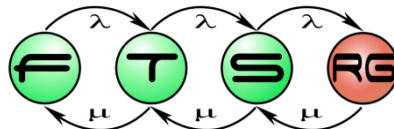


# Rendszerintegráció és –felügyelet laboratórium

Bartha Tamás,  
Bergmann Gábor, Hegedüs Ábel, Izsó  
Benedek, Kocsis Imre, Polgár Balázs



# A tárgy célja

- Alapjai a „Szolgáltatásintegráció” és “Autonóm és hibatűrő rendszerek” tárgyak
- Folyamat alapú rendszertervezés/integrációs technológiák
  - Komponens leírás, vezérlés, kommunikáció
  - Modell alapú folyamat tervezés
  - Monitorozás, elemzés, vizualizáció
- Elosztott üzleti alkalmazások fejlesztése

# A tárgy célja

- Saját üzleti folyamaton kipróbálni a technológiákat
  - Absztrakt munkafolyamat leképezése az eszközökre
  - Üzleti logika: „ $X=X+1$ ”,
  - Adatmodell van: néhány osztály, attribútumok
  - Vezérlés: elágazás, fork-join
- Előfeltétel
  - Java alapismeretek
  - Angol dokumentáció megértése
- Cél: egyszerű legyen implementálni a folyamatot a különböző technológiákban, ugyanakkor megértessük az egyes technológiák működését egy (a szerkezetét tekintve) realiztikus folyamaton.

# Követelmények

- Csak olyan mérésre kapható érdemjegy,
  - amelyen a Hallgató részt vesz,
  - a mérés előtt az írásos ellenőrző kérdéseket legalább elégséges szinten megválaszolja,
    - a beugrót komolyan vesszük! Aki felkészületlenül érkezik, az nem kezdheti el a mérést, és pótmérésen kell pótolnia!
  - valamint a mérésen elvégzett munkáját a laborvezető jóváhagyja.

# Követelmények

- A félévközi ellenőrzés módja a laboratóriumi mérési feladatok sikeres elvégzése.
- Értékelés:
  - Minden laboratóriumi mérést külön pontszámmal értékelünk, ez
    - a laboratóriumi tevékenység, és
    - a jegyzőkönyv együttes értékeléséből adódik.
  - A félévközi jegy megszerzéséhez minden mérést legalább elégséges szinten (40%) kell teljesíteni!
  - A félévközi jegyet a méréseken szerzett pontszámok átlagából számítjuk.
    - 40% – 55% – 70% – 85%

# Feladatok beadása

- Csoportthonlapra feltöltendő ZIP:
  - Forrás (program kód, konfigurációs leírók, modellek)
  - Jegyzőkönyv (PDF)
- Mérési jegyzőkönyvek:
  - igényes forma (követhető, rendezett)
  - tartalom: mi a feladat, mi okozott kihívást, hogyan lehetett megoldani
  - mindig tartalmazza a munkafolyamatot, leírását, és az adatmodellt
- Az elkészítésre 1 hét áll rendelkezésre.

# Pótlás

- Pótlási célból
  - a szorgalmi időszakban egy pótmérési alkalmat biztosítunk,
  - ezen túl a pótlási időszakban is lesz egy pótmérési alkalom.
- A pótmérésen vagy a pótlási időszakban **összesen egy** mérés pótolható.
- A pótmérésre a megadott határidőig jelentkezni kell!

# A mérések időzítése

- Első alkalom (laborismertető): **február 13.**
- Hallgatói mérések:

Időpont	Mérés neve	Mérésvezető
február 20.	Munkafolyamatok megvalósítása Java nyelven	Izsó Benedek
március 6.	Rendszerfelügyelet komplexesemény-feldolgozással	Bergmann Gábor
március 13.	Megbízható üzenetküldés IBM WebSphere MQ alapon	Hegedüs Ábel
március 20.	Kommunikáció JMS és JMX technológia segítségével	Hegedüs Ábel
március 27.	OSGi szolgáltatások fejlesztése	Izsó Benedek
április 10.	Modell alapú eszközintegráció	Polgár Balázs
április 24.	Felügyeleti adatok vizuális elemzése	Kocsis Imre

- Pótmérés a szorgalmi időszakban: **május 8.**
- Pót-pótmérés: **május 22.** (pótlási hét).

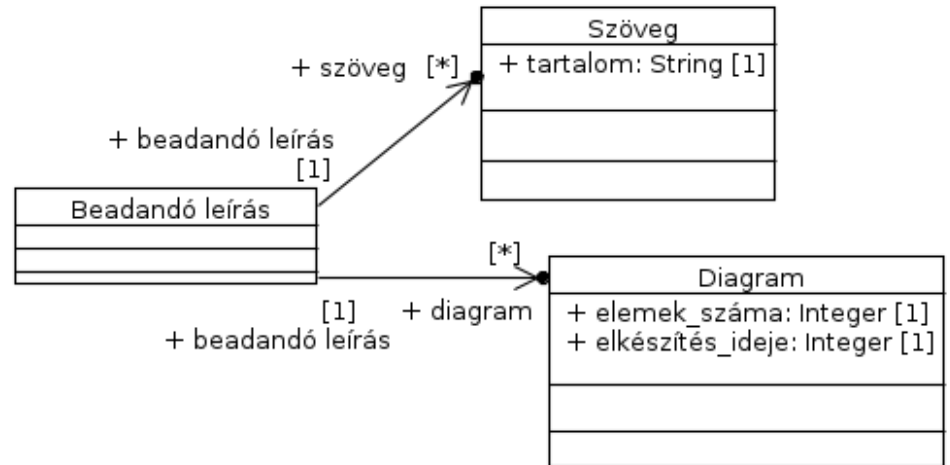
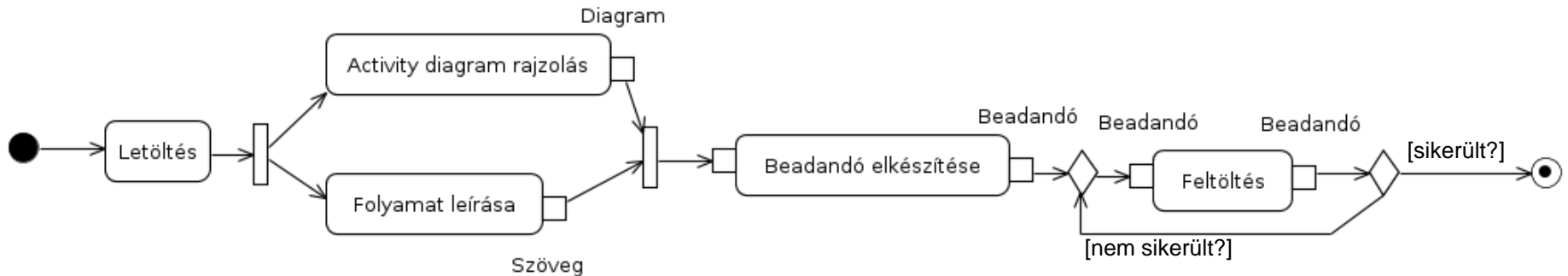


# A munkafolyamat

- „Munkadarab megy végig a gyártósoron”
  - Több elosztott erőforrás dolgozik rajta
  - Pl. Rendelés feldolgozás, hitelkérelem elbírálás, felvételi jelentkezés továbbítása, online útrendelés, egyenleg-lekérdezés, ...
  - Eszközintegráció
- Nem állapotgép jellegű, hanem munkafolyamat (vagy adatfolyam) orientált

