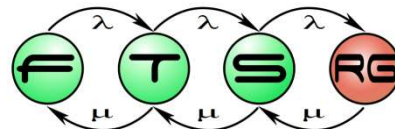
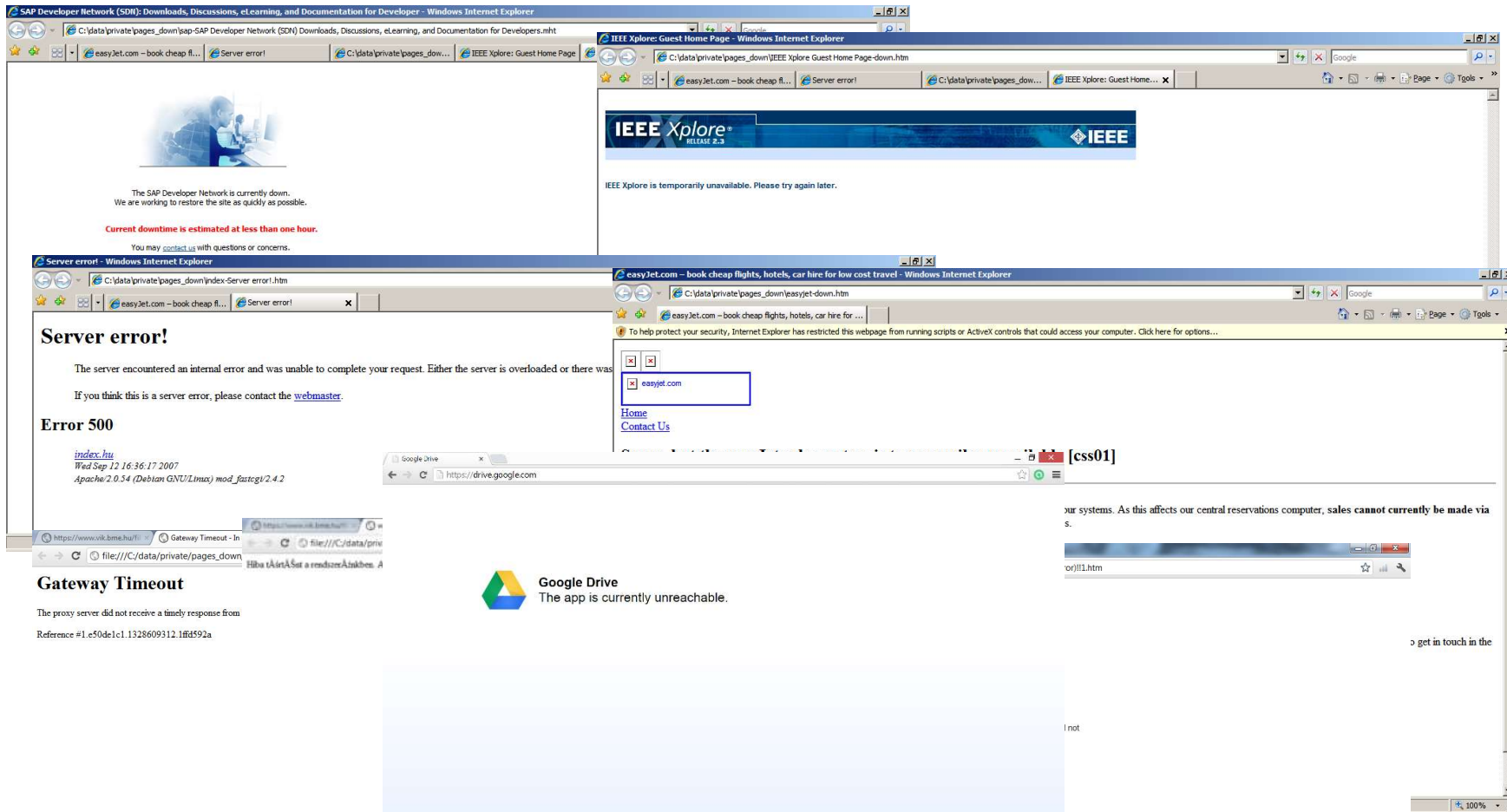


Kapacitásstervezési metodika

**Budapest University of Technology and Economics
Fault Tolerant Systems Research Group**



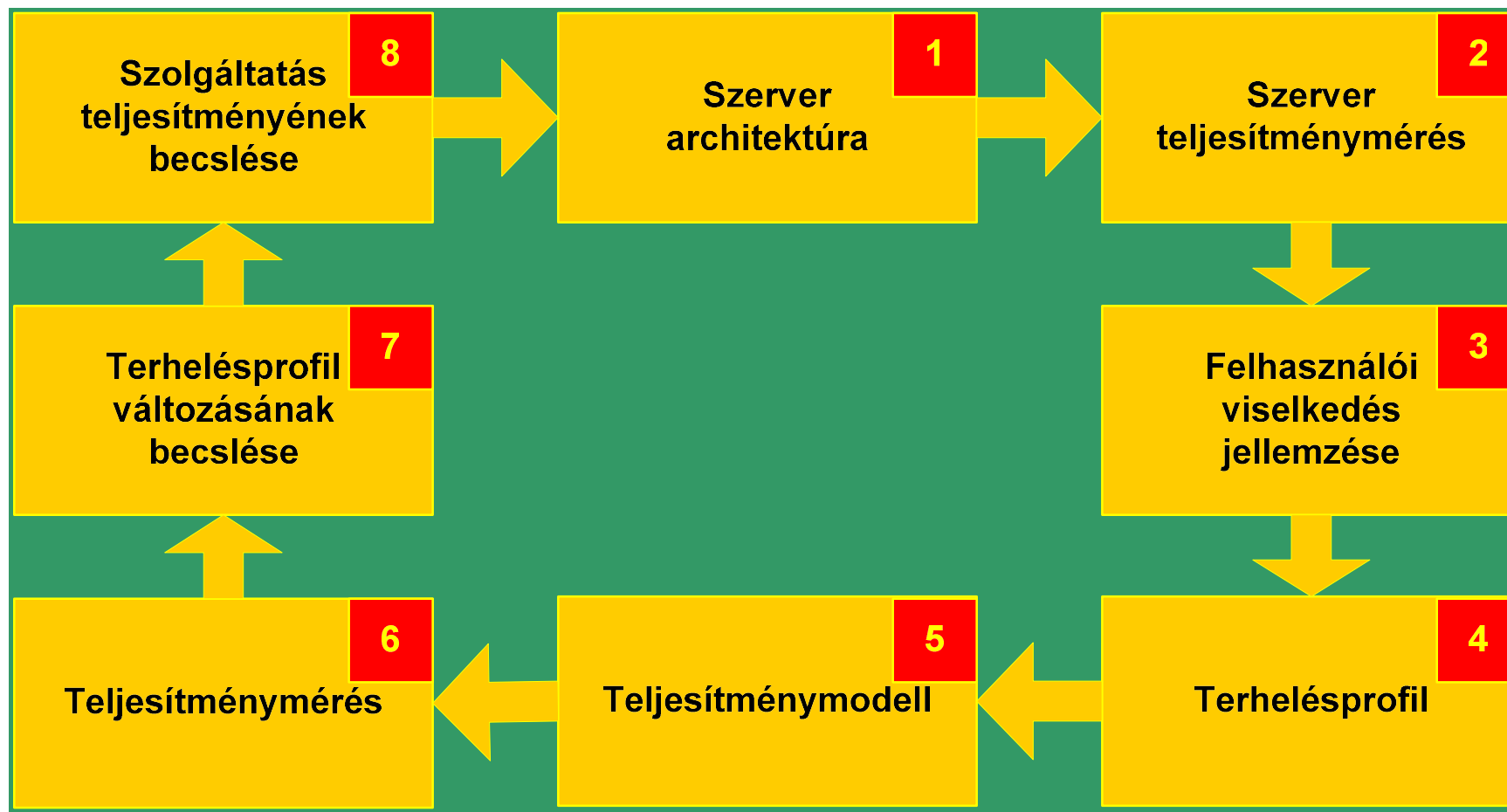
Motiváció



Kapacitástervezés

- „Annak **becslése**, hogy a rendszer mikor **telítődik** a terhelés hatására”
- „A leginkább **költséghatékony** módszer megtalálása, mellyel a rendszer **túlerhelése** a lehető legjobban **késleltethető**”
- Figyelembe veszi a nyújtani kívánt szolgáltatás szintjét

A mennyiségi analízis lépései



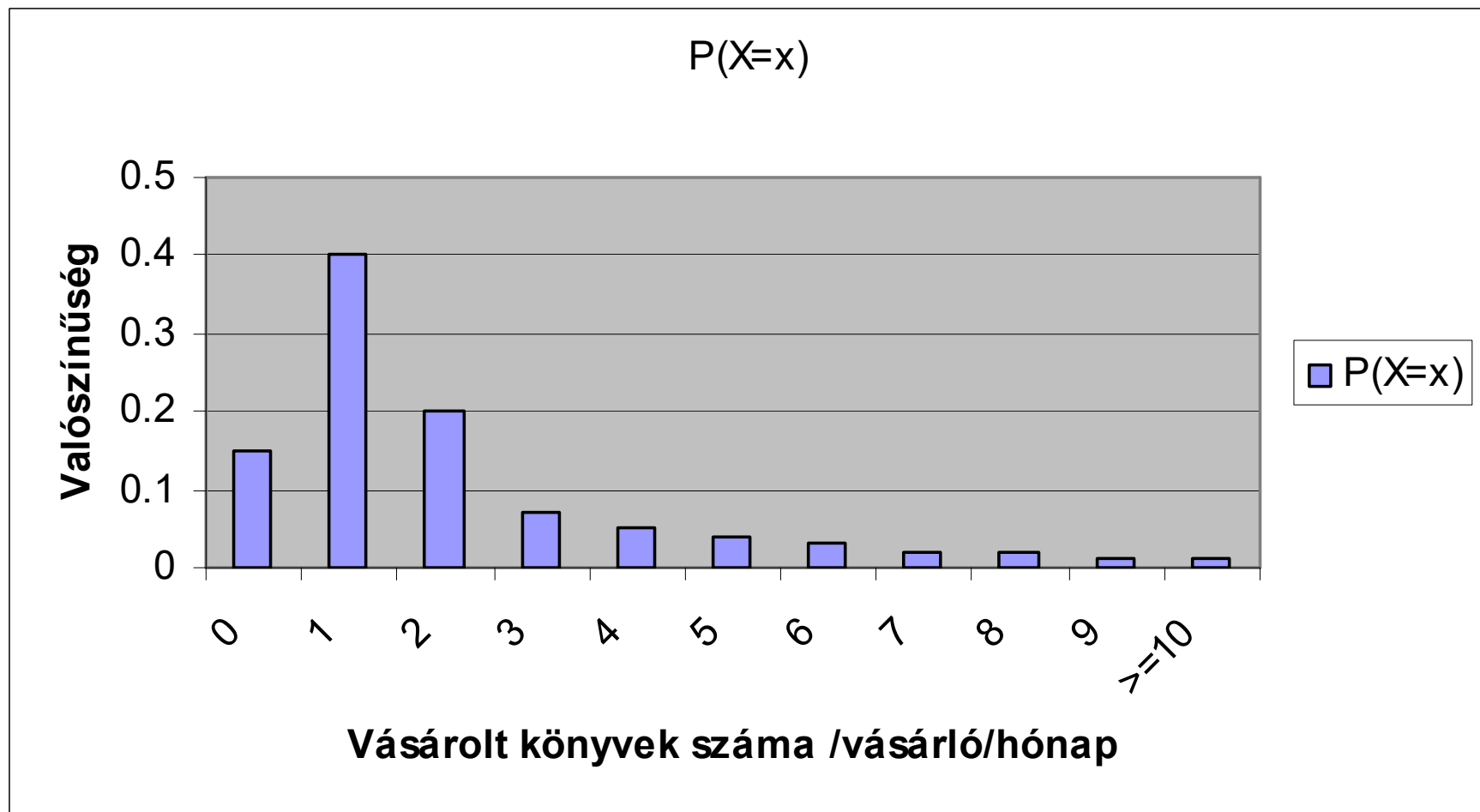
Mitől függ az elvárt kapacitás?

- Service Level Agreement (SLA):
 - A vezetés által meghatározott mérőszámok (válaszidő, elérhetőség, átbecsátóképeség)
- Használt technológiák, szabványok
- Pénzügyi lehetőségek
- Kapacitástervezés:
 - **kvantitatív** szemlélet
 - cél: ne kelljen túl sűrűn változtatni a rendszeren

Mitől nőhet a terhelés?

- A rendszer változatlan, nő az átlagos terhelés
 - pl. eddigi 10000 helyett 15000 látogató
- Új alkalmazások/szolgáltatások
 - pl. könyvesbolt: szemelvények a könyvekből
- Változik a felhasználók viselkedése
 - hirtelen változás az informatikai rendszeren kívül (pl. hirdetési kampány, 9.11.)
 - navigációs minták változnak (többen keresnek)

Online könyvesbolt vásárlási mintája



Felhasználói viselkedés

- CBMG megalkotása
- Statikus: oldalak szerkezet alapján
- HTTP logok alapján
- Hasonló viselkedésű felhasználók csoportosítása
- Hogyan változik a modell az új funkciók bevezetése után?
 - Pl. a Home oldalról egyenesen mehet a Vásárláshoz?

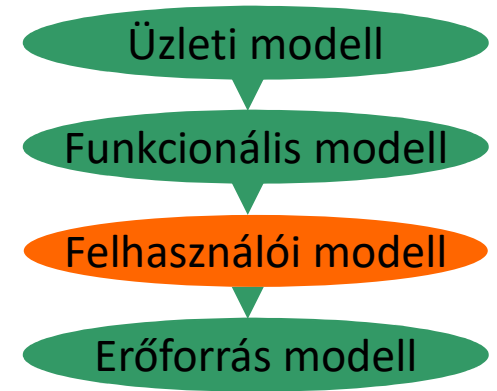
Felhasználói, terhelés és erőforrás modellek

■ Felhasználói modell

- navigációs minták: felhasználhatók a későbbi terhelés előrejelzéséhez (mi történik, ha hirtelen több felhasználó jelenik meg, többet keresnek, stb.)
- eszköz: **Customer Behavior Model Graph** (CBMG) ,
Customer Behavior Model Statechart (CBMS)
- terhelési paraméterek meghatározása: ha csak a jelenlegi terhelési modell felírása a cél, elég egy kevésbé részletes felírás
- eszköz: **Customer Visit Model** (CVM), kevésbé részletes, nem használható előrejelzésre
- mindig egy session-t vizsgálunk (azonos felhasználótól egy látogatás alatt érkezett kérések sorozata)

Felhasználói modell – CBMG

- Irányított gráf az oldalak közti lehetséges átmenetek és valószínűségeik ábrázolására
- n csúcs, ahol
 - 1. csúcs: **Entry** állapot, absztrakt belépési pont, minden felhasználó innen indul ide nem tér vissza
 - n . csúcs: **Exit** állapot, absztrakt kilépési pont, a folyamat vége (nem mindig ábrázoljuk)
 - a többi csúcs megfelel a felhasználó által elérhető szolgáltatásoknak (oldalaknak)
- élek: az oldalak szerkezete határozza meg a csúcsok közt lehetséges átmeneteket



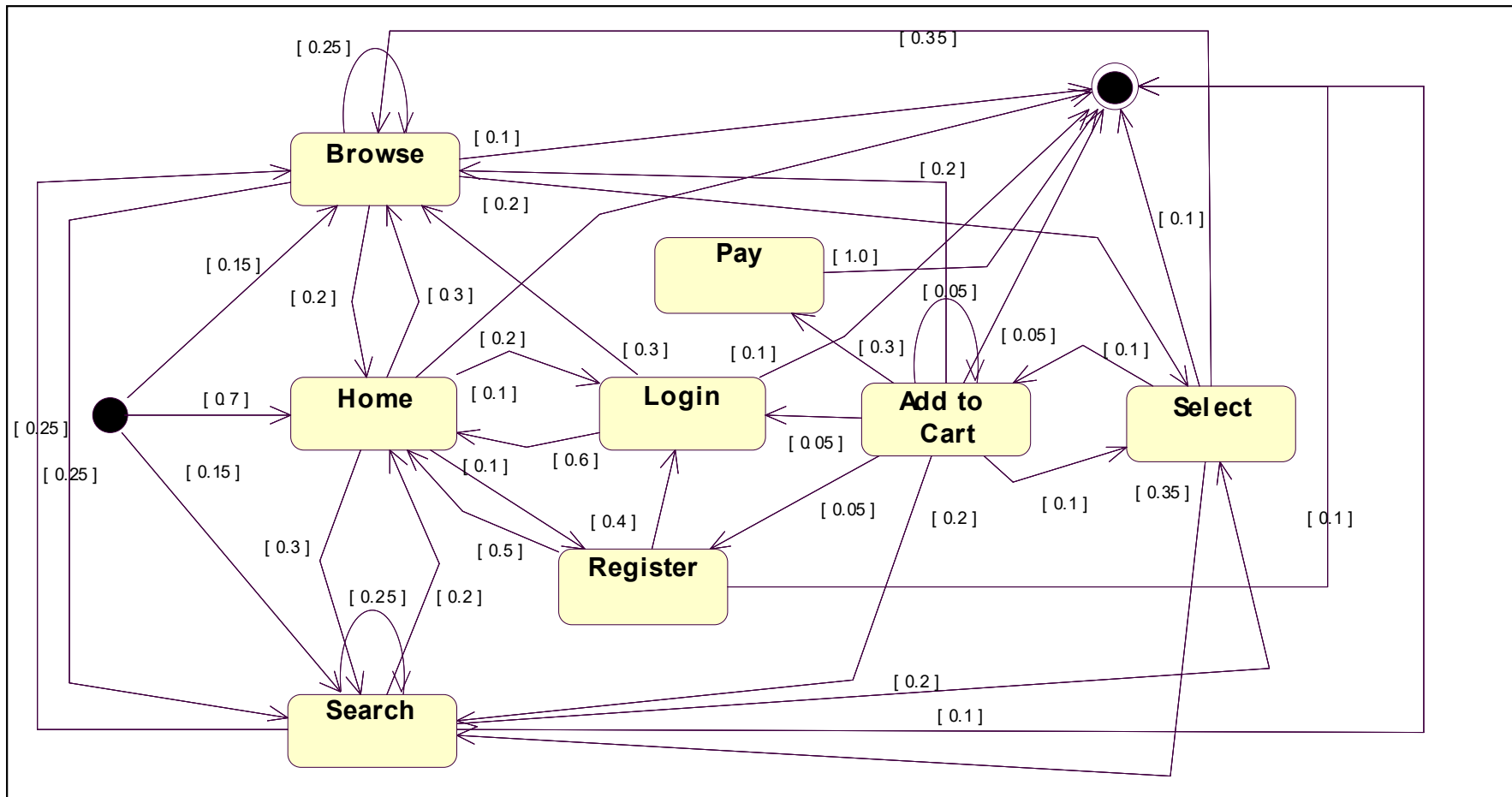
CBMG meghatározása

- Statikus CBMG: oldalak és köztük lévő lehetséges átmenetek
 - az oldalak által nyújtott szolgáltatások meghatározása
 - (pl. Login, Register, Add Item, Remove Item, Pay, Get Quotes, Download, Subscribe, stb.)
 - a szolgáltatások halmazának finomítása – az infrastruktúrát különböző mértékben terhelő szolgáltatásokra
 - (pl. Download szétválasztása Download Audio, Download Video szolgáltatásokra)
 - lehetséges átmenetek meghatározása – az oldalak megjelenítésének vizsgálatával
 - (linkek, formok, menüpontok, stb.)

