Cloud Based Analytics for Cloud Based Applications

András Pataricza, with contributions of
Imre Kocsis[1], Gábor Horváth, Ágnes Salánki

Dept. of Measurement and Information Systems, BME, Hungary

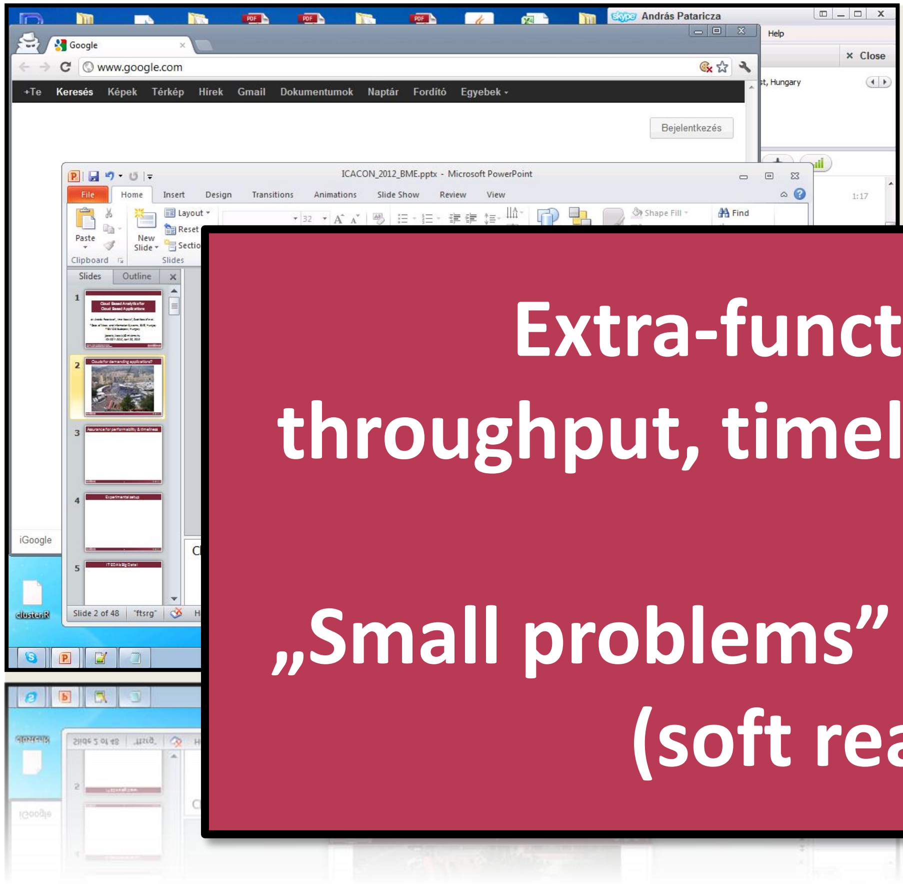
Standard infrastructure for demanding applications?