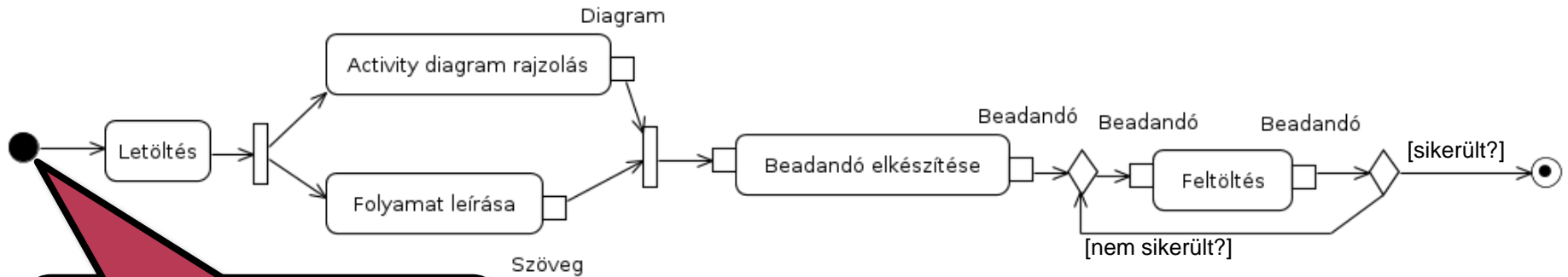
# A munkafolyamat

- Munkafolyamat ábrázolása: activity diagram
- Adatszerkezet ábrázolása: class diagram



# A munkafolyamat

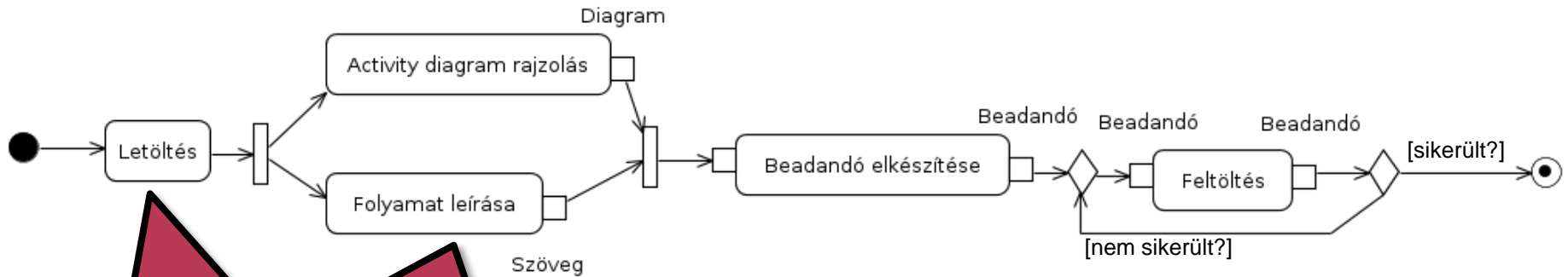
- Munkafolyamat ábrázolása: activity diagram
- Adatszerkezet ábrázolása: class diagram



**Initial node:**  
Itt kezdődik a futás

# A munkafolyamat

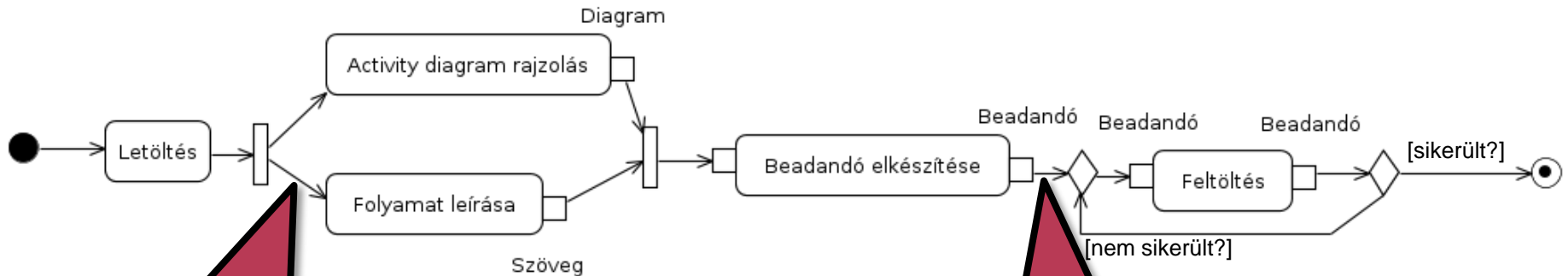
- Munkafolyamat ábrázolása: activity diagram
- Adatszerkezet ábrázolása: class diagram



**Action:**  
Feladat végrehajtása

# A munkafolyamat

- Munkafolyamat ábrázolása: activity diagram
- Adatszerkezet ábrázolása: class diagram

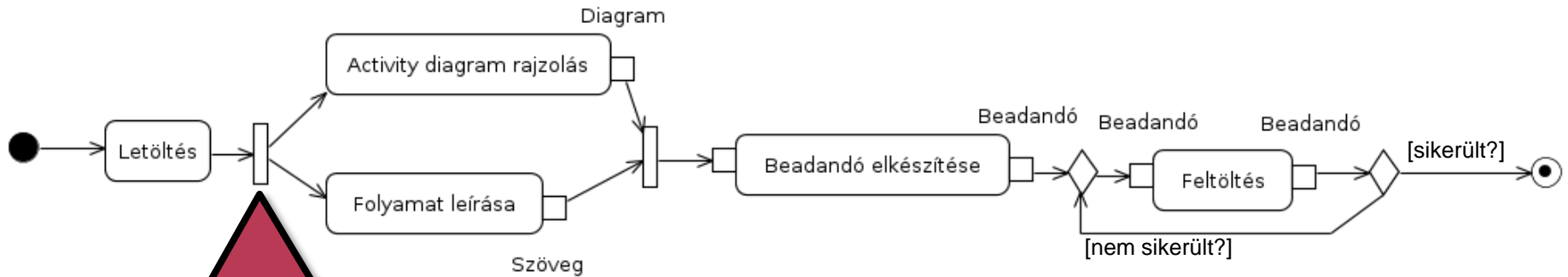


**Control flow:**  
Futás vezérlése

**Object flow:**  
Objektum tokenek  
vándorlását mutatja.  
(Ilyen legyen végig.)

# A munkafolyamat

- Munkafolyamat ábrázolása: activity diagram
- Adatszerkezet ábrázolása: class diagram

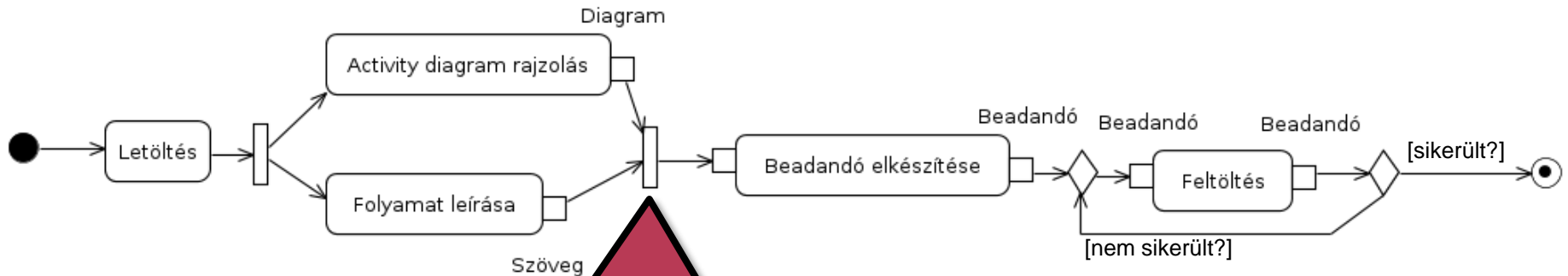


## Fork node:

- Bejövő token továbbküldése összes kimenő élen.
- Örfeltétel korlátozhatja.

