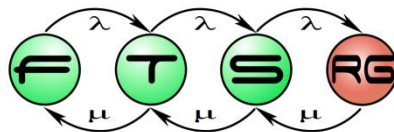


- Hoszt gép – **CoolerMaster** gépek
 - felhasználó: meres
 - jelszó: LaborImage
- Virtuális gép:
 - vmware-images: CSI - Lab - 2017
 - “I copied it” opció
 - felhasználó: meres
 - jelszó: LaborImage
 - A felhasználó sudo joggal rendelkezik



Kritikus rendszerek integrációja laboratórium

Dr. Bergmann Gábor, Honfi Dávid,
Klenik Attila, Semeráth Oszkár,
Dr. Szatmári Zoltán, **Szárnyas Gábor**, Tóth Tamás



A tárgy célja

- Oktatott téma
 - Szolgáltatás alapú rendszerintegráció
 - Komponensek leírása, vezérlése, kommunikációja
 - Monitorozás, elemzés, vizualizáció
 - Elosztott üzleti alkalmazások fejlesztése
 - Modell alapú folyamattervezés
 - Párhuzamos programozás
- Előfeltétel
 - Java alapismeretek
 - Angol dokumentáció megértése
- Oktatási módszer
 - Egyes technológiák kipróbálása egy egyszerű feladaton

A tárgy célja

- Féléven végighúzódó implementációs példafeladat
 - „Szöveghasonlóság-vizsgálat üzletileg kritikus ajánlórendszerben”
 - Részletes specifikáció a weben:
<https://github.com/FTSRG-CSI/csi-lecture-notes/wiki/Running-Example>
 - Az egyes mérések építenek a korábbiakon elkészült megoldásra!
- Segítséget adunk
 - Témánként bevezető segédanyag (nem tankönyv)
<https://github.com/FTSRG-CSI/csi-lecture-notes/wiki>
 - Személyes konzultáció a mérési alkalmak során
 - Virtuális gép telepített fejlesztő- és futtatókörnyezettel
 - Mérőpáronként GitHub repository
 - Automatikus builddel, integrációs teszttel

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!
 - a mérésen (mérőtárssal közösen) elvégzett munkáját a laborvezető személyes bemutatás alapján jóváhagyja,
 - és az elkészült megoldásról kellő részletességű jegyzőkönyvet ad be

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

- A mérőpár részére kiadott Git repositoryba
 - Forrás (program kód, konfigurációs leírók, modellek)
 - Jegyzőkönyv (ld. később)
 - Minden mérési feladat külön Git branch!
- Beadás: Git tag elhelyezése határidőig (vasárnap)
- Előtte a kész megoldás személyes bemutatása!
 - Ideálisan még a mérésen, de legkésőbb péntekig
 - A jegyzőkönyv itt még nem kell
- Travis CI működése kötelező (2 pont)
 - GitHub / Travis / Maven repository outage-ot leszámítva

Jegyzőkönyv

- Dokumentáció: docs / könyvtárba
 - Pl. `1-basic-java-implementation.md`
 - Sima GitHub Markdown is jó, de persze lehet PDF
 - Igényes forma (követhető, rendezett)
- Tartalma:
 - A feladat (röviden)
 - Telepítési útmutató („Getting Started”)
 - Fejlesztői útmutató („Developer’s Guide”):
 - architektúra
 - kihívások és ezek megoldása
 - Technológia és labor értékelése, jobbító javaslatok

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!

Időzítés

- Egy mérési alkalom ütemezése

H	K	Sze	Cs	P	Szo	V
Segédlet elérhető kb. egy hétrel előbb						
MÉRÉS 8 ³⁰ -12 ⁰⁰ Beugró + munka				Személyes bemutatás határideje (ha a mérésen nem lett kész)		Végleges kód, jegyzőkönyv beadása (tag)