Felhasználói modell: CBMS

- Customer Behavior Model Statechart
- UML állapotdiagram
- Hasonló információt hordoz, mint CBMG, de mindezt szabványos módon jeleníti meg
- Állapotok: az oldal funkciói (állapotai)
- Lehetséges átmenetek: rendszer modellje alapján
- Átmenetekhez valószínűség tartozik (dinamikus viselkedés)

Felhasználói viselkedés állapotdiagram modellje



Felhasználói viselkedés vizsgálata

- Mérőszámok (felhasználói modell és egyéb statisztikák alapján):
 - *Revenue Throughput*: az oldal által elért átlagos bevétel
 - *Potential Loss Throughput*: mennyibe kerül a szolgáltatás kiesése egy adott időszakban
 - *Visit Ratio*: átlagosan hányszor vesznek igénybe egy adott szolgáltatást (egy session alatt)
 - *Buy to Visit Ratio*: átlagosan hányszor vásárolnak egy session alatt (tényleges eladási tanzakció)
 - *Average Session Length*: egy session átlagosan hány szolgáltatást vesz igénybe (nem időt mér)
 - mindezen mérőszámok változása, ha a bemenő paraméterek megváltoznak (pl. egyes átmenetek valószínűségei)

Felhasználói viselkedés vizsgálata

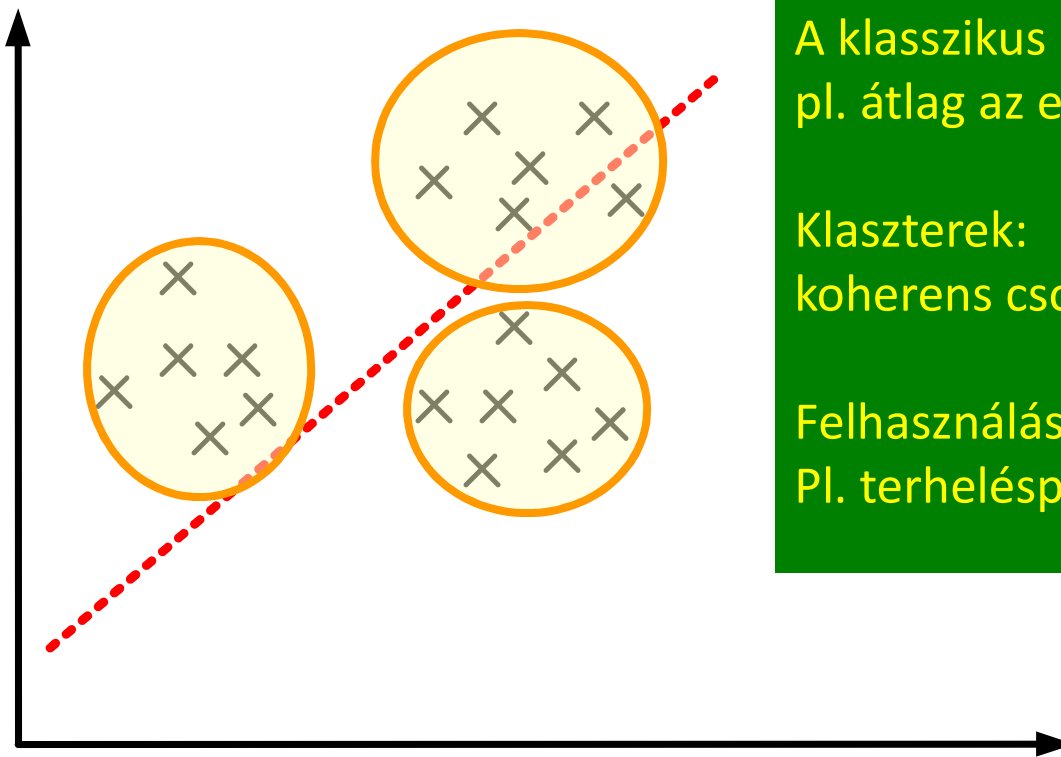
- Az előbbi példából kinyerhető adatok :
 - *Buy to Visit Ratio (BV)*: **Pay** állapot előfordulásának várható értéke, $\sum_k [\prod P(i,j)]$ az összes lehetséges **Entry-Pay** útra (k darab), *BV* értéke itt 0.058 (5.8 %)
 - *Végrehajtott eladási tranzakciók száma*: ha naponta 100000 látogató (session) van, akkor átlagosan $100000 * 0.058 = 5800$
 - *Átlagos session hossz* (average session length): az állapotok átlagos előfordulásainak összege, itt. 7.998 (**Exit** és **Entry** állapotok előfordulási gyakorisága mindig 1, itt nem számítanak)
 - mindezen adatok kiszámolhatóak más átmeneti valószínűségek esetén is (pl. ha megnő a keresés után vásárlók száma)

Customer Visit Model (CVM)

- CVM: különböző típusú felhasználók viselkedését jellemző vektorok (melyik állapotban hányszor járt)
- A felhasználók csoportosítása klaszterezési technikákkal történhet
- Nem jelzi az egyes állapotok közti átmenetek gyakoriságát, kevésbé alkalmazható előrejelzésre

	Session1	Session2	Session3
Home	1	2	3
Browse	4	8	4
Search	5	5	3
Login	0	1	1
Pay	0	0	1
Register	0	0	1
Add to Cart	0	2	1
Select	3	3	2

Klaszter technikák



A klasszikus statisztikai jellemzők,
pl. átlag az egész mintát jellemzik.

Klaszterek:
koherens csoportok

Felhasználás:
Pl. terhelésprofilok meghatározása

Klaszter alapfogalmak

- Klaszter:
 - összetartozó, hasonló értékek csoportja
- Klaszter középpontja: centroid
 - ezzel helyettesíthető az adott csoport
- Különböző (n dimenziós) távolságfogalmak
 - nem csak euklideszi
- Különböző algoritmusok
- Algoritmus + távolság definíció
→ klaszter technika

Session azonosítás

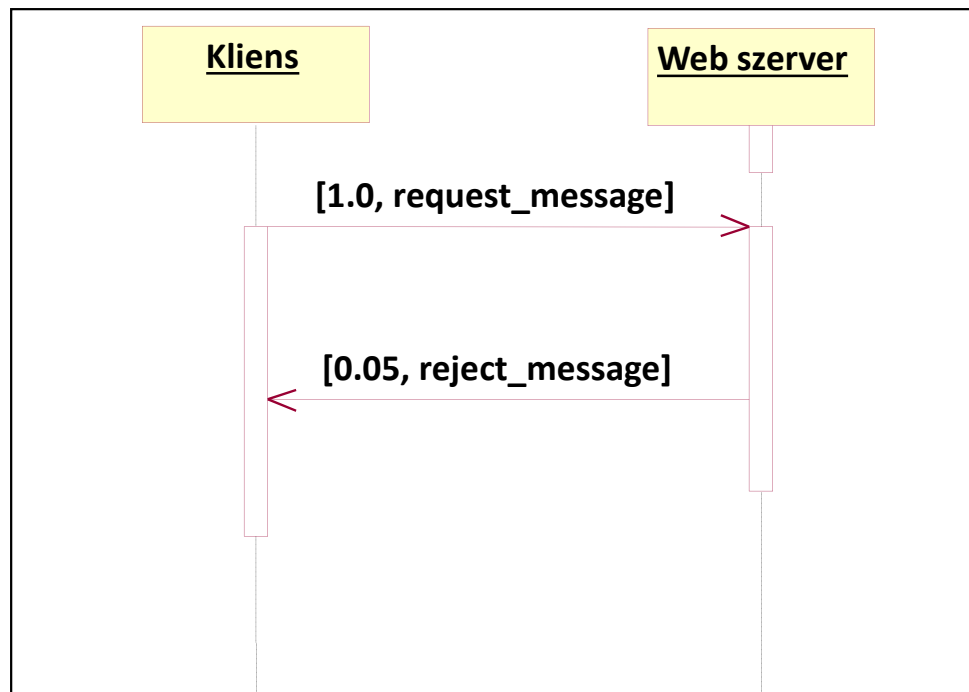
- Cookie-k használatával
 - szerver generál egy ID-t, amit a kliens minden kéréssel elküld, az kliens mellett az alkalmazás állapotát is tárolhatja (pl. bevásárlókocsi)
- Autentikációs mechanizmusok, rejtett mezők HTML oldalak formjaiban, dinamikus URL-ek, stb.
- Szerver logok alapján, várakozási küszöbérték (threshold) használatával
 - ha ennél hosszabb a szünet két kérés (request) elküldése közt, akkor két külön session

C/S Interaction Sequence Diagram (CSISD)

- Szekvenciadiagramok minden lehetséges esetre
- Objektumok: kliens és a különböző szerverek (erőforrások)
- Nyilak: üzenetküldés
- Üzenetek: $[p, m]$ párok, ahol p az üzenet küldésének valószínűsége, m a mérete (byte)
 - pl. $[0.05, \text{reject_message}]$: 5% valószínűséggel reject_message méretű üzenetet küldünk
- Minden lehetséges végrehajtási szekvencia kliensből indul ki és kliensben végződik
- CSID: C/S Interaction Diagram
 - a szekvenciák összessége

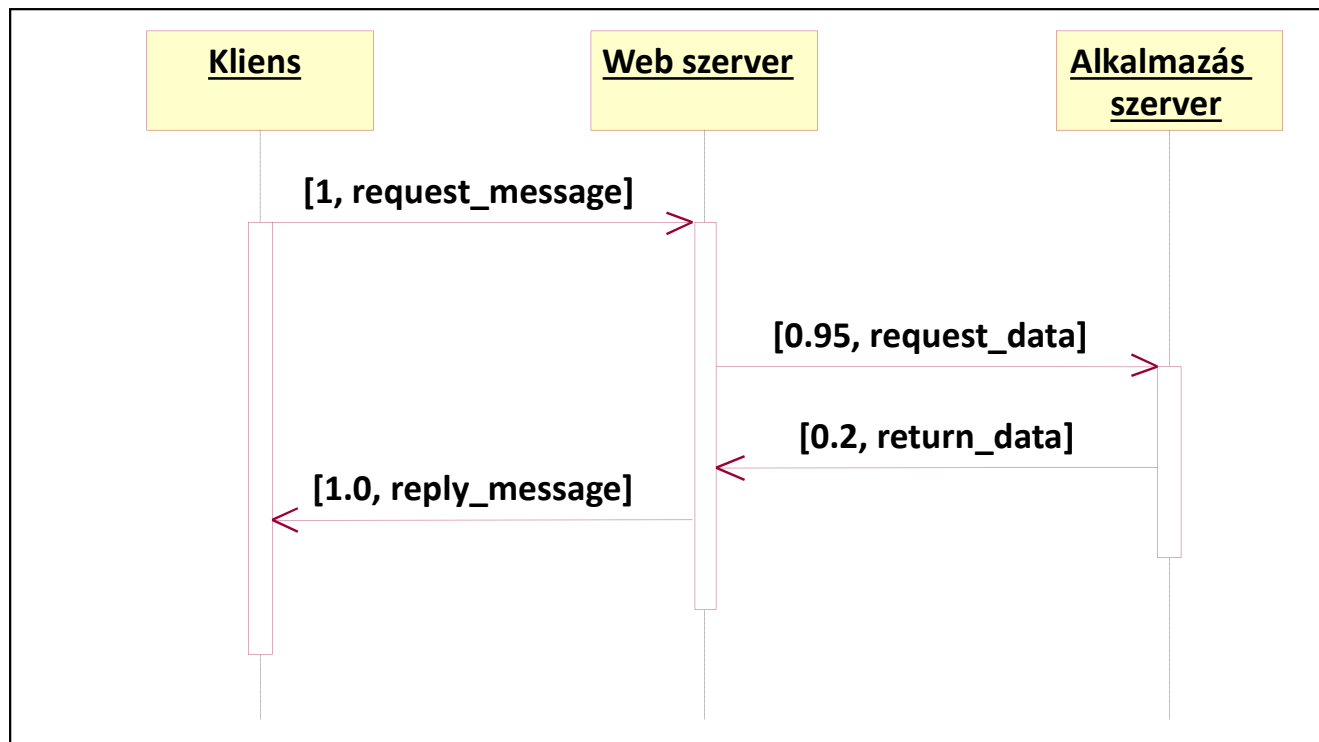
CSISD példa 1.

- Kliens kér valamilyen adatot a Web szervertől, 3 lefutási lehetőség
- 1. eset: A Web szerver túlterhelt: elutasítja a kérést



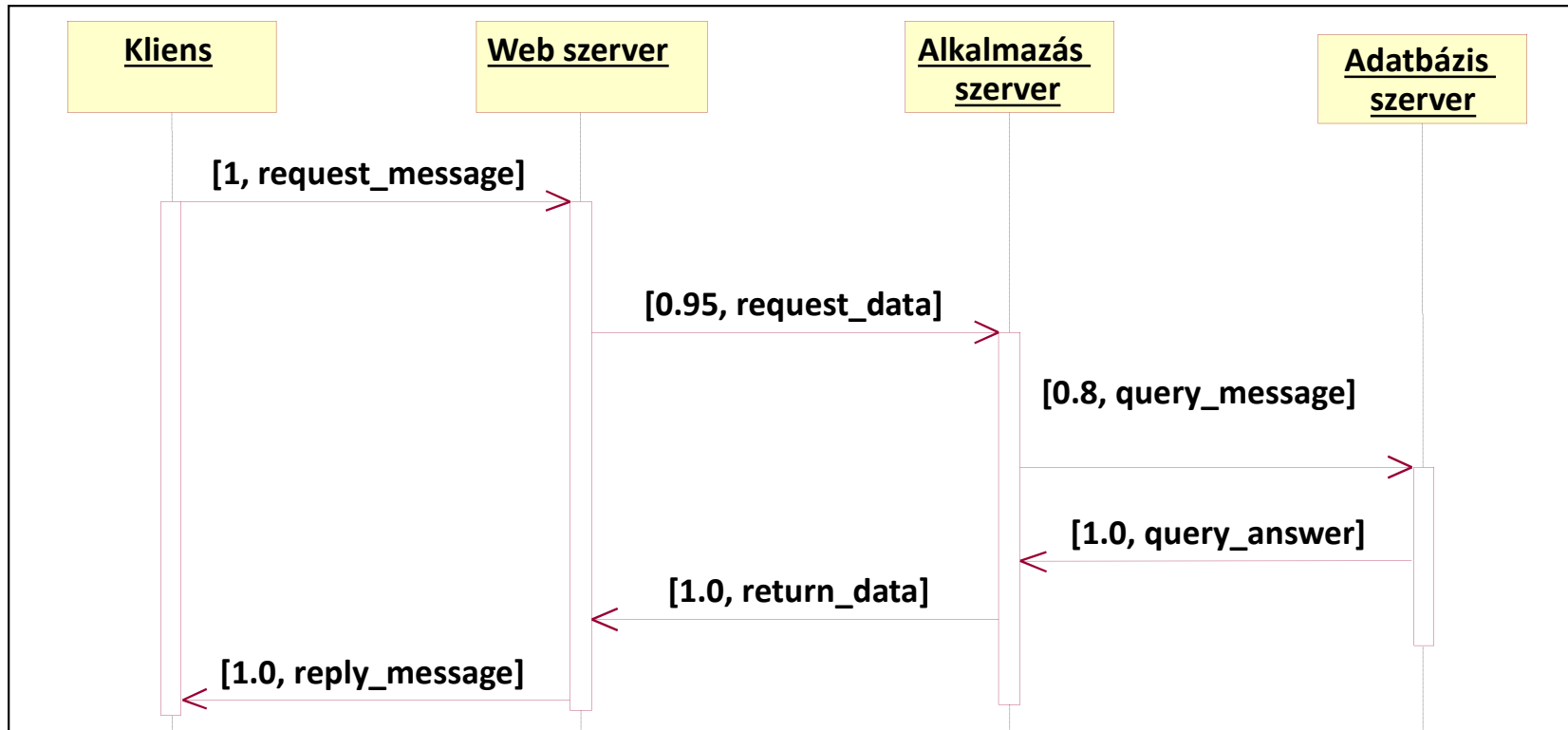
CSID példa 2.

- 2. eset: A Web szerver továbbküldi a kérést valamilyen alkalmazásnak (alk. szerver), az alkalmazás cache-ből visszaadja az adatot



CSID példa 3.

