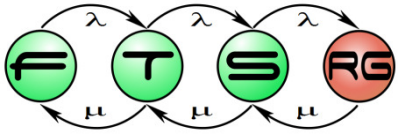
Szabályalapú rendszerek, eseményfeldolgozás

Gönczy László
gonczy@mit.bme.hu
Az OptXware Kft. , Bergmann Gábor és Dávid István anyagainak felhasználásával



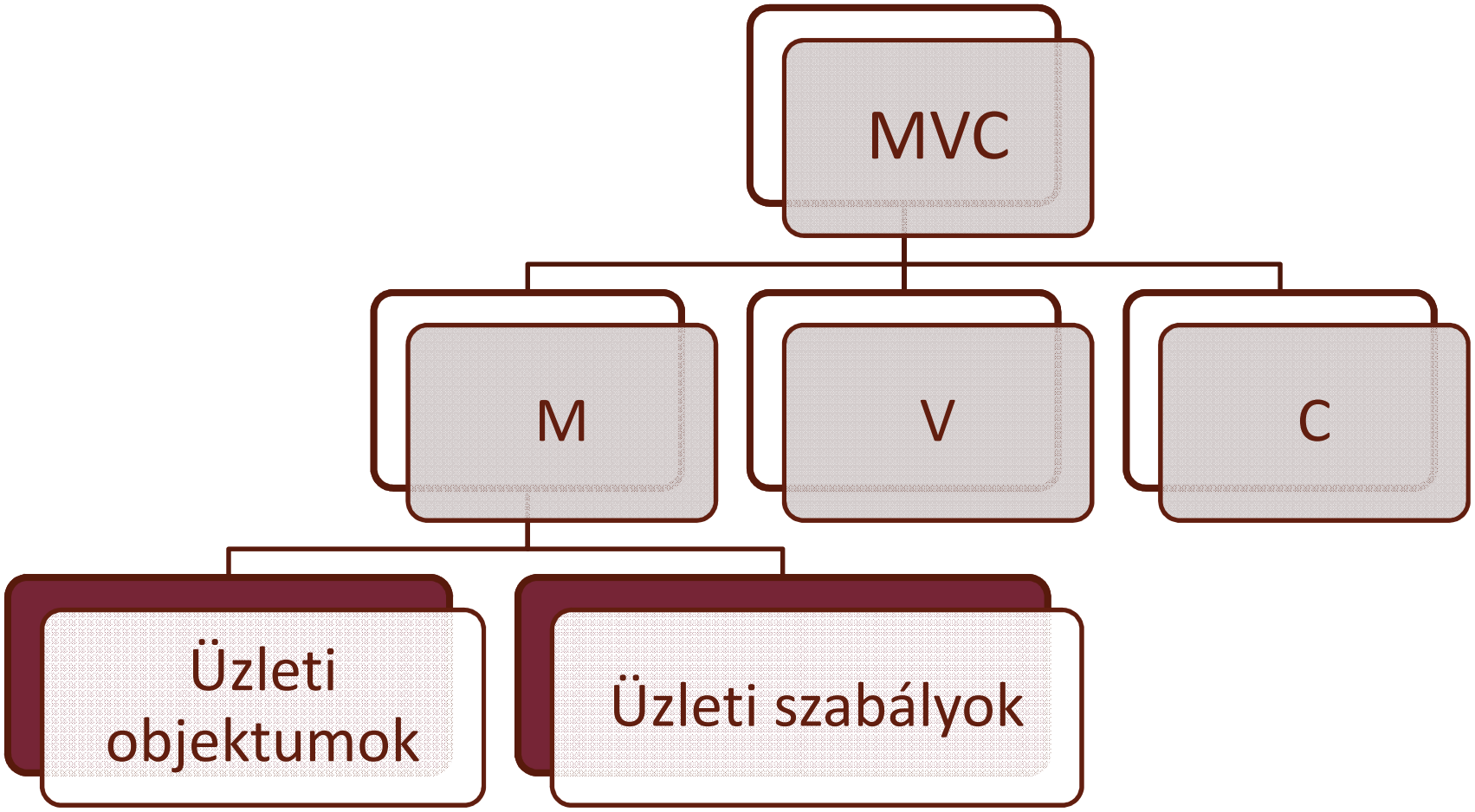
Tartalom

- Szabályalapú logika
- Szabályalapú megközelítés felhasználása: komplex eseményfeldolgozó rendszerek
- Esettanulmány: CoMiFin
- Szabályok modellalapú tervezése

Szabályalapú üzleti logika

Business Rule Systems

- Szabály alapú üzleti logika



Üzleti szabályok

- **Üzleti logika** „kiszervezésére” végrehajtató modell
- **Üzleti objektumok**at 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”

Szabály alapú üzleti logika előnyei

- Dedikált szabálytár
  - 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
- Hatékony végrehajtás
- Cserélhető körülötte az architektúra
- „Externalizáció”, karbantarthatóság
- Eszköztámogatás

Felhasználási területek - példák

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

BRMS – szolgáltatások

- **Szabálytár**
  - Kereshető, automatizáltan módosítható
  - Verziózás
- **Végrehajtó** könyvtár, végrehajtó szerver, SOA
- **Tool support**
  - IDE, webes felület
  - Template lehetőség
  - Magasabb granularitású szabályok
  - Tesztelési támogatás, gyors próba
  - Üzleti szótár építése meglévő adatokból

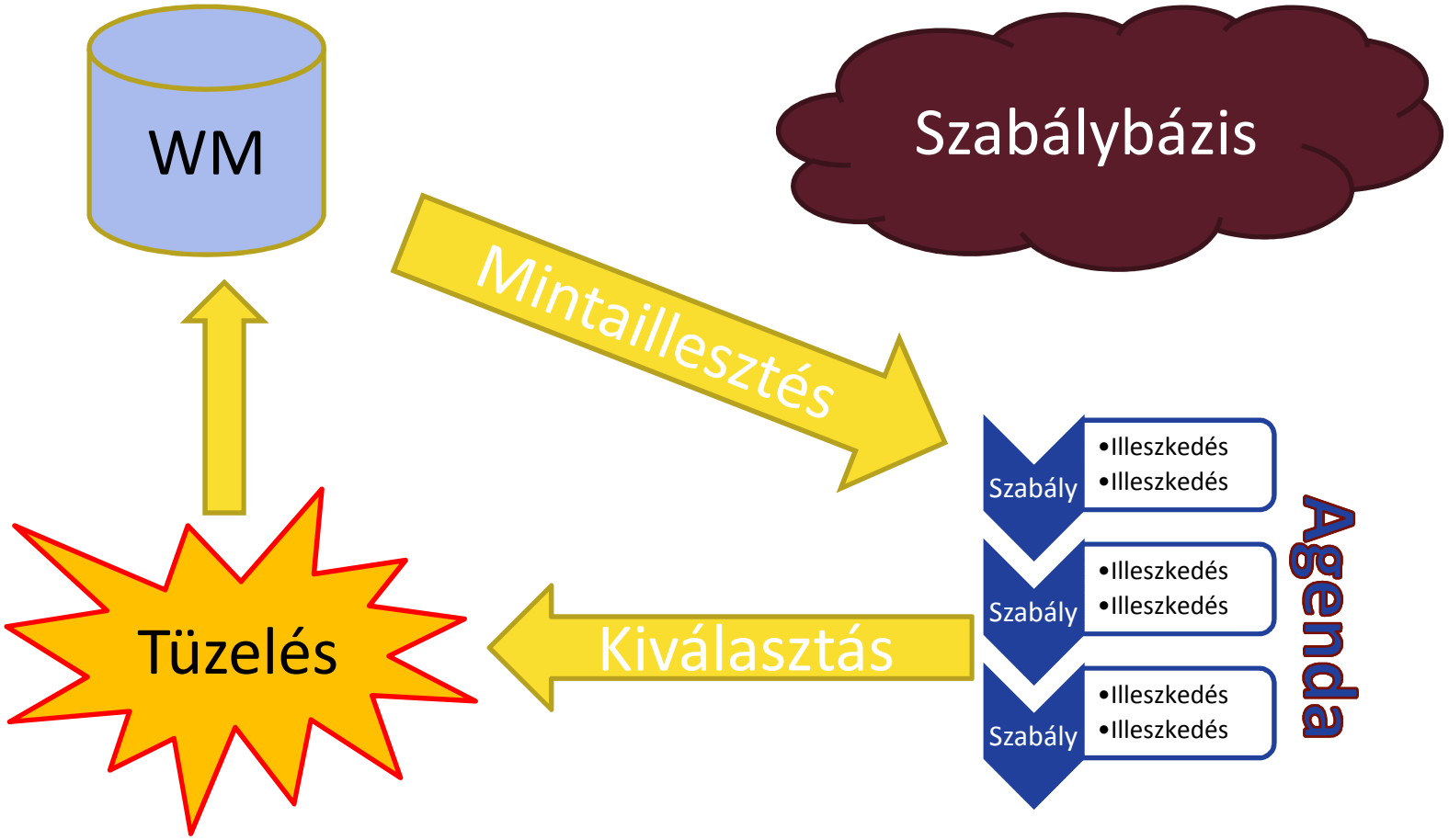
Szabály alapú rendszerek

- Adottak tények
  - „**Ténybázis**” / munkamemória
  - Változatos felépítés
- Adottak szabályok, amelyekkel új tényeket lehet kapni
  - „**Szabálybázis**”
  - Felépítés: ha-akkor
- Végül egy következtető mechanizmus

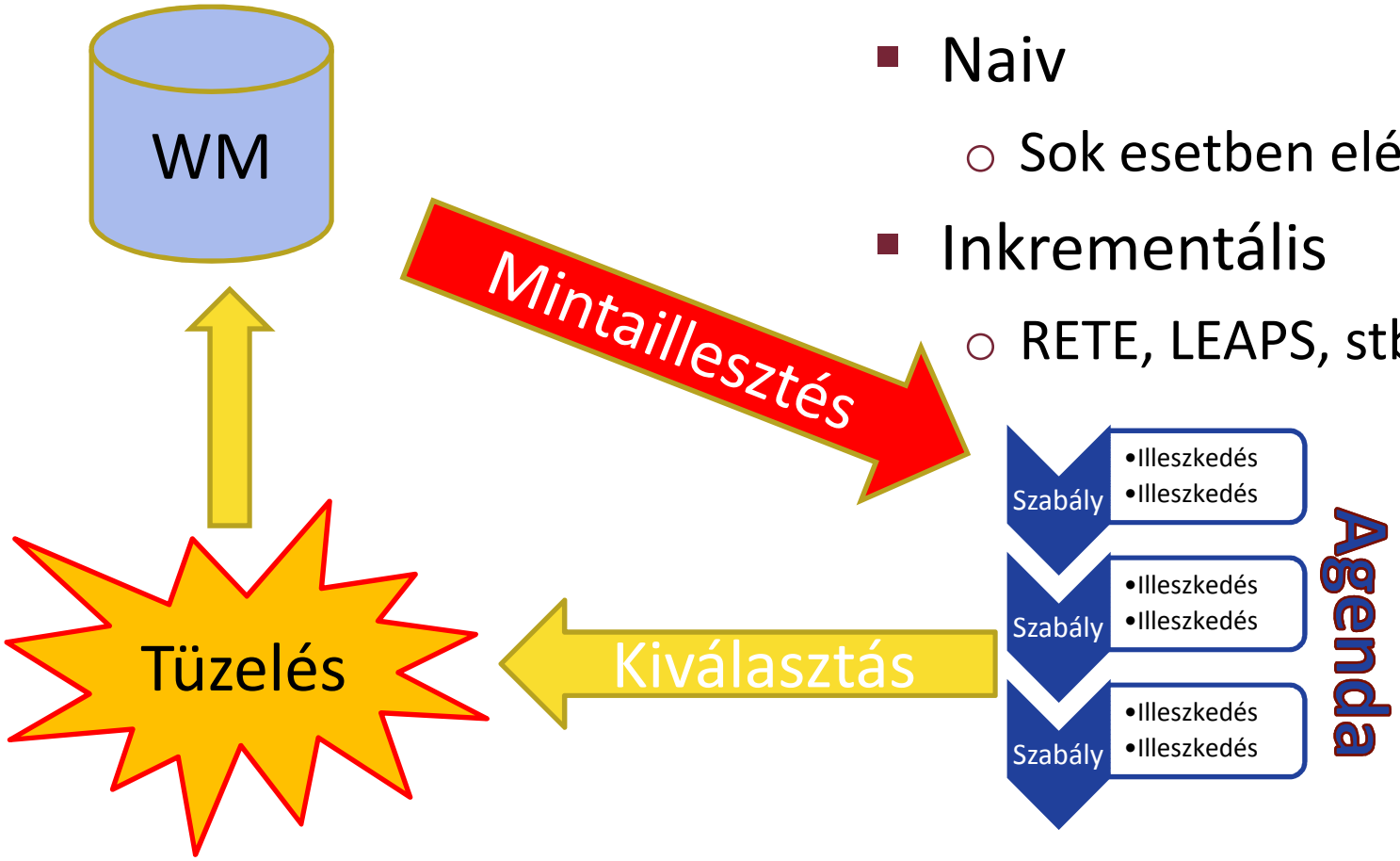
Következtetés

- **Előre** láncoló (induktív/produkciós, adatvezérelt)
  - A tényekből újabb tényeket képez
  - Analógia: generatív nyelvtan
  - Ilyenek például a üzleti szabályrendszerek
- **Hátra** láncoló (deduktív, igényvezérelt)
  - Egy cél-állítást próbál visszavezetni alaptényekre
  - Analógia: parser
  - Ilyen például a *Prolog* és számos szakértői rendszer