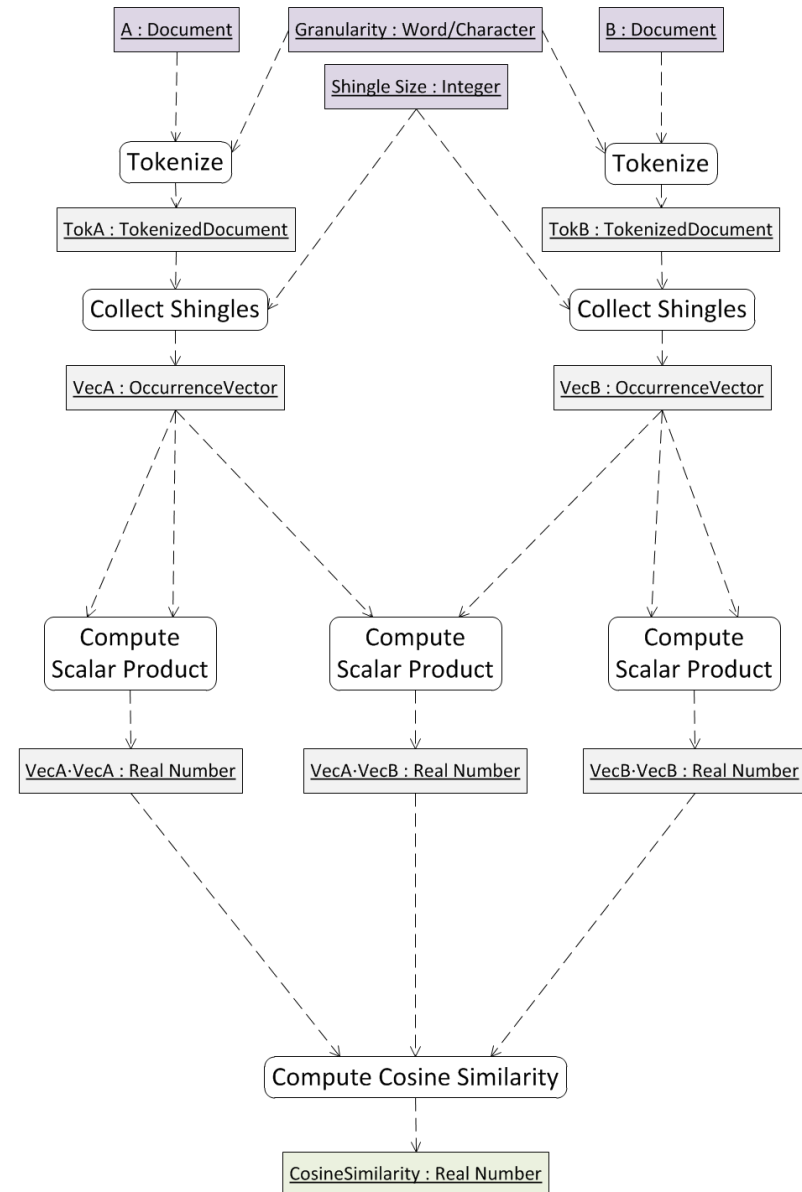
MÉRÉSEK BEMUTATÁSA

1. PÉLDAFELADAT MEGVALÓSÍTÁSA JAVA NYELVEN

Mérésvezető: Szárnyas Gábor

Példafeladat

- Részletes specifikáció:
<https://github.com/FTSRG-CSI/csi-lecture-notes/wiki/Running-Example>



2. JAVA ALKALMAZÁSOK PÁRHUZAMOSÍTÁSA

Mérésvezető: Semeráth Oszkár

Java alkalmazások párhuzamosítása

■ Követelmények

- Minden tevékenység külön szálon fusson
- Kommunikáció a szálak között
 - Elosztottan vagy központi szállal
 - Várakozás vagy értesítést
 - `concurrent package`
- Adatáramlás – egyszerre több adat is mehet a munkafolyamatban
- Kivételek legyenek kezelve
- Működés naplózása (folyamatok kezdése és befejezése)
- Fork és join műveletek szálak közötti kommunikációval

Párhuzamos programozás

- Your multi-threaded code is broken (and I can prove it): <https://bss.sch.bme.hu/video/simonyi-konferencia-2015-your-multi-threaded-code-is-broken-and-i-can-prove-it>
- Concurrency: It's Harder (and Easier) Than You Think:
<http://www.ustream.tv/recorded/61447706>