# A munkafolyamat

- Munkafolyamat ábrázolása: activity diagram
- Adatszerkezet ábrázolása: class diagram

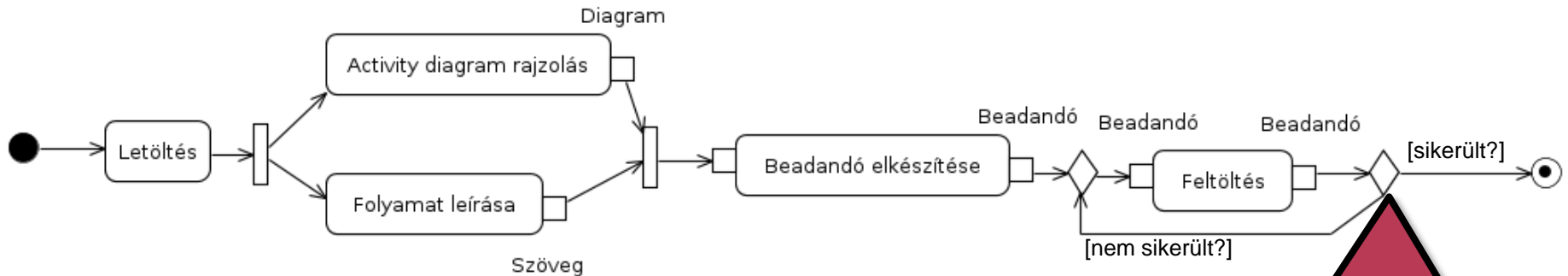


## Join node:

Minden bejövő élen tokenek várása  
**(szinkronizáció)**, majd kimenő élen továbbítás.

# A munkafolyamat

- Munkafolyamat ábrázolása: activity diagram
- Adatszerkezet ábrázolása: class diagram



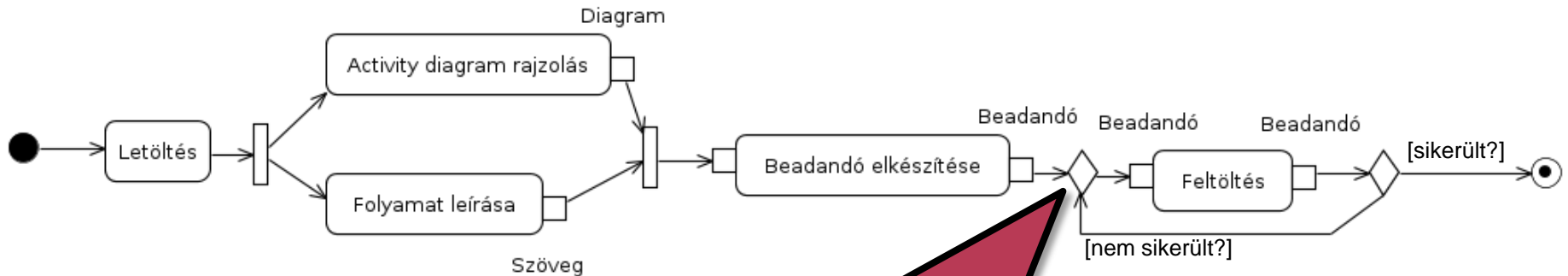
## Decision:

Minden bejövő tokent pontosan 1 kimenő élen továbbít (őrfeltételek szerint).



# A munkafolyamat

- Munkafolyamat ábrázolása: activity diagram
- Adatszerkezet ábrázolása: class diagram

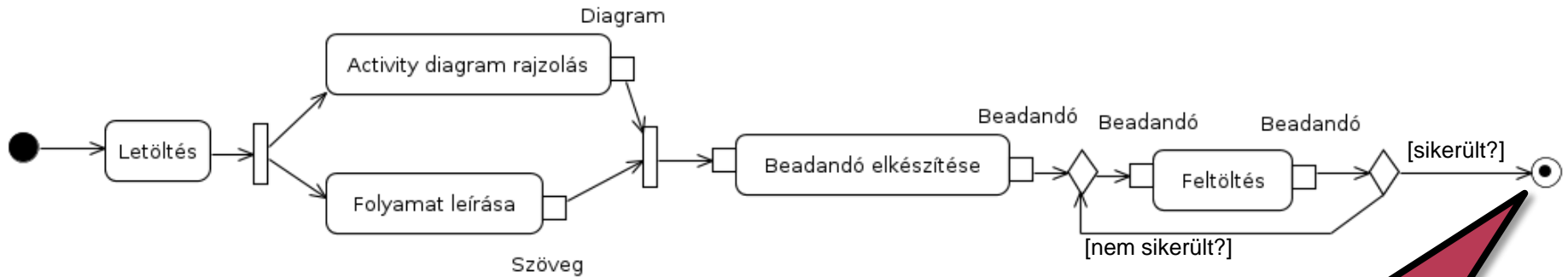


## Merge:

Minden bejövő élen  
elfogadja a tokent, és  
**szinkronizálás nélkül**  
továbbít.

# A munkafolyamat

- Munkafolyamat ábrázolása: activity diagram
- Adatszerkezet ábrázolása: class diagram



## Flow final:

A folyam megszakad: a tokenek elnyelődnek. (Az aktivitás párhuzamosan ettől még futhat.)

## Activity final:

Az egész aktivitás futása megszakad

# Mérések bemutatása

# 1. Munkafolyamatok megvalósítása Java nyelven

Mérésvezető: Izsó Benedek

# Munkafolyamatok megvalósítása Java nyelven

## ■ Feladat

- Az előre elkészített workflow megvalósítása
  - Java alaptechnológiákkal (Swing + Thread)

## ■ Követelmények

- Minden workflow csomópont külön szálon fusson
- Minden egyes csomópontot külön ablak jelképezzen
- Fork és join műveletek szálak közötti kommunikációval
- Exception mentes működés

# Munkafolyamatok megvalósítása Java nyelven

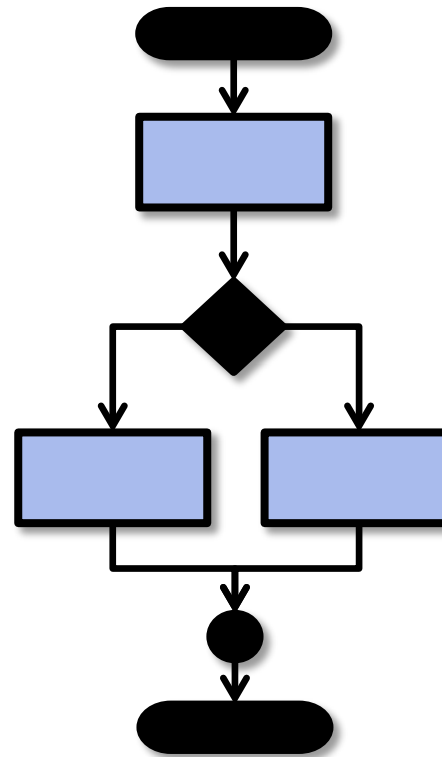
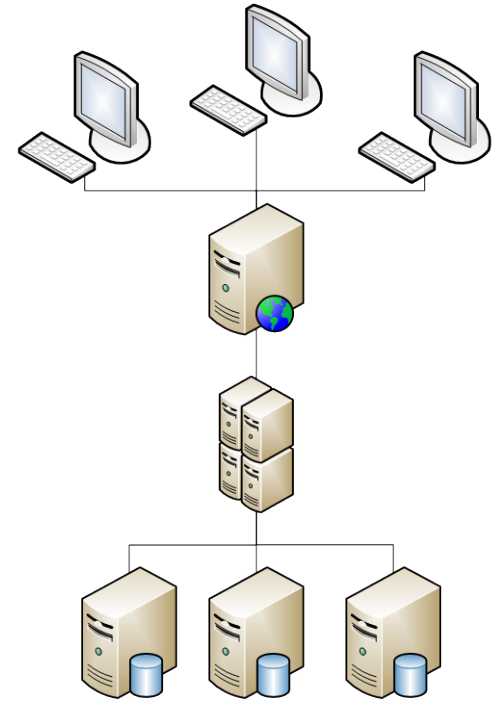
- Munkafolyamat
  - **Lépések:** külön szálak egy alkalmazáson belül
  - **Vezérlés, szinkronizáció :** elosztottan, lépések egymást hívják vagy központi szállal
  - **Adatáramlás:** szálak közötti kommunikációval
- Monitorozás
  - Külön ablak minden csomóponthoz

## 2. Rendszerfelügyelet támogatása komplexesemény-feldolgozással

Mérésvezető: Bergmann Gábor

# IT rendszerek felügyelete

- Üzleti folyamatok, szolgáltatások
- Kiszolgáló infrastruktúra
- Kiszolgáló infrastruktúra erőforrásainak felügyelete
  - Erőforrás gazdálkodás
  - Szűk keresztmetszetek
- Alkalmazás szintű események