Egyszerű előre láncoló rendszer

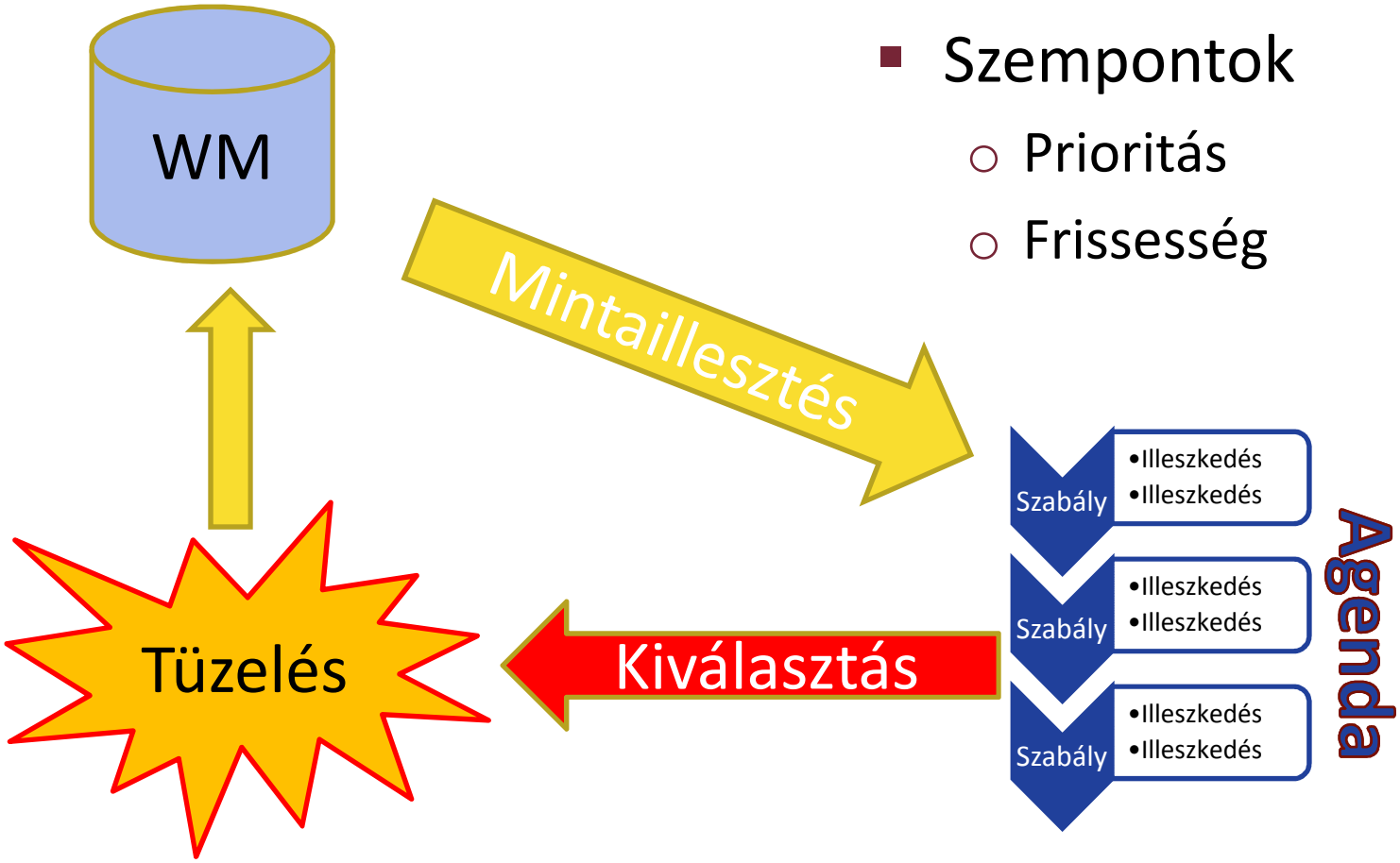


Egyszerű előre láncoló rendszer



- Naiv
  - Sok esetben elég
- Inkrementális
  - RETE, LEAPS, stb.

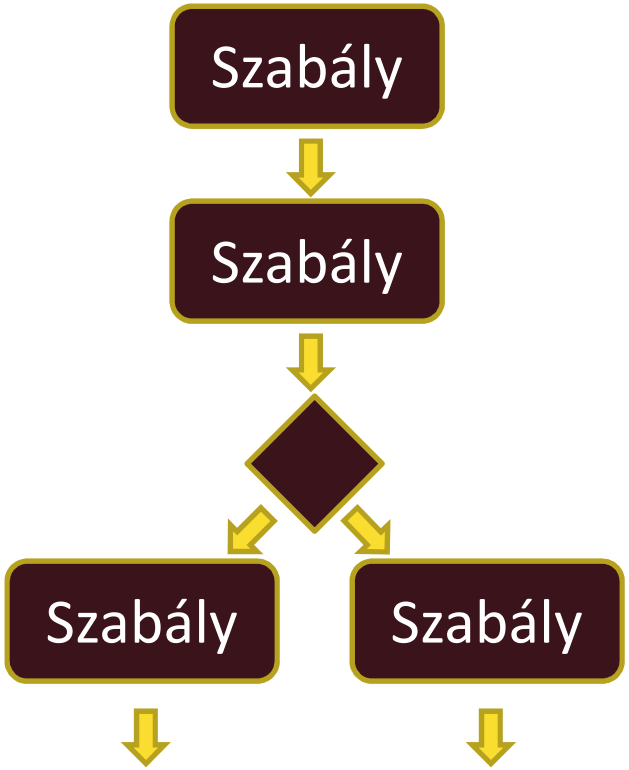
Egyszerű előre láncoló rendszer



- Szempontok
  - Prioritás
  - Frissesség

Előre láncoló rendszer vezérlése

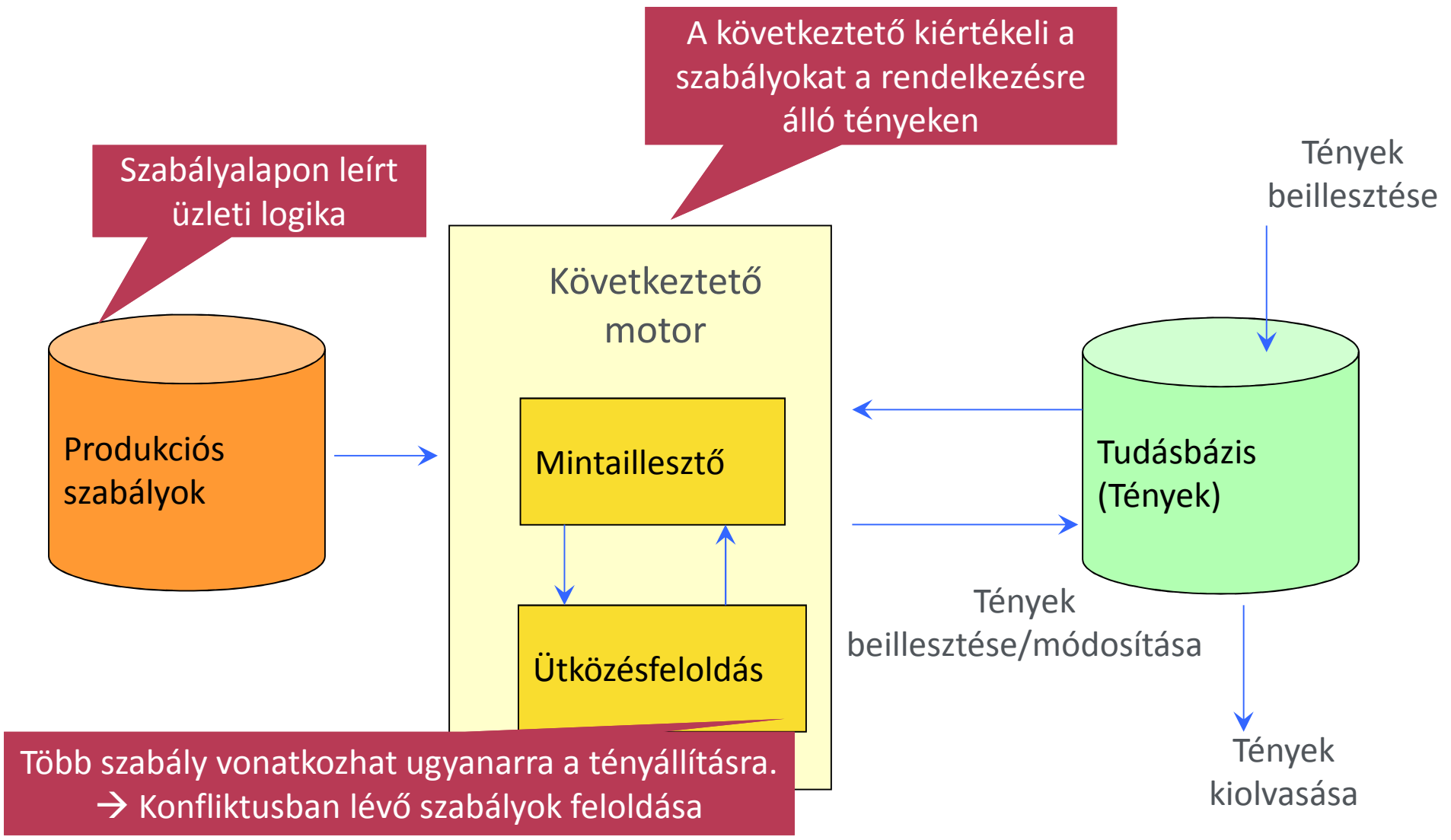
- Leállás
  - STOP szabály után
  - Ha nincs több tüzelhető szabály
- Komplex rendszer: vezérlési folyam
  - Pl. UML Activity Diagram
  - Kiválthatja a bemutatott ciklust
- Eseményvezéreltség is elképzelhető
  - „Alvó” szabályok
  - Külön utasítás nélkül



BRMS

- BRMS = **Business Rule Management System**
- Számos termék
  - *G2, JBoss Rules, IBM ILOG (J)Rules, Blaze Advisor, MS BRE, TIBCO iProcess,*

