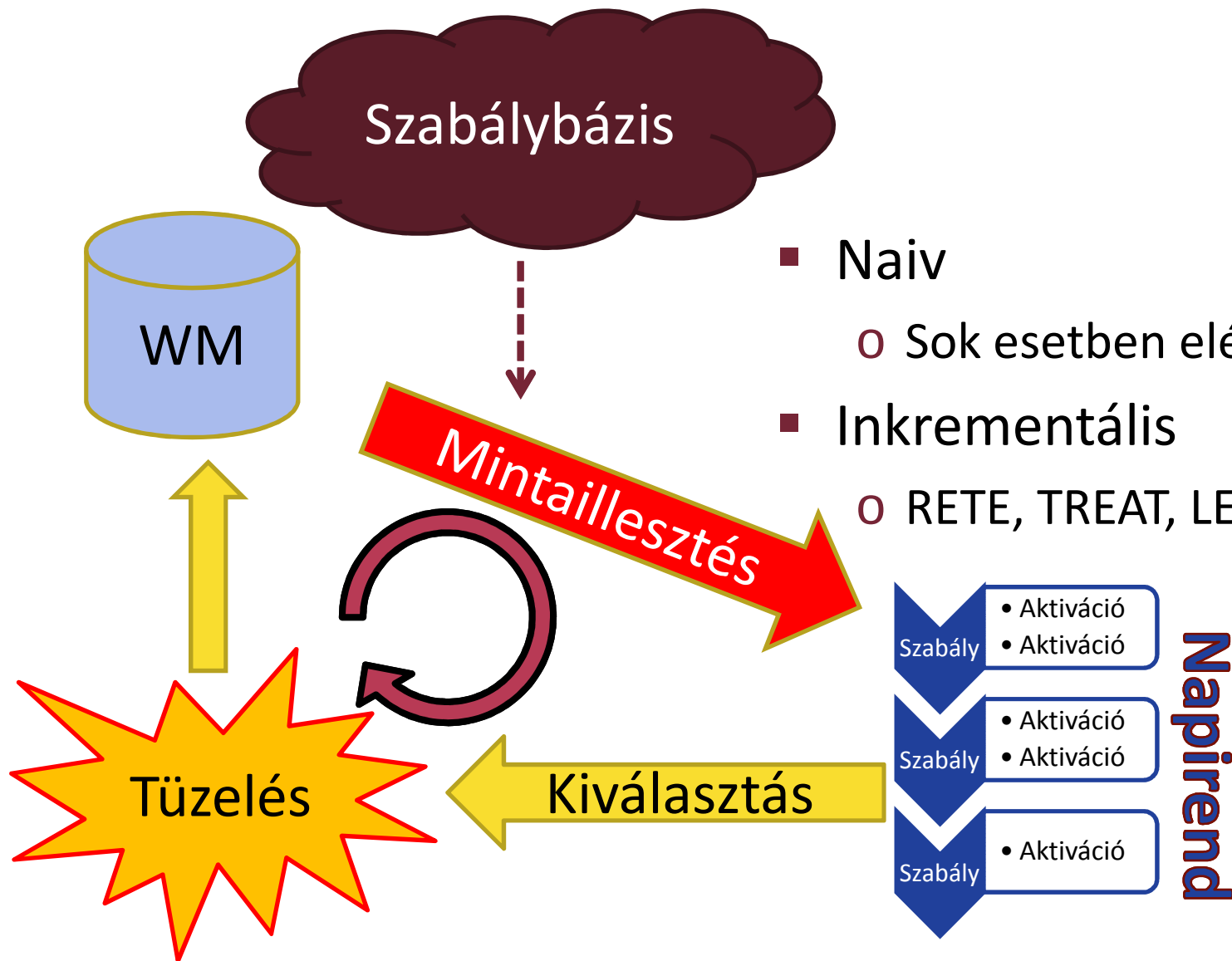
Produkciós rendszer fogalomtár

- 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

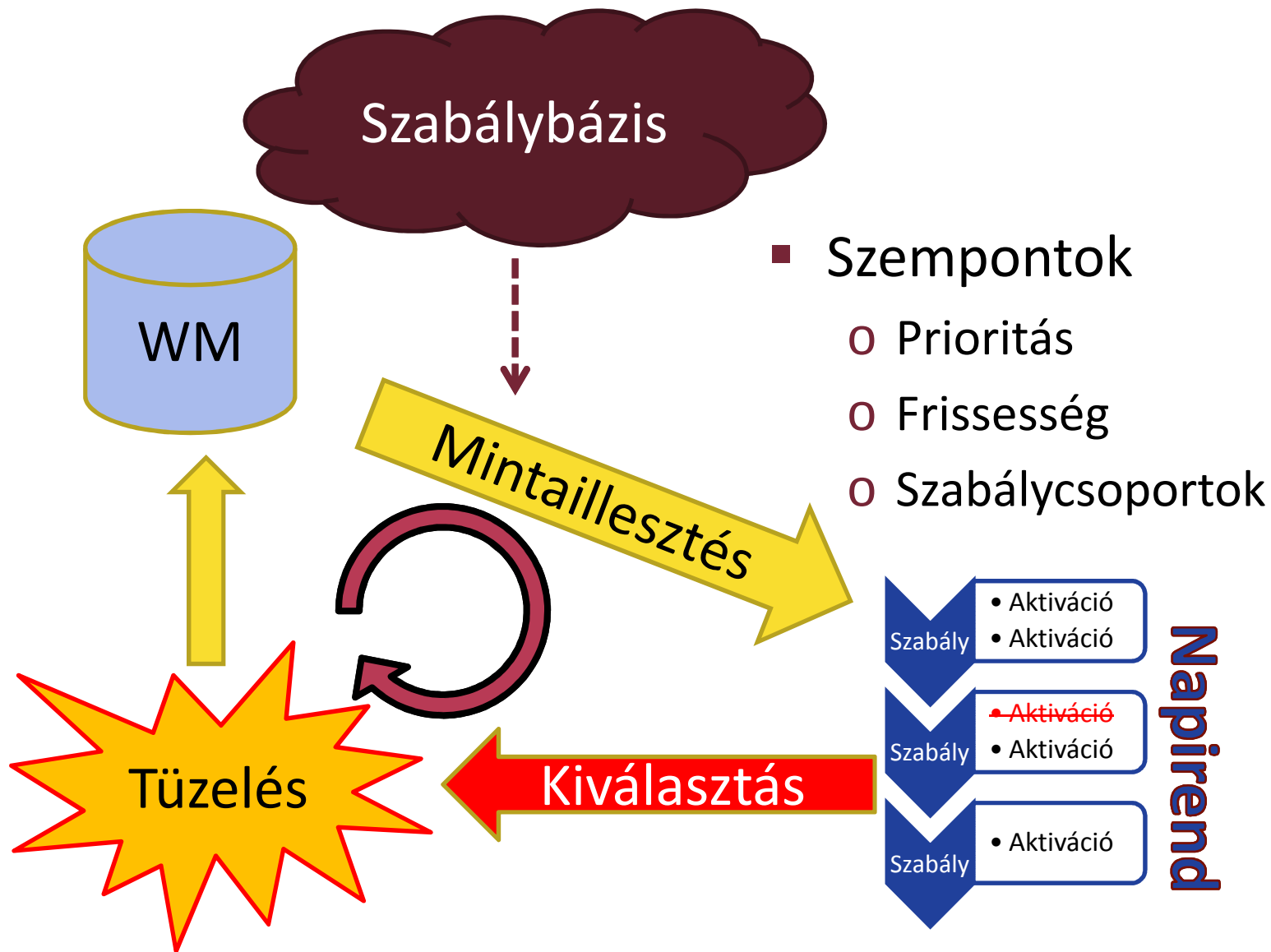
Egyszerű produkciós rendszer



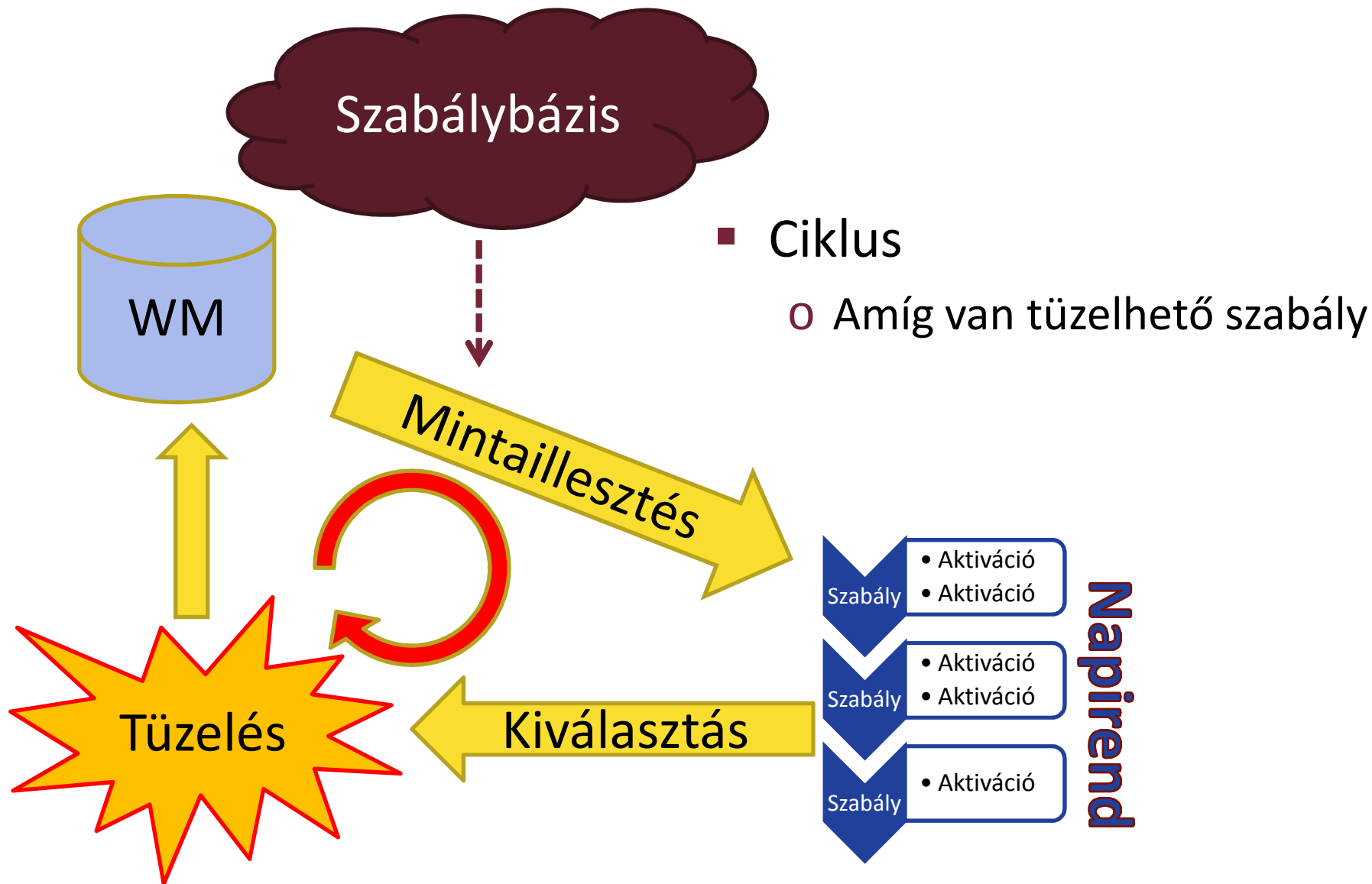
Egyszerű produkciós rendszer



Egyszerű produkciós rendszer



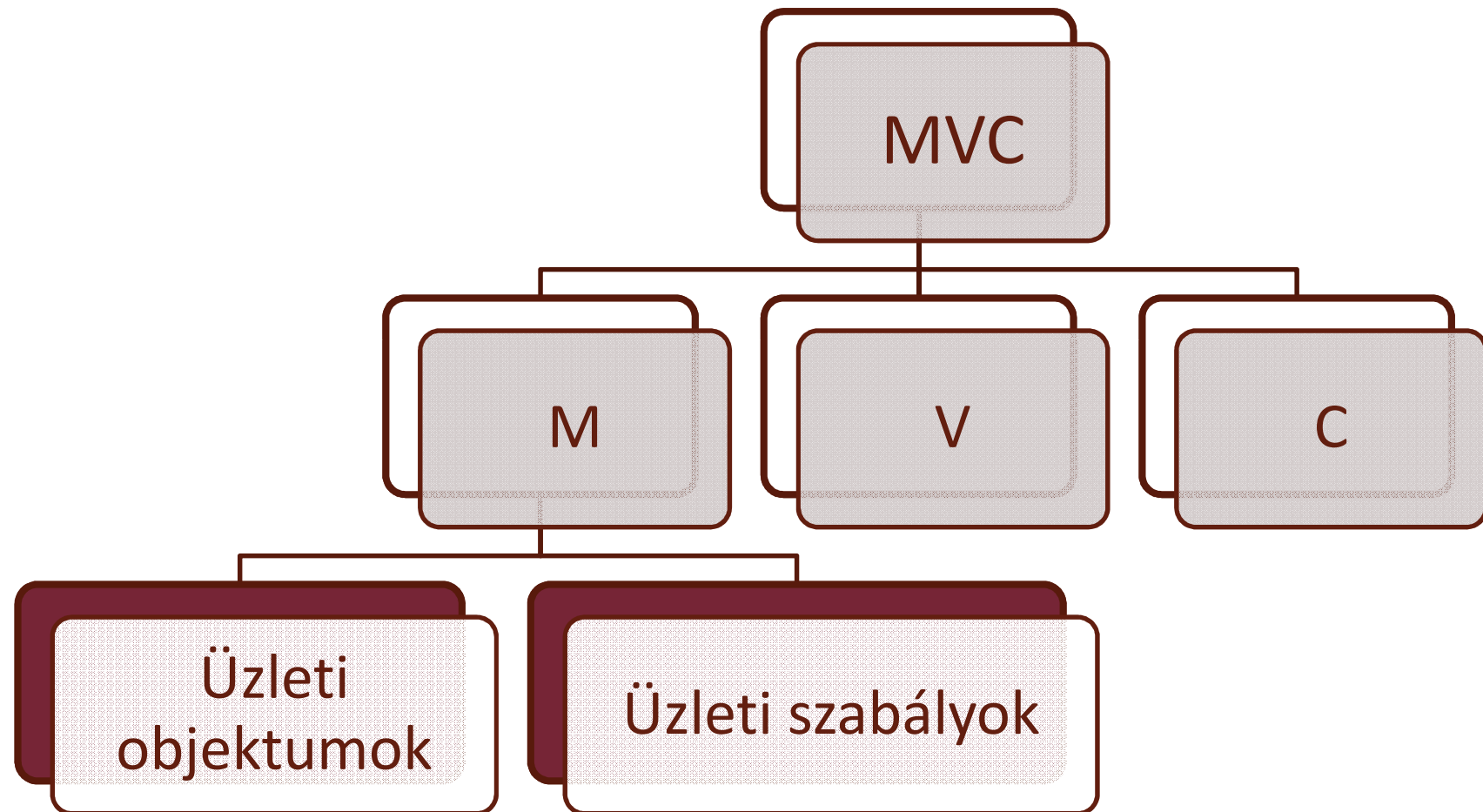
Egyszerű produkciós rendszer



Üzleti szabályrendszerek

Business Rule Systems

- Szabály alapú üzleti logika