3. MUNKAFOLYAMAT KERETRENDSZER MEGVALÓSÍTÁSA

Mérésvezető: Semeráth Oszkár

Munkafolyamatok megvalósítása Java nyelven

■ Követelmények

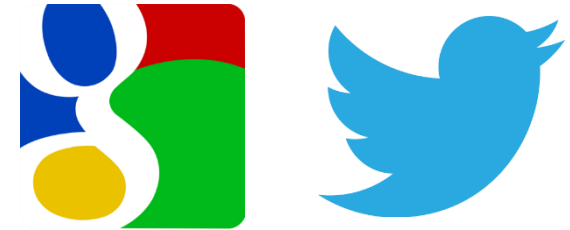
- Munkafolyamat keretrendszer megvalósítása
 - Többszálú megoldás (ld. 2. labor)
 - Többszálú viselkedés elfedése a csomópontok implementációjában
 - Átkonfigurálhatóság, újrafelhasználhatóság
- Java Generics megismerése
 - Típushelyes megoldás
 - Indokolatlan típusmegszorítások nélkül
 - Type cast nélkül
 - Warning nélkül (@SupressWarnings nem ér 😊)

4. AKTOR MODELL KONKURENS ALKALMAZÁSOK KÉSZÍTÉSÉRE: AKKA

Mérésvezető: Tóth Tamás

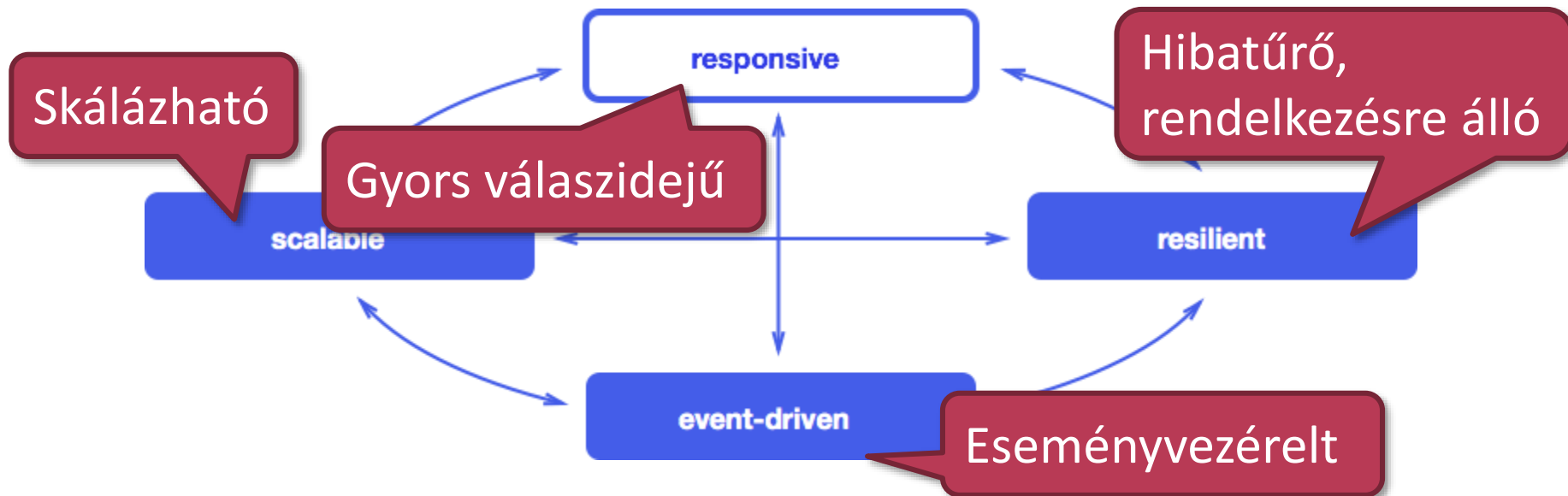
Reaktív alkalmazások

- Alkalmazások
 - Web 2.0: Google, Twitter
 - Pénzügy, telekommunikáció



- Reaktív rendszer:

<http://www.reactivemanifesto.org/>



Skálázhatóság

Moore-törvény

Amdahl-törvény

Tranzisztorok száma

Gyorsulási faktor

Intel CPU Trends

(sources: Intel, Wikipedia, K. Olukotun)

Dual-Core Itanium 2

Pentium 4

Pentium

386

Órajel

Év



SPEEDUP

95% parallel

90% parallel

75% parallel

50% parallel

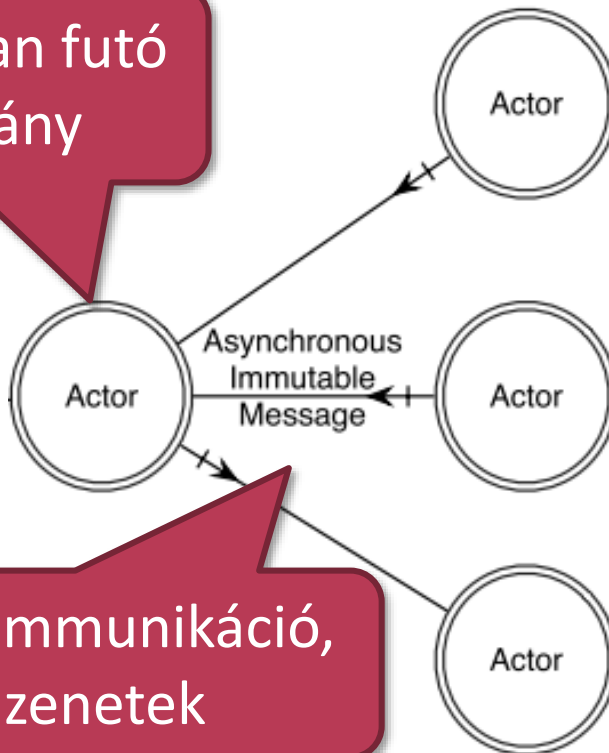
Processzorok száma

PROCESSORS

Párhuzamosítás

- Megszokott módszer: zárok alkalmazása
- Alternatív módszer: **aktor modell** (1973)
- Az aktorok **elosztottan** is futtathatók