BRMS működés



Összetett 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
```

Szabályalapú megközelítés alkalmazása

Complex Event Processing

Stream Processing

CEP alapelvek

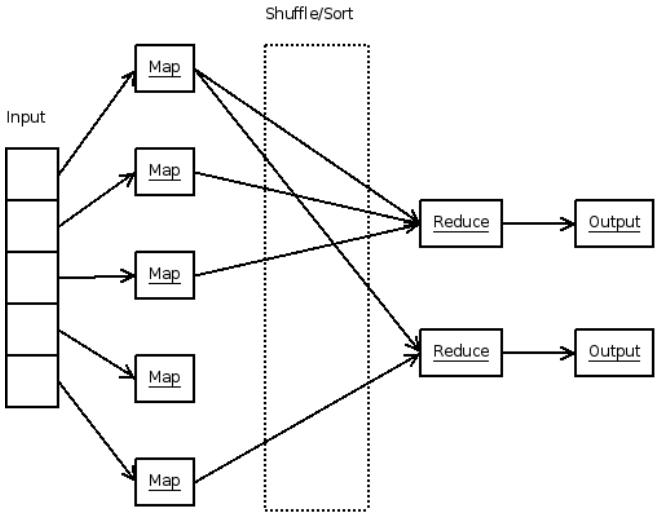
- „Komplex esemény”
  - Több elemi esemény összekapcsolása
- Tulajdonságok
  - Időzítések figyelembevétele (pl. csúszóablak)
  - Aszinkron működés
  - Oksági kapcsolatok, hierarchikus események
  - Korreláció
- SQL-szerű query nyelvek
  - Pl. EPL: Event Processing Language
  - Feldolgozási folyamatba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

CEP alkalmazási területek

- Üzleti alkalmazások
  - Tőzsde, befektetések
  - „Treasury”
  - Kockázatkiértékelés
  - 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/

Map/Reduce algoritmus

- Map lépés
  - adat felosztása
- Reduce lépés
  - adat feldolgozása
- Példa
  - szöveg felosztása szavakra, szavak számának megállapítása
- Számos programnyelven
- Apache implementáció
  - Elosztott megoldás
  - Hadoop (+ Hadoop Distributed File System)
  - Ütemezés : Job Tracker, Task Tracker



CEP eszközök

- Esper
- Drools Fusion
- IBM InfoSphereStreams (System S)
- OpenESB - Intelligent Event Processor
- Apache Hadoop + ráépülő projektek
- Döntési szempontok
  - Eseményfeldolgozási logika
  - Áteresztőképesség
  - Elvárt válaszidő („low latency”)

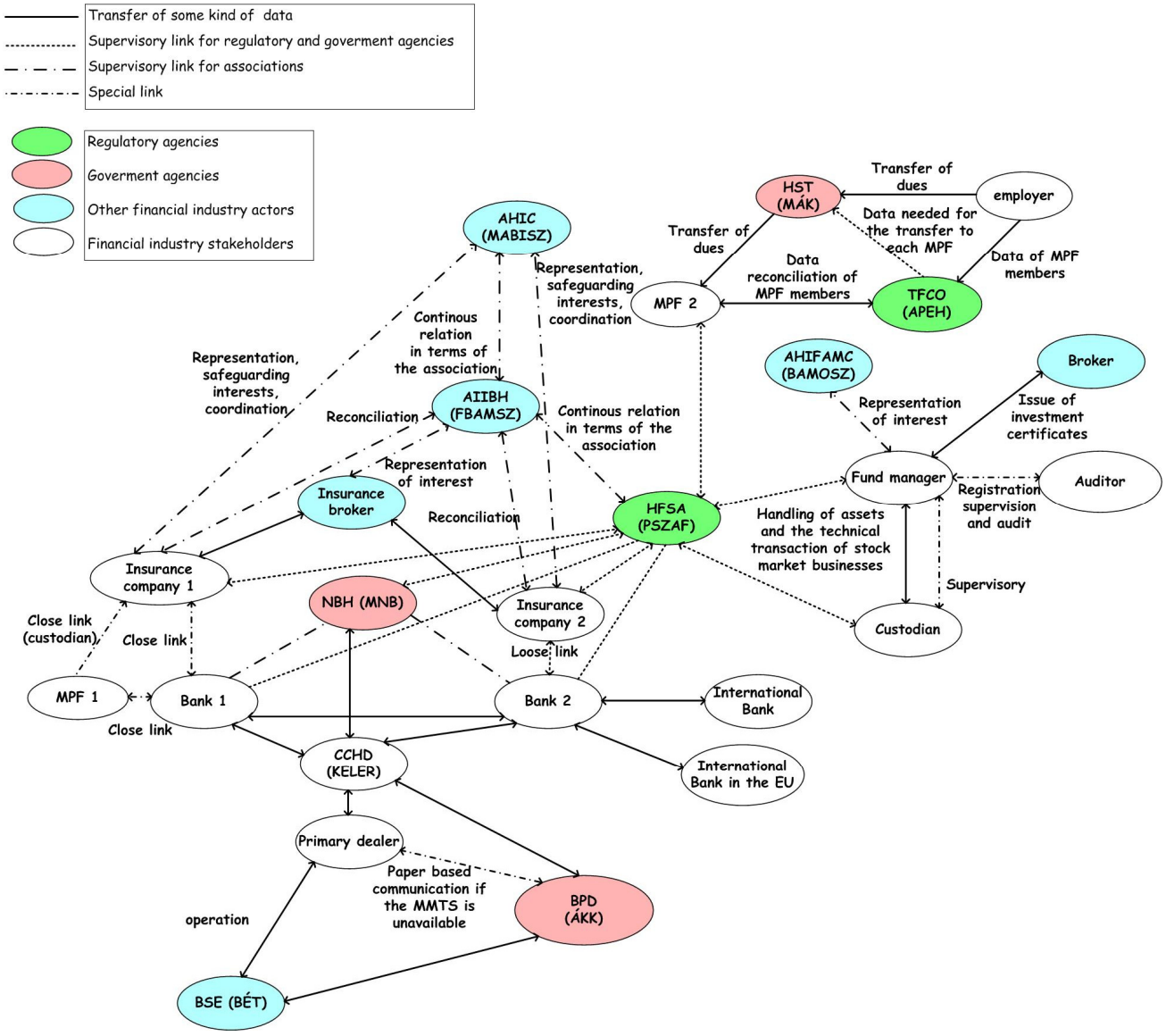
Esettanulmány: CoMiFin

Szolgáltatásalapú rendszerek, modellvezérelt fejlesztés, komplex eseményfeldolgozás,…

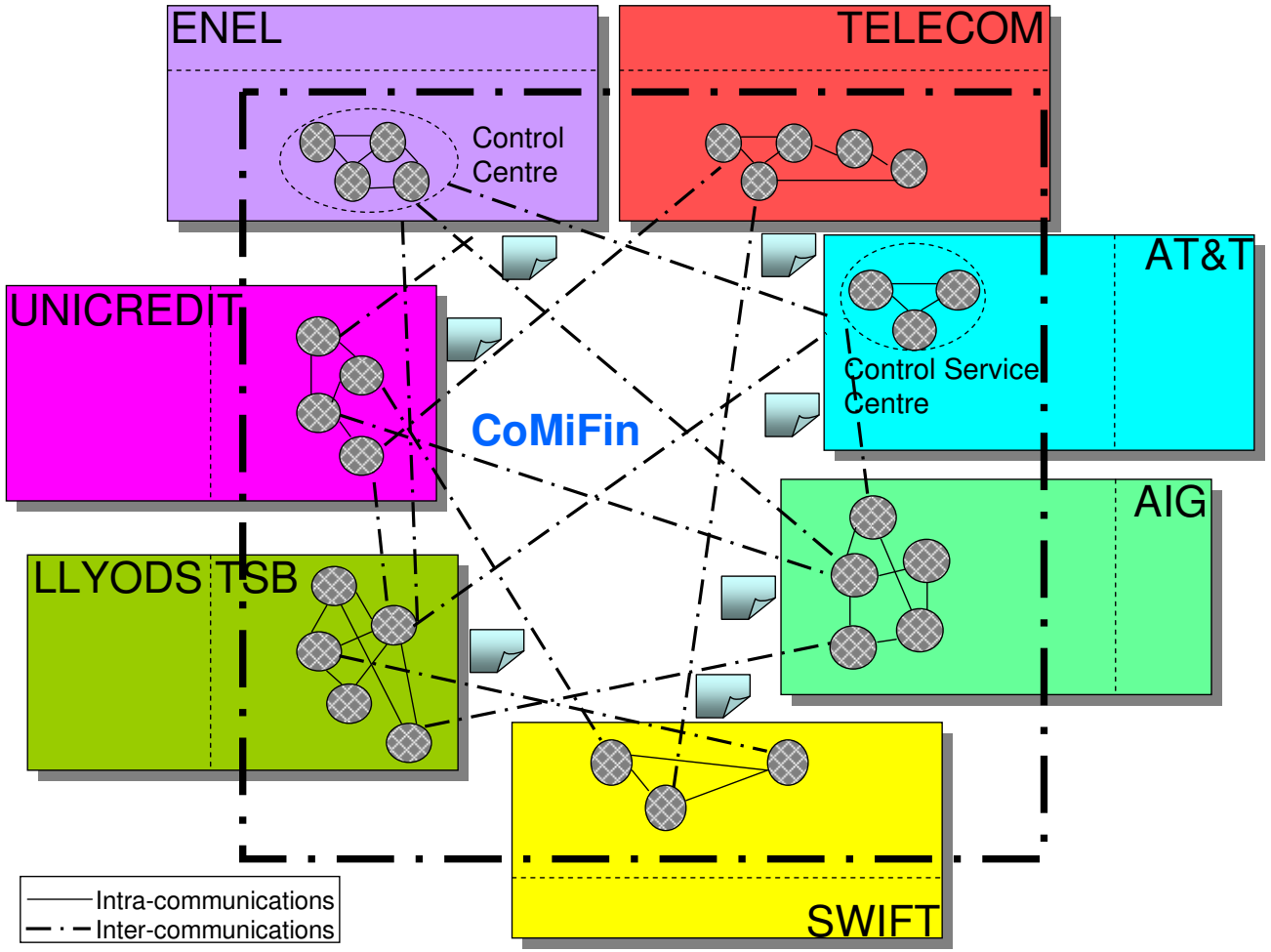
Esettanulmány: CoMiFin

- „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

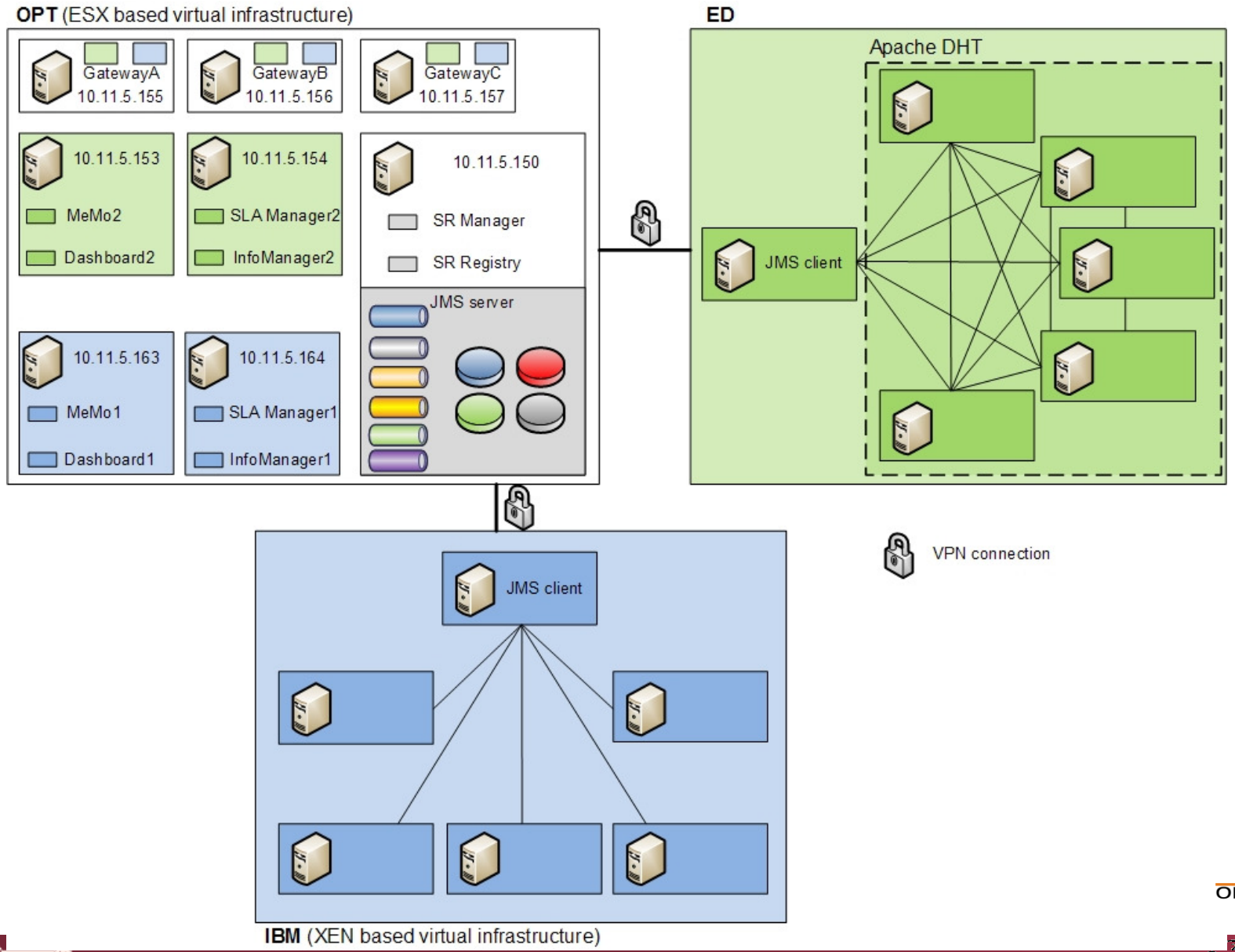
Példa: magyar infrastruktúra



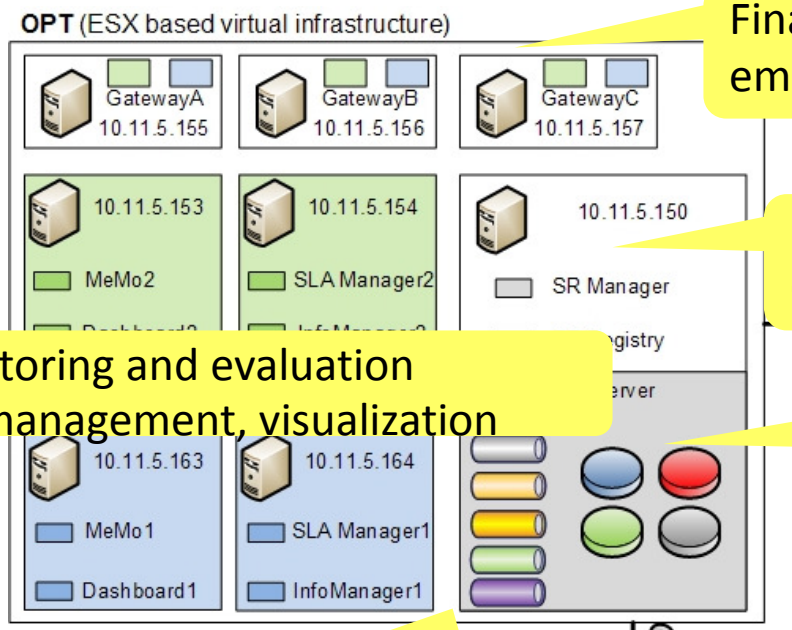
Logikai architektúra



Architektúra