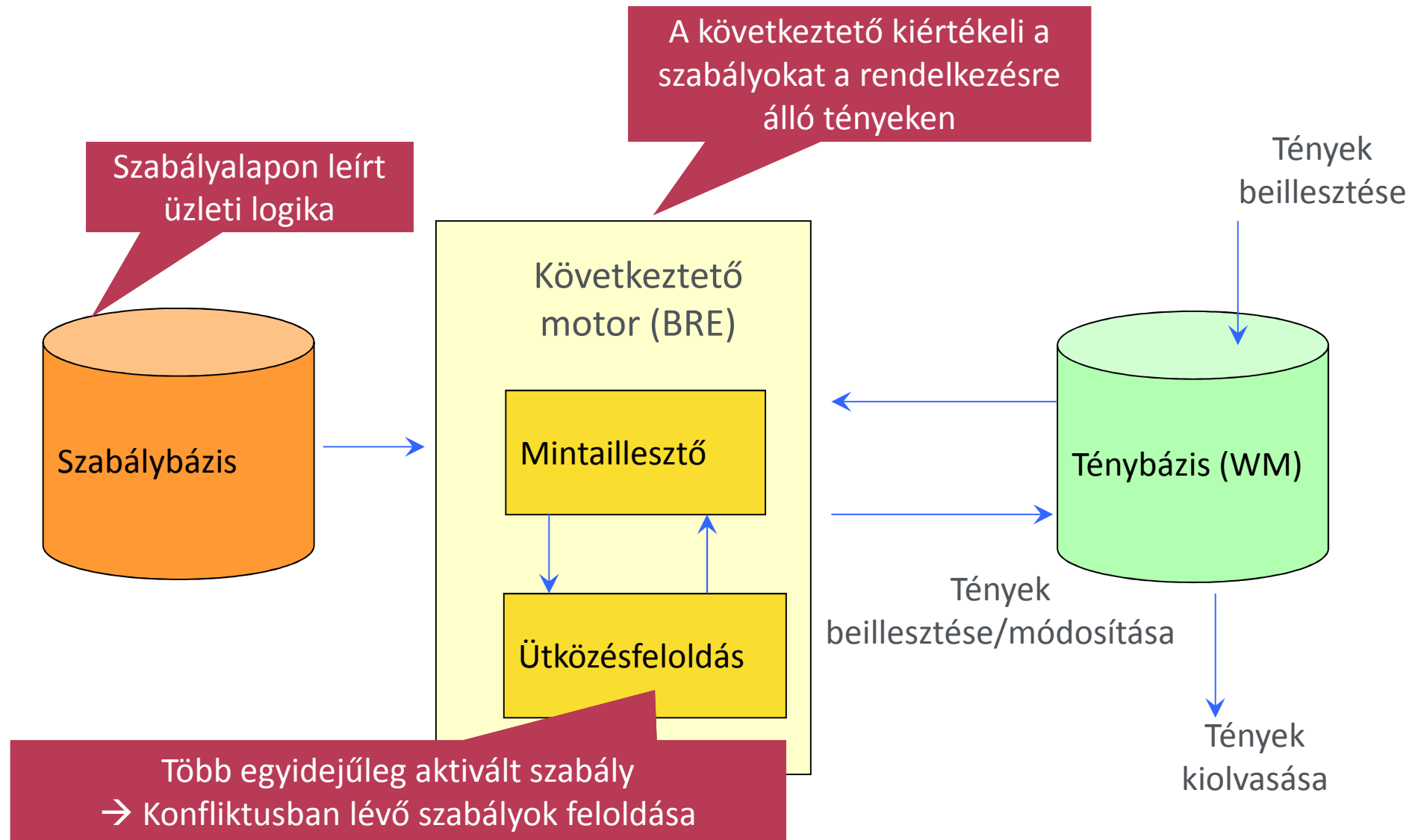
Üzleti szabályok

- **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

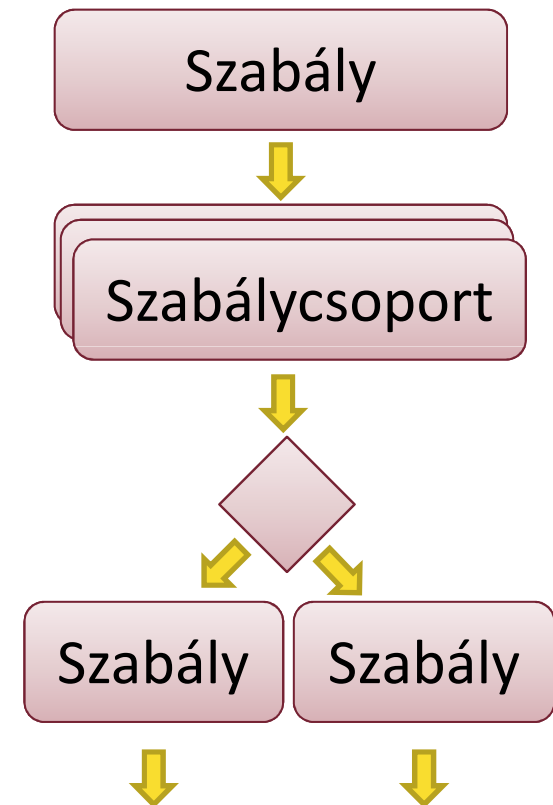
- Üzleti szabályok produkciós rendszer szemszögéből
 - „Tények” → üzleti objektumok
 - Kvázi produkciós szabályok, de RHS tetszőleges **akció**
 - Nem (feltétlen) logikai következtetés
 - Érvénytelenné váló feltétel, akció hatása mégis megmarad
 - Egy aktiváció többször is tüzelhet (pl. addig jár a korsó...)