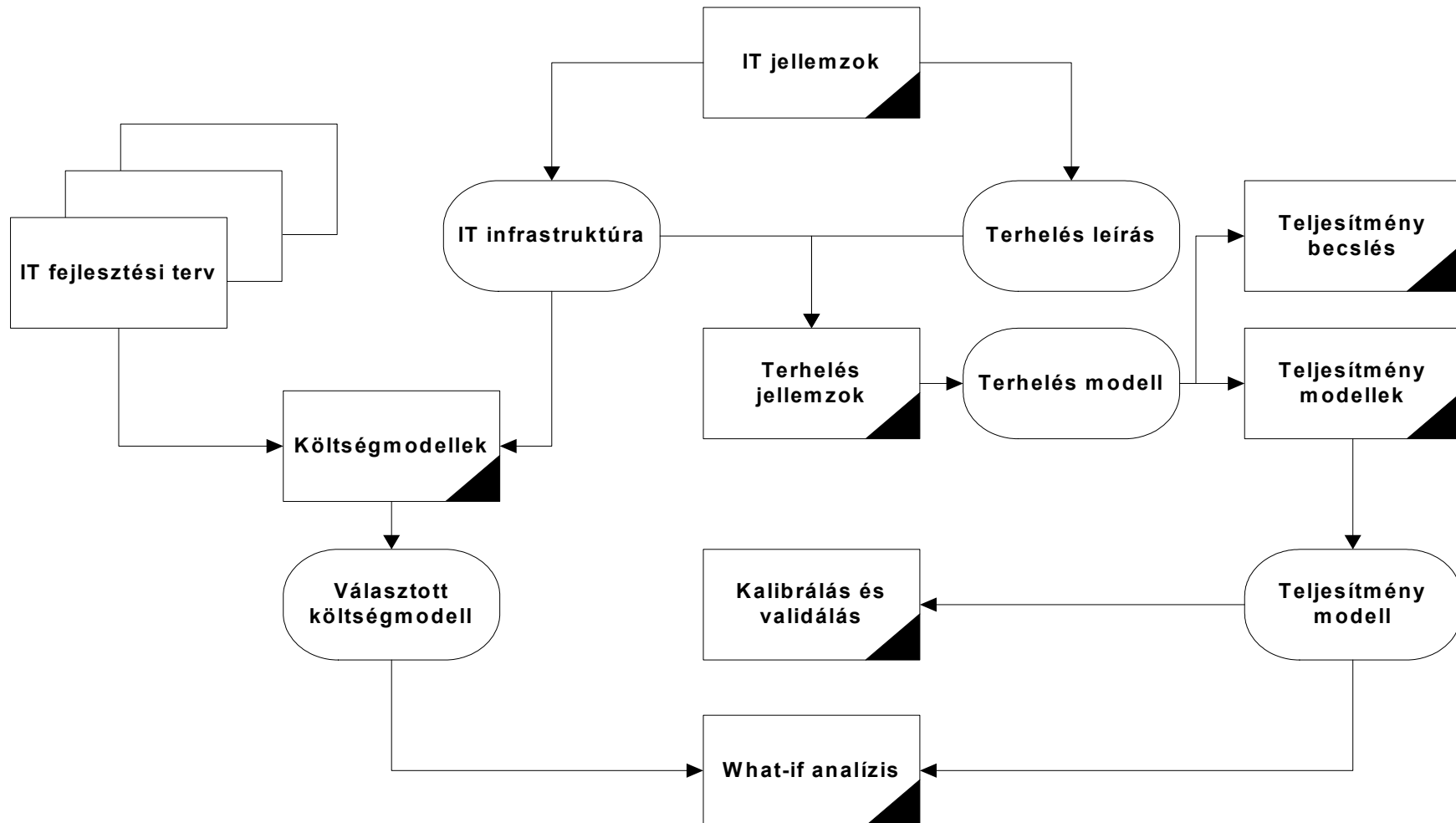
- 3. eset: A Web szerver továbbküldi a kérést, az alkalmazás lekérdezést hajt végre az adatbázisszerveren, majd visszaadja az adatot



CSISD elemzése

- mennyi a lokális hálózat forgalma, ha minden szerver ugyanazon a LAN-on van? $\Sigma (p^*m)$ minden szerverek közti üzenetre
- Mi történik, ha a Web szervert leválasztjuk?
 - (Web szerver:LAN1, többi LAN2)
 - (Szerver oldali) hálózati késleltetés meghatározása sávszélesség és üzenetméret alapján
- Milyen lesz a teljesítmény, ha minden szerver ugyanazon a gépen fut?

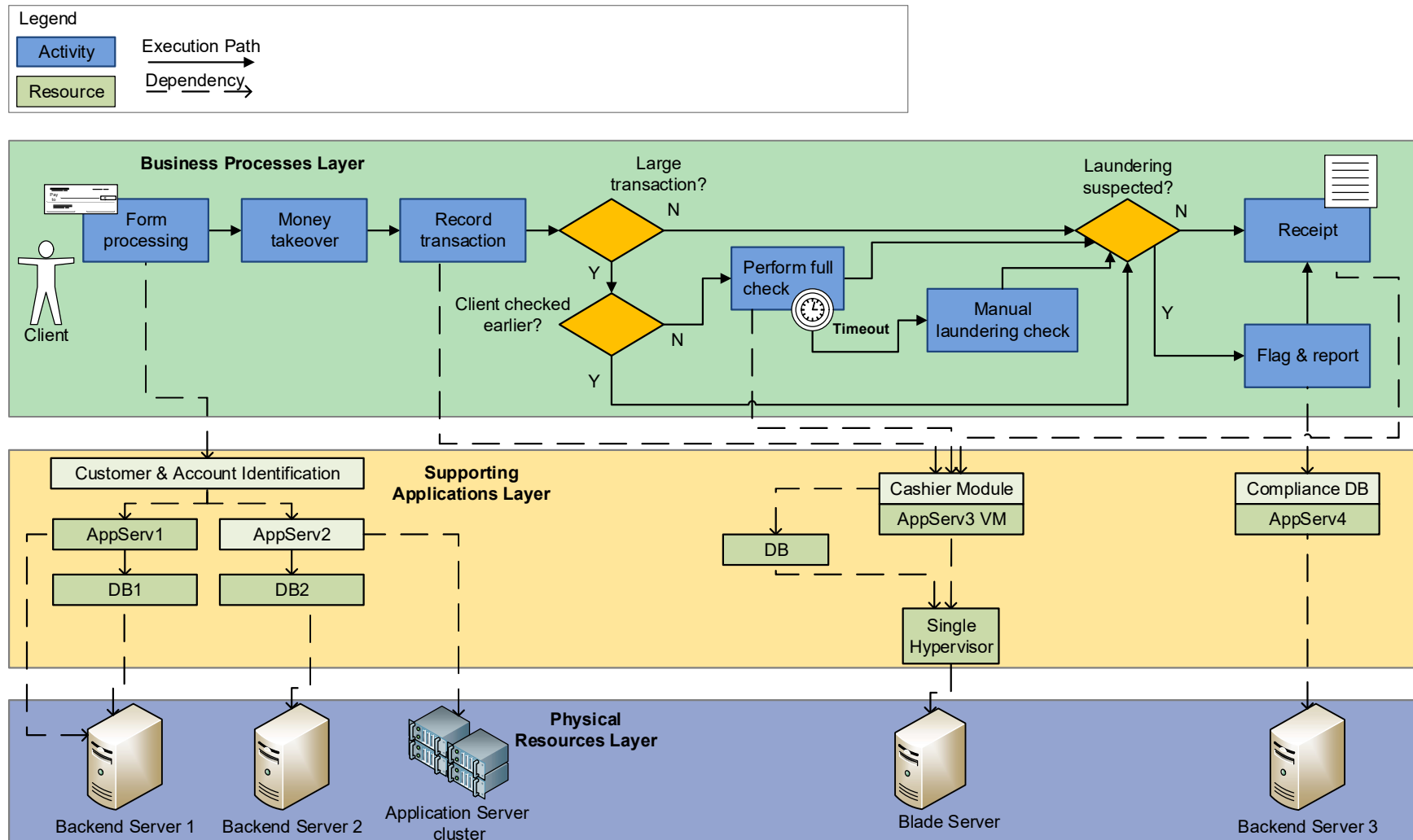
Erőforrás szintű kapacitástervezés lépései



Erőforrás szint

- Felhasznált IT erőforrások konkrét meghatározása az előző modellek alapján
- Informatikai környezet meghatározása
 - infrastruktúra és
 - terhelés leírása
 - felhasználói funkciókhoz tartozó programok meghatározása
- Infrastruktúra:
 - hardver (szerver gépek, diszk farmok, routerek, tűzfalak...)
 - szerverek (Web ~, alkalmazás ~, adatbázis ~, DNS...)
 - szoftverek (OS, middleware, adatbáziskezelő...)
 - Hálózati kapcsolat, hálózati protokoll
 - Fizetési szolgáltatás (Payment service)

Példa



Erőforrás szintű terhelés meghatározása

R: adott erőforrás

F: funkciók halmaza

$$E[\text{terhelés}_R] = \sum_{|F|} (\text{gyakoriság}_F * E[\text{terhelés}_{F,R}])$$

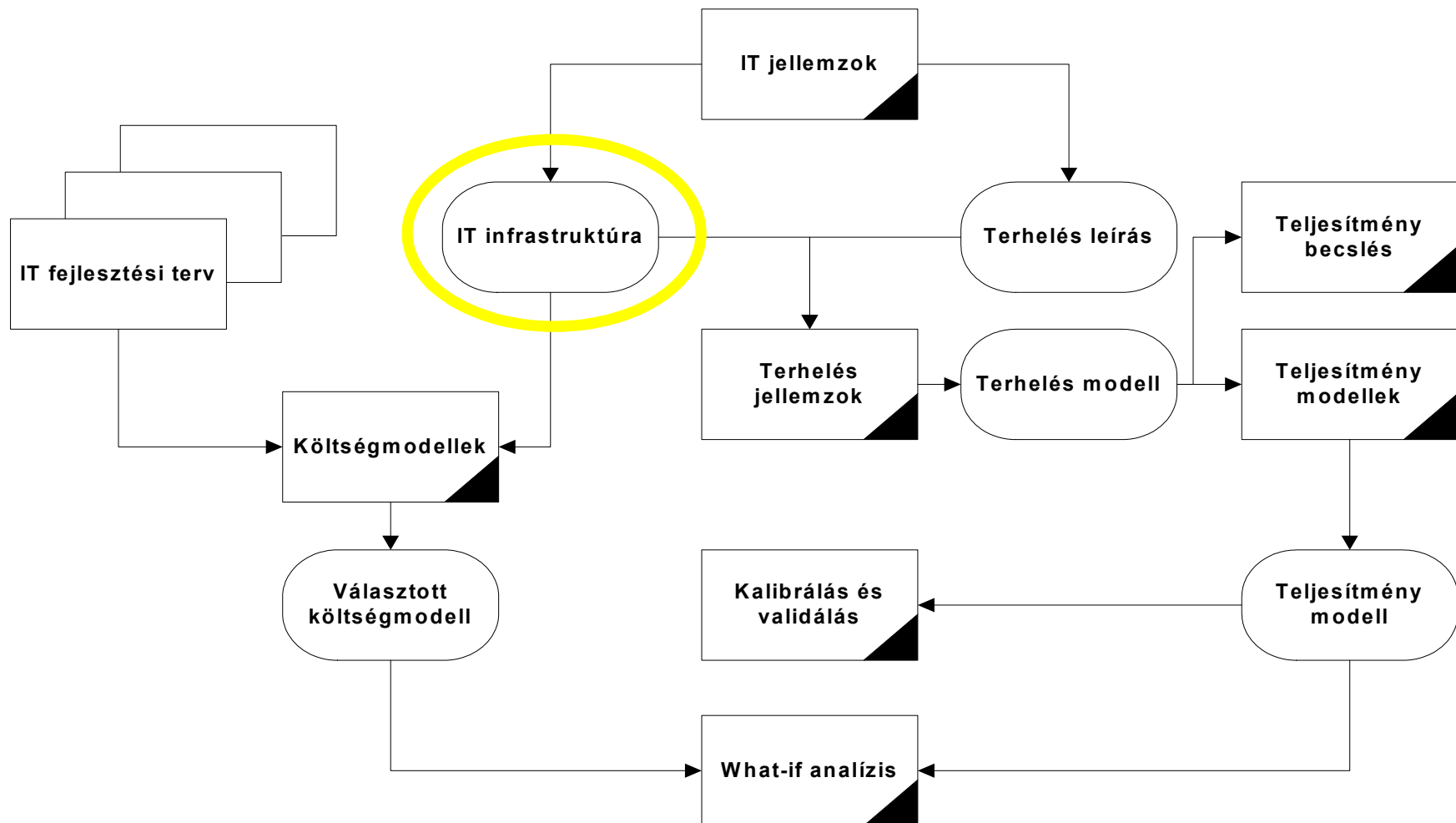
**felhasználói
(nem inf.) mérték**

... csal, mert

- nem additív a terhelés
 - Taszkváltás, cache frissítés, stb.
- lehetnek kiugró erőforrásigények

**informatikai
(műszaki) mérték**

Infrastruktúra leírása

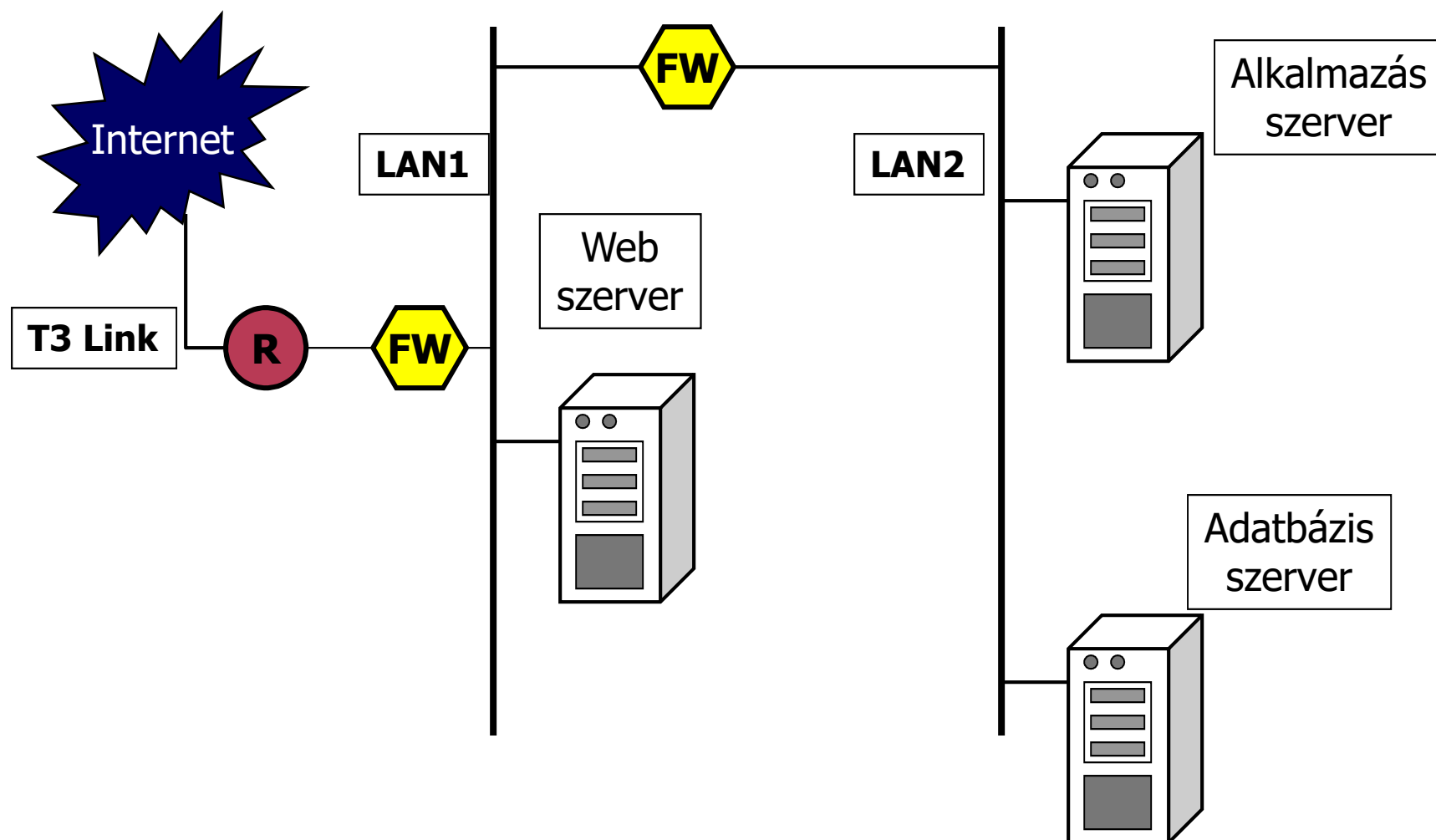


Informatikai környezet meghatározása

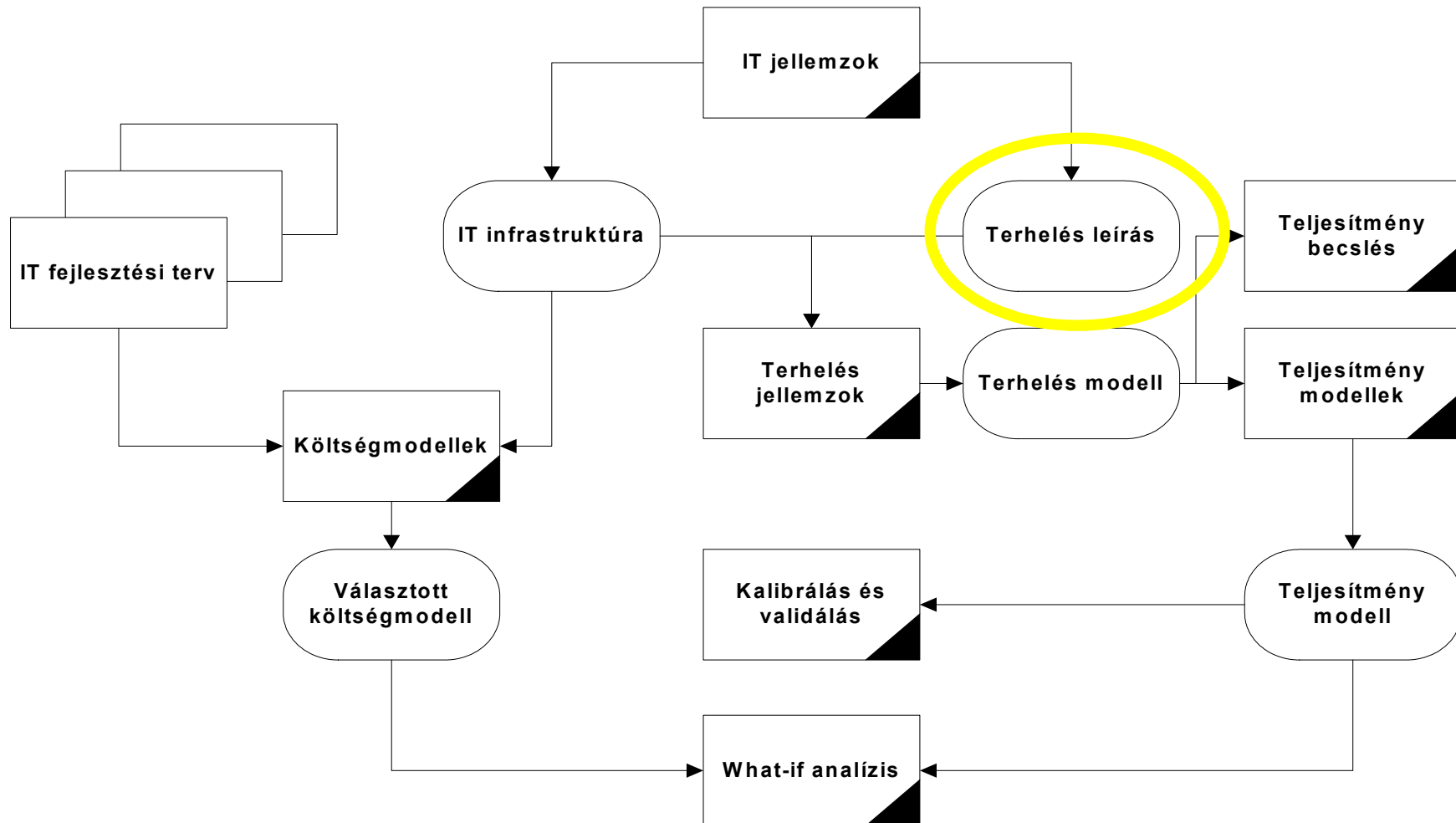
- Cél: meghatározni a funkciókhoz tartozó tranzakciókat (elemi lépéseket) és az ezek által használt erőforrásokat