Architektúra

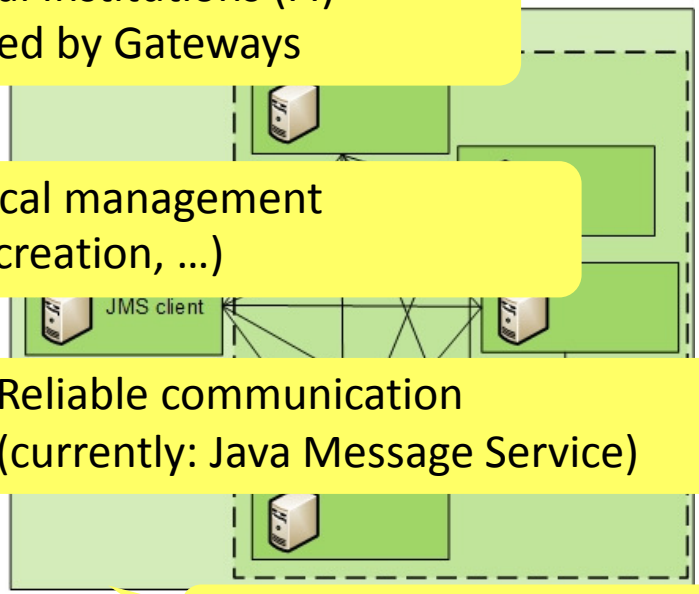


Financial Institutions (FI)
emulated by Gateways

Logical management
(SR creation, …)

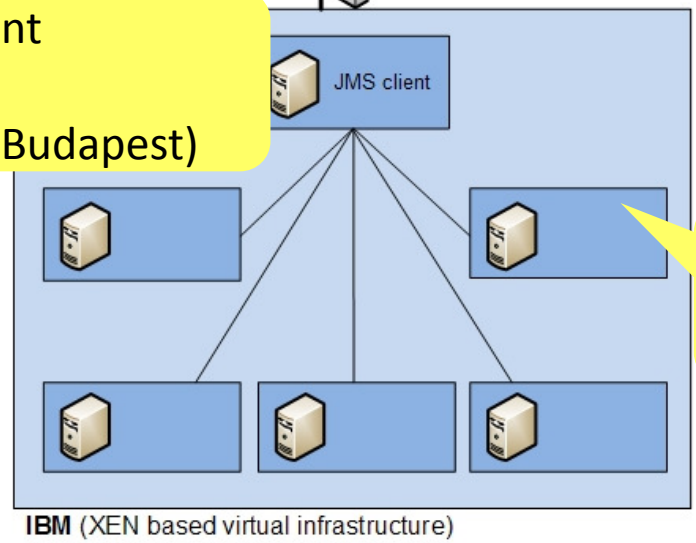
Monitoring and evaluation
SLA management, visualization

Reliable communication
(currently: Java Message Service)



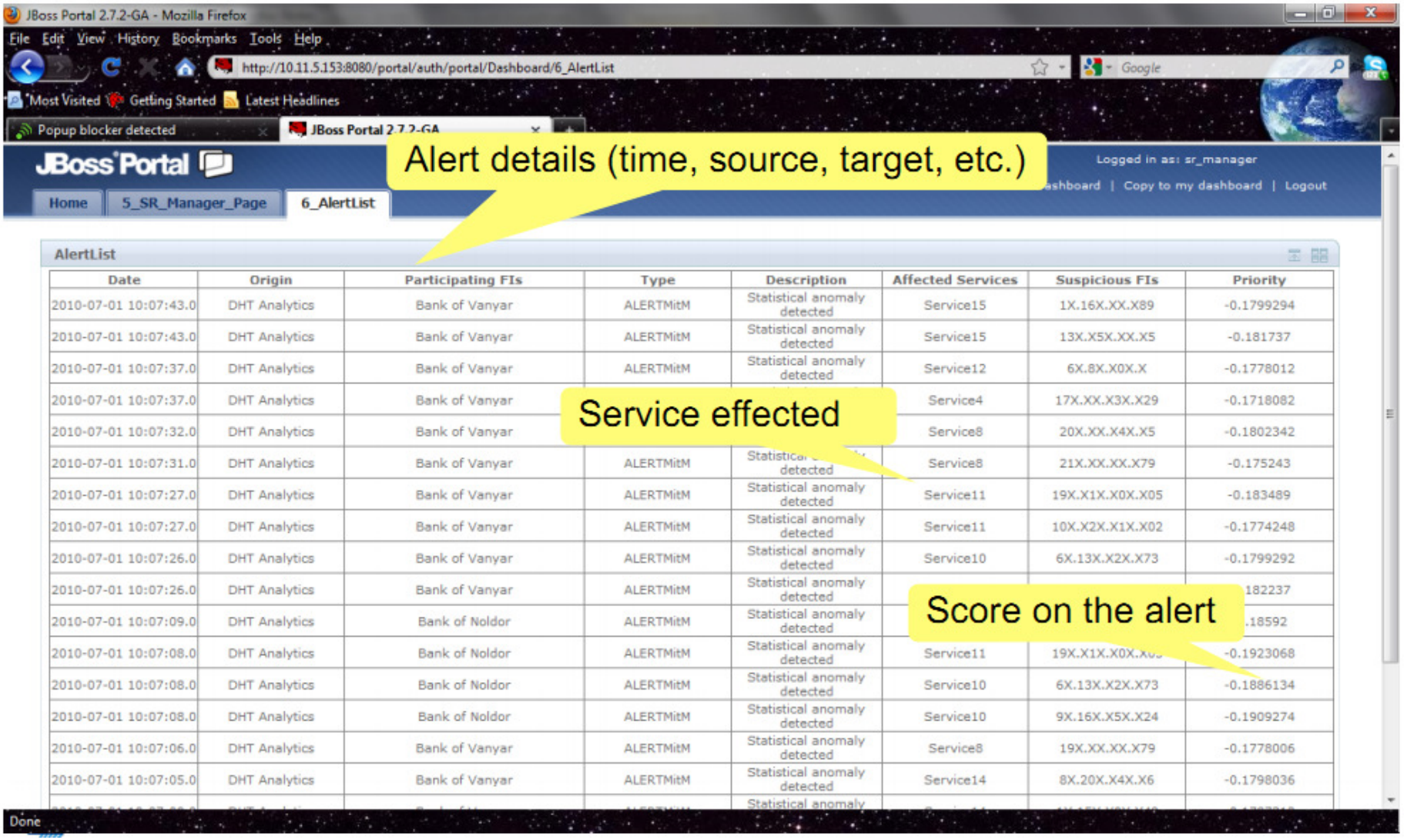
ED Event Processing (DHT)
(ED Testbed, Rome)

CoMiFin management
components
(OptXware testbed, Budapest)



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

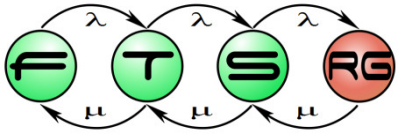
Eredmények megjelenítése



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			Service4	17X.XX.X3X.X29	-0.1718082
2010-07-01 10:07:32.0	DHT Analytics	Bank of Vanyar			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			182237
2010-07-01 10:07:09.0	DHT Analytics	Bank of Noldor	ALERTMitM	Statistical anomaly detected			.18592
2010-07-01 10:07:08.0	DHT Analytics	Bank of Noldor	ALERTMitM	Statistical anomaly detected	Service11	19X.X1X.X0X.X	-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

Modellalapú fejlesztési módszer komplex események feldolgozásához

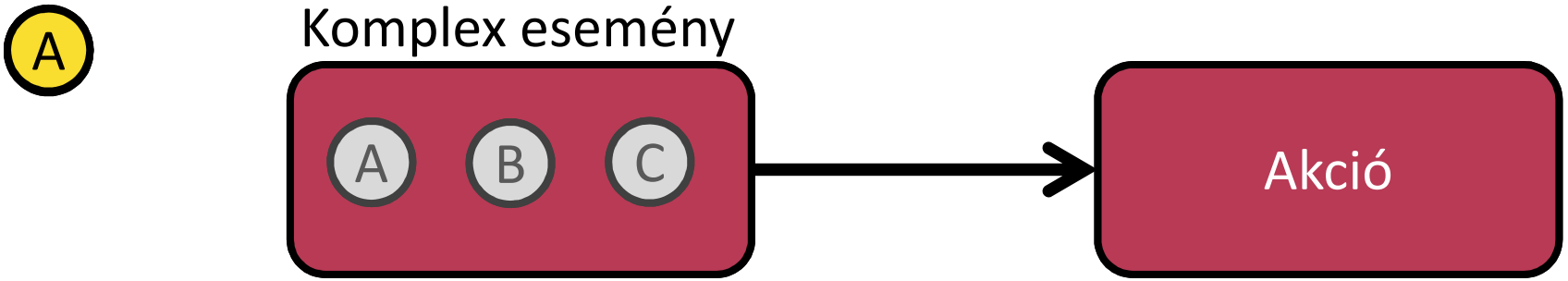


Budapesti Műszaki és Gazdaságtudományi Egyetem
Méréstechnika és Információs Rendszerek Tanszék

MŰEGYETEM 1782

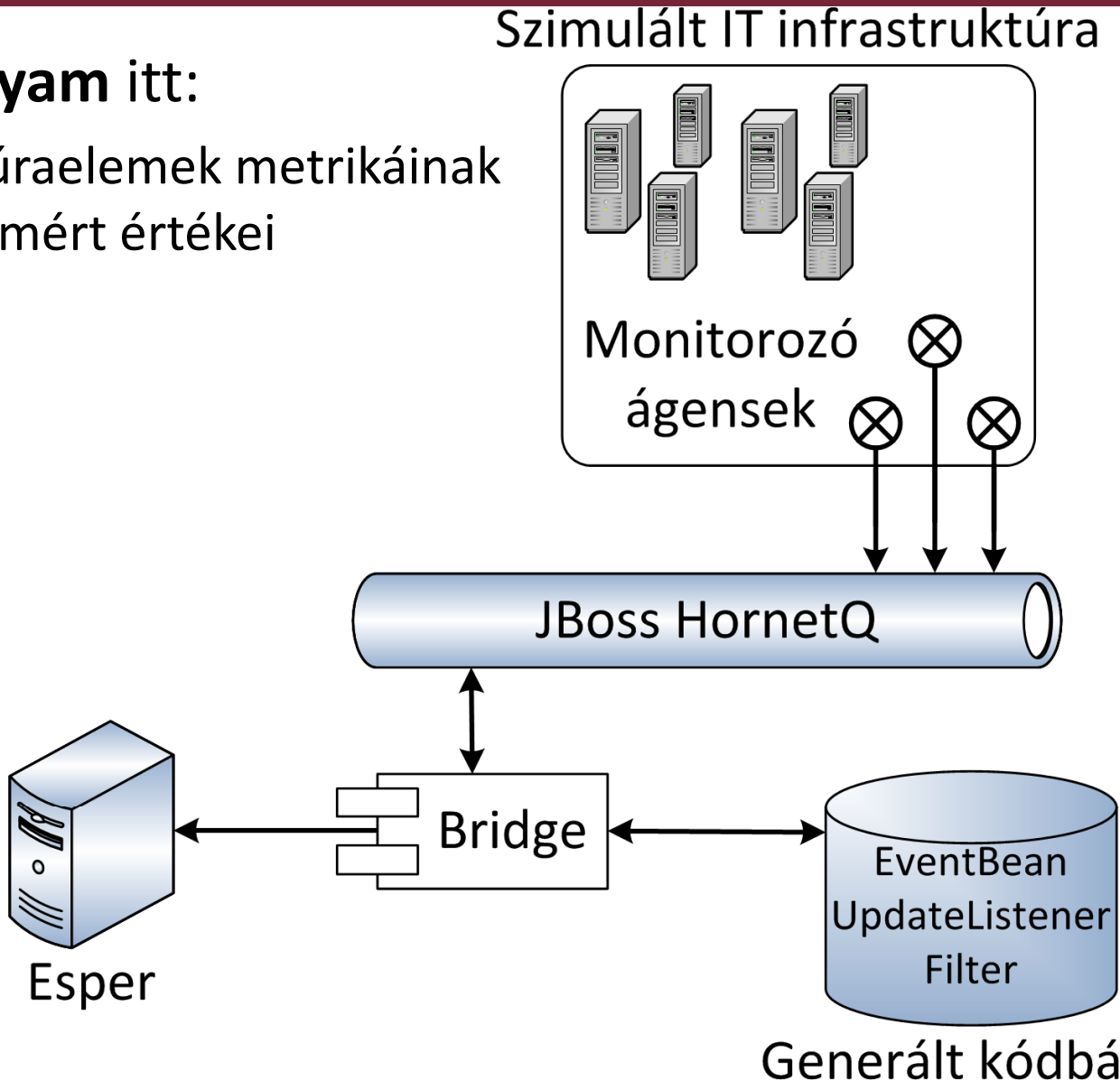
Motiváció

- Események feldolgozása – alapja: a mintafelismerés
  - Az érdekes események összetettek lehetnek
- Komplex esemény: elemi eseményekből álló struktúra
  - Complex Event Processing – CEP
- Példák
  - *„A webszerver terheltsége 90% és az adatbázis szerveren lekérdezést futtatnak.” ⇨ „Vonjunk be tartalék erőforrást.”*
  - *„Az X részvények értéke jelentősen csökkent, és a vele korreláló Y részvények is elkezdtek esni.” ⇨ „Adjunk el Y részvényeket.”*