- 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 szabálmotor működése



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

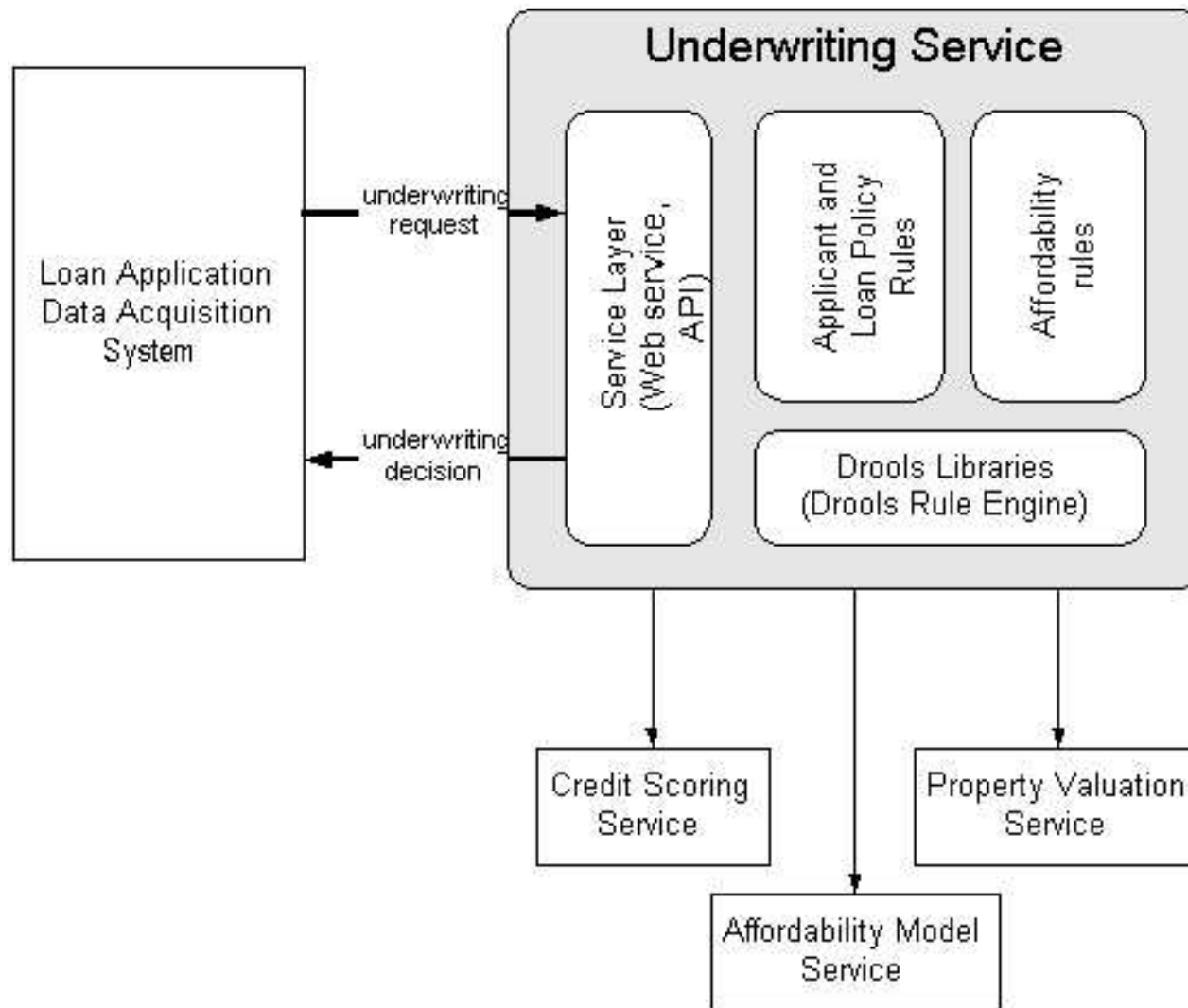
```
rule "We have an honest Politician"  
  salience 10  
  when  
    exists( Politician( honest == true ) )  
  then  
    insertLogical( new Hope() );  
end
```

```
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 = **B**usiness **R**ule **M**anagement **S**ystem
- BRE + kapcsolódó szolgáltatások
- Számos termék
 - *G2, JBoss Rules (**Drools**), IBM ILOG (J)Rules, Blaze Advisor, MS BRE, TIBCO iProcess, stb.*

Microsoft®



BRMS – szolgáltatások

- **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

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	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	30	1	risk assessment		Safe and mature	Old driver Approved
12						
13						
14						
15						
16						

Kitekintés: komplex eseményfeldolgozás (CEP)

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
- Elosztott adatforrások
 - Adatbázisok, beérkező kérések, megfigyelt események, stb.
- Skálázhatóság
 - Cloud környezet

CEP alkalmazási területek

- Üzleti alkalmazások
 - Tőzsde, befektetések
 - „Treasury”
 - Kockázatkértékelés
 - 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/>

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
    
```