Aktor: izoláltan futó
programpéldány



Aszinkron kommunikáció,
immutable üzenetek



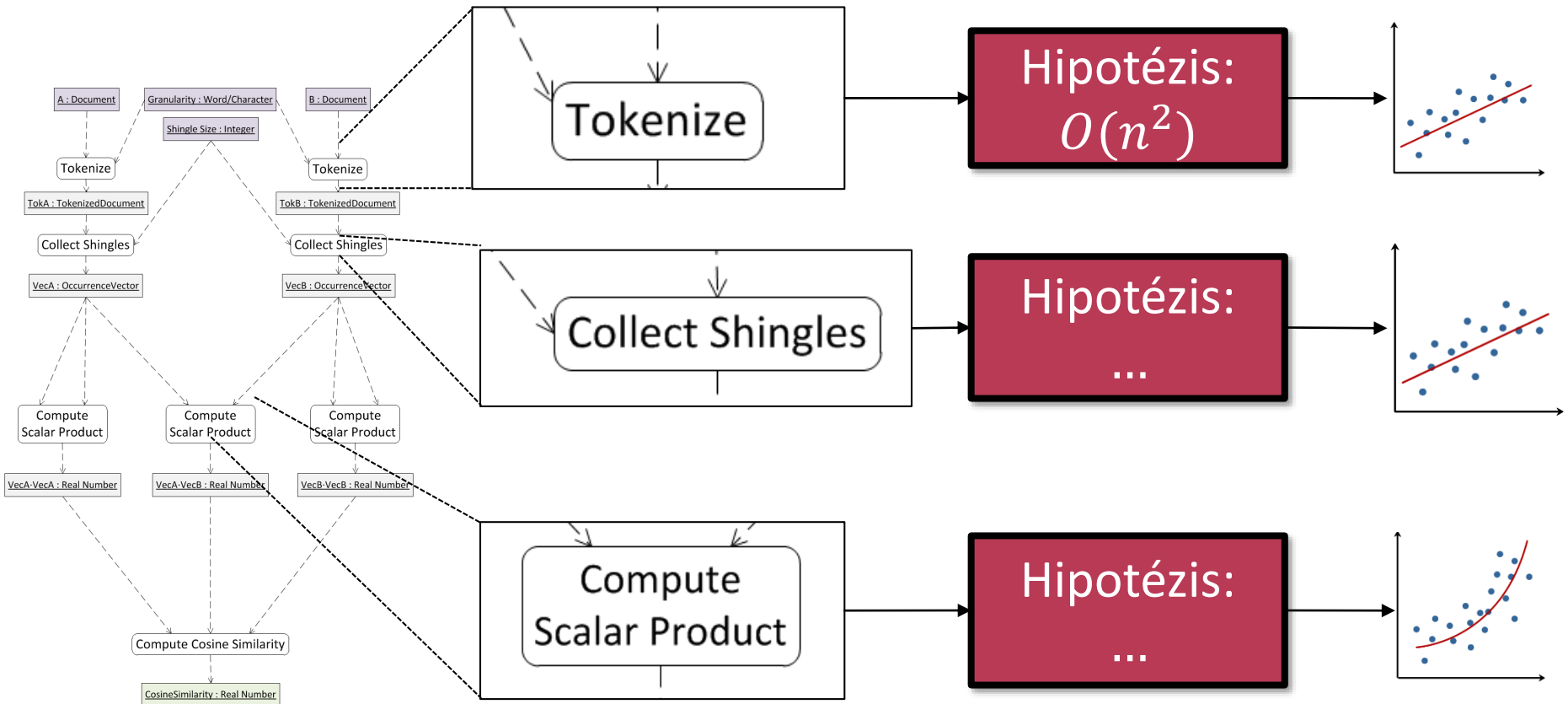
The logo for Akka, featuring a stylized blue mountain range above the word "akka" in a bold, black, lowercase sans-serif font.

5. TELJESÍTMÉNYELEMZÉST SEGÍTŐ ADATELEMZÉSI TECHNIKÁK

Mérésvezető: Klenik Attila

Mérésalapú elemzés

■ Implementációk teljesítményének mérése



Mérésalapú elemzés

- Mérés céljai
 - Hipotézis ellenőrzése
 - Szűk keresztmetszetek azonosítása
 - Implementációk összehasonlítása

- Környezet
 - Statisztikai elemzés R-ben
 - Alapvető vizualizációs technikák
 - Regresszió



Felkészülés

- Implementáció/mérés rész elkészítése
 - Eredmény CSV formátumba
- Segédanyag elolvasása
 - A valószínűségszámítás és a statisztika alapfogalmait ismertnek feltételezzük, de linkelünk forrásokat is
- R szintaxisának megtanulása
 - [DataCamp](#)



DataCamp

Amire szükségünk lesz

■ Némi statisztika

- Kísérlettervezési alapok
- Nagy számok törvénye
- Hipotézistesztelés alapjai

■ Kód instrumentáció

- **Workflow automatikus lejátszása adott késleltetéssel**
- Szimuláció → megismételhetőség!

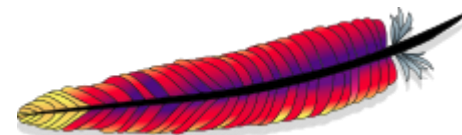
■ R

6. INTEGRÁCIÓ SZABVÁNYOS TÁVOLI ELJÁRÁSHÍVÁS PLATFORMON

Mérésvezető: Dr. Bergmann Gábor

Apache Thrift

- Platformfüggetlen szolgáltatás-leírónyelv (.thrift)
 - Adatstruktúrák
 - Szolgáltatás interfészek (RPC)
- Kódgenerátor
 - Adatstruktúra megvalósítása
 - Sorosítás és kommunikáció megvalósítása
 - Szerver és kliens váza
 - Különféle esetek (pl. szinkron / aszinkron, nemblokkoló stb.)
 - Célplatformok: Java, C++, C#, Python, Erlang, JS, ...
 - Szerver és kliens nyelve eltérhet!



7. MEGBÍZHATÓ ÜZENETKÜLDÉS RABBITMQ ALAPON

Mérésvezető: Honfi Dávid

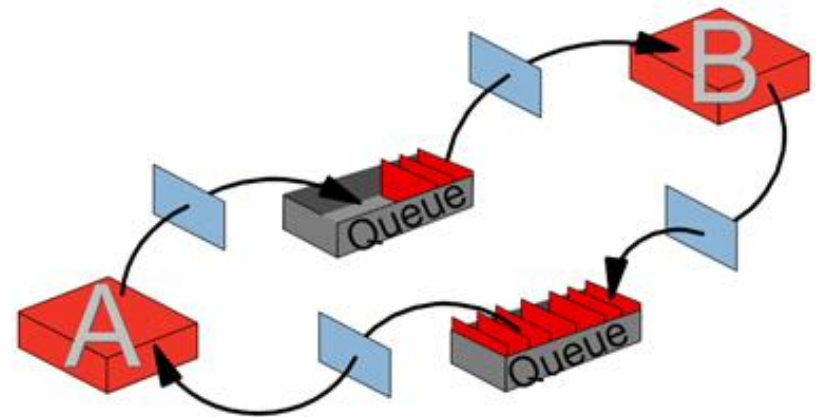
Megbízható üzenetküldés RabbitMQ alapon

■ A mérés céljai

- Üzenetsorok használatának elsajátítása
- RabbitMQ eszköz megismerése

■ Feladatok

- Munkafolyamat kommunikációjának átalakítása üzenet alapúvá
- RabbitMQ használata az üzenetkezeléshez

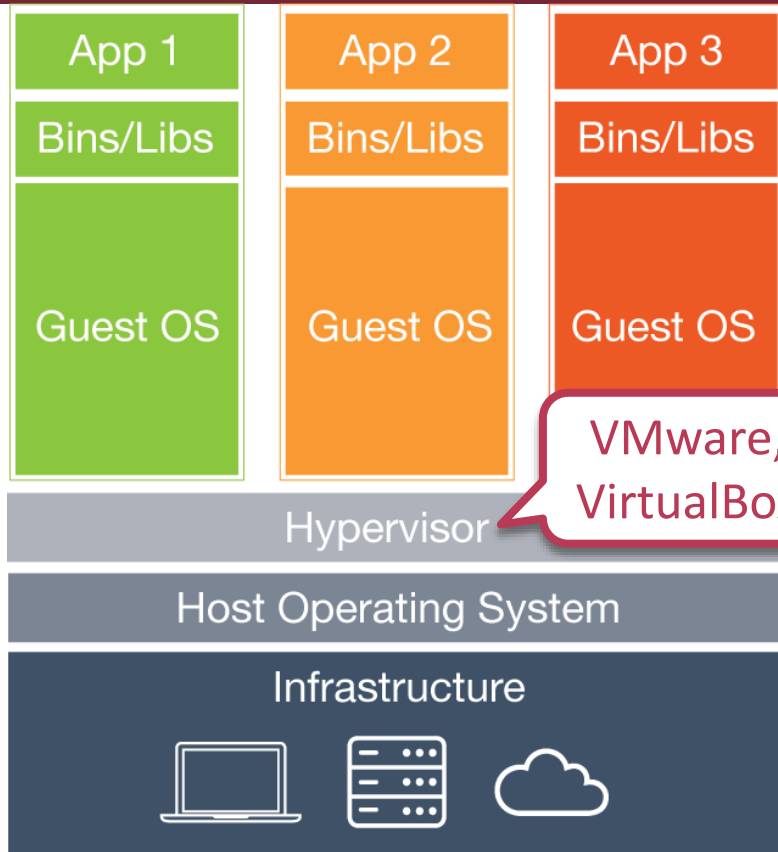


 RabbitMQ™

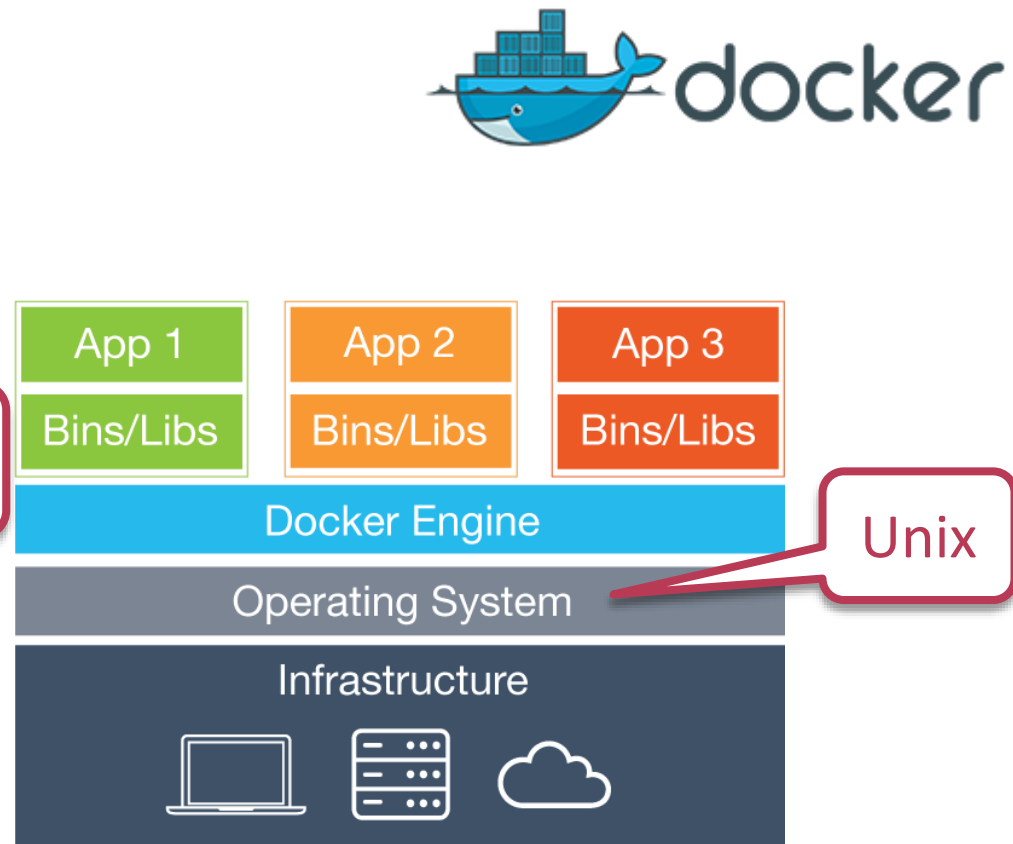
8. DEKLARATÍV ALKALMAZÁSKONTÉNER PLATFORM

Mérésvezető: Dr. Szatmári Zoltán

Docker



VMware,
VirtualBox



Unix

Platform virtualizáció
+ Teljes izoláció
– Nagy overhead

Alkalmazásszintű virtualizáció
+ Minimális vagy nulla overhead
– OS függő

Docker

■ Docker alapismeretek

- Nyílt forráskódú, Go nyelven készült
- Dockerfile: egy képfájlt definiál
- Konténerek parancssorból építhetők és indíthatók
- Docker Hub: központi hely képfájlok tárolására

■ Előző feladat

- Elosztott munkafolyamat Akka segítségével

■ Új feladat

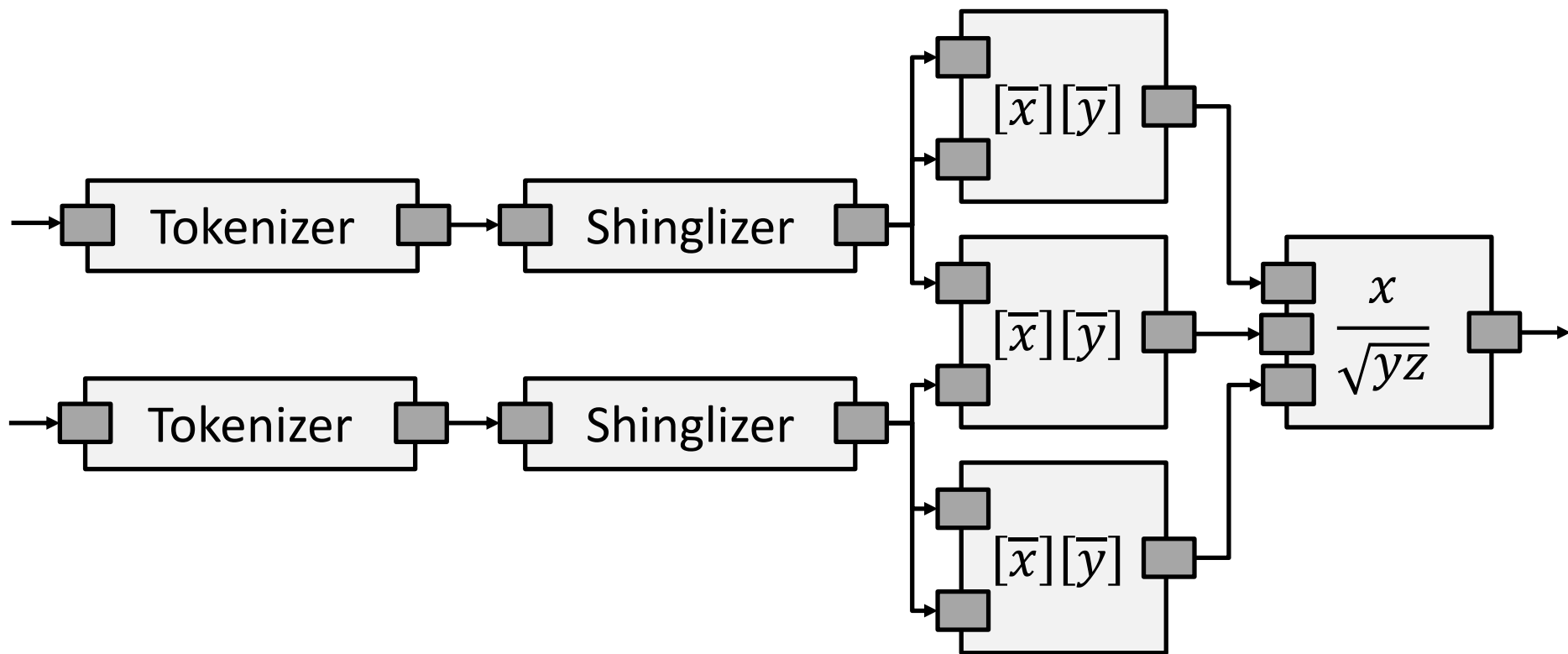
- Docker képfájlok készítése aktorok futtatásához
- Ezek telepítése és futtatása (deployment)

9. PLATFORMFÜGGETLEN MODELLEZÉS GENERATÍV PROGRAMOZÁSSAL

Mérésvezető: Dr. Bergmann Gábor

- Célok
 - Különböző technológiák integrációja/lecserélése
 - Folyamat átstrukturálása automatikusan
- Bonyolult + sok + ismétlődő összekötő kód
- Feladat: Folyamatmodellező nyelv + kódgenerálás
 1. Szétválasztani: generálható \leftrightarrow manuális kódrészeket
 2. Modellezni: a generálható részt
 3. Kódgenerátort készíteni: a modellező nyelvhez
 4. Összekötni: generált + manuális részeket
- Technológiák: EMF, Xtend, Eclipse fájlkezelés

Platformfüggetlen modellezés generatív programozással



AKTUÁLIS TEENDŐK

Aktuális teendők

- Mérőpárok alakítása: <http://tiny.cc/csi-lab-2017>
- Mai óra hátralevő része
 - GitHub repository, build rendszer megismerése
 - Megoldandó feladat elolvasása, megértése
 - Erősen javasolt: megoldás elkezdése
 - Sima, egyszálú Java program elég
- Jövő héttől minden alkalomra:
 - segédanyag elolvasása, felkészülés a beugróra