Mintakörnyezet

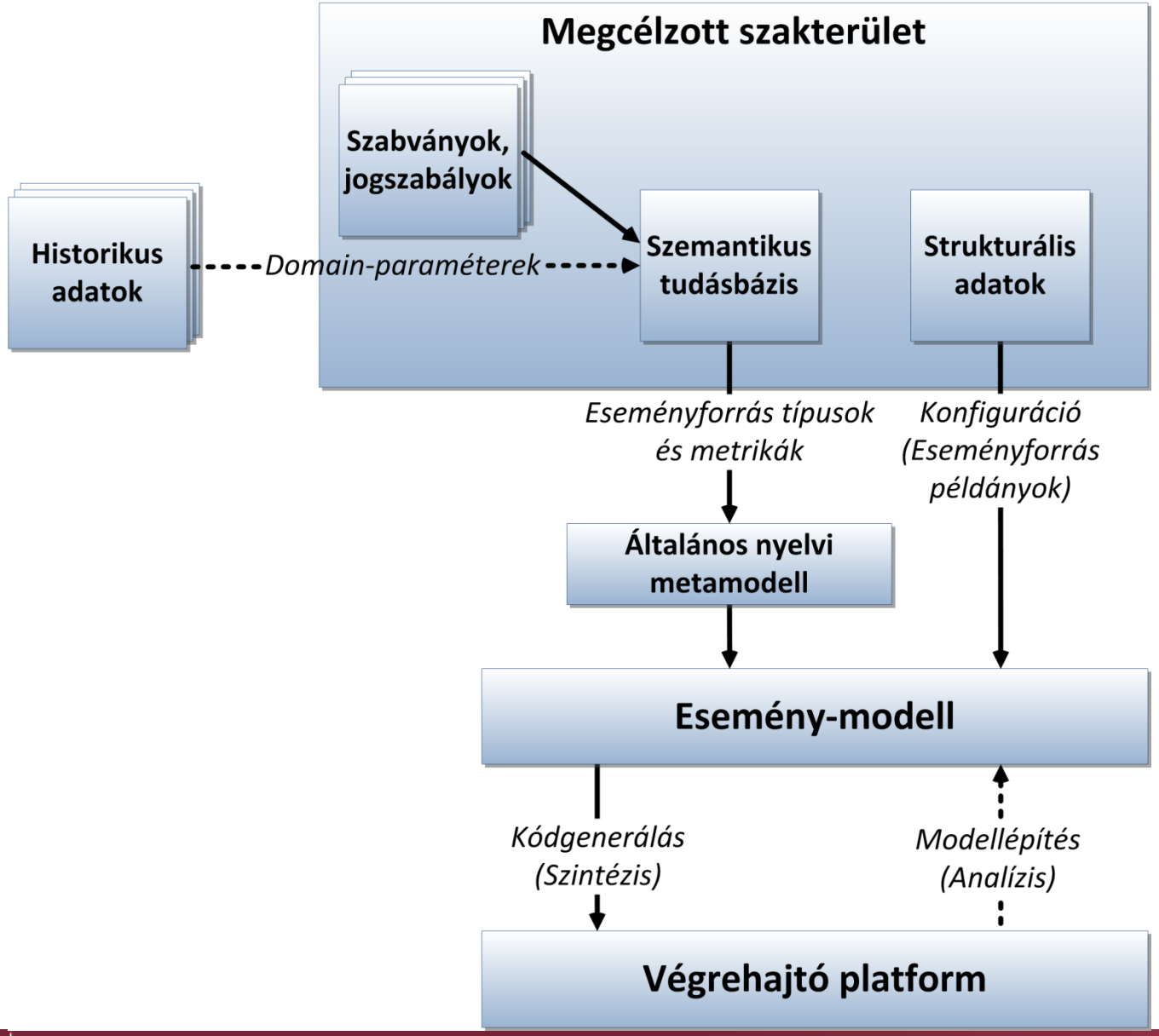
- Az **eseményfolyam** itt:
  - Az infrastruktúraelemek metrikáinak folyamatosan mért értékei



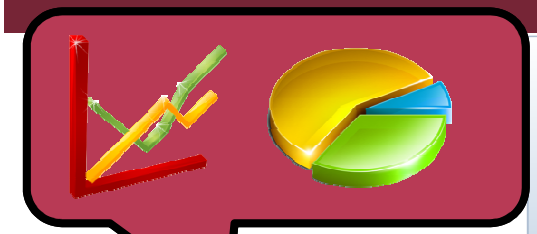
Problémafelvetés

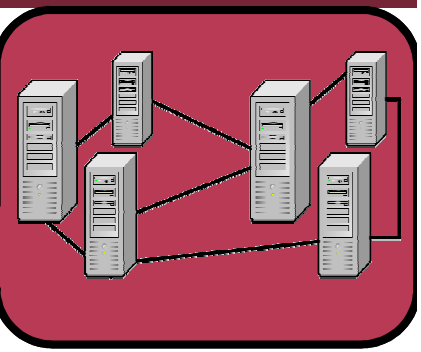
- Különböző platformok, saját nyelvvel – nincs közös szabvány
- Nehezen kezelhetőek a nagy eseményminták
  - Platformközeli nyelv → sok „kézi kódolás”
  - Egymást fedő eseményminták problémája
- Magas szintű követelményeknek való megfelelés biztosítása
  - Ad-hoc konfigurációk
  - Eseti hibakezelés
  - Heterogén információforrások
- Modellalapú megközelítés
  - Magasabb logikai szinten történő ellenőrizhetőség
  - Generált kód → a szintaktikai hibák megszüntethetőek
    → több célplatform is megcélozható egy modellel

A kialakított megoldás struktúrája



A kialakított [illegible] szerkezete





$\mathcal{L}_1$ = { „webszerver”, „szolgáltatás” …}

$\mathcal{L}_2$ = { „kritikus”, „bizalmas” …}

Szabványok, jogszabályok

Historikus adatok

Domain-[illegible]améterek

Szemantikus tudásbázis

Strukturális adatok

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

Eseményforrás típusok és metrikák

Konfiguráció (Eseményforrás példányok)

Általános nyelvi metamodell

$\mathcal{L}$ = { Event, ComplexEvent, Source …}

Esemény-modell

Kódgenerálás (Szintézis)

Modellépítés (Analízis)

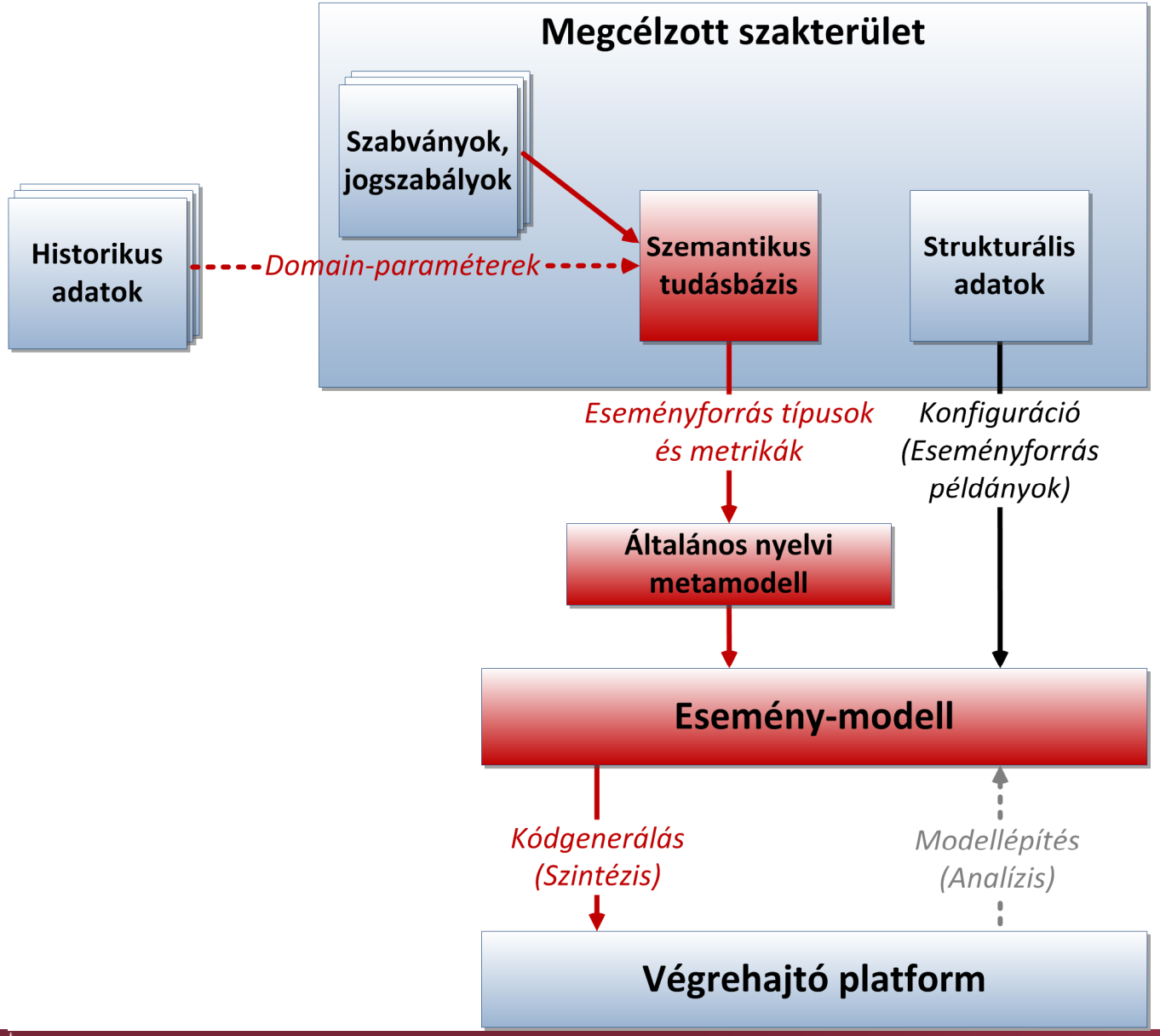
Végrehajtó platform

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

A

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

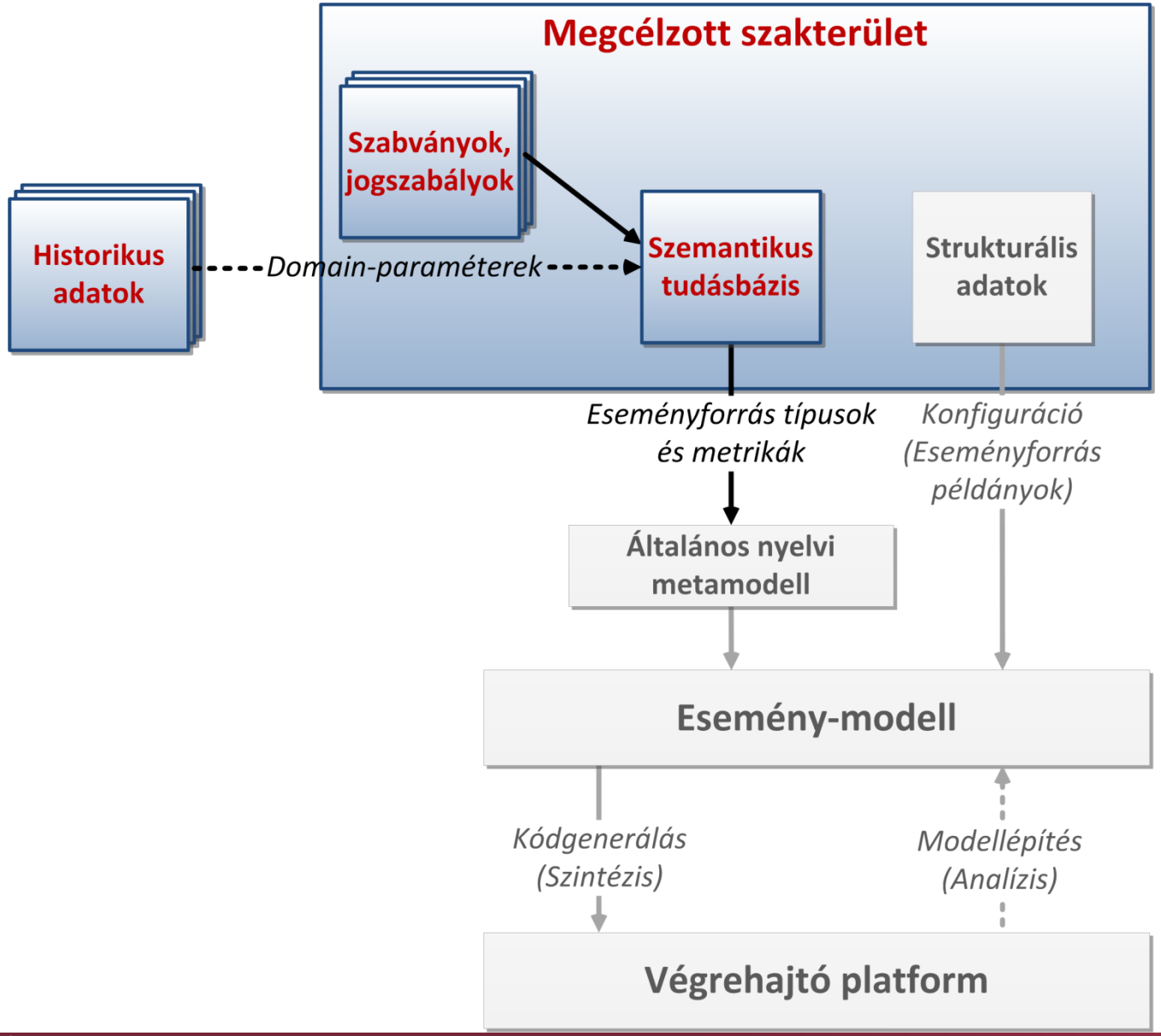
A kifejlesztett módszer



A kifejlesztett módszer

- Módszer a magas szintű követelmények formalizálására
- Módszer a historikus adatok felhasználására
- Szemantikus tudásbázis mintaséma