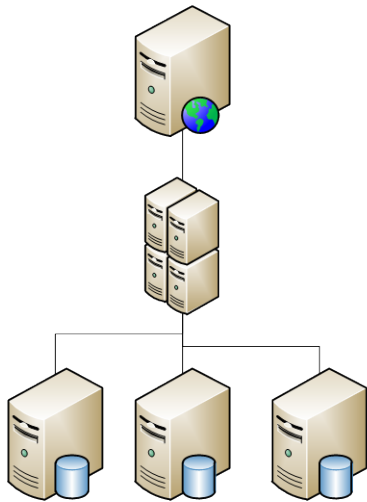




# Felügyelet / Monitoring

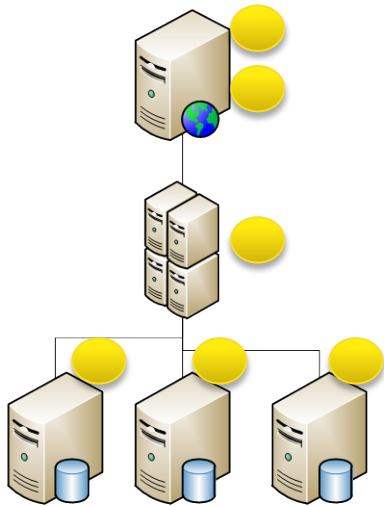
- IT rendszert „**felműszerezzük**”
- Mérőszámokat, jelzéseket **gyűjtünk**
- **Komplex eseményeket** figyelünk / számolunk
  - „ha a CPU>80%, de a szolgáltatás nem válaszol...”
  - „ha a RAID tömb konzisztenciát javít, és közben egy arra épülő szolgáltatásra bejelentkezik valaki...”
  - „ha az elmúlt öt percben feltűnően gyakran volt sikertelen tranzakció...”
- Felügyelet **eseményfolyam analízissel**