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
 - Áteresztőképeség
 - 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

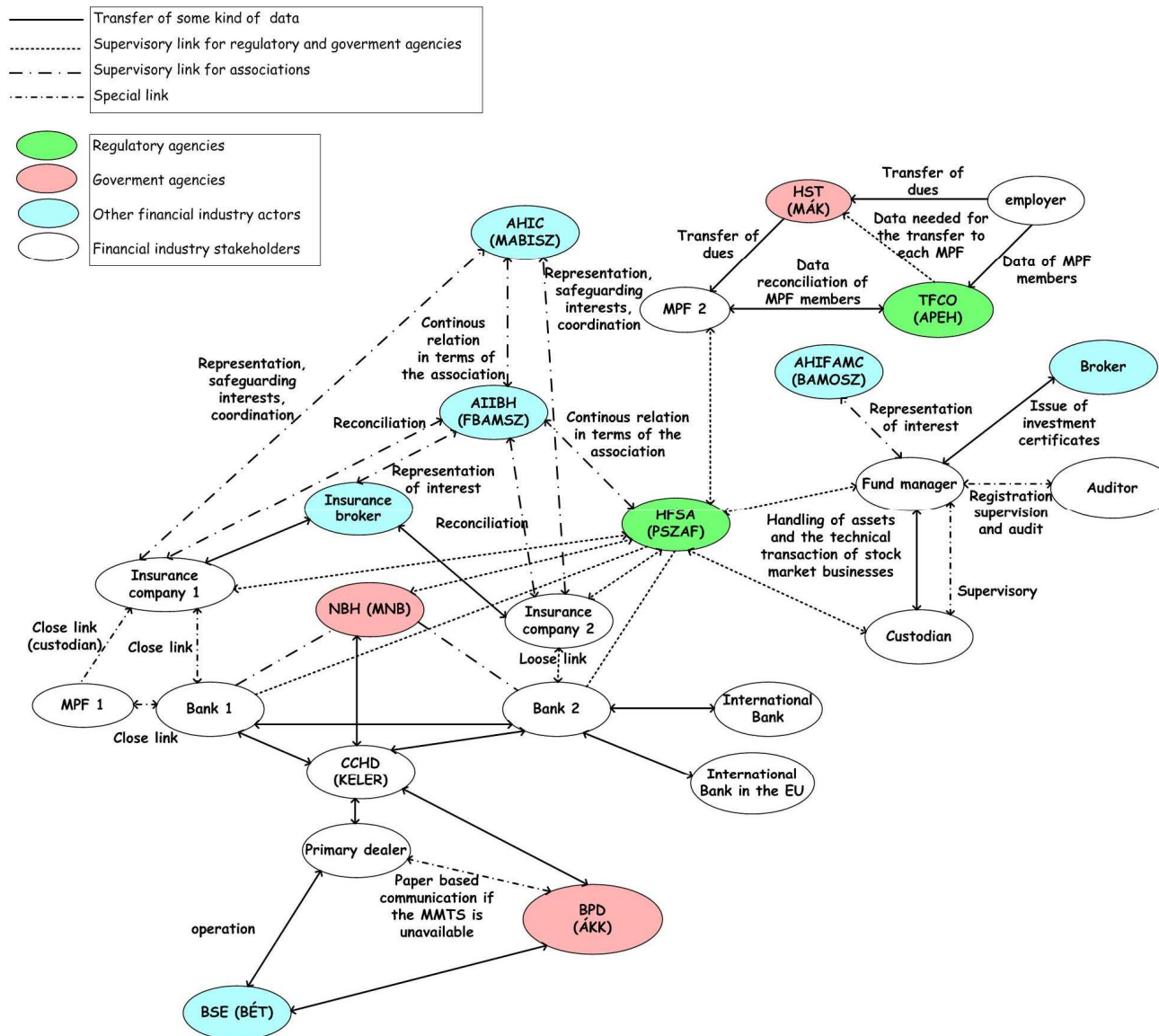
Esettanulmány: CoMiFin

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

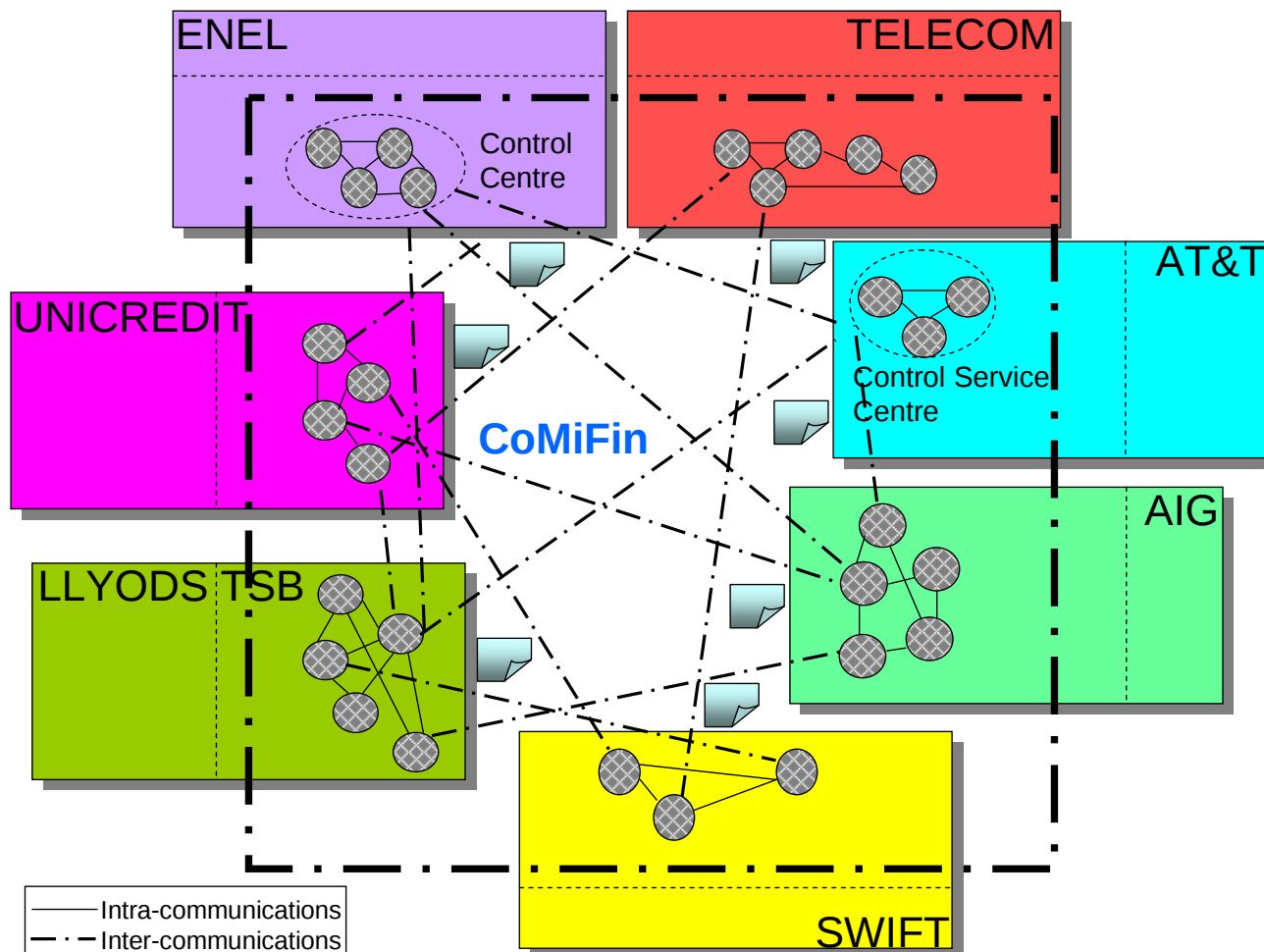
Esettanulmány: CoMiFin