- Módszer a szemantikus adatok kinyerésére
- Modellezőnyelv prototípus
- Fejlesztőeszköz prototípus
  - Modellvalidáció
  - Kódgenerálás

Szemantikus tudásbázis

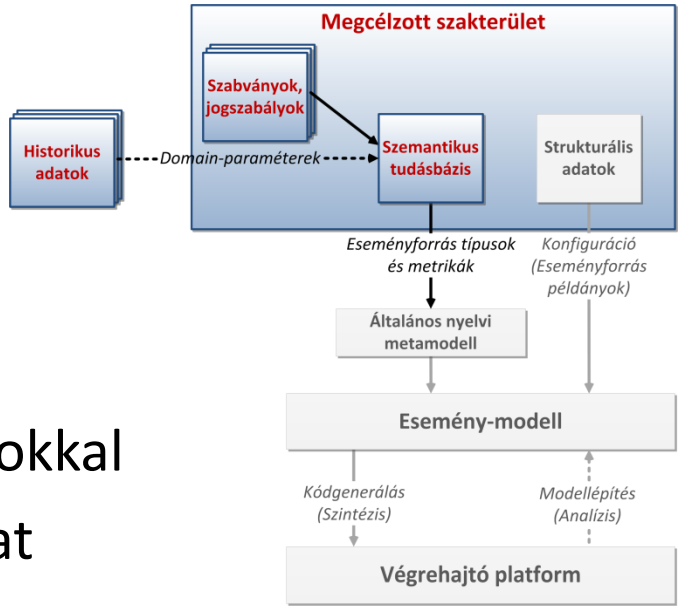


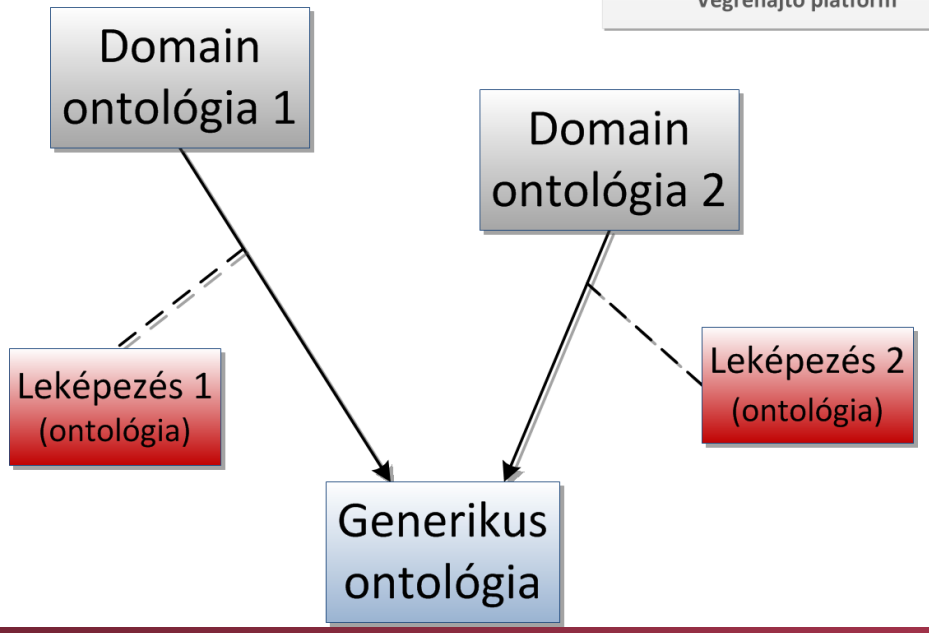
Szemantikus tudásbázis

- Jellemző gyakorlati probléma:
  (túl) sok mérhető metrika
- Jó lenne formalizálni a
  magas szintű követelményeket
  - Ezeket pedig összekötni a mérhető metrikákkal
- Szemantikus tudásbázis: egy ontológia
  - Következtető logikákkal kimutatható összefüggések
  - *„Létezik-e minden X kritikus erőforráshoz két példány, amelyek közötti kapcsolat a Tartalék típus példánya? Figyeljük meg ezeket.”*
- A szemantikus tudásbázis tehát:
  - Lehetőséget nyújt a magas szintű kritériumok formalizálására
  - Lehetőséget nyújt bizonyos szintű ellenőrzésekre

Szemantikus tudásbázis

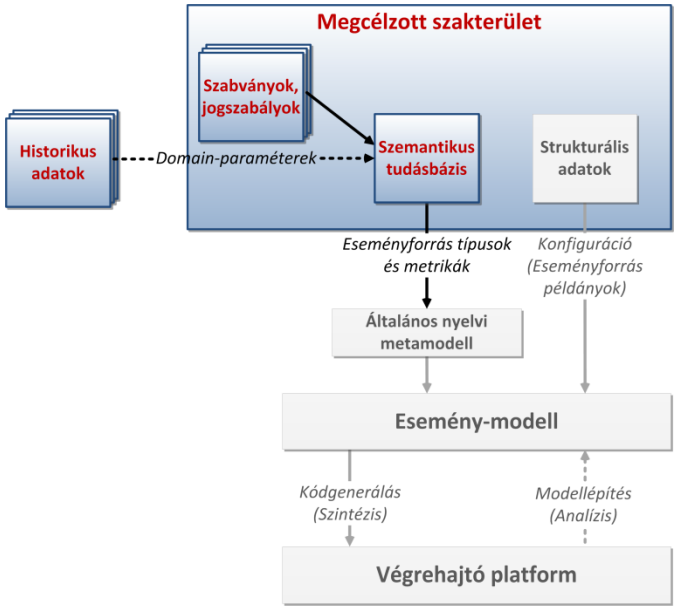
- Domain ontológia
  - Az adott szakterültet írja le
  - Minden szakterületen más
- Generikus ontológia:
  - A modellező nyelv alapján, rögzített típusokkal
  - Erre lehet leképezni a domain ontológiákat
  - Minden esetben azonos

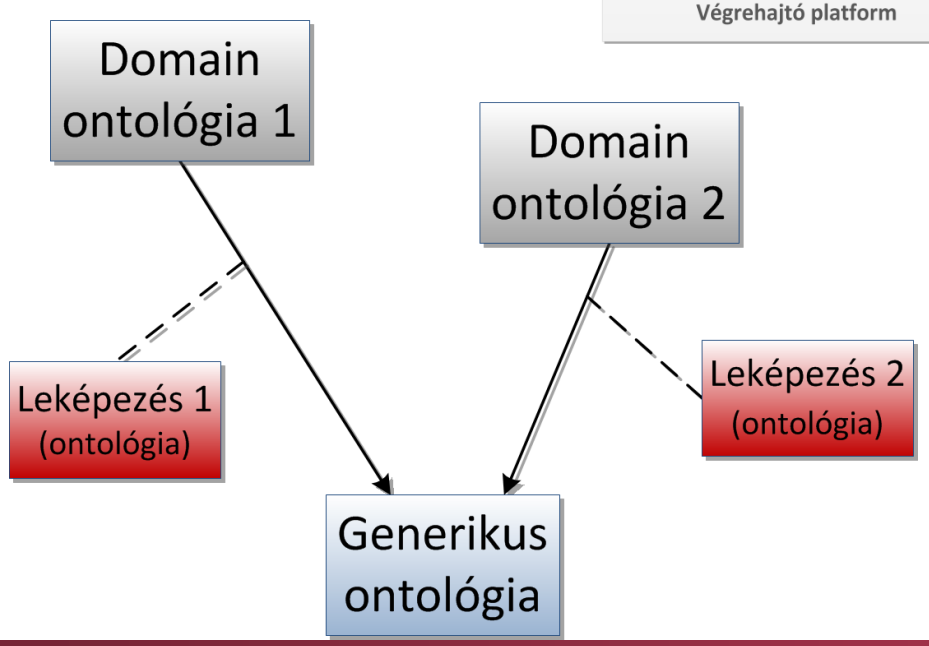




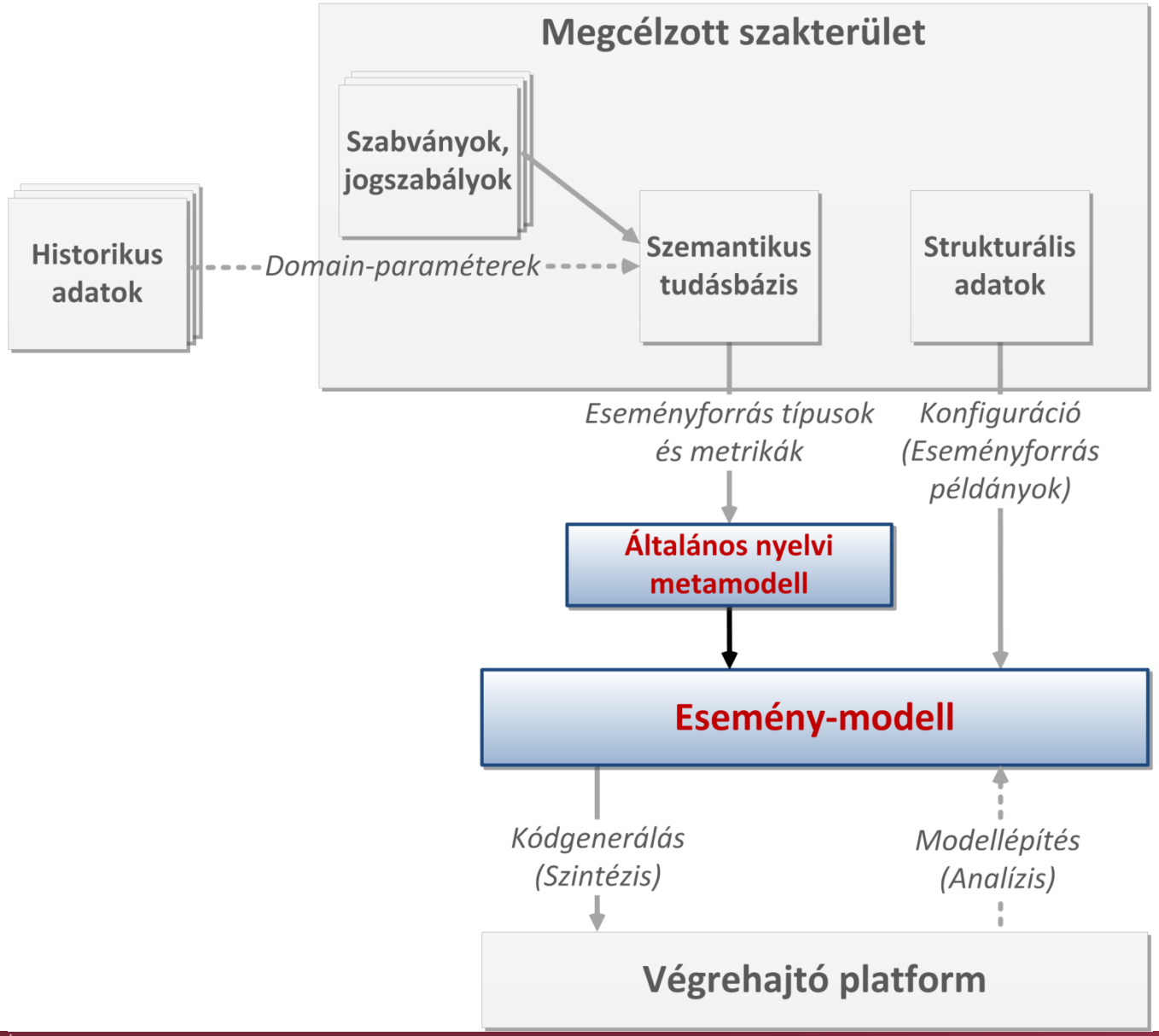
Szemantikus tudásbázis

- Példa leképezések (IT):
  - Webszerver ⇨ *SourceType*
  - Terheltség ⇨ *Quantifier*
  - m(Webszerver, terheltség) ⇨ *Metrics*
  - kritikus(m, [0.9;1]) ⇨ *Qualifier*



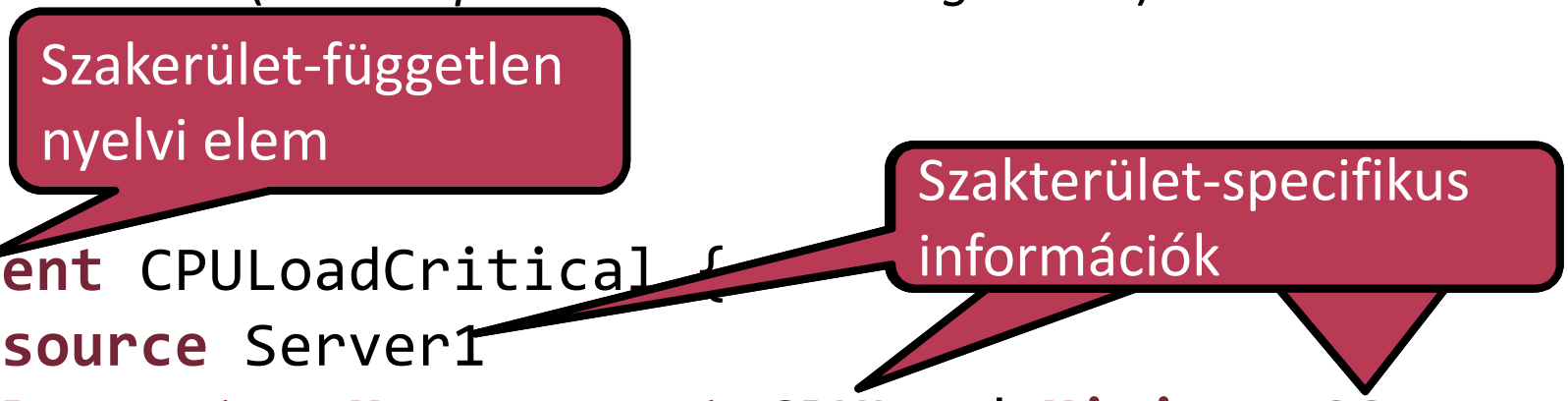


Complex Event Description Language



Complex Event Description Language

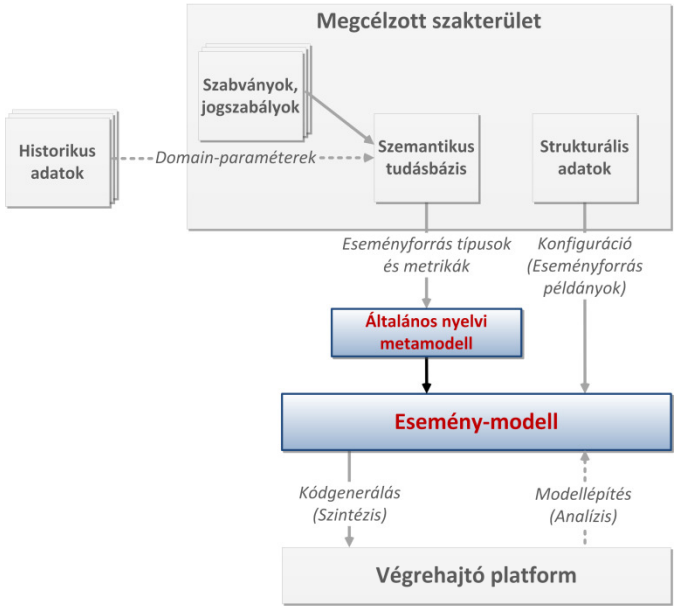
- Szakterület-független modellre épül
- A szakterület információival egészül ki
  - Eseményforrás típusok (*webszerver*)
  - Eseményforrás példányok (*a Server1 webszerver*)
  - Metrikák (*kritikus processzor terheltség ≥ 90%*)