# Complex Event Processing



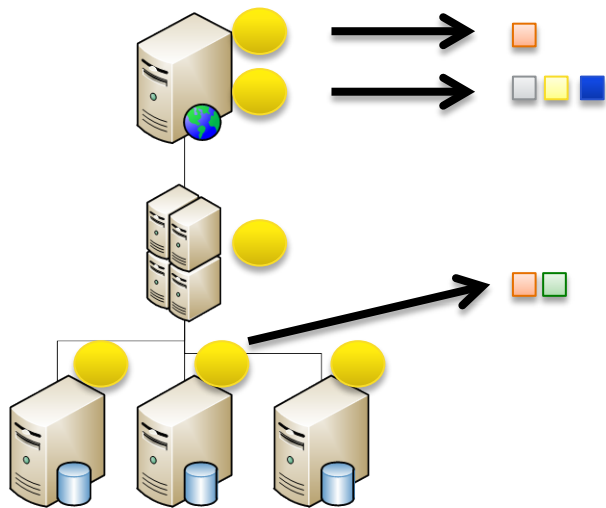
# Complex Event Processing

## Eseményforrások

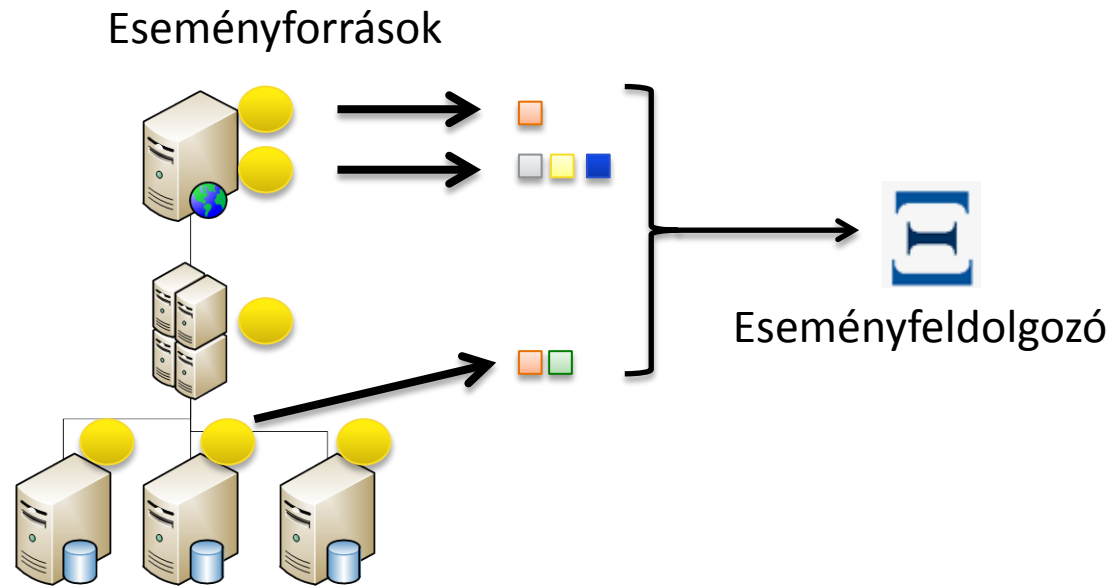


# Complex Event Processing

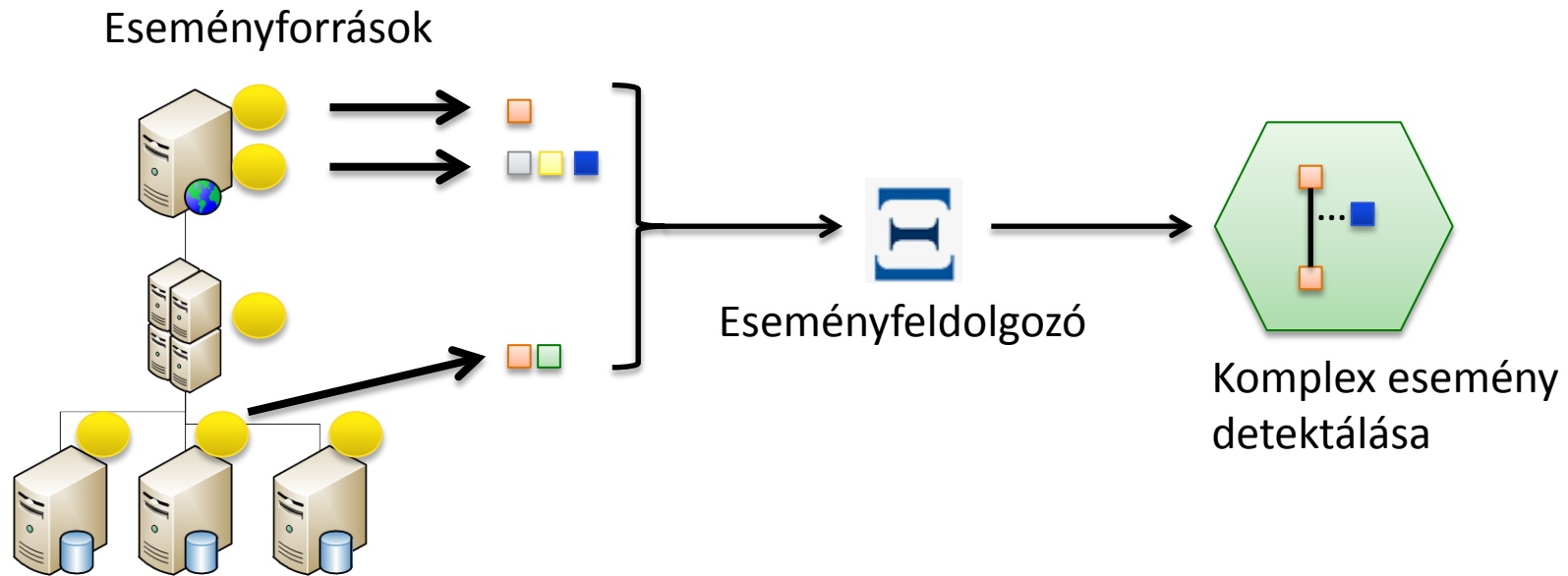
Eseményforrások



# Complex Event Processing



# Complex Event Processing



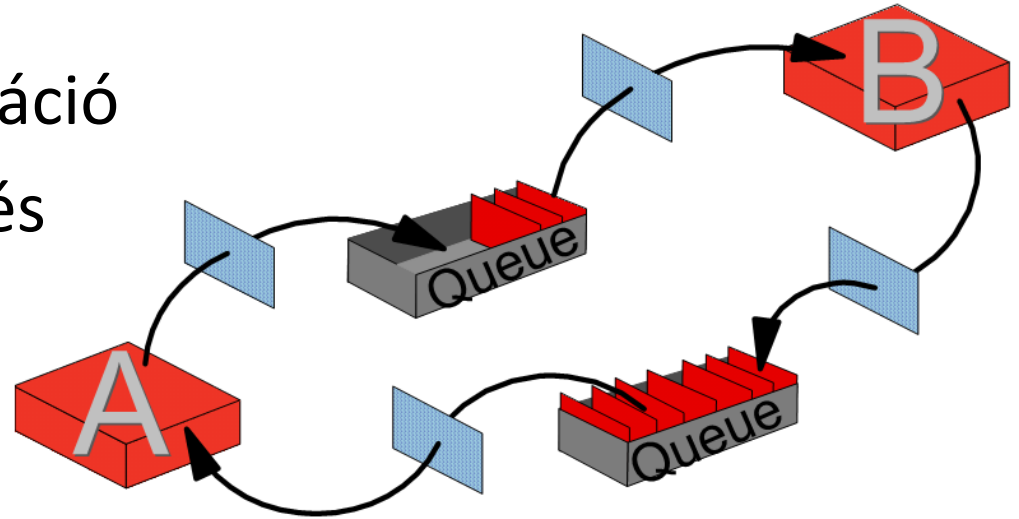
# 3. Megbízható üzenetküldés Websphere MQ alapon

Mérésvezető: Hegedüs Ábel

# Megbízható üzenetküldés IBM WebSphere MQ alapon

## ■ Üzenetsorok:

- Aszinkron kommunikáció
- Lazán csatolt működés
- Megbízható



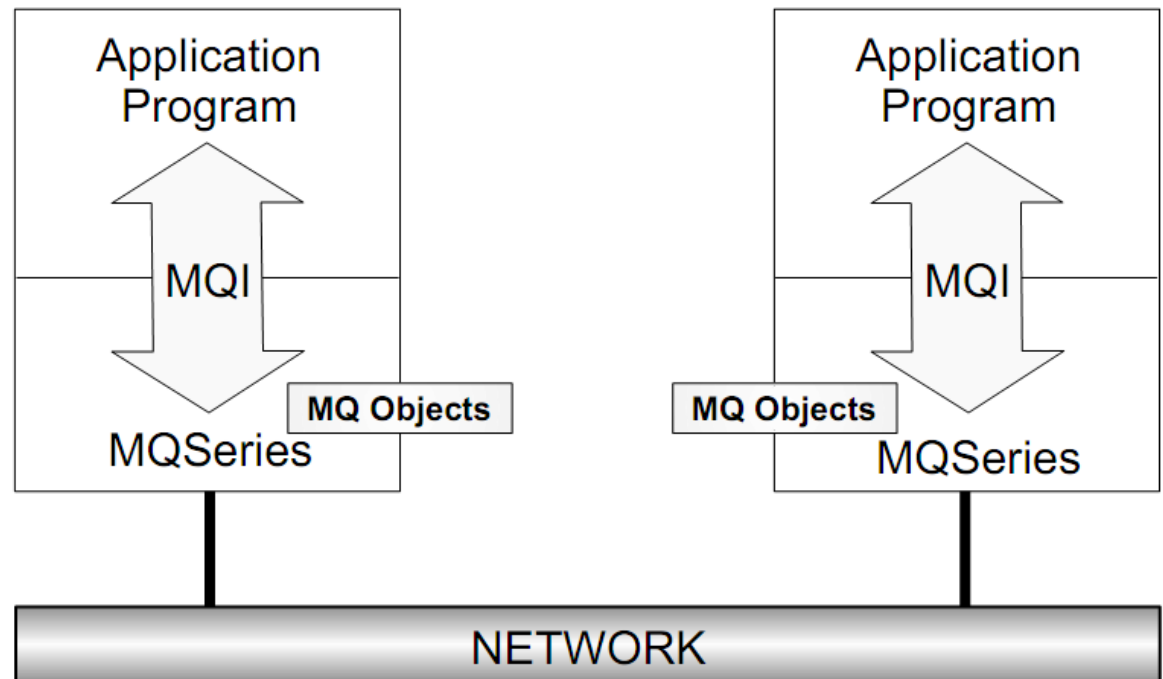
## ■ Munkafolyamat:

- lépések = komponensek
- vezérlés, szinkronizáció = üzenetek
- adatáramlás = üzenet tartalma



# Megbízható üzenetküldés IBM WebSphere MQ alapon

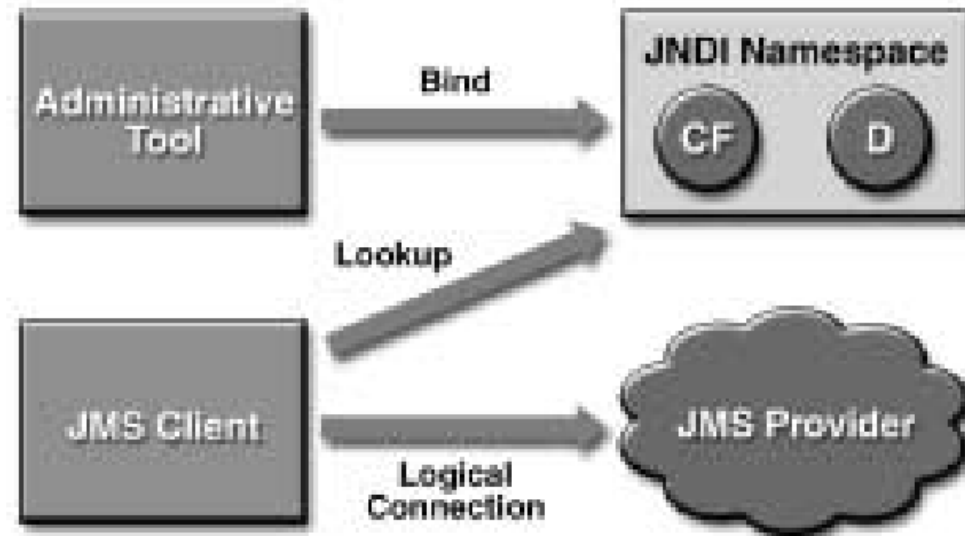
- IBM Websphere MQ:
  - Kommunikációs middleware
  - Multi-platform
  - API különböző nyelvekhez
  - Üzenetsorok
    - létrehozása
    - kezelése
    - felügyelete