E-Business szolgáltatás	Tranzakciók	Szerver
Könyv kiválasztás	ShowBooks	WS
	DisplayBooks	AS
	SearchBooks	DB
	LaunchShowBooks	WS
	DisplayBooksByAuthor	AS
	SearchBooksByAuthor	DB
	SendReply	WS
Option Selection	LaunchShowOptions	WS
	DisplayBookOptions	AS
	SearchBookOptions	DB
	SendReply	AS

Példa: a kereskedés informatikai infrastruktúrája



Terhelés leírása



Terhelés leírása

- Az előző lépésben meghatározott modellt veszi alapul
 - melyik szolgáltatás milyen erőforrásokat használ
- Alapvető komponensek: tranzakciók
- CSISD (szekvenciák) felírása, minden szerveroldali objektumhoz tranzakciónév
- Komponensekhez gyakoriság (érkezési ráta) és erőforrásigények meghatározása
- Cél: Szolgáltatások erőforrásigényeinek meghatározása

TERHELÉS MODELLEK, ELŐREJELZÉS

Valós (történelmi) terhelés példa (iwiw)



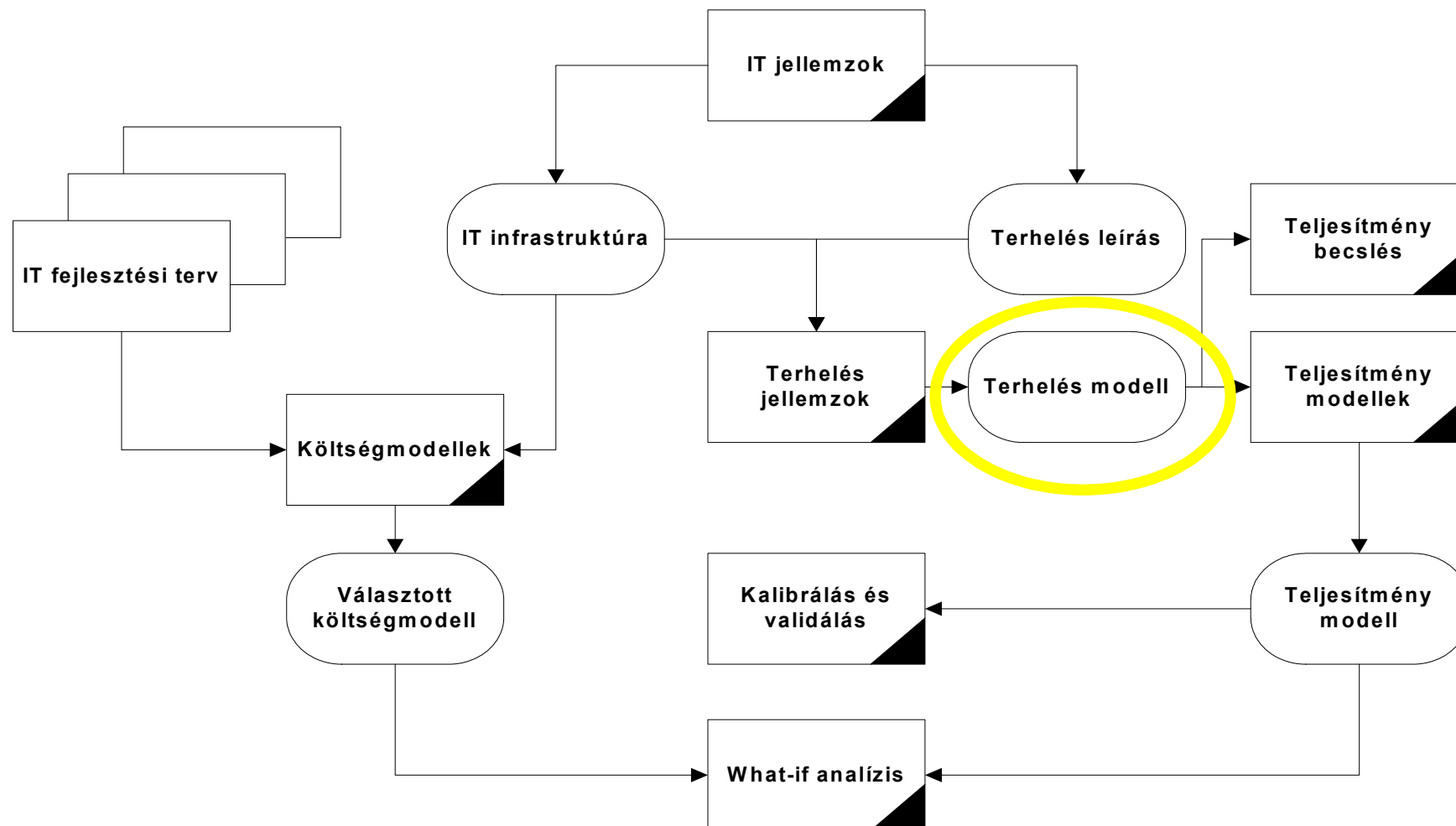
Szakaszokra becslőfüggvény
illeszthető

- Lineáris, exponenciális, logaritmikus
- Regresszió



Forrás: <http://www.sg.hu/cikkek/42924/>

Terhelés modell



Terhelés modell meghatározása

- Adatgyűjtés
 - benchmarkok (részletesen Benchmark ea.)
 - ökölszabályok
 - „best practices”
 - mérések
- Monitorozás
 - belső (szerver)
 - külső (kliens, hálózat)
- Adatok rendszerezése
 - Pl. klaszterezési technikák, feltáró adatelemzés
 - Változókiválasztás, faktoranalízis....

Terhelés modell minta

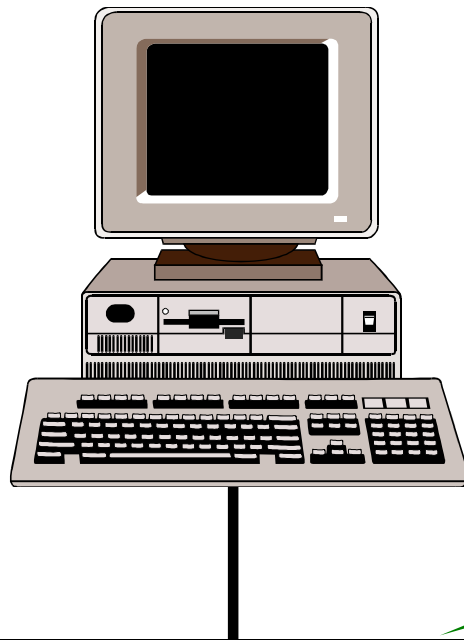
1 napi mérés adatai											
Érkezési ráta	20,000	kérés/óra	LAN 1	100	Mbps	LAN 2	100	Mbps	T3 Link	1.5	Mbps
Egy kérés szolgáltatás igénye											
E-business Funkció	Kérések száma	WS CPU (ms)	WS IO (ms)	AS CPU (ms)	AS IO (ms)	DB CPU (ms)	DB IO (ms)	LAN 1 (ms)	LAN 2 (ms)	T3 Link (ms)	Össz. szolg. igény
általános info	20,000	5.2	9.5	25.0	15.0	10.0	20.0	0.492	0.532	16.4	132.1
könyv keresés	18,900	4.8	8.5	18.0	14.0	13.0	40.0	0.328	0.352	12.0	91.0
könyv böngészés	14,120	4.9	8.2	13.0	12.0	13.0	40.0	0.287	0.328	12.0	83.7
bejelentkezés	8,020	5.1	8.4	12.0	10.0	13.0	20.0	0.295	0.492	11.5	80.8
regisztráció	892	32.0	15.0	16.0	30.0	15.0	20.0	0.655	0.000	32.8	126.4
bevásárlókosár	670	32.0	14.0	18.0	24.0	0.0	0.0	0.410	0.000	19.1	107.5
fizetés	584	31.0	15.0	35.0	90.0	30.0	80.0	0.819	0.901	43.7	326.4

Honnan lesznek adataink?

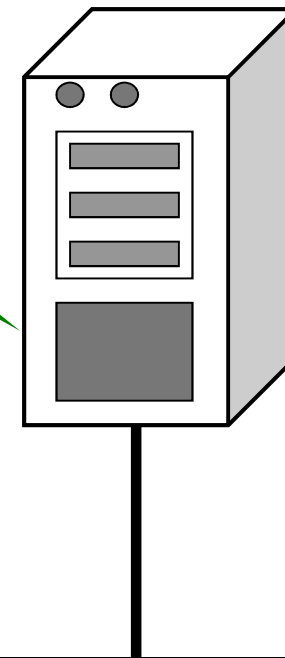
- Hol mérünk?
 - Felhasználó szemszögéből
 - Üzemeltető szintjén
 - Alkalmazástól függően
 - Mérés/monitorozás
 - Többszintű architektúra
 - Virtualizált környezetek
 - Mintavételezés
 - Mérés többletköltsége, mérési hibák
 - Nagyszámú változó
 - Topológia változása
 - Sok mért érték kezelése (ld. virtualizált példa)
- Valós környezetben nehéz feladat

Terhelés mérése ellenőrzött környezetben

**Kliens gépen futó
tesztelő szkript**



Dedikált szerver



**Teljesítmény
monitorozás**

**Teljesítmény
monitorozás**

DEDIKÁLT HÁLÓZAT

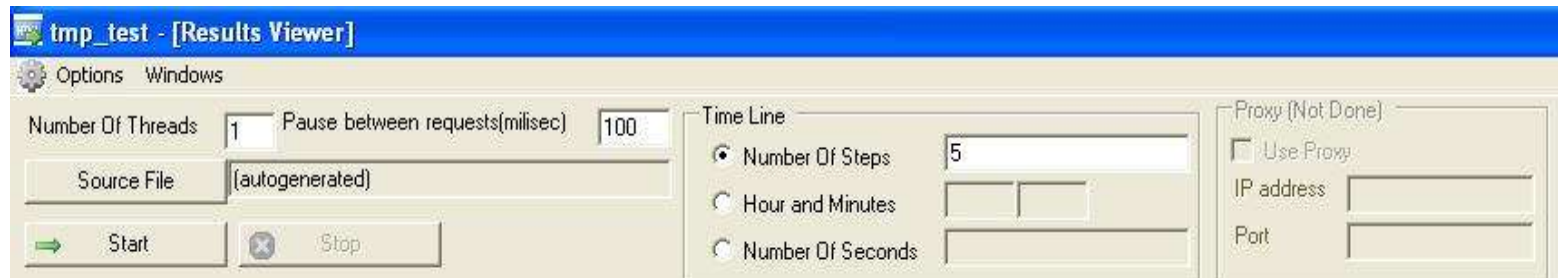
Terhelésgenerálás

- Felhasználói viselkedés rögzítése
 - Cookie kezelés
 - SSL kezelés
- Visszajátszás tipikus paraméterei
 - Párhuzamosan futó kliensek száma
 - Lekérdezések száma
 - „Gondolkodási idő”
 - „Felpörgési idő”
 - Timeout
 - Cache használata/tiltása
- Emulált kliensek
 - Web Performance Tools
 - ASP.NET – ANTS Profiler
 - Rational Performance Tester
 - Jmeter
 - Selenium
 - <http://www.softwareqatest.com/qatweb1.html>
 - <http://www.pyload.org/>
 - <http://www.opensourcetesting.org/performance.php>

Példa: terhelésgenerálás beállítása (WPT)

Clients	1
ThreadsPerClient	2
TotalPageRequests	1000
TimedRun	3600 (seconds)
RampUp	20 seconds
ThinkMinimum	2 seconds
ThinkMaximum	20 seconds
ErrorLog	error.log
UseRandomPage	
CookieCache	on
PurgeCookieCache	on
CookieLog	cookie.log
CookieIgnoreDefaultExpires	on
ReplaceHost	host0 localhost
RequestTimeout	30 seconds

Példa: terhelésgenerálás (FWPTT)



Tesztet beállítás

	RequestAddress	Response	Duration(sec)	Page Size
89	http://localhost/cs/themes/hawaii/style/DynamicStyle.aspx	200	0	6576
90	http://localhost/cs/forums/	200	0,031	15576
91	http://localhost/cs/themes/hawaii/style/DynamicStyle.aspx	200	0	6576
92	http://localhost/cs/	200	0,046	17144
93	http://localhost/cs/themes/hawaii/style/DynamicStyle.aspx	200	0	6576
94	http://localhost/cs/logout.aspx	200	0	6341
95	http://localhost/cs/themes/hawaii/style/DynamicStyle.aspx	200	0	6576

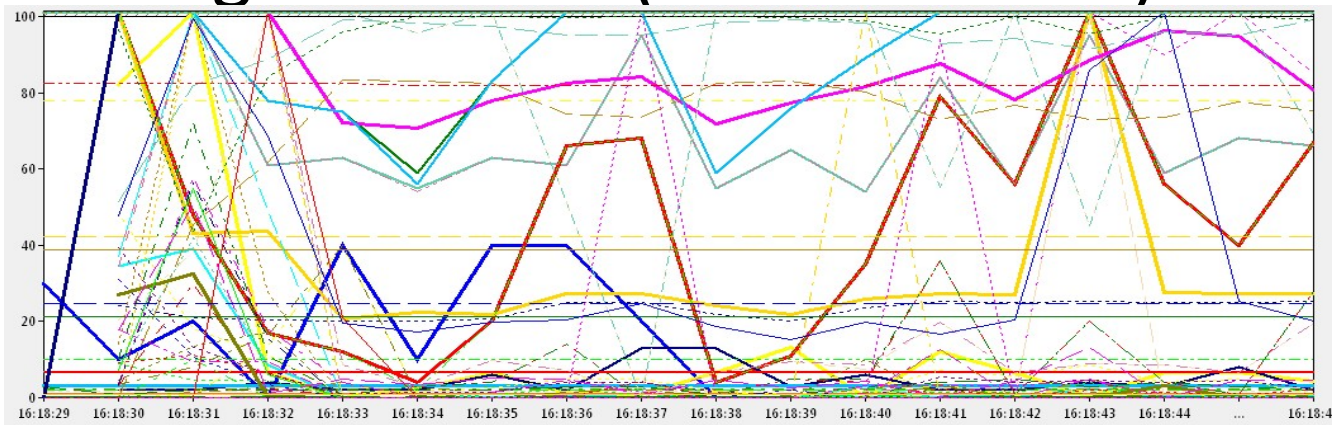
Terhelésgenerálás

Példa: Terhelés mérés eredmény- WPT

[Fri May 13 10:12:12 2005]	Uptime: 0 hours 1 minutes 2 seconds
[Fri May 13 10:12:12 2005]	Number of Clients: 1
[Fri May 13 10:12:12 2005]	Pages Attempted: 1000
[Fri May 13 10:12:12 2005]	Pages To Be Attempted: 1000
[Fri May 13 10:12:12 2005]	Pages per second: 16.13
[Fri May 13 10:12:12 2005]	Requests completed: 5000
[Fri May 13 10:12:12 2005]	Requests per second: 80.65
[Fri May 13 10:12:12 2005]	Failed Connections: 0
[Fri May 13 10:12:12 2005]	Incorrect response codes: 0
[Fri May 13 10:12:12 2005]	Content verification failed: 0
[Fri May 13 10:12:12 2005]	Request write failures: 0
[Fri May 13 10:12:12 2005]	Number of early server closes: 0
[Fri May 13 10:12:12 2005]	Number of timeouts: 0
[Fri May 13 10:12:12 2005]	SSL handshake failures: 0

Windows teljesítményszámlálók

- Átlagszámlálók (Average counters)
- Különbségszámlálók (Difference counters)
- Pillanatnyi érték számlálók (Instantaneous counters)
- Százalékszámlálók (Percentage counters)
- Gyakoriság számlálók (Rate counters)



Kapui Ákos: Webalkalmazások teljesítményvizsgálata és skálázhatósága, 2008.

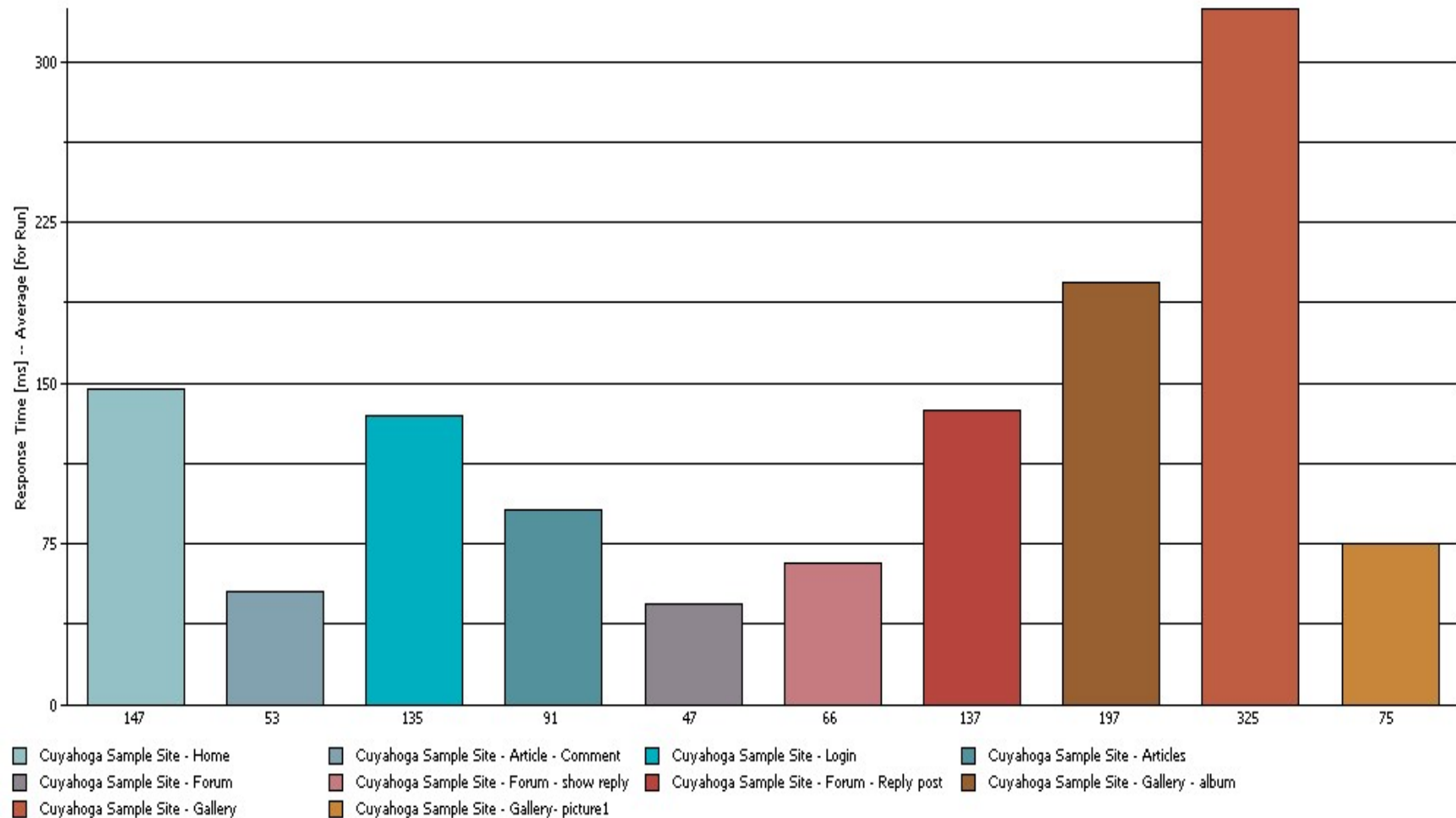
Konkrét teljesítményszámlálók

- Processor(_Total)\% Idle Time
 - A processzor szabad erőforrásának mértéke
- Processor(_Total)\% Processor Time
 - A processzor aktuális kihasználtsága
- ASP,NET v2,0,50727\Request Execution Time
 - Web kérések végrehajtási ideje
- SQLServer:Databases(_Total)\Transactions/sec
 - Adatbázis tranzakciók másodpercenként
- SQLServer:General Statistics\Logins/sec
 - Adatbázis bejelentkezések száma
- Memory(_Total)\Available Memory
 - Rendelkezésre álló memória
- ASP,NET\Requests Current
 - Aktuális kérések száma

Példa: Rational Performance Tester

Mérési eredmények

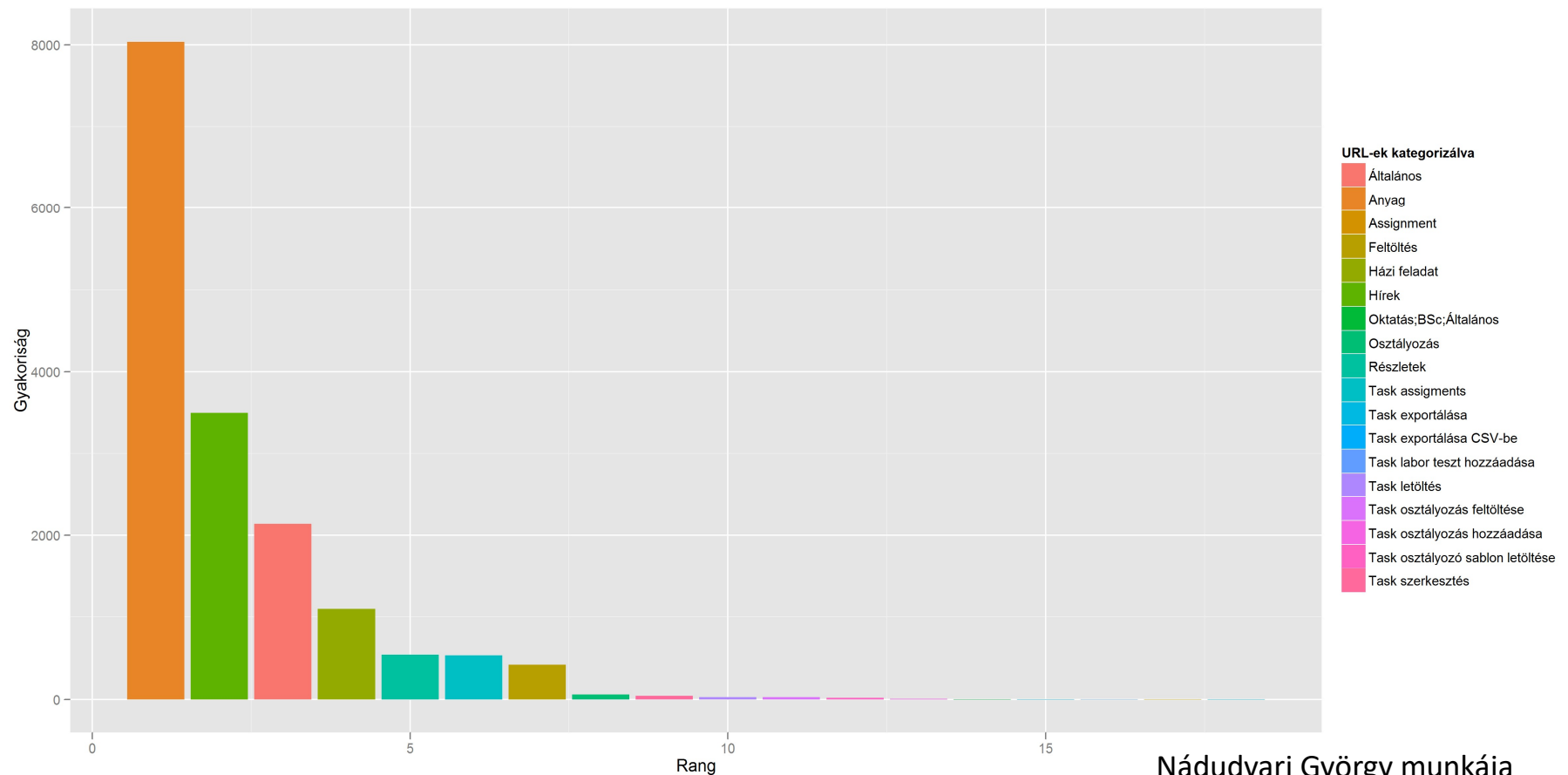
Average Page Response Time for Run (Filter applied: Count Filter: 10 highest)



Kapui Ákos: Webalkalmazások teljesítményvizsgálata és skálázhatósága, 2008.

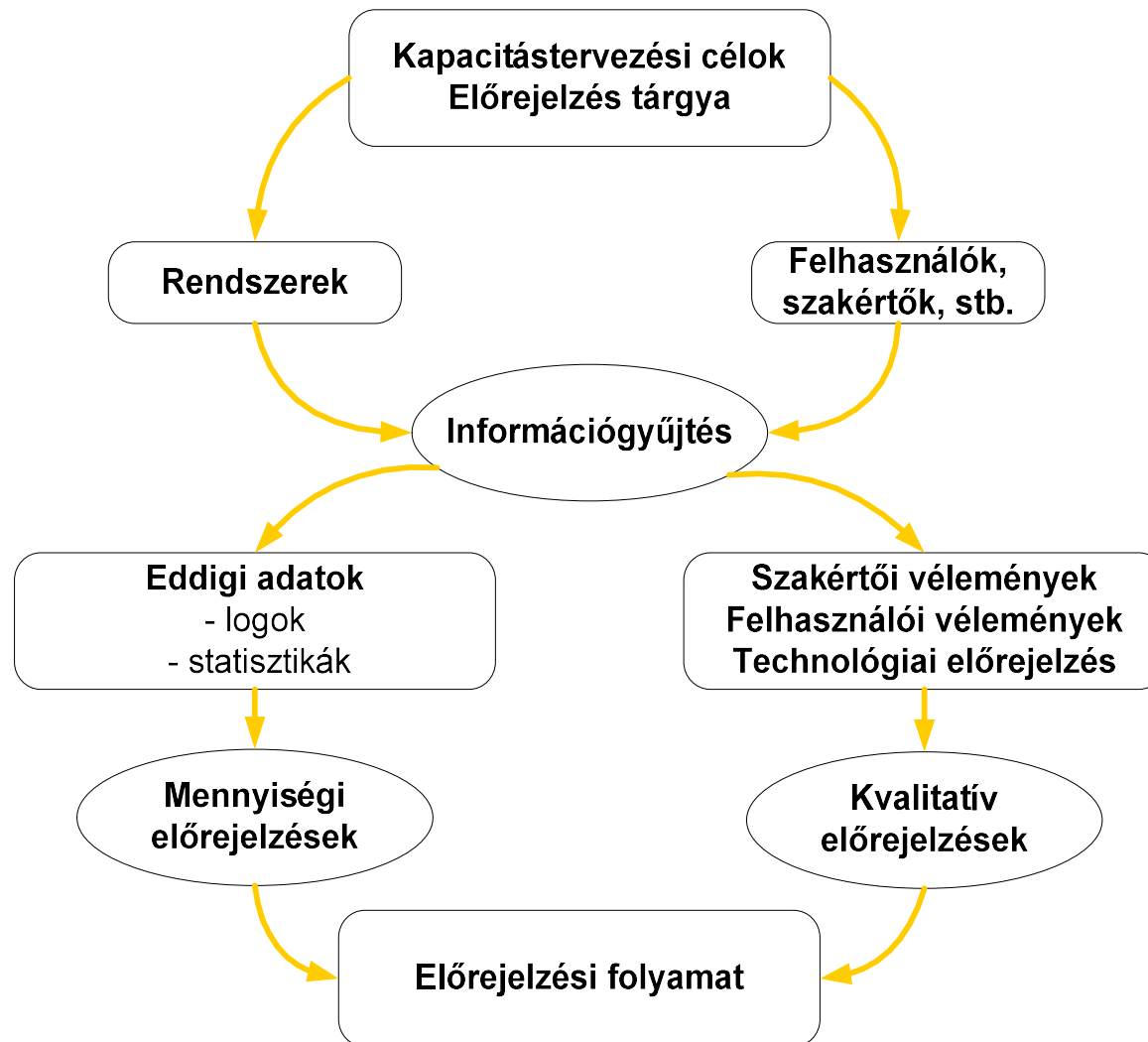
Példa: Tanszéki honlap

- A Rendszermodellezés tárgy oldalainak látogatottsága



Nádudvari György munkája

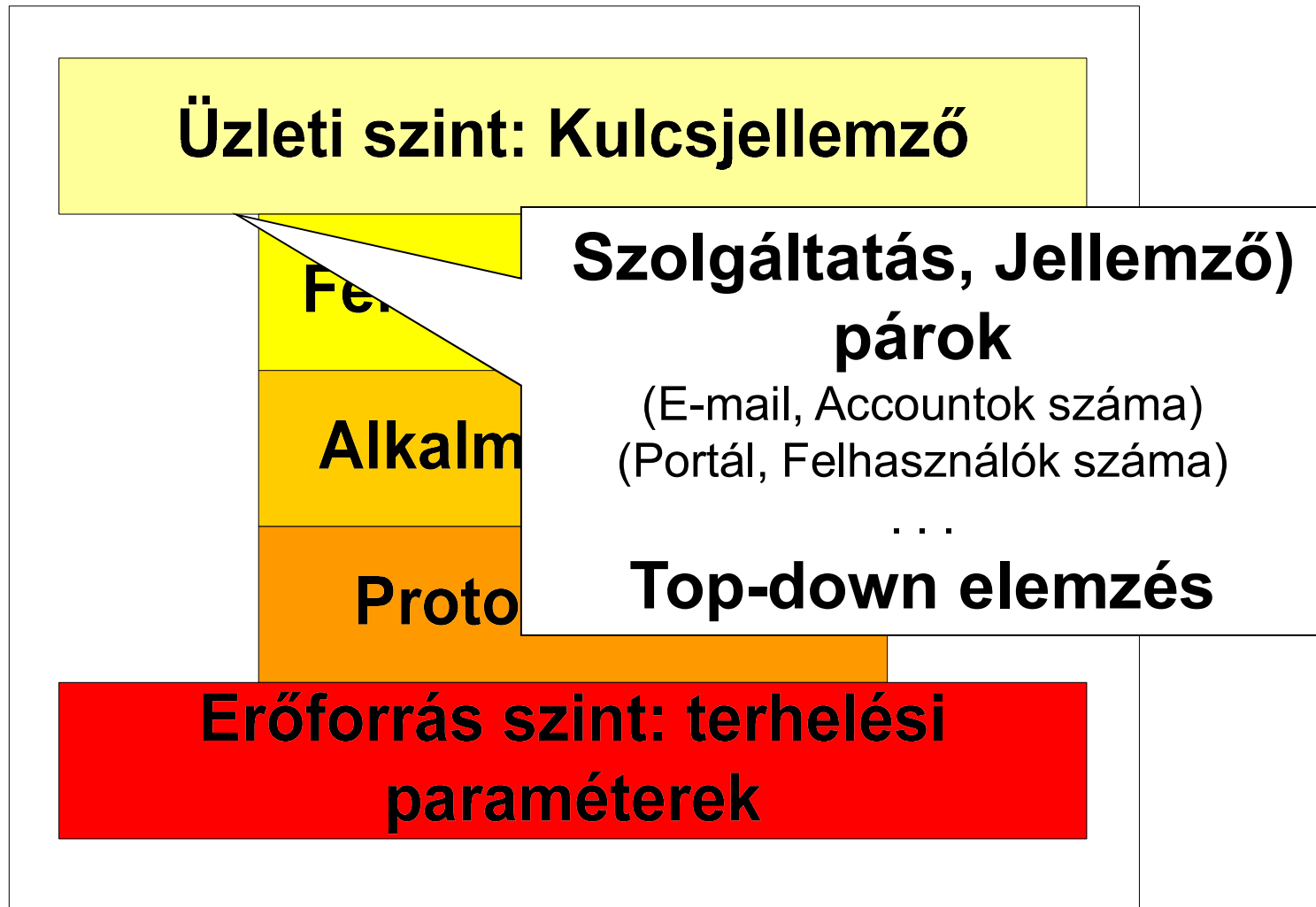
Terhelés előrejelzési stratégia



Hierarchikus előrejelzési modell

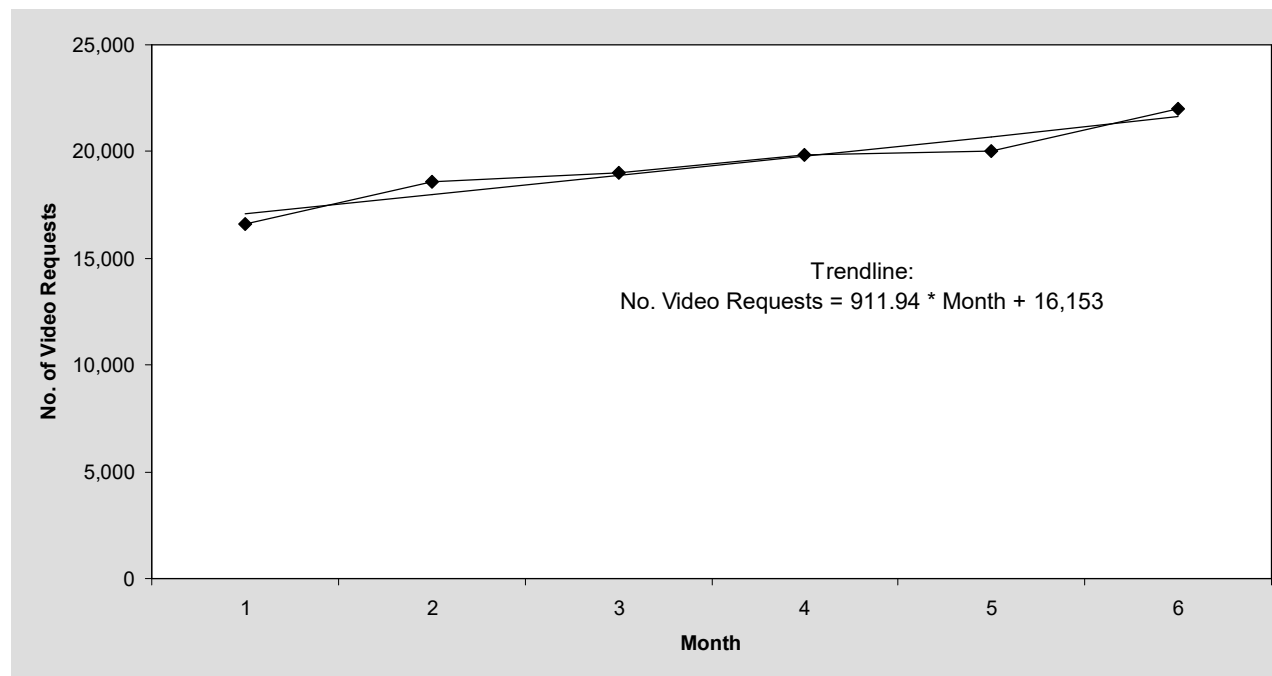


Hierarchikus előrejelzési modell



Terhelés előrejelzés

- A terhelési modell várható változása
- Többféle technika létezik
- Példa: lineáris regresszió



Terhelés típusai

- Terhelési minta azonosítása
- Trend
 - Állandó jellegű változás, pl. folyamatos növekedés
- Időszakos
 - Visszatérő, periodikus hatások
- (Statikus)
- (Véletlenszerű)
- Előrejelzési módszer a konkrét mintához
 - Más területekről
 - Statisztika, meteorológia, tőzsde, stb.

Terhelés előrejelzési technikák

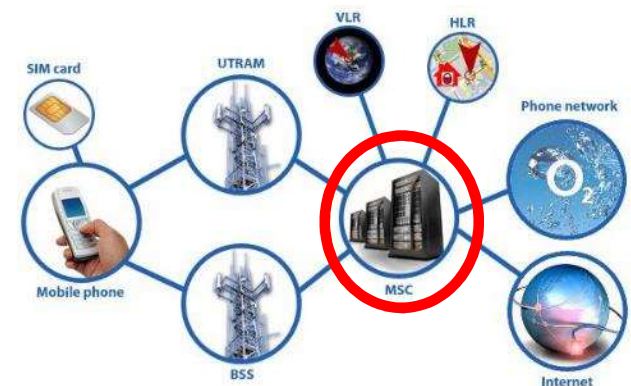
- Regresszió
 - Lineáris
 - Exponenciális
 - (Logaritmikus)
- Mozgó átlagok
- Exponenciális csúszóablak
- Könnyen automatizálhatók
 - Matlab, R
 - MS Excel
 - ...

ESETTANULMÁNY: KRITIKUS SZOLGÁLTATÁS

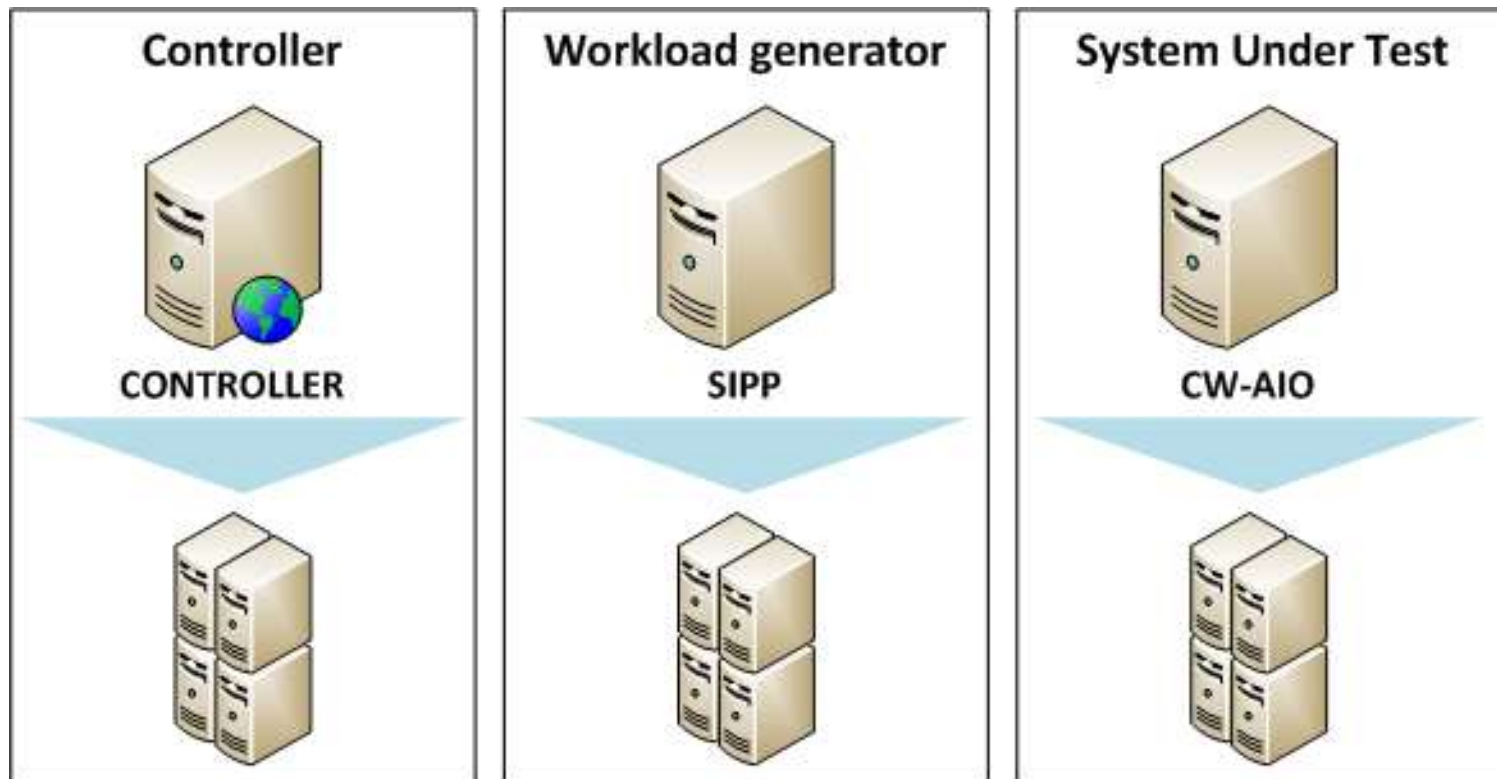


Felhő alapú szolgáltatások

- Telekommunikációs szolgáltatások felhőben
- Az alkalmazásnak működni kell...
 - Mit mérjünk? Mire számítsunk?
- Migrálás, hibatűrés támogatása
- Környezet változhat
- Mikor hibás a rendszer?
 - Kritikus szolgáltatásoknak működniük kell
- Ritka események...



Példa: digitális “telefonközpont”



(Clearwater IMS)

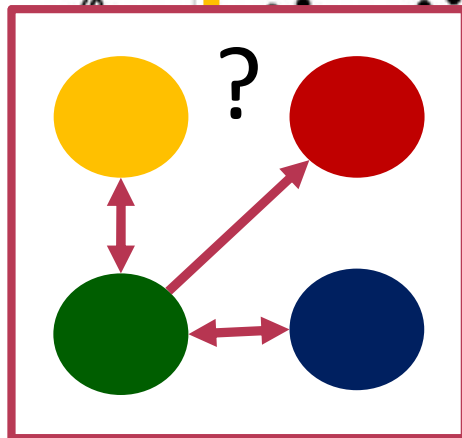
Példa: terhelés vs. kihasználtság

Kiugró értékek/
Háttérműveletek?

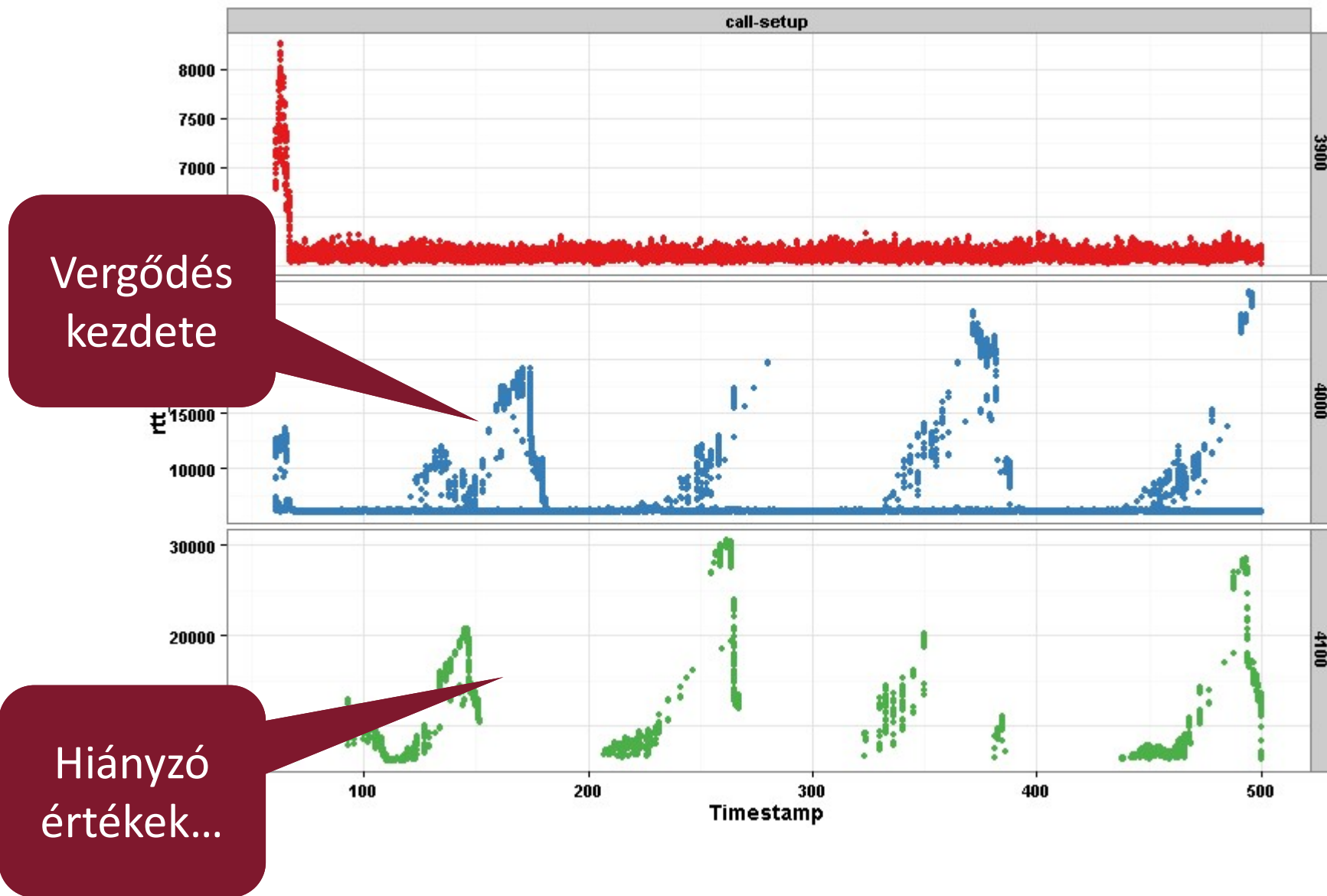
Eltérések?

Lineáris
kapcsolat

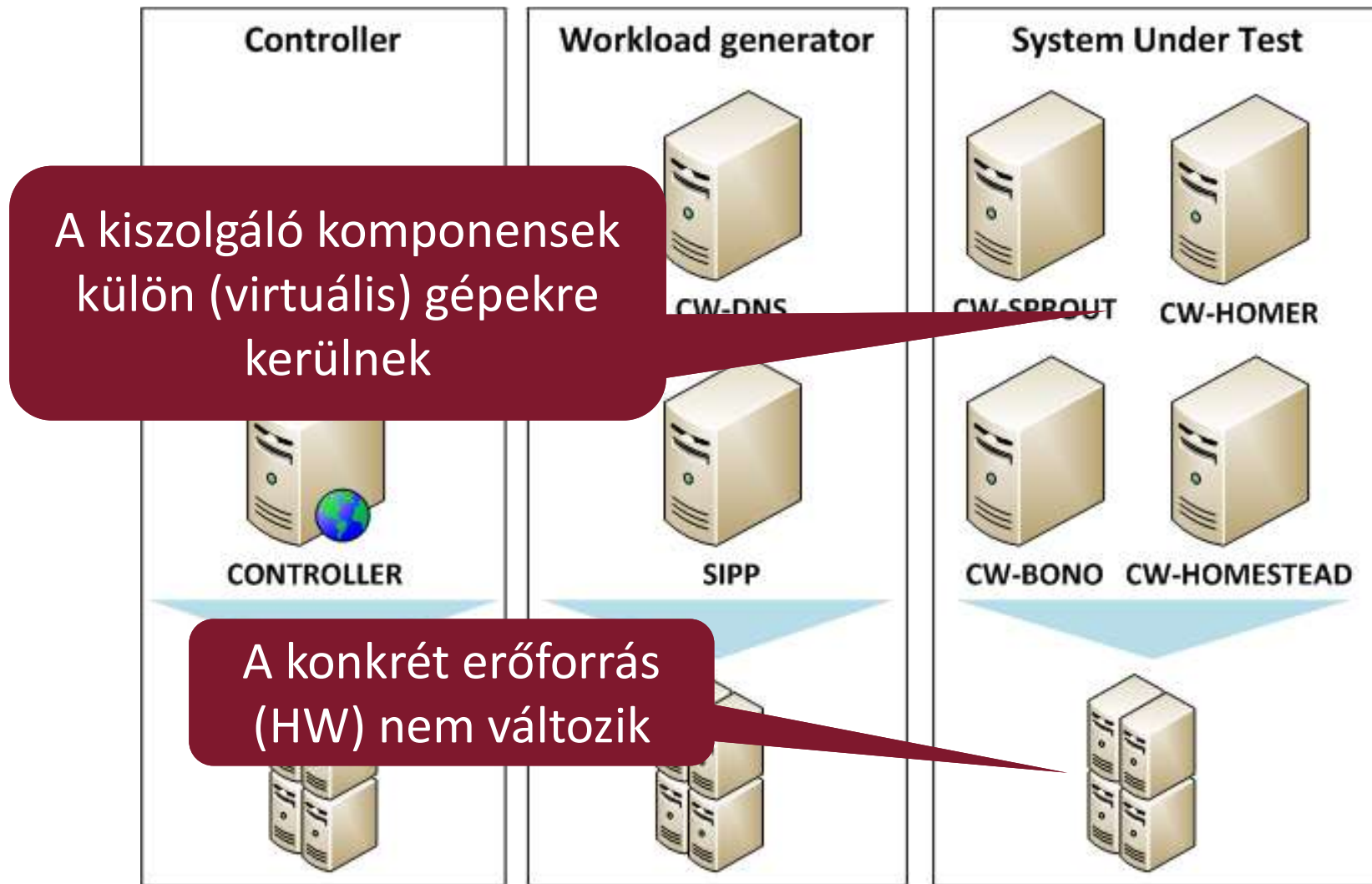
Nagy terhelésnél
nem jósolható
működés



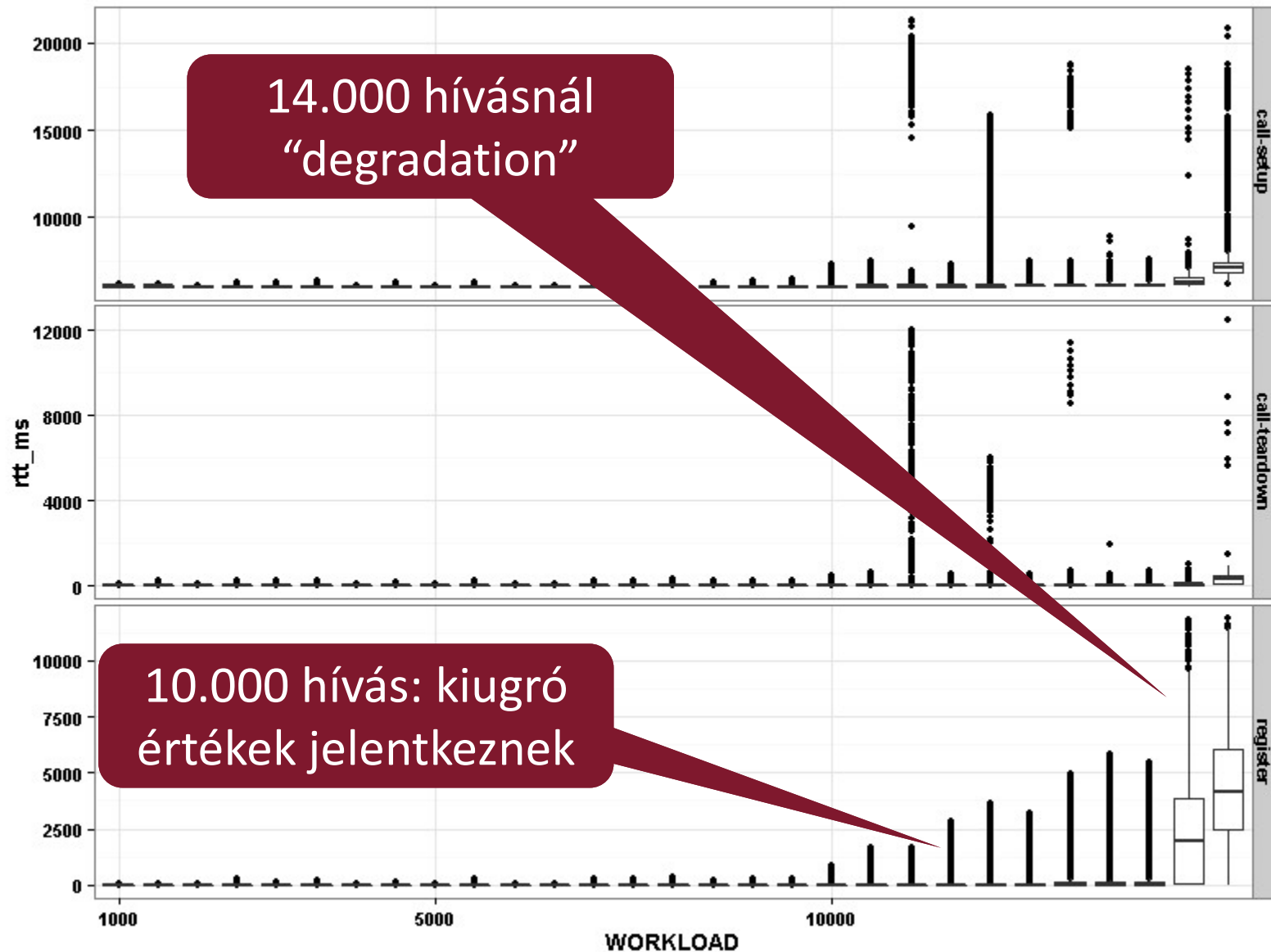
Idősorelemzés



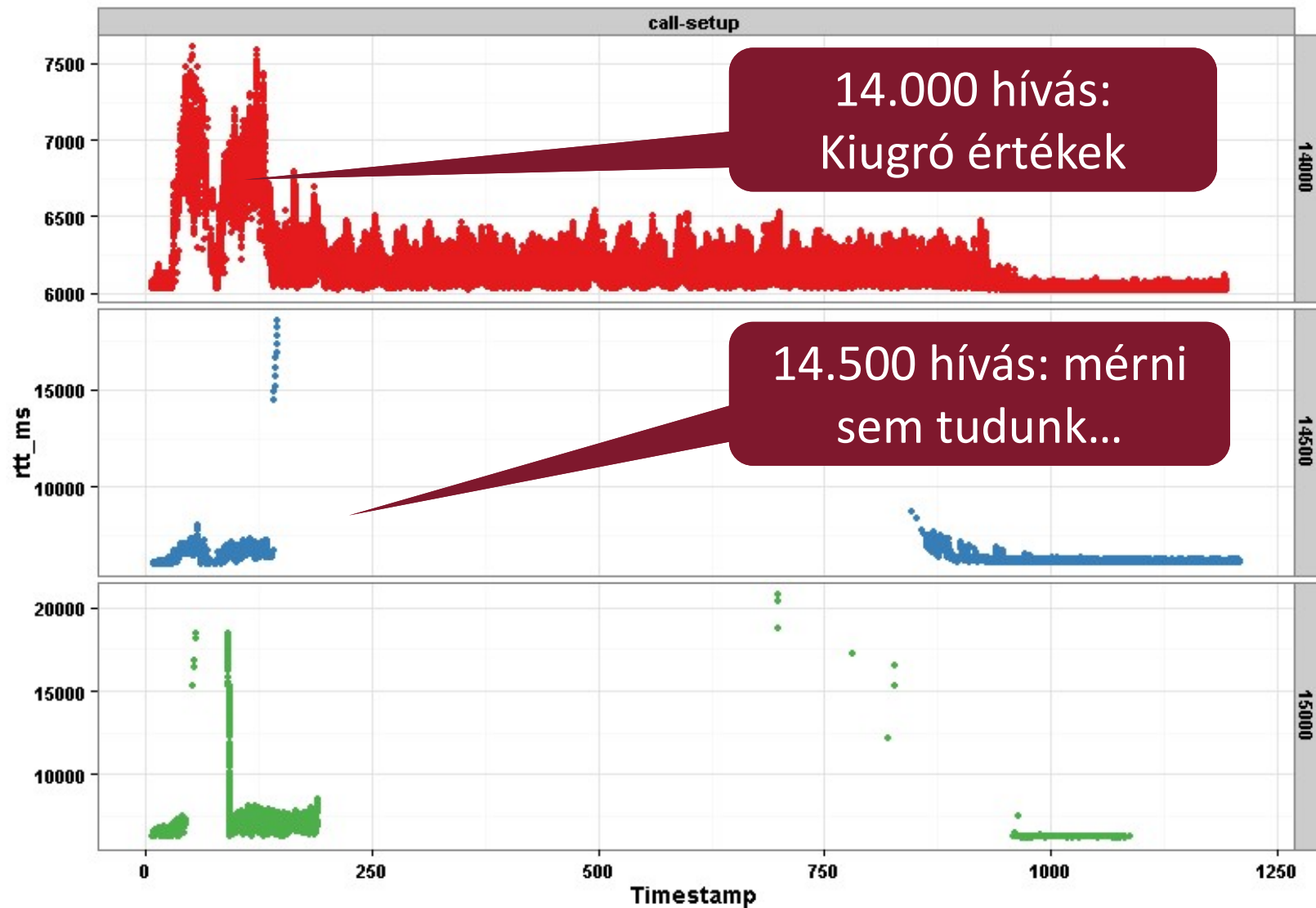
Osszuk szét a feladatokat!



Válaszidő alakulása különböző műveletekre



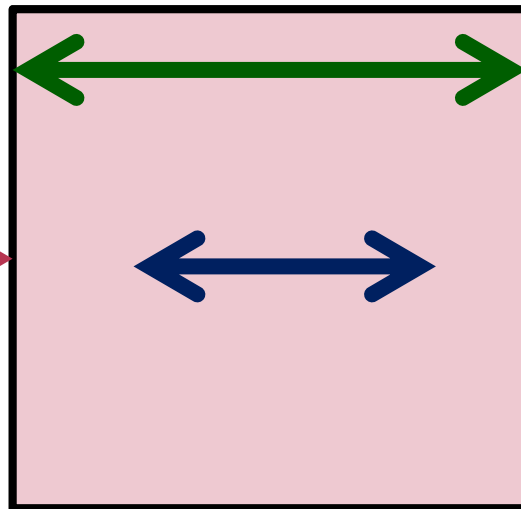
Idősor elemzés



Little törvény példa

CurrentCall
(#total calls in the system)

CallRate
(average)



ResponseTime
(time spent in the system)

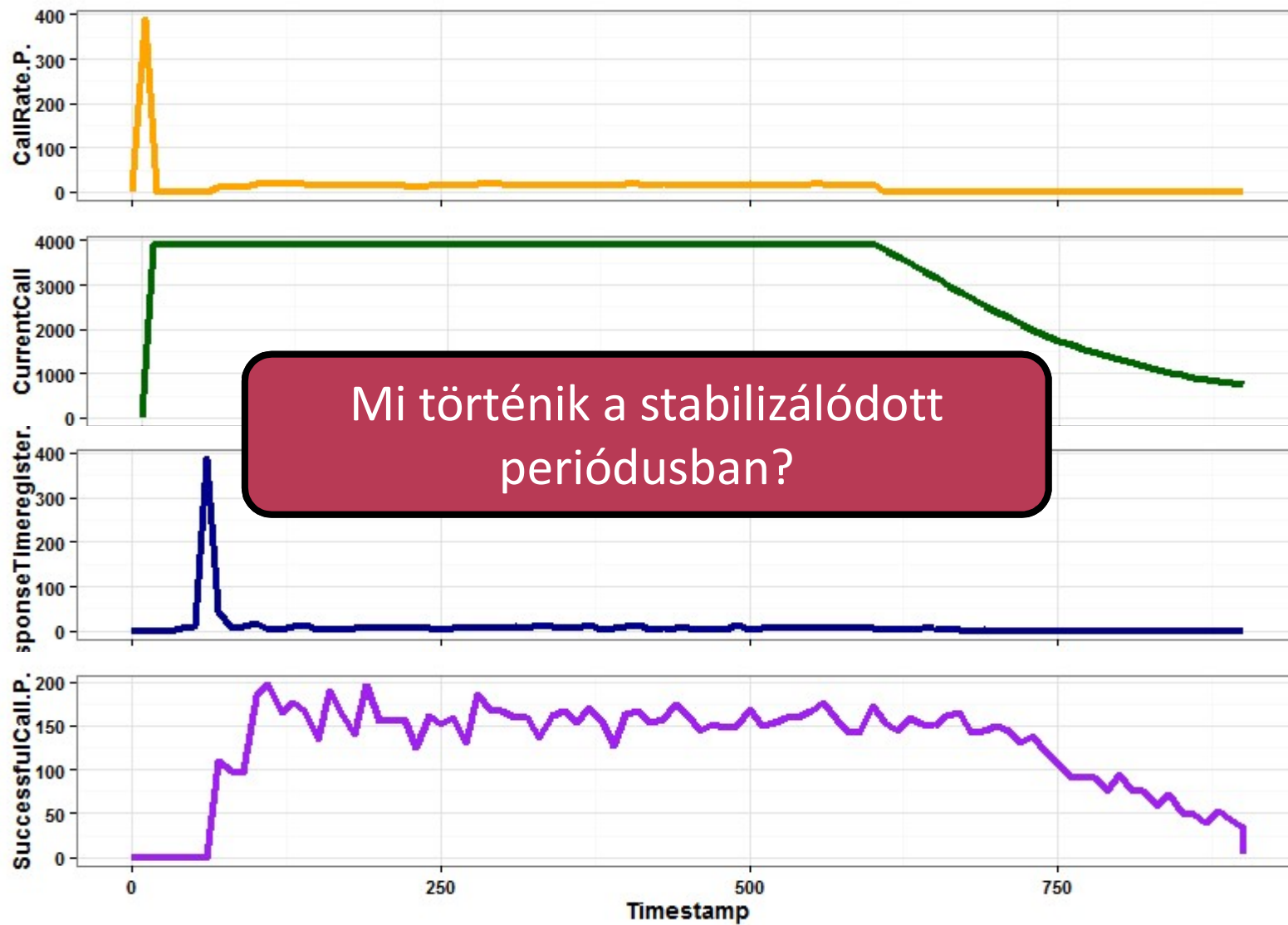
SuccessfulCalls
(# of succ. calls)



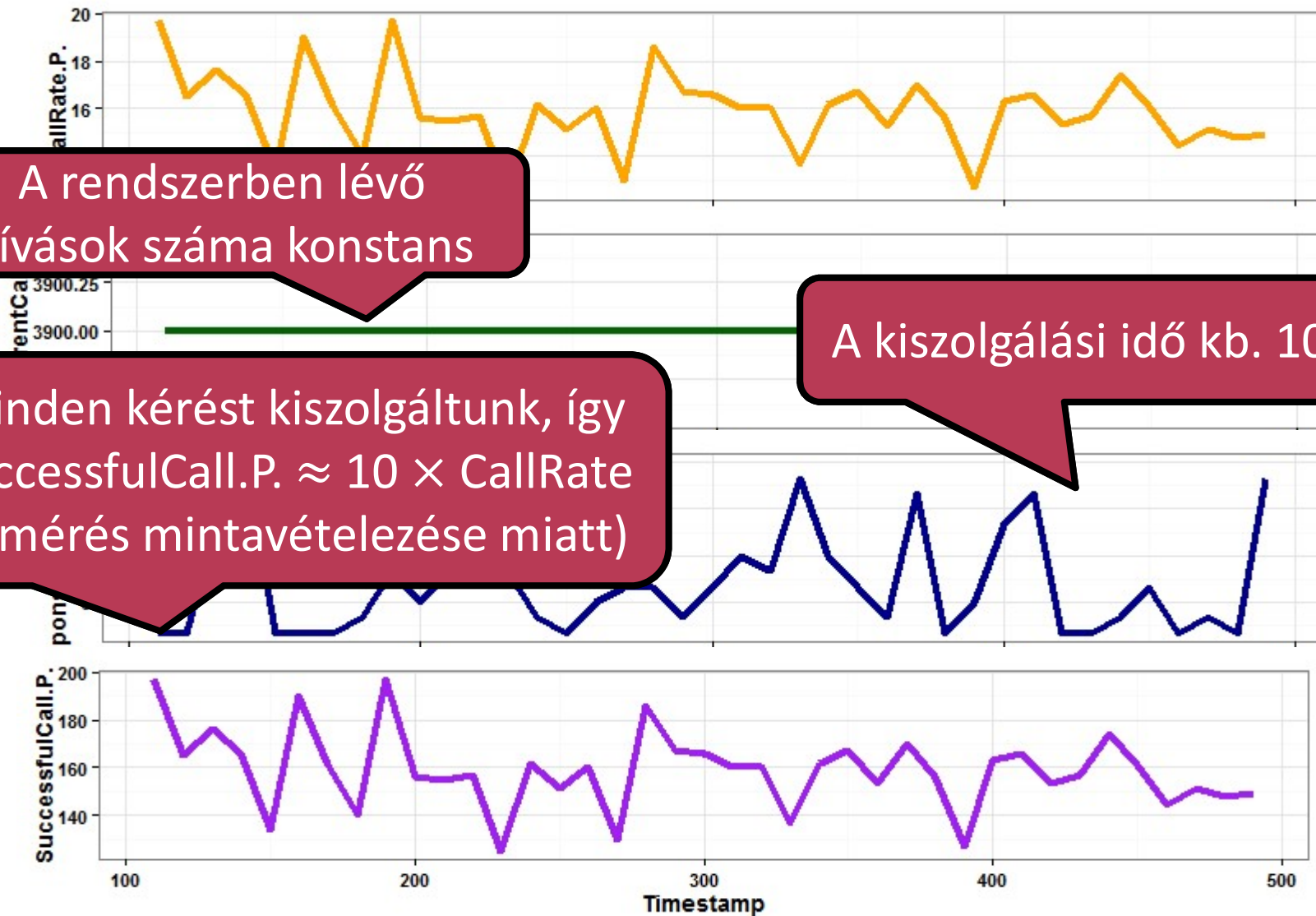
FailedCalls
(# of failed calls)



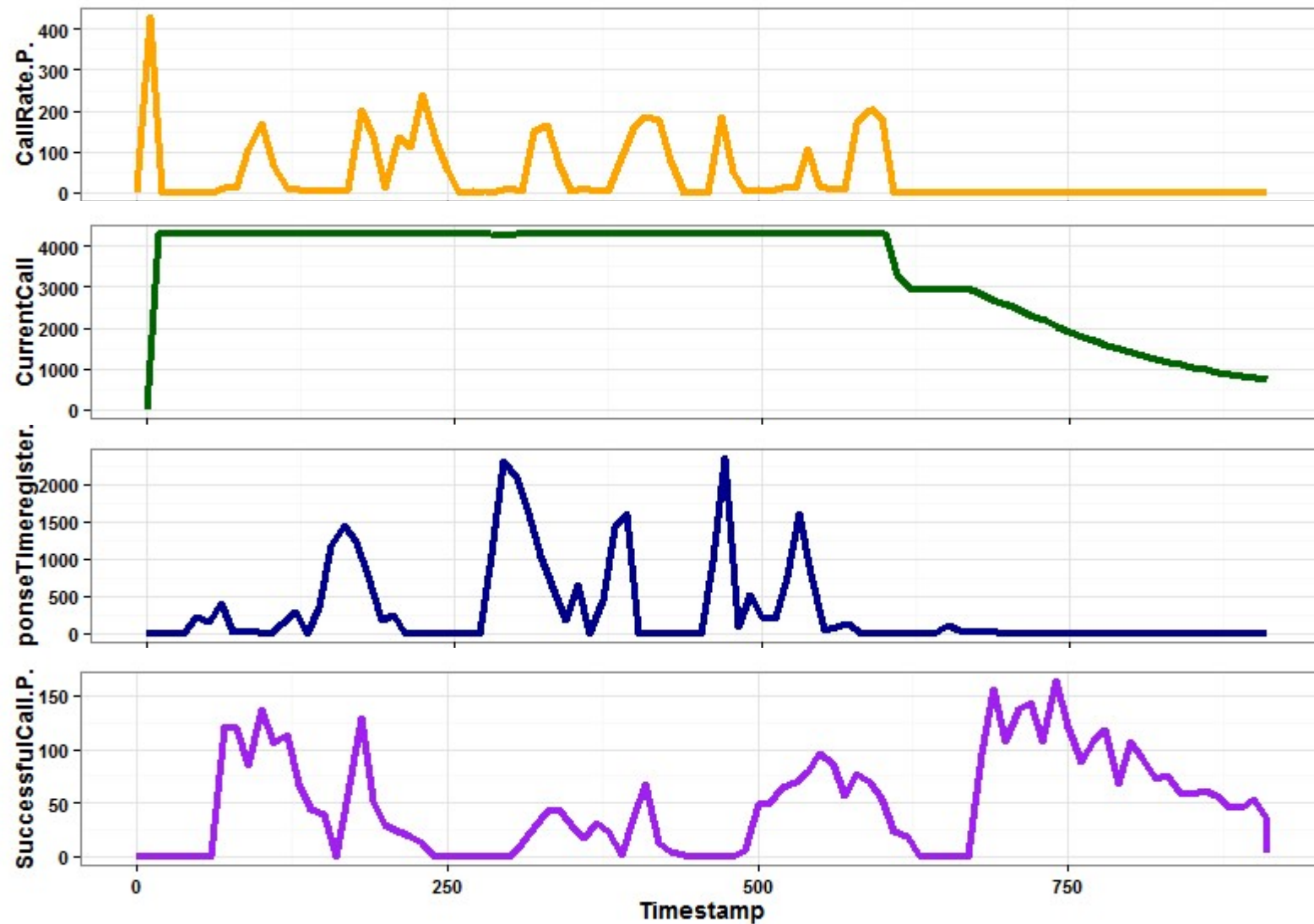
Visszacsatolós terhelés



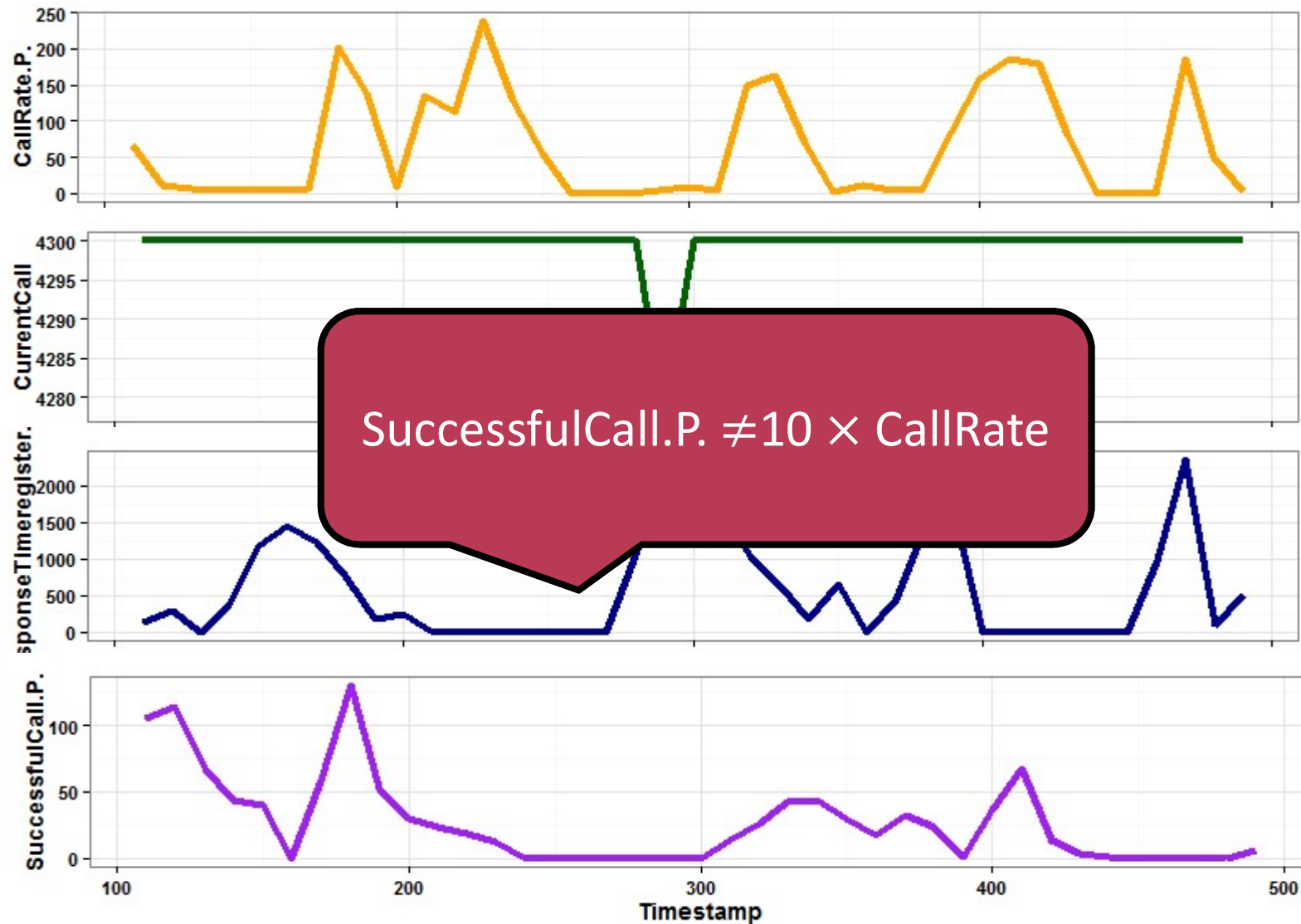
A mérés állandósult állapotban



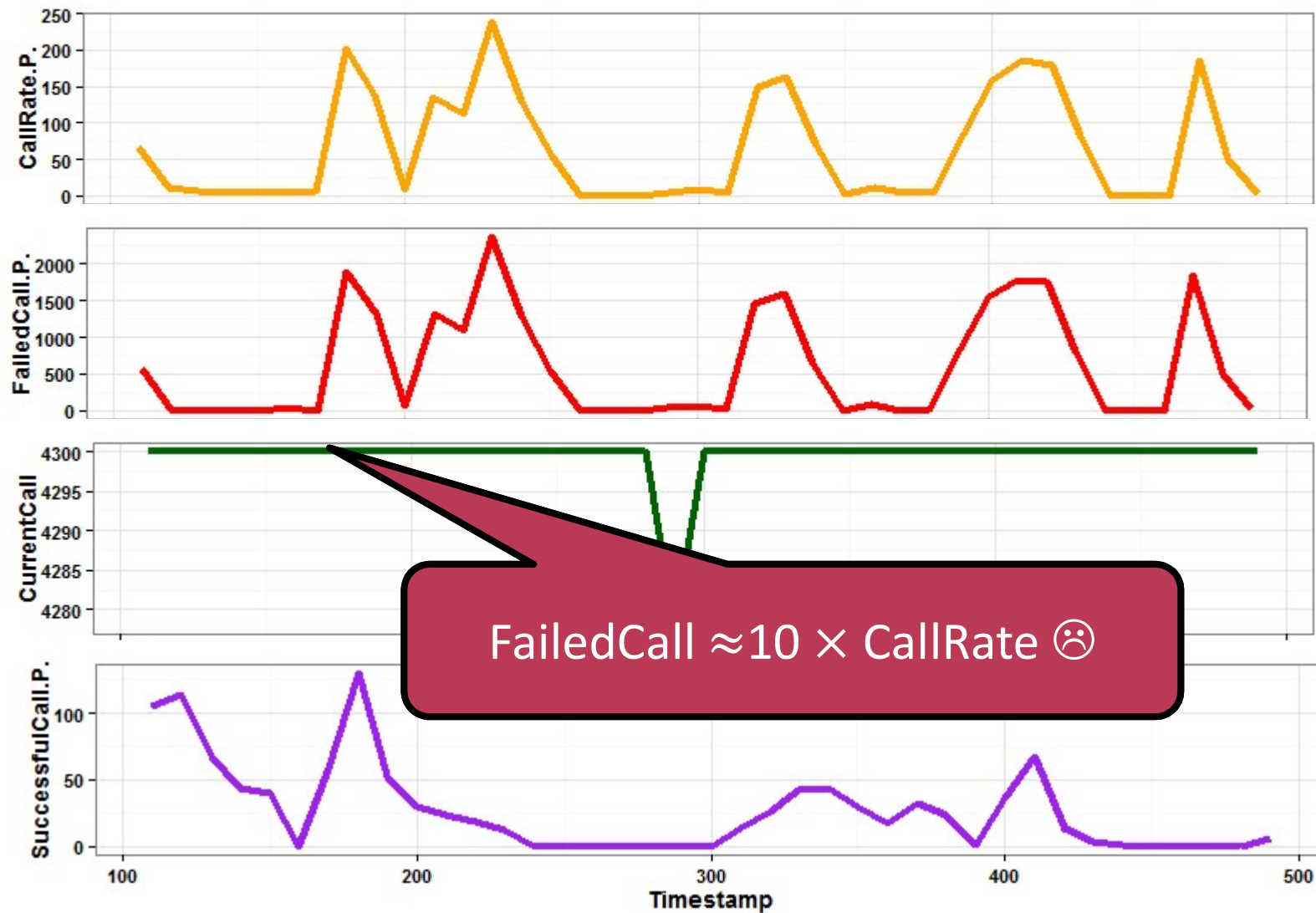
Visszacsatolás, túlterhelés



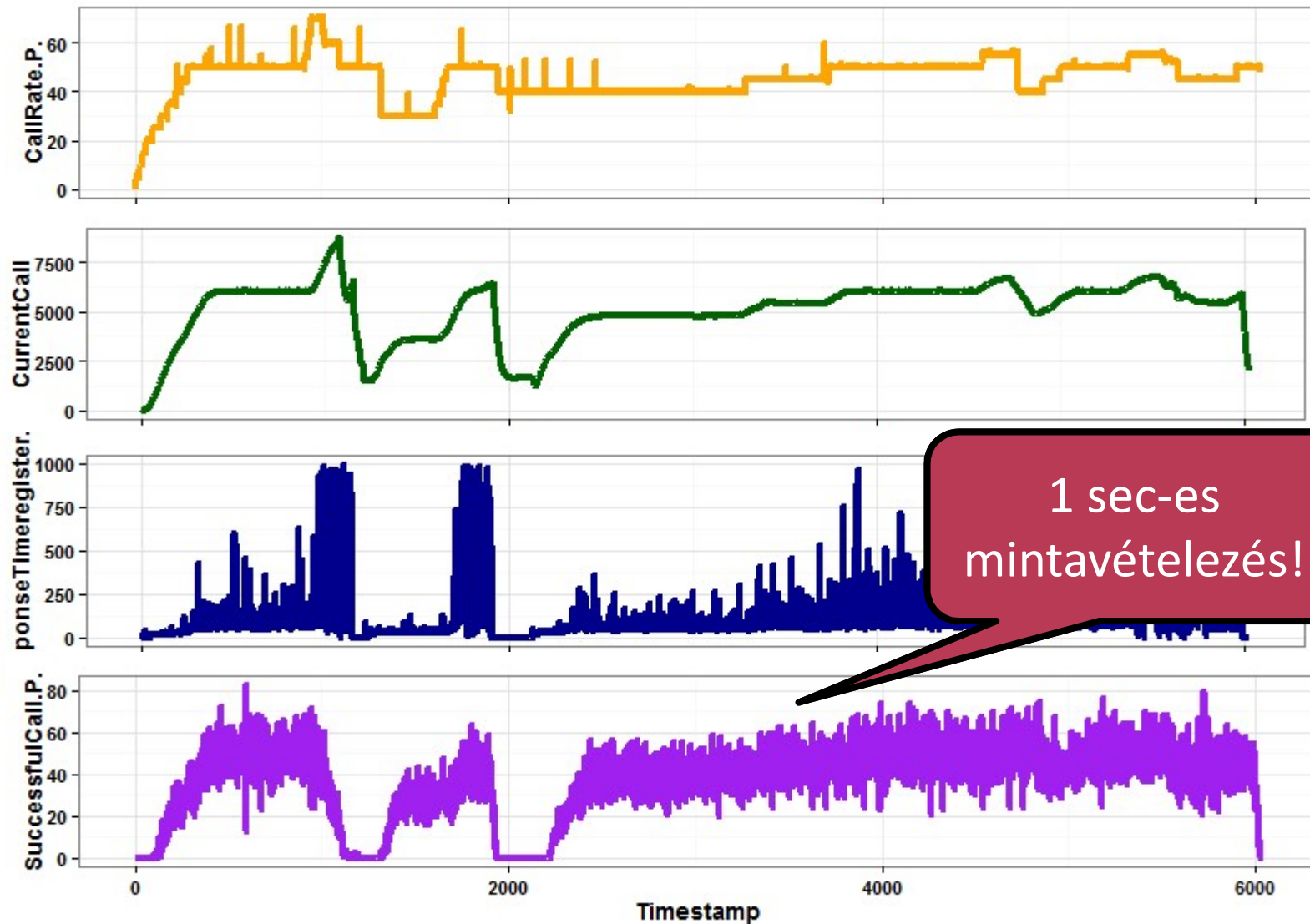
Hol sérül az állandósult állapot feltétele?



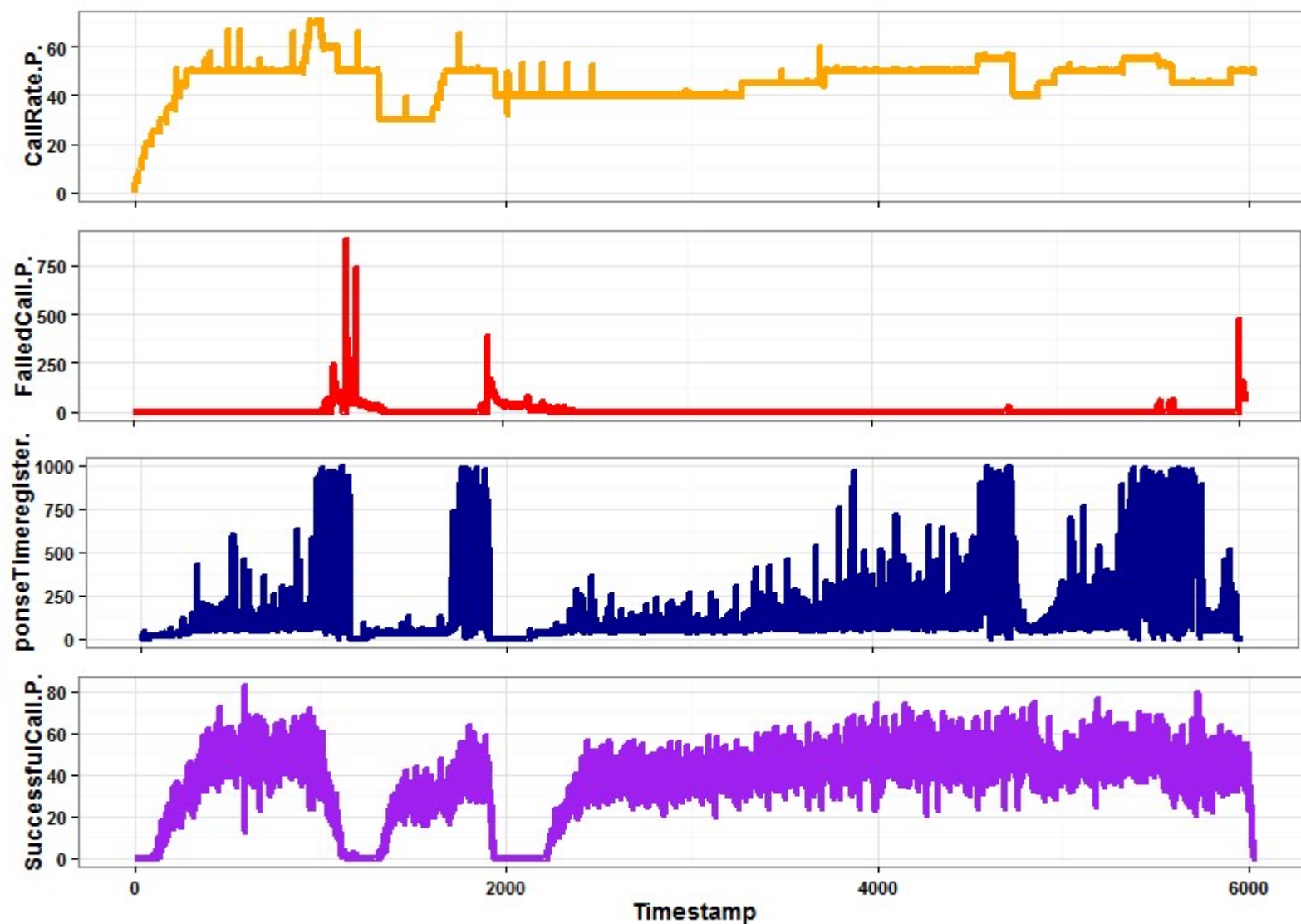
Túlterhelés: Szinte minden kérés hibás...



Nincs visszacsatolás, állandó terhelés



Finomabb mintavételezés

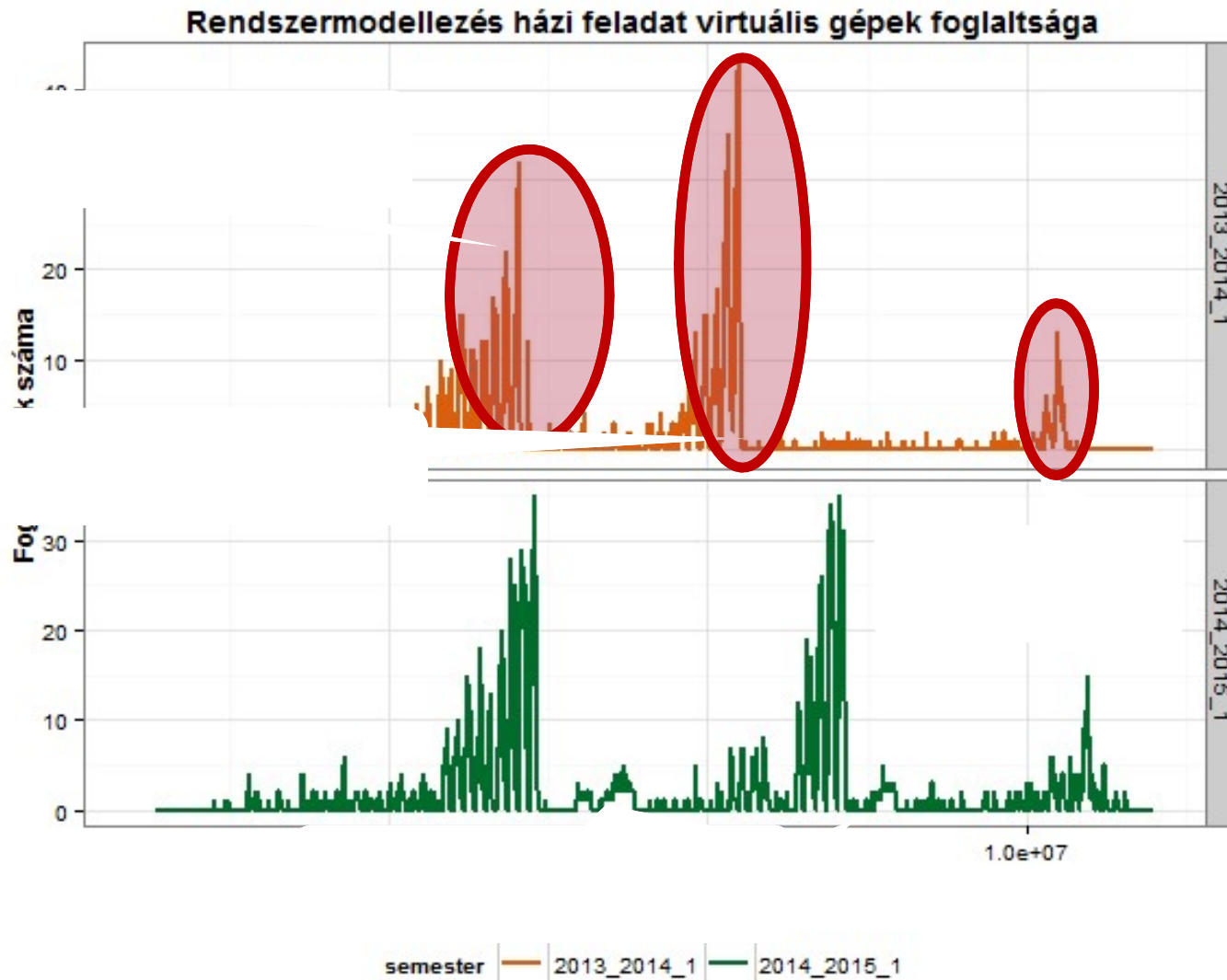


Tanulságok

- Az elvi teljesítménymodell elemeit a gyakorlatban nehéz meghatározni
- Az erőforrások skálázásának az alkalmazások is korlátot szabnak
- Az időbeli változás szerepe kritikus
- Sok változó, sok mérés → adatelemzés
 - Ld. később...

KITEKINTÉS: KAPACITÁS/KÖLTSÉGSZÁMÍTÁS OKTATÁSI FELHŐHÖZ

Rendszermodellezés (7. félév) a felhőben



Hogyan becsüljük költséget? (DEMÓ)