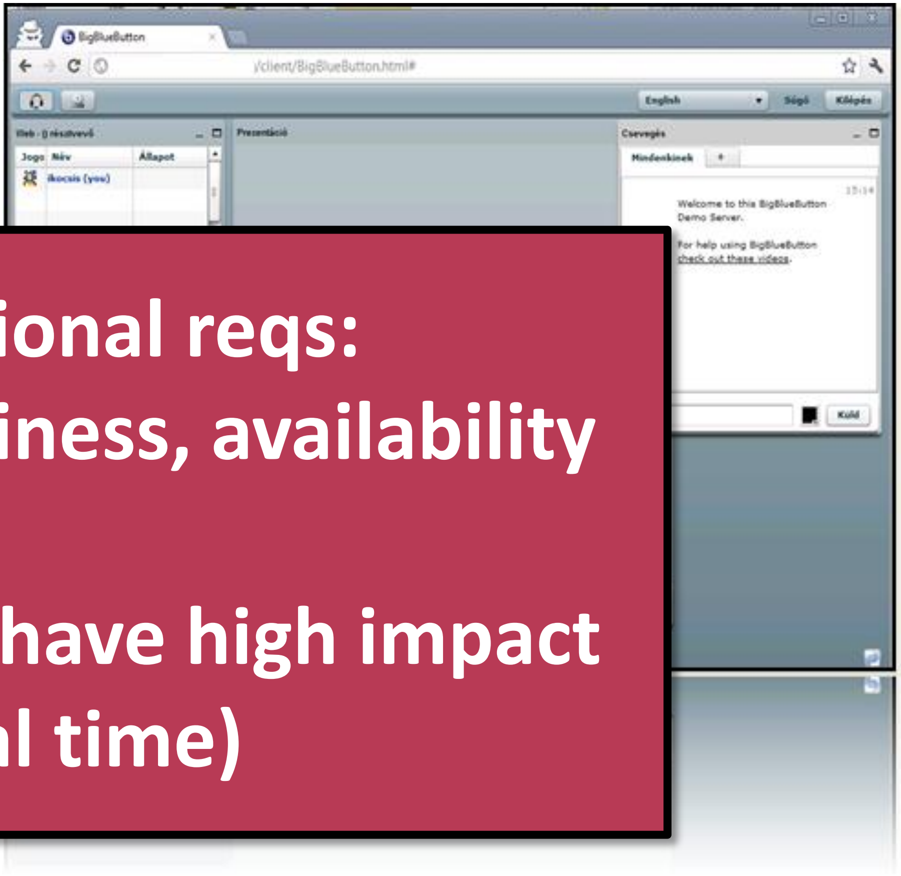
Power supply to an ER

- COTS energy source replicated -> ->
- Warm backup

- Power supply to an ER
- COTS energy source replicated
- Warm backup

Clouds for demanding applications?


András Patariczа
Google
www.google.com
+Te Keresés Képek Térkép Hírek Gmail Dokumentumok Naptár Fordító Egyebek
Bejelentkezés
ICACON_2012_BME.pptx - Microsoft PowerPoint
File Home Insert Design Transitions Animations Slide Show Review View
Paste Layout Reset New Slide Shape Fill Find
Clipboard Slides
Slides Outline
iGoogle
Slide 2 of 48 "ftsrg"
Help
Close



Welcome to this BigBlueButton Demo Server.

For help using BigBlueButton check out these videos.

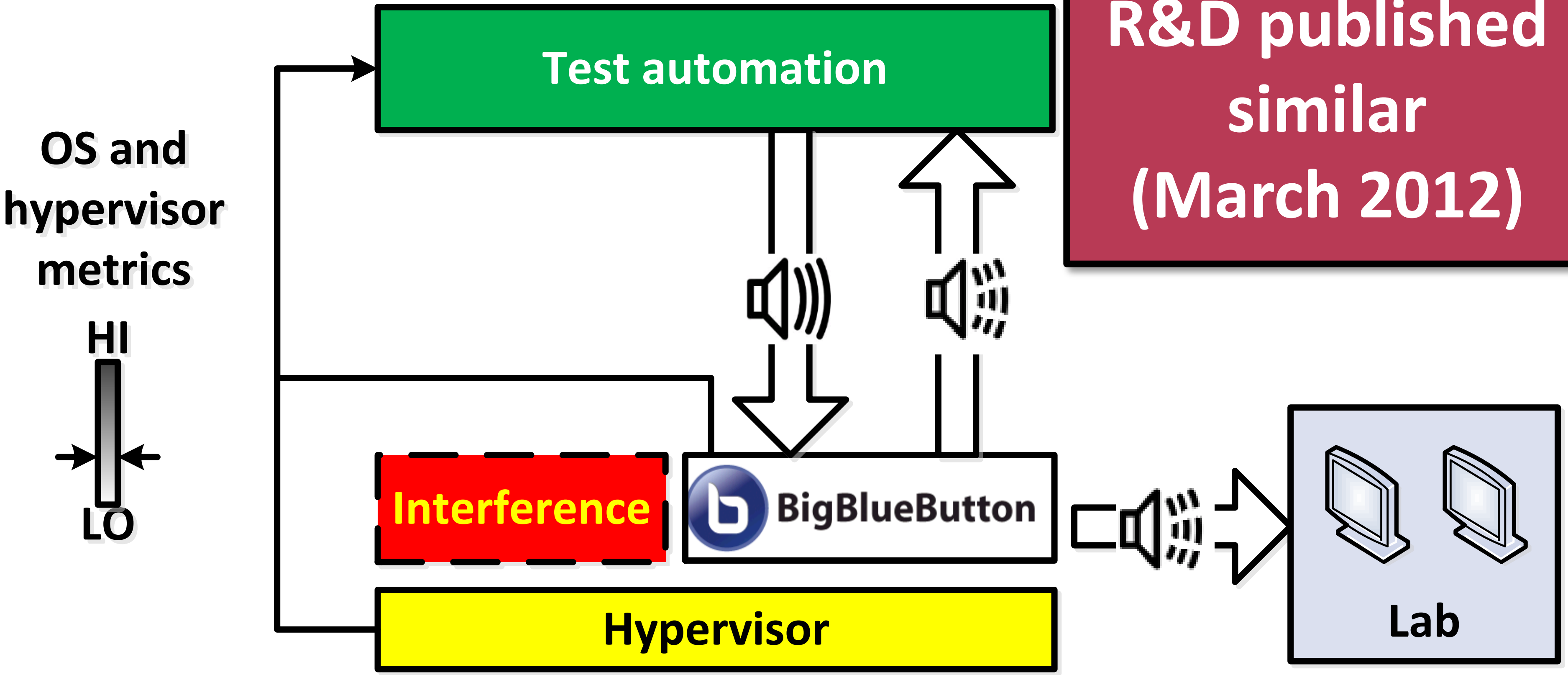
Extra-functional reqs: throughput, timeliness, availability

„Small problems" have high impact (soft real time)

Virtual Desktop Infrastructure

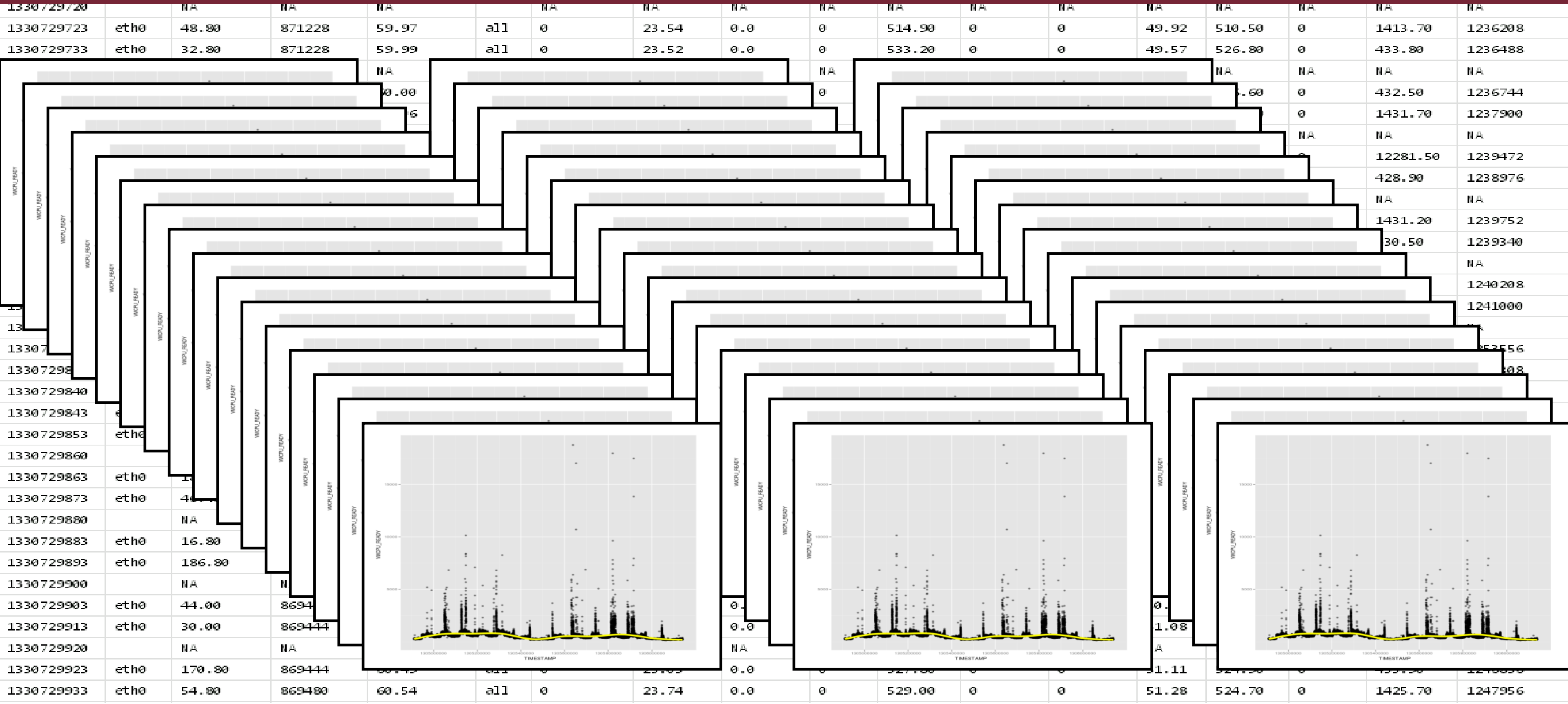
Telecommunications

Experimental setup



N.B.: VMware R&D published similar (March 2012)

IT EDA is Big Data!



1330729720		NA	NA	NA		NA	NA	NA	NA	NA	NA	NA	NA	NA	NA	NA	NA
1330729723	eth0	48.80	871228	59.97	all	0	23.54	0.0	0	514.90	0	0	49.92	510.50	0	1413.70	1236208
1330729733	eth0	32.80	871228	59.99	all	0	23.52	0.0	0	533.20	0	0	49.57	526.80	0	433.80	1236488
				NA					NA				NA		NA	NA	NA
				0.00					0					.60	0	432.50	1236744
				6											0	1431.70	1237900
															NA	NA	NA
																12281.50	1239472
																428.90	1238976
																NA	NA
																1431.20	1239752
																30.50	1239340
																	NA
																	1240208
13																	1241000
13																	
13307																	53556
1330729																	08
1330729840																	
1330729843																	
1330729853	eth0																
1330729860																	
1330729863	eth0	1															
1330729873	eth0	4															
1330729880		NA															
1330729883	eth0	16.80															
1330729893	eth0	186.80															
1330729900		NA	N														
1330729903	eth0	44.00	8694					0.					0.				
1330729913	eth0	30.00	869444					0.0					1.08				
1330729920		NA	NA					NA					A				
1330729923	eth0	170.80	869444					0.0					1.11				
1330729933	eth0	54.80	869480	60.54	all	0	23.74	0.0	0	529.00	0	0	51.28	524.70	0	1425.70	1247956

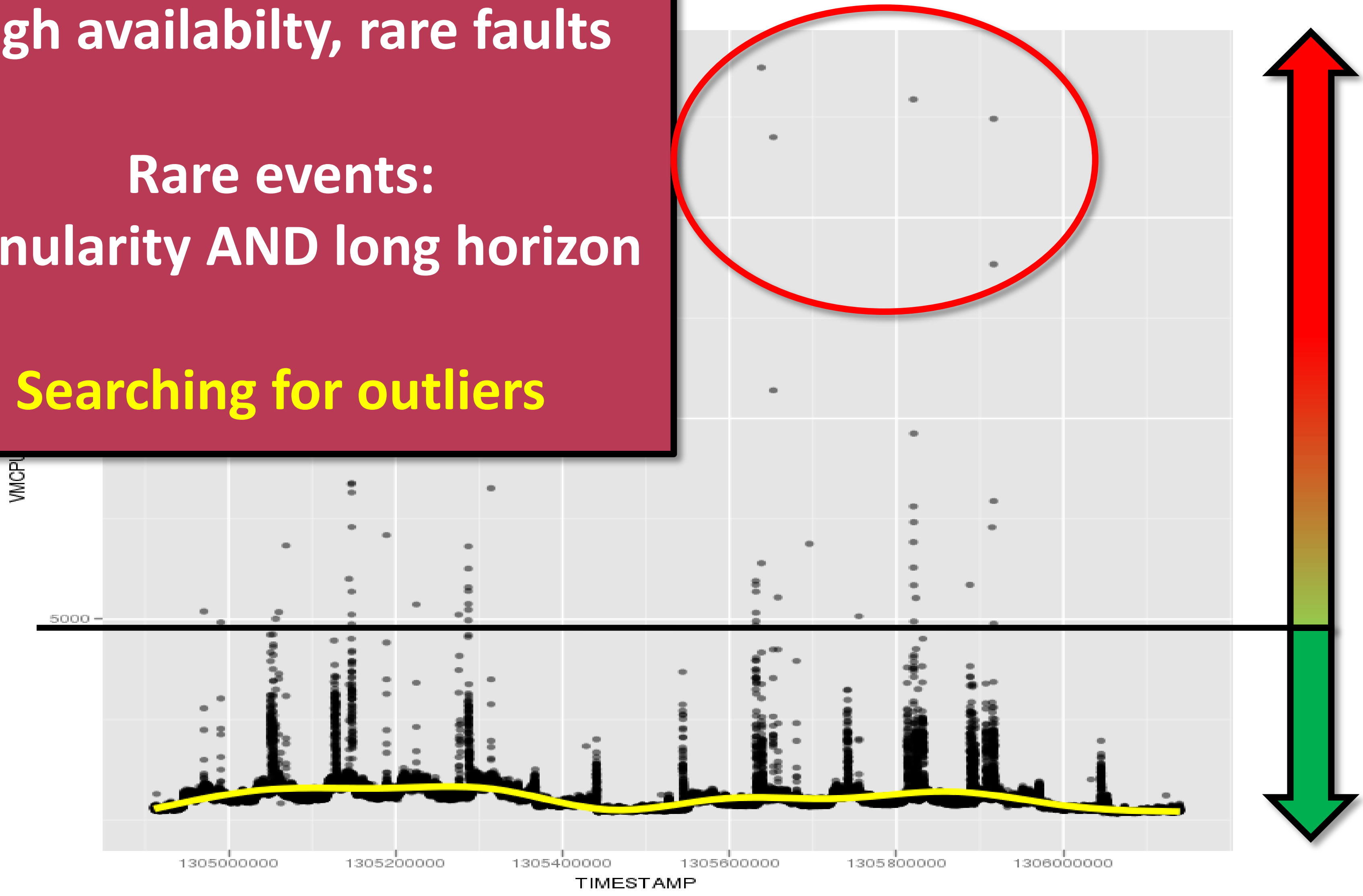
Which determine the QoS?

IT EDA is Big Data!

High availabilty, rare faults

Rare events:
granularity AND long horizon

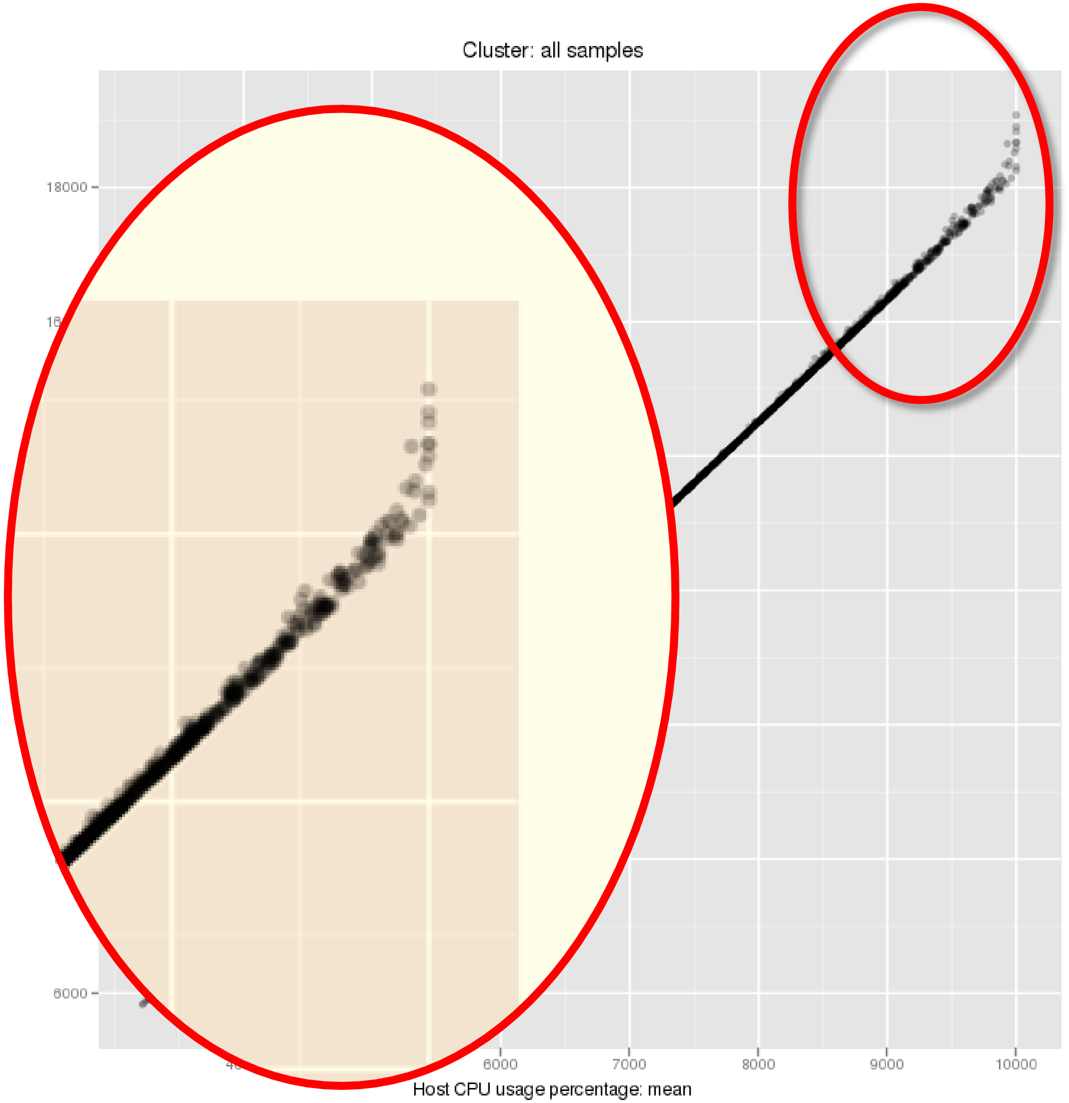
Searching for outliers



Rare events: lot of sand, a few pellets

Typically sand: gold mining ≠ data mining

Visual analytics = causal **insight**



Computing power use = CPU use × CPU clock rate (const.)

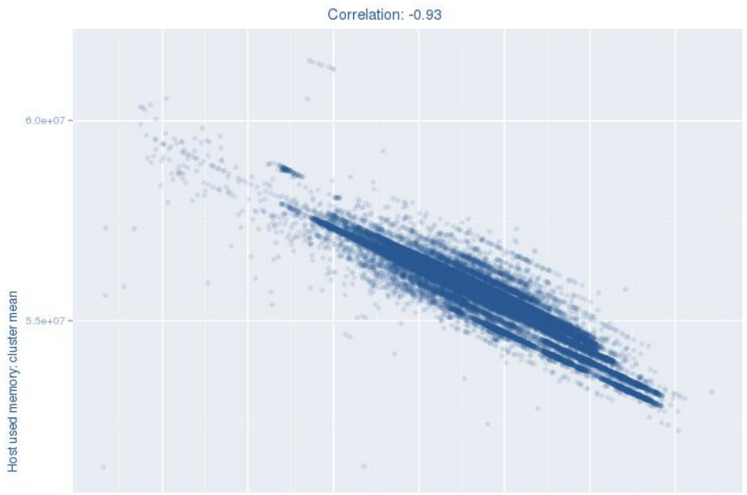
Should be pure proportional

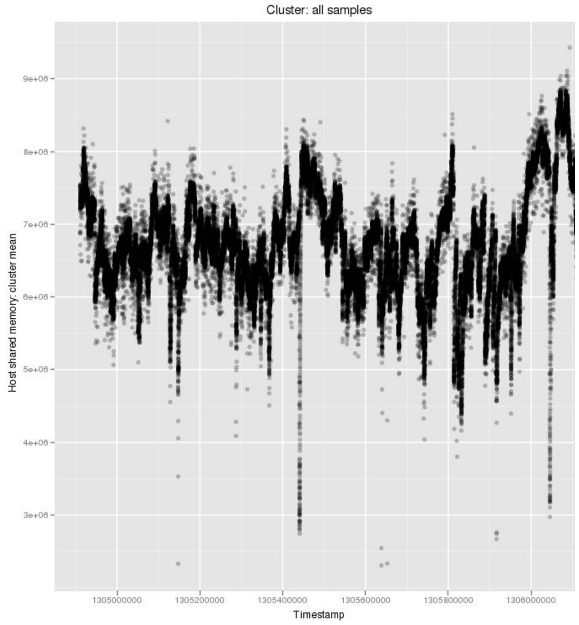
Correlation coefficient: 0.99998477434137

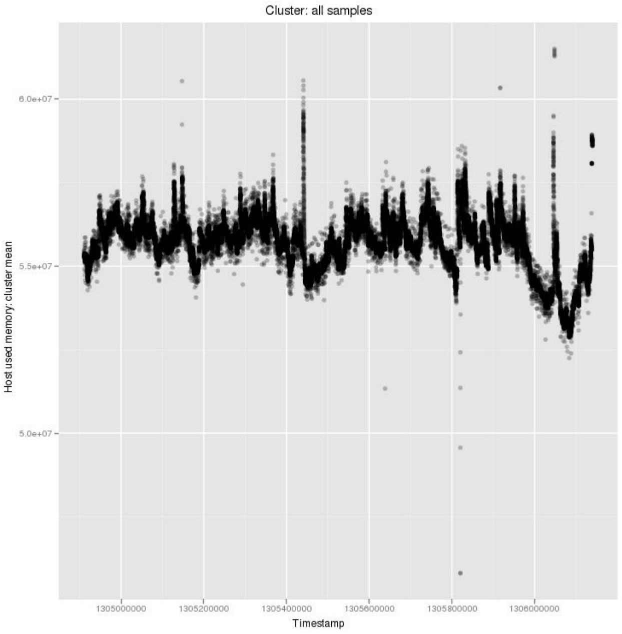
Well-visible, but numerically suppressed

Origin???

Visual analytics







Noisy…

High frequency components dominate

But they correlate (93%!)

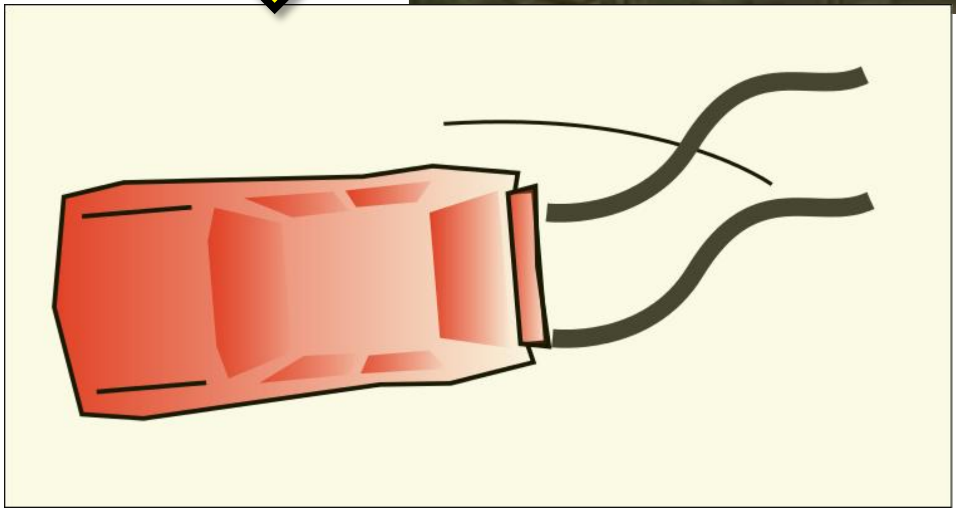
YOU DON’T SEE IT

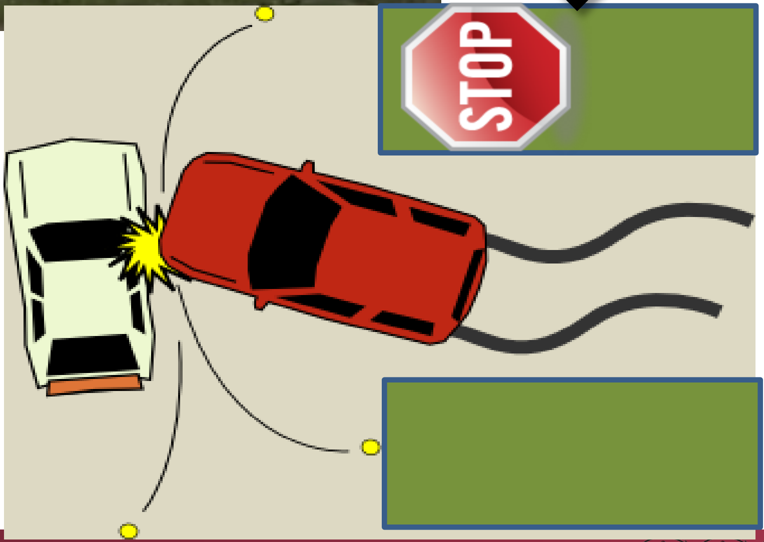
Dangers in a standard cloud for demanding apps?

Impacts of resource sharing?