# 4. Kommunikáció JMS és JMX technológia segítségével

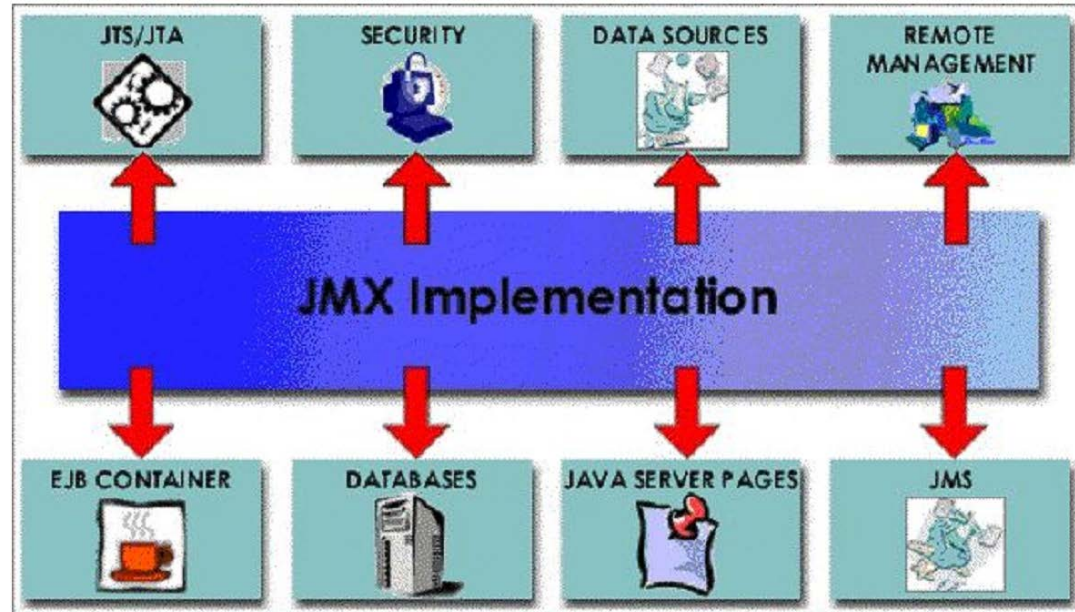
Mérésvezető: Hegedüs Ábel

- Java Messaging Service (JMS):
  - Általános megvalósítás üzenetsorok használatára
  - Platform-független
    - a kód hordozható különböző MQ middleware-ek között
  - Kapcsolódás JNDI segítségével
  - Mérésen:
    - Websphere MQ megvalósítás átalakítása JMS-hez



# Kommunikáció JMS és JMX technológia segítségével

- Java Management Extension (JMX):
  - Management és integrációs keretrendszer
  - Mérésen:
    - JMS szolgáltatások kezelése
    - üzenetsorok létrehozása, megfigyelése



# 5. OSGi szolgáltatások fejlesztése

Mérésvezető: Izsó Benedek

# OSGi keretrendszer

- Plug-in keretrendszer Javához
  - Moduláris fejlesztés
  - Cél
    - Külön fejlesztett modulok -> közös alkalmazás
- Dinamikus
  - Szolgáltatások elindulnak/leállnak
  - Plug-inek megjelennek/eltűnnek

# Ismert OSGi alkalmazások

- IDE
  - Eclipse
  - Netbeans
- Alkalmazás szerverek
  - GlassFish (V3)
  - Websphere (IBM)
  - Weblogic (Oracle)
- ...

# OSGi szolgáltatások fejlesztése

- Munkafolyamat
  - **Lépések:** külön OSGi kötetek (bundle), egy OSGi alkalmazás, szolgáltatások deklaratívan megadva
  - **Vezérlés, szinkronizáció:** külön kötetben „kézzel”
  - **Adatáramlás:** OSGi szolgáltatás paraméterén keresztül
- Monitorozás
  - Plug-inek és szolgáltatások elérhetősége



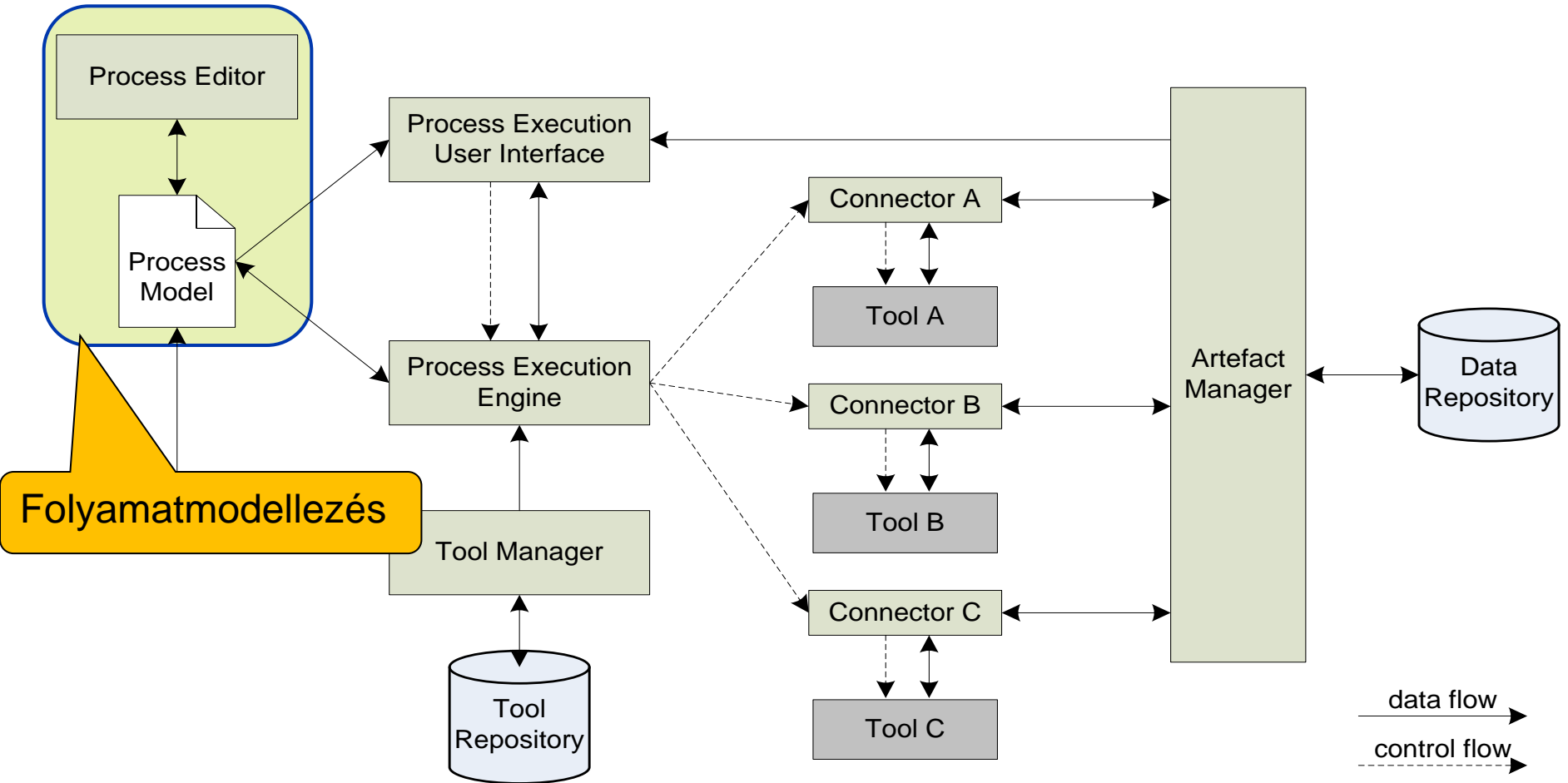
# 6. Modell alapú eszközingegráció elosztott környezetben