- „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

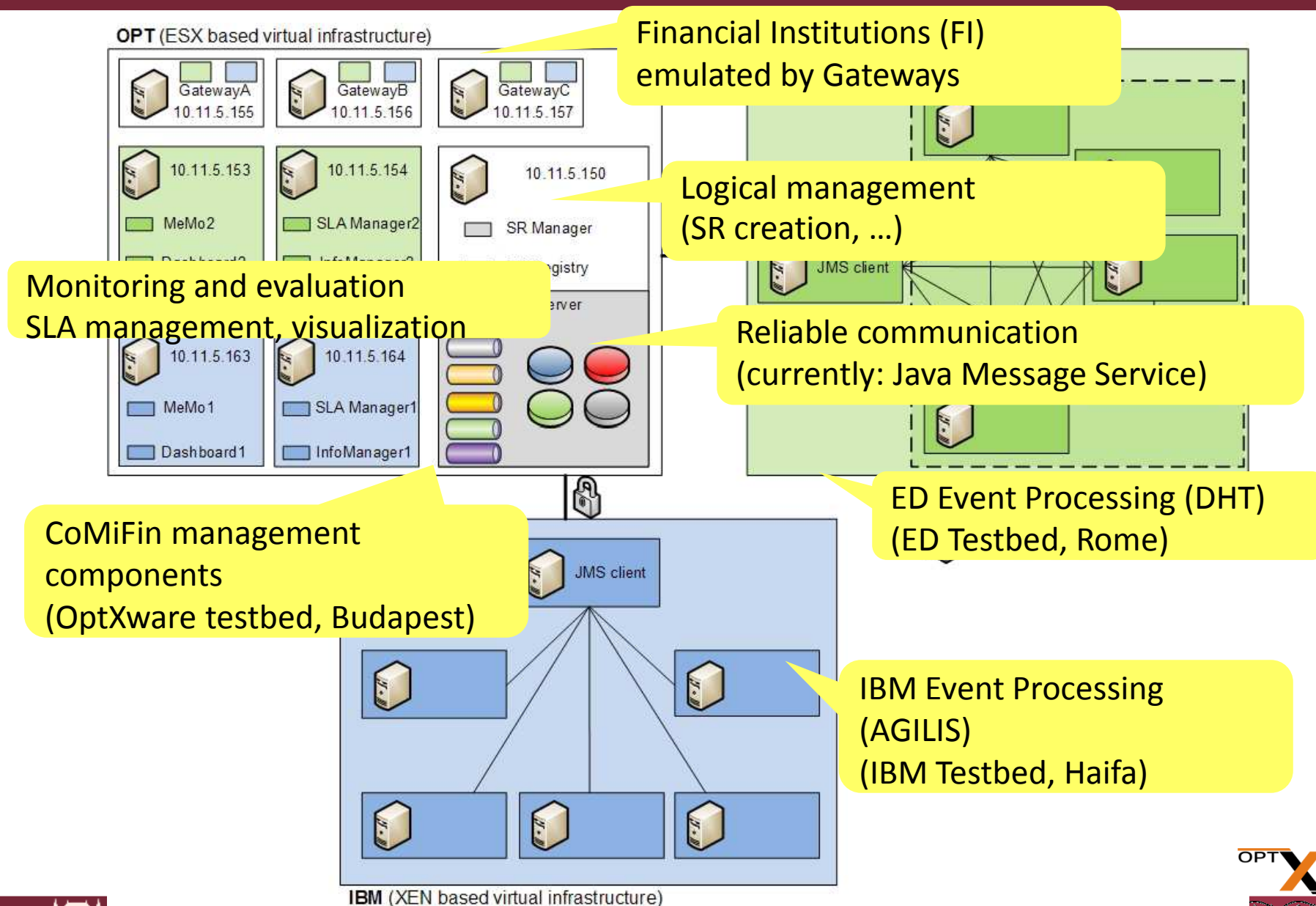
Példa: magyar infrastruktúra



Logikai architektúra



Architektúra



Eredmények megjelenítése

JBoss Portal 2.7.2-GA - Mozilla Firefox

http://10.11.5.153:8080/portal/auth/portal/Dashboard/6_AlertList

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

Logged in as: sr_manager

Home S_SR_Manager_Page 6_AlertList

AlertList

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	-0.1909274
2010-07-01 10:07:09.0	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

Service effected

Score on the alert

Done