```
Event CPULoadCritical {
  source Server1
  PercentageMeasurement CPULoad Minimum 90
}
```

Complex Event Description Language

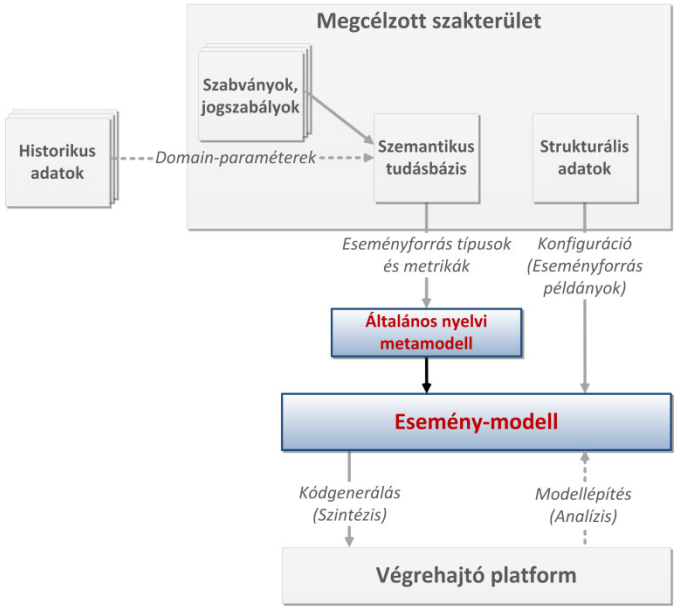
- Nyelvi alapok:
  - Komplex operátorok az elemi események relációinak leírásához:
    EXISTS, FOLLOWS, CONCURRENT
  - Időablak – TIMEWIN(T)

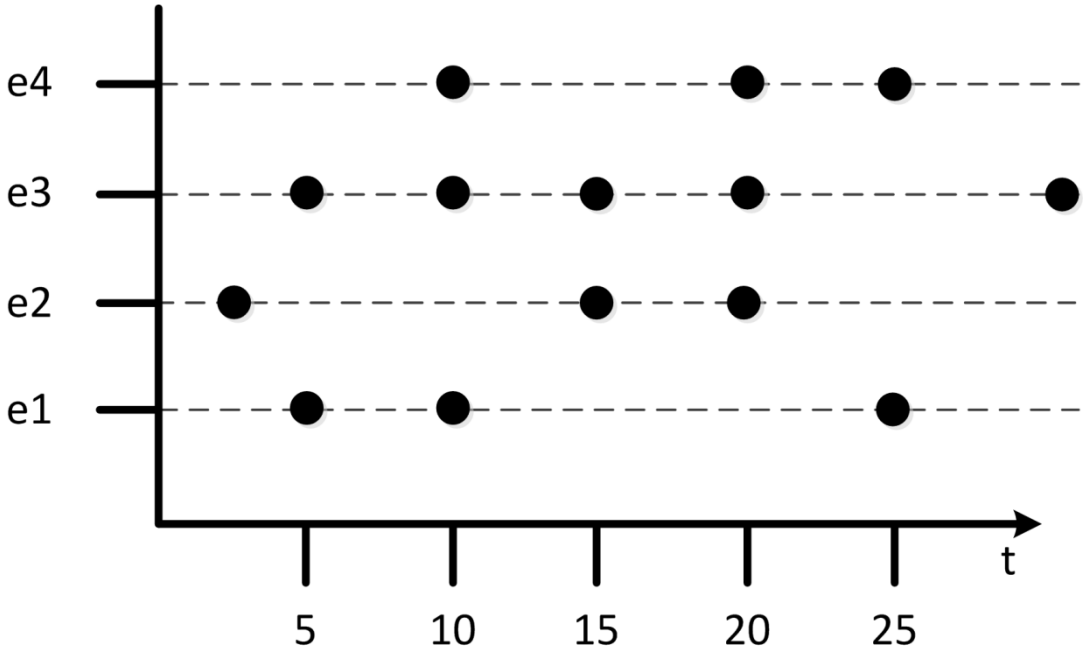


- Ezen felül még:
  - Aggregációk
  - Strukturális tervezési minták (OO-szerű)
  - Végrehajtható akciók
  - Minták alkalmazása erőforrások csoportjaira
  - Többféle metrika: százalék, skalár, szöveges, intervallum
  - Névterek kezelése, telepítési konfigurációk kezelése
  - Környezeti (nem-funkcionális) paraméterek kezelése

Complex Event Description Language

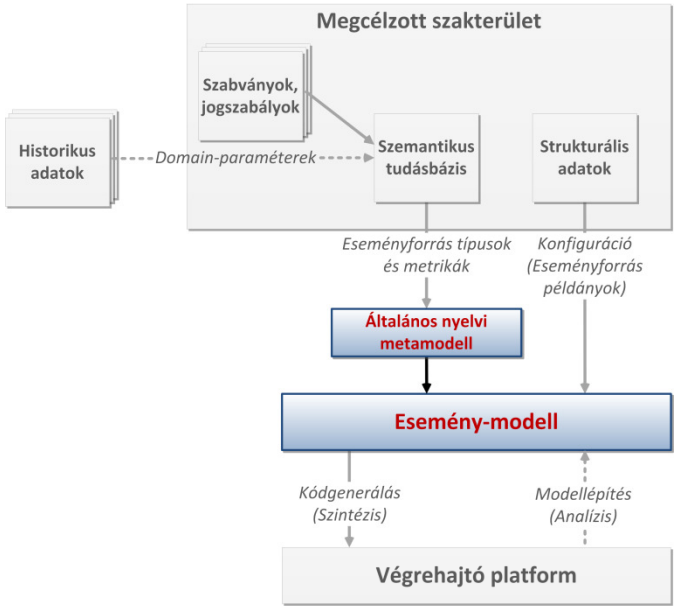
- Nyelvi alapok:
  - Komplex operátorok az elemi események relációinak leírásához:
    EXISTS, FOLLOWS, CONCURRENT
  - Időablak – TIMEWIN(T)

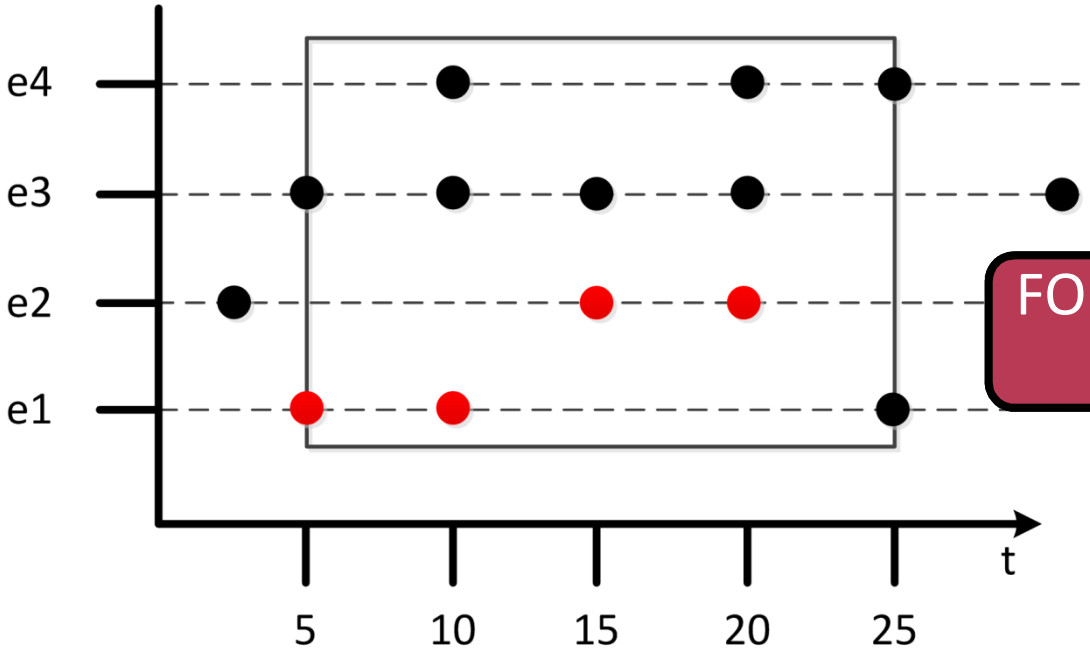




Complex Event Description Language

- Nyelvi alapok:
  - Komplex operátorok az elemi események relációinak leírásához:
    EXISTS, FOLLOWS, CONCURRENT
  - Időablak – TIMEWIN(T)

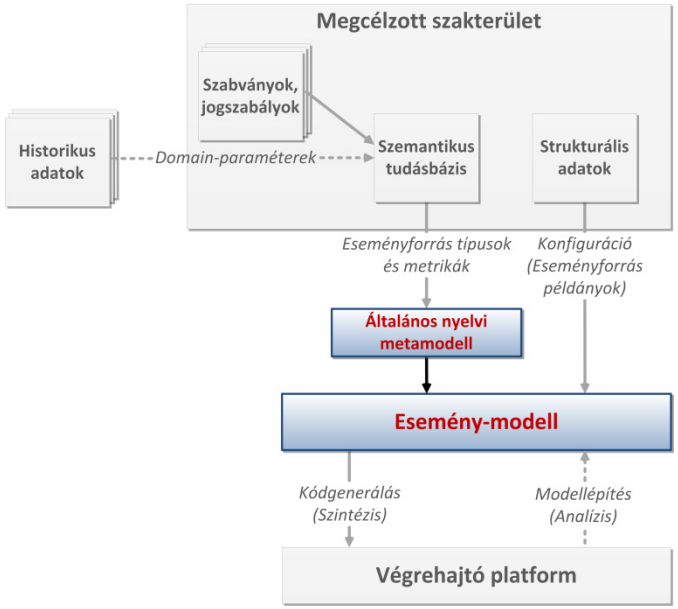




FOLLOWS_T(e1(2), e2(2);
T: Maximum 20)

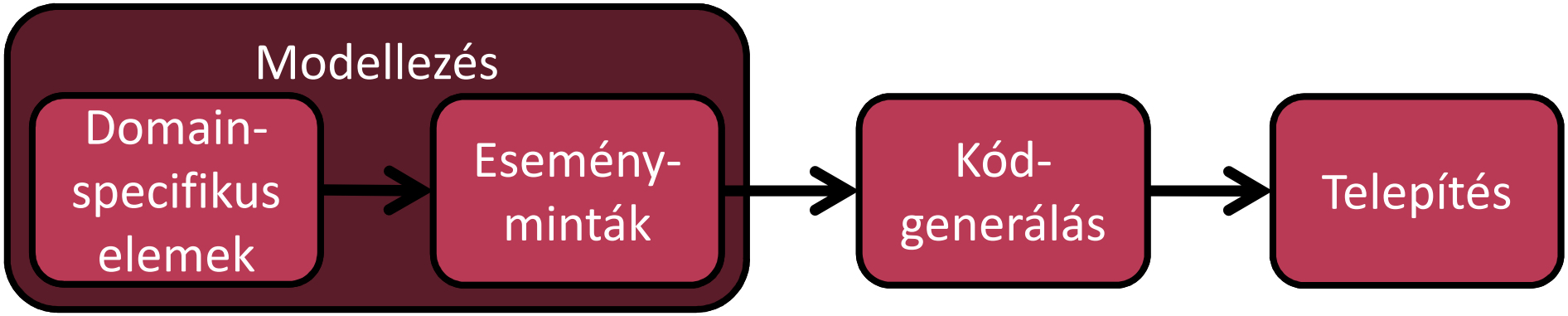
Complex Event Description Language

- Nyelvi alapok:
  - Komplex operátorok az elemi események relációinak leírásához:
    EXISTS, FOLLOWS, CONCURRENT
  - Időablak – TIMEWIN(T)



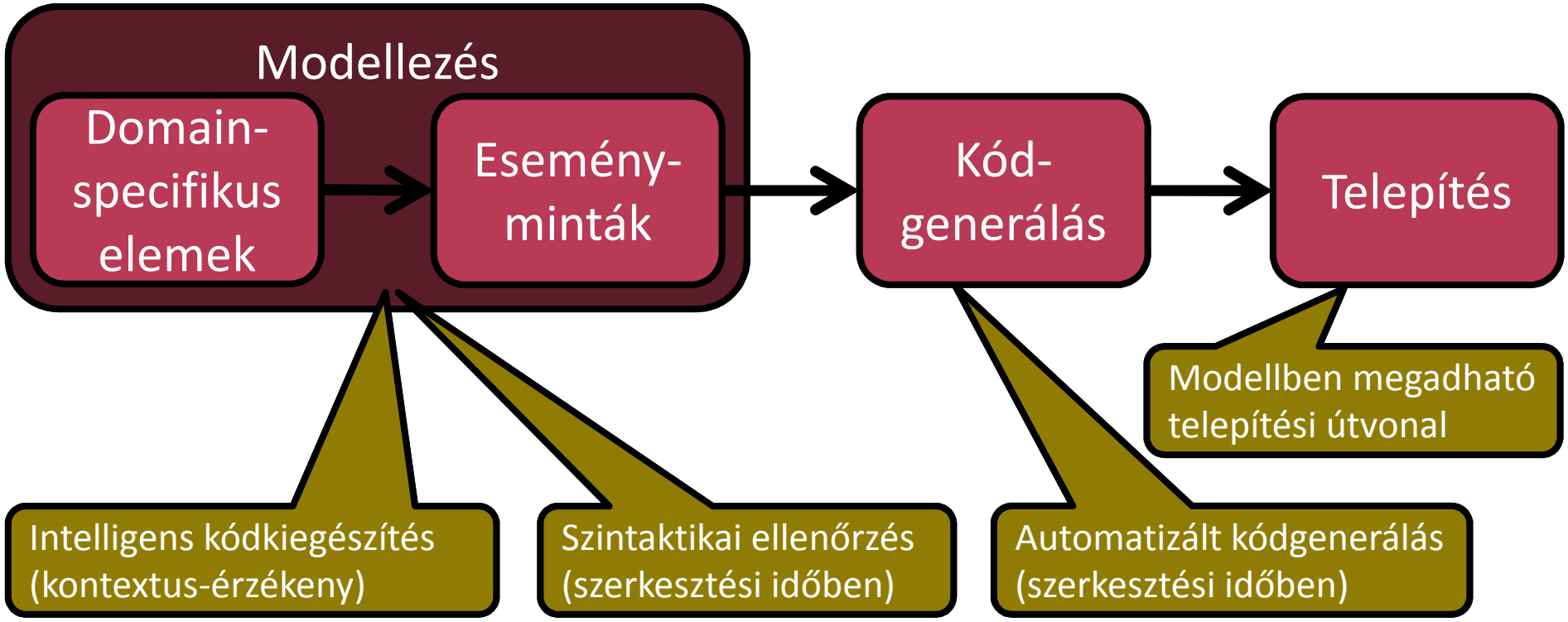