Mérésvezető: Polgár Balázs

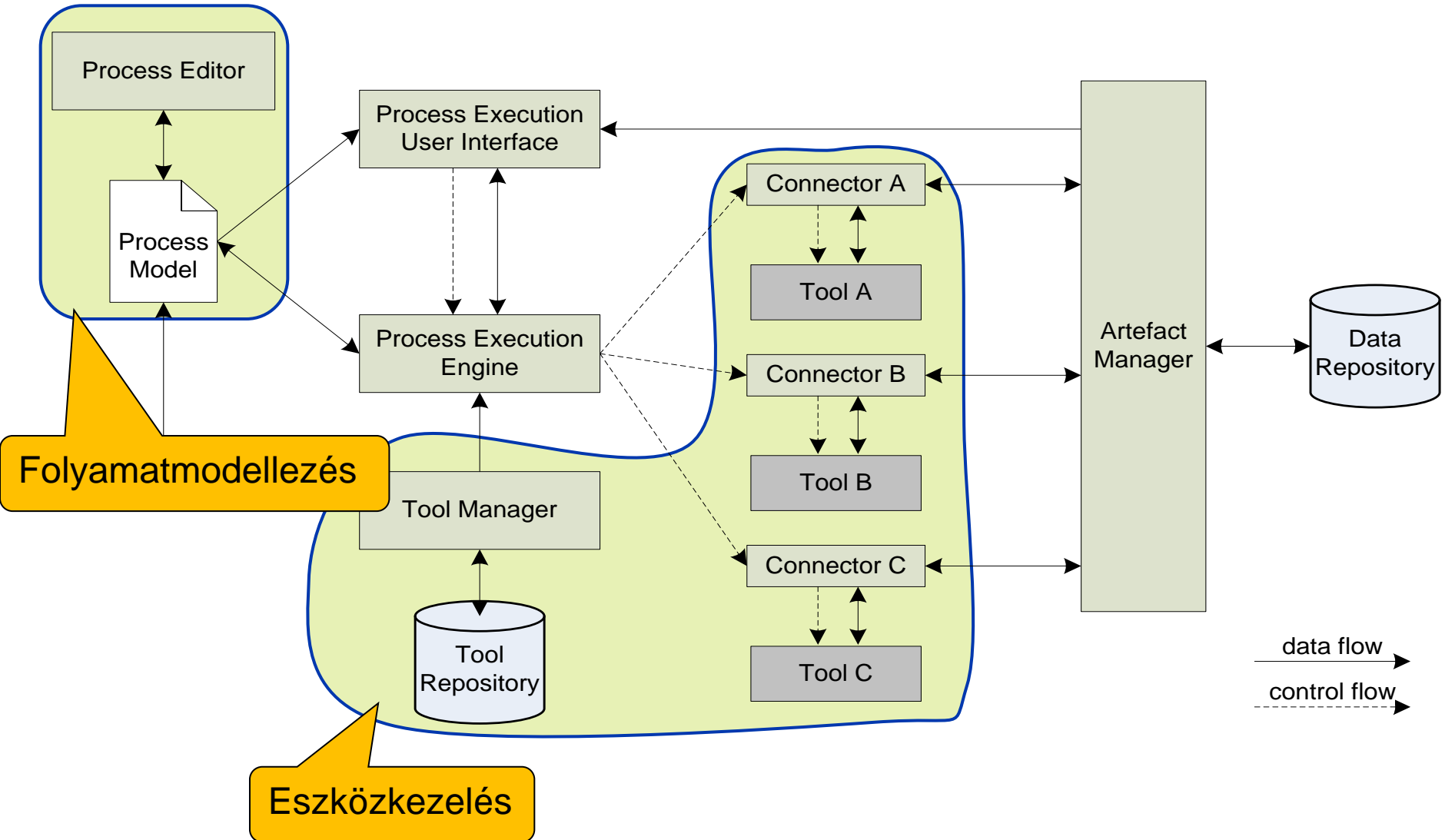
# INTEGRA

- Szolgáltatás-orientált platform (fejlesztői) eszközök integrációjára
  - Eszközök funkcionalitása szolgáltatásként definiálva
  - Eclipse (OSGi) alapú
- Szolgáltatásokból munkafolyamat szervezhető
  - Pl. alkalmazás modelljének külső (matematikai) eszközzel való ellenőrzése
- Folyamat központilag futtatható, monitorozható
- Támogatás a folyamat során kezelt adatok perzisztálására