```
ComplexEvent CriticalServer {
  CONCURRENT_T(CPULoadSuspicious,
               BackupLoadSuspicious; T: Minimum 30)
  action {
    Action of Type sendWarning
  }
}
```

Tervezési-fejlesztési munkafolyamat



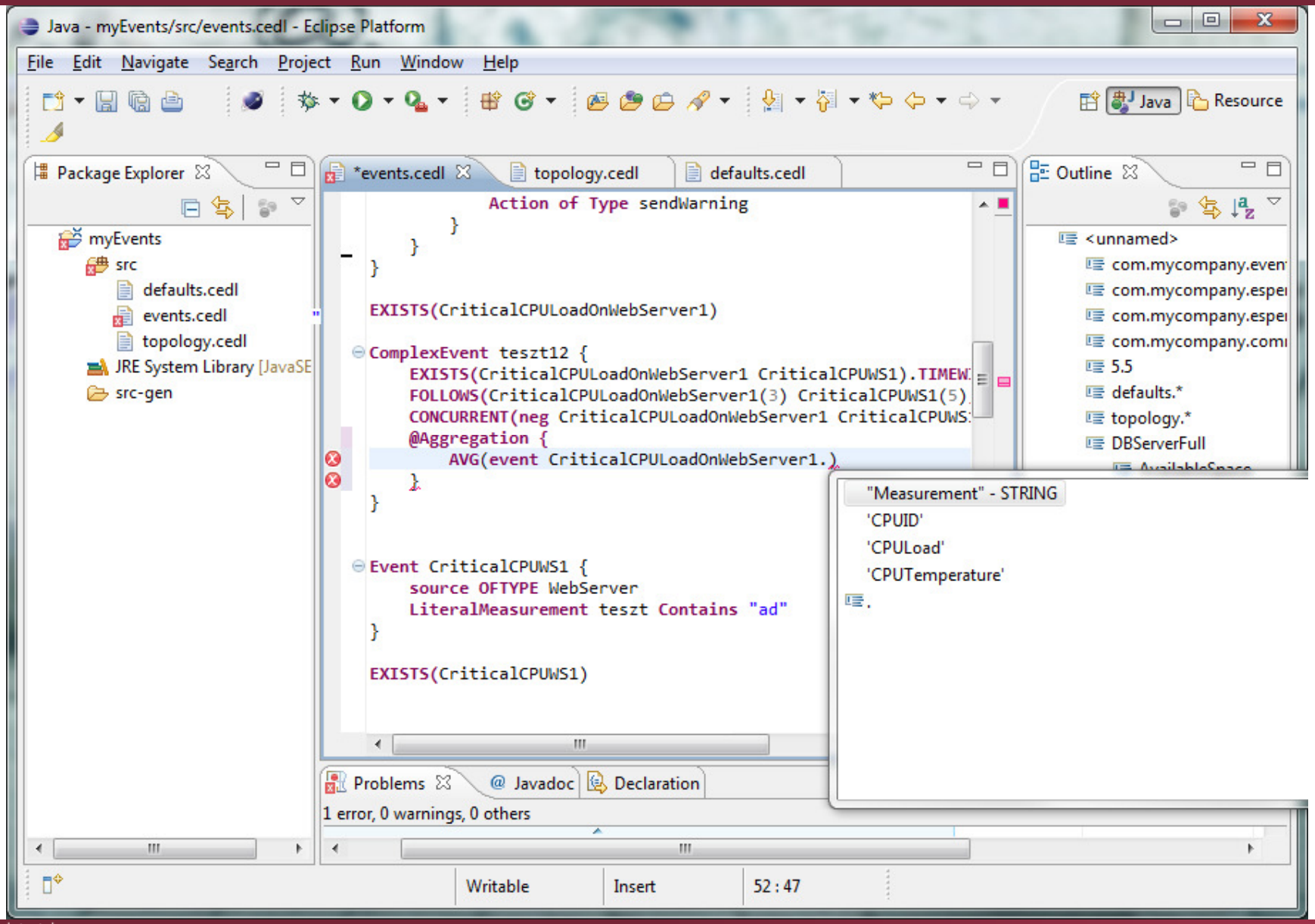
- Generált kód: nem csak eseményminta definíciók!
  - Esemény-definíciós osztályok *(EventBean*), feldolgozást végző osztályok (*UpdateListener*), interfész definíciók, konfigurációs állományok
  - Mindezt **egy** modellben definiáltuk
- Esper esetében: háromszor több utasítás a generált kódban, mint a modellben
  - Nagyobb rendszereknél ez az arány várhatóan növekszik

Integrált fejlesztőkörnyezet

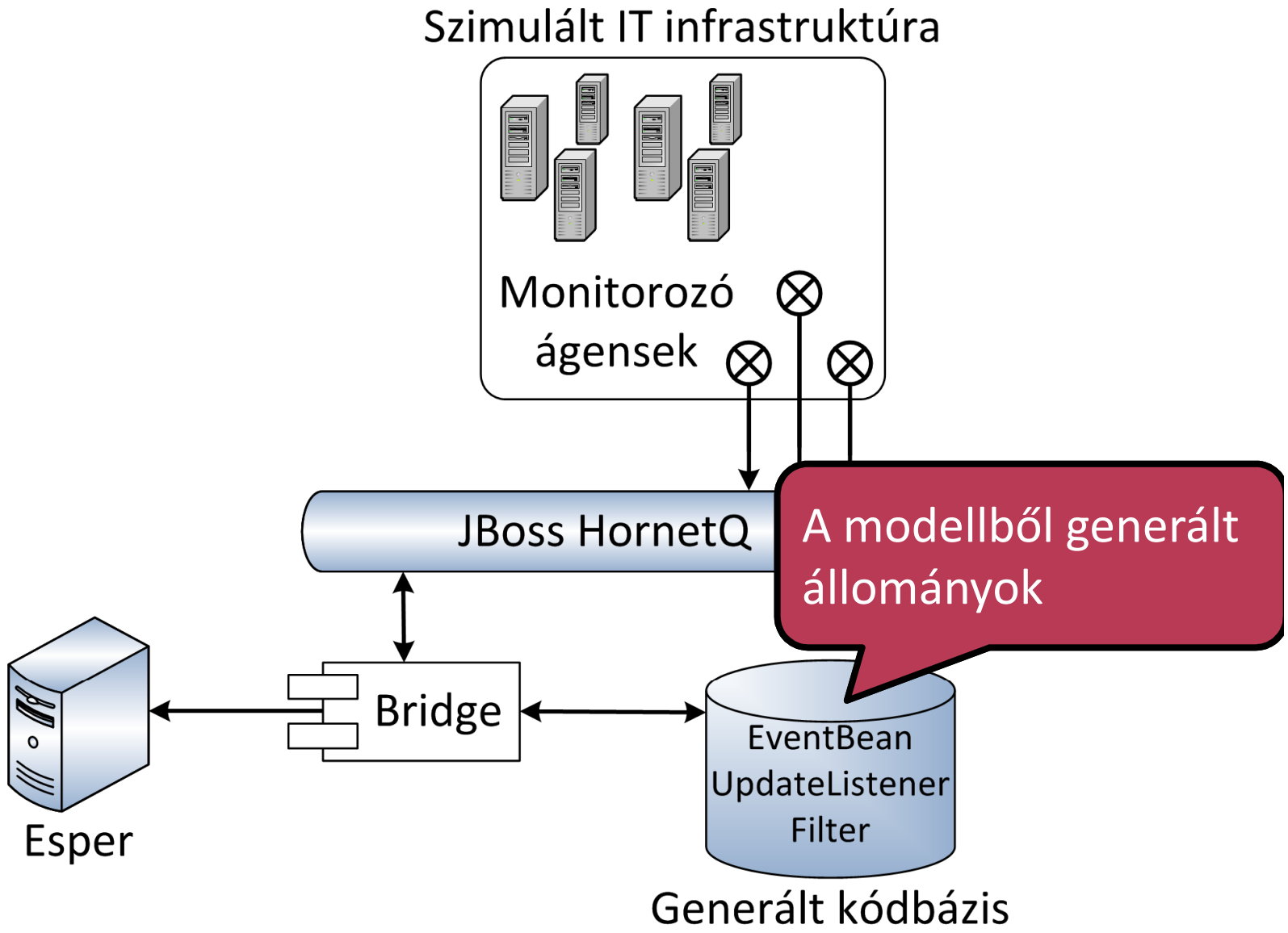


- Domain-specifikus fejlesztőeszköz
- Kifejezetten komplex események modellezéséhez
- A munkafolyamat támogatása több ponton

Integrált fejlesztőkörnyezet



Tesztkörnyezet



Továbbfejlesztési lehetőségek

- Analízis és integráció támogatása
  - Modell létező kódbázisból
- Grafikus felület
- Proaktív irányítórendszerek fejlesztésének támogatása
  - Historikus adatok visszacsatolása a szemantikus tudásbázisba
  - Valósidejű intelligens feldolgozás (klaszterezés, osztályozás)
  - Lásd: MAPE-K szabályozás
  - Példa alkalmazás: algoritmikus kereskedés modellalapú támogatása