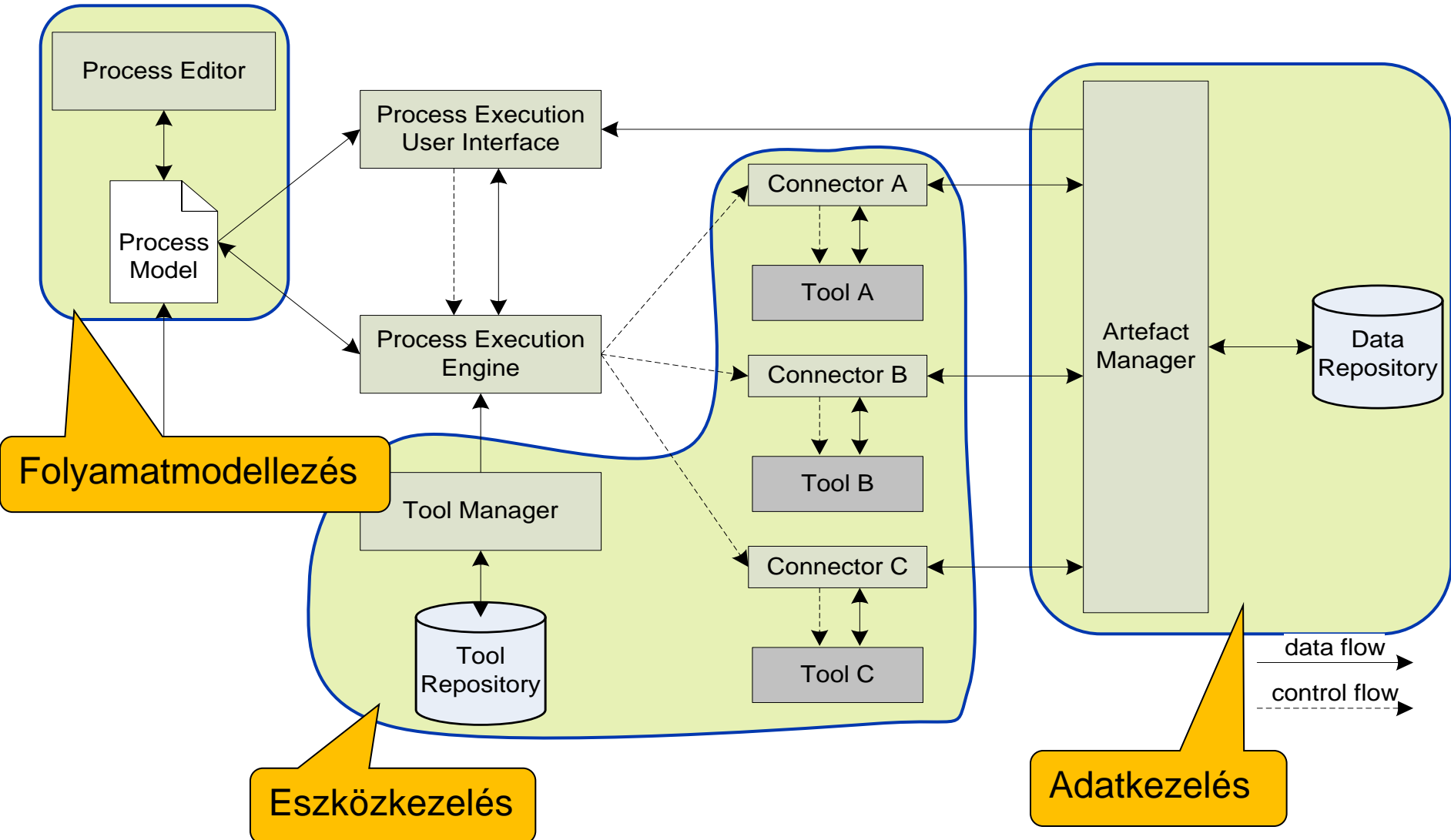
# Architektúra



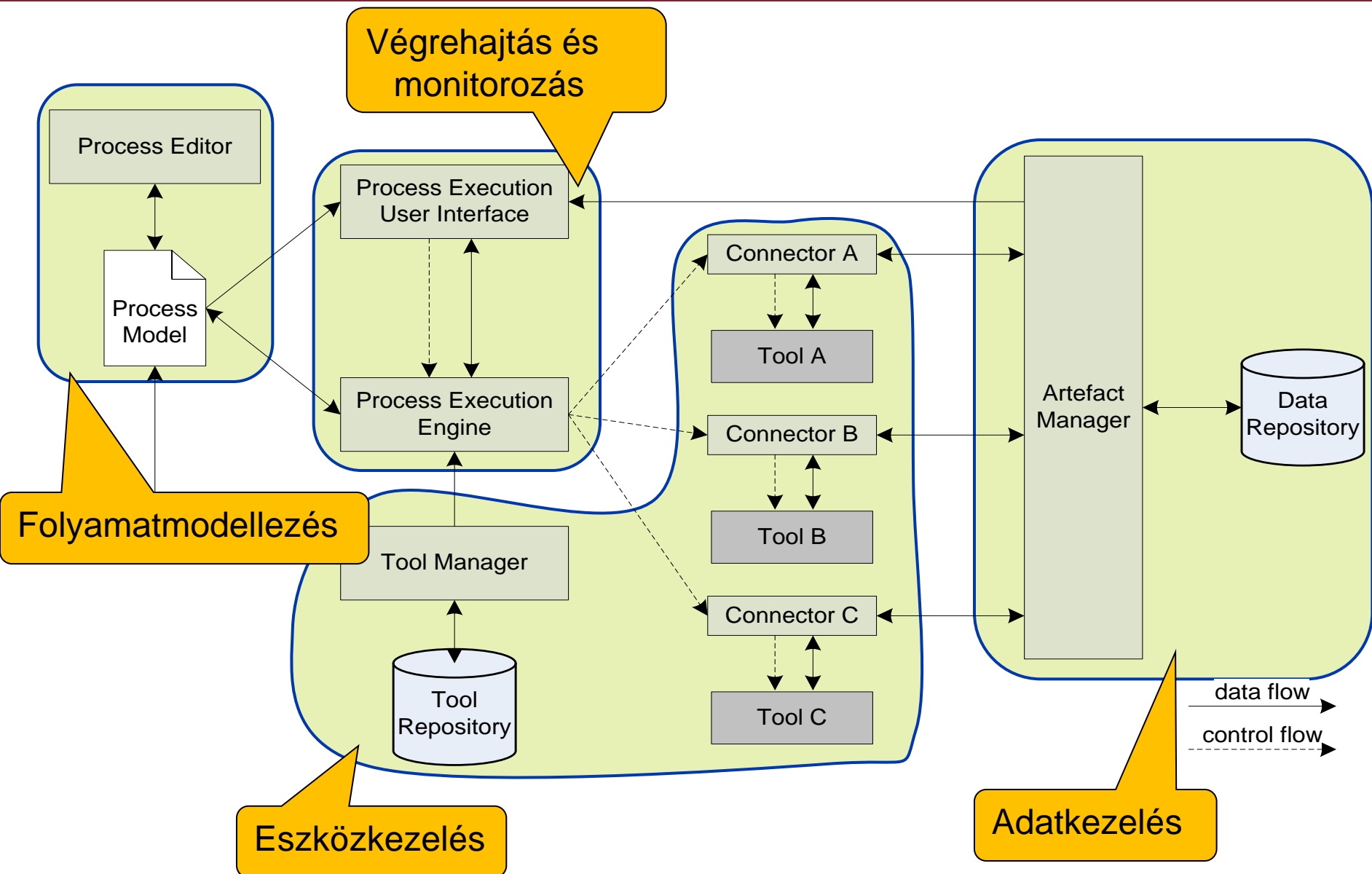
# Architektúra



# Architektúra



# Architektúra



# Modell alapú eszközintegráció

- Munkafolyamat
  - **Lépések:** OSGi szolgáltatások
  - **Vezérlés, szinkronizáció:** modell alapján központi végrehajtó motorral
  - **Adatáramlás:** OSGi szolgáltatás paraméterén, ill. perzisztens adattáron keresztül
- Monitorozás
  - Folyamatmodell alapján végrehajtó felületen

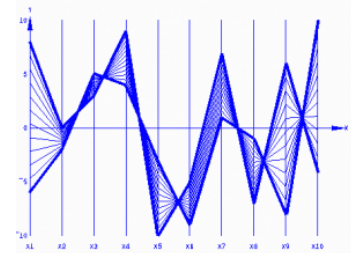
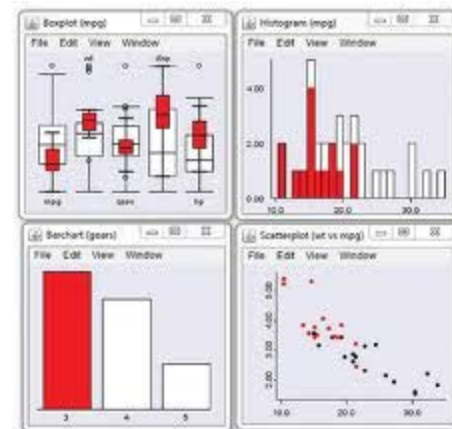
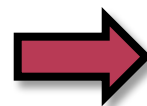
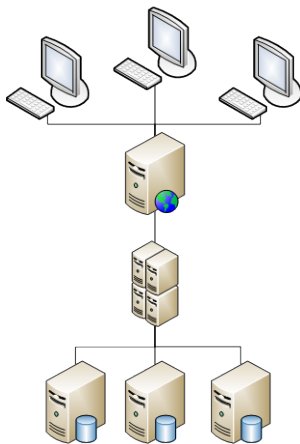
# 6. Felügyeleti adatok vizuális elemzése

Mérésvezető: Kocsis Imre



# Motiváció

- NAGY és heterogén infrastruktúra
- NAGYON sok adat
- sok forrásból
- Tároljuk
  - Naplózzuk
- Feldolgozzuk
  - Vizuális analízis
    - Ember számára emészthető módon
    - Jó kezdeti analízis / becslés
    - Ökölszabályok szakterület részletes ismerete nélkül
    - Jól prezentálható 😊



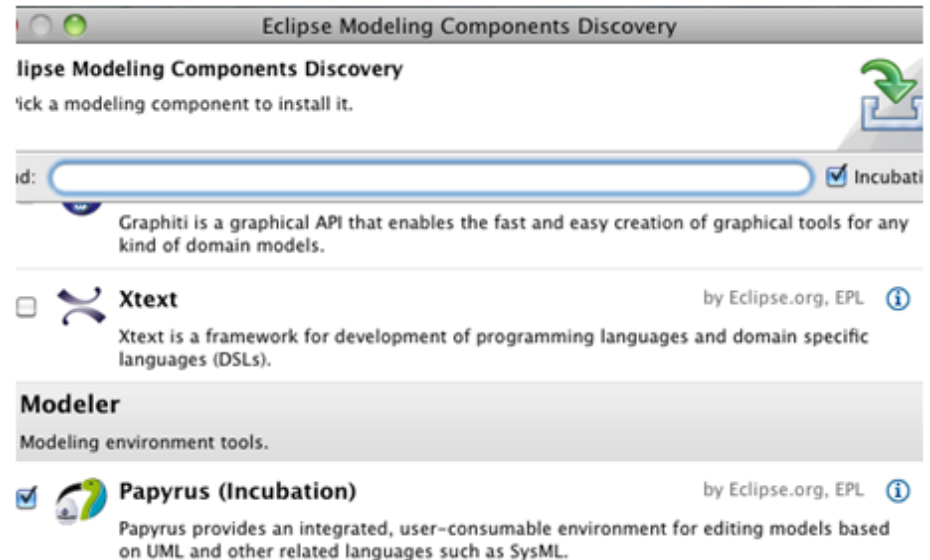
# Aktuális teendők

# 0. házi feladat

- Munkafolyamat és adatszerkezetének megtervezése
- Csoportonlapon keresztül feltölteni (ZIP)
  - Forrás: munkafolyamat, adatszerkezet (modell project)
  - Jegyzőkönyv (PDF): rövid leírás a munkafolyamatról (hogyan működik, mi történik az egyes lépésekben)
- Határidő: 2013. február 19. 12 óra

# Papyrus UML - Letöltés

- Eclipse Modeling Tools
  - Help/Install Modeling Components
  - Papyrus -> Finish



- Visual Paradigm UML editor (Community Edition):
  - <http://www.visual-paradigm.com/product/vpuml/>

# 0. házi feladat

- Üzleti folyamat követelményei
  - Felépítés:
    - 6-8 lépés
    - Legyen benne: elágazás, párhuzamos lépések
    - Ne legyen benne hurok
    - A lépések ne akarjanak pl. adatbázist elérni (impl.)
  - Minden lépés hatása megfigyelhető legyen az adatokon
- Adatszerkezet követelményei: néhány (1-3) osztály, attribútumokkal (pl.: int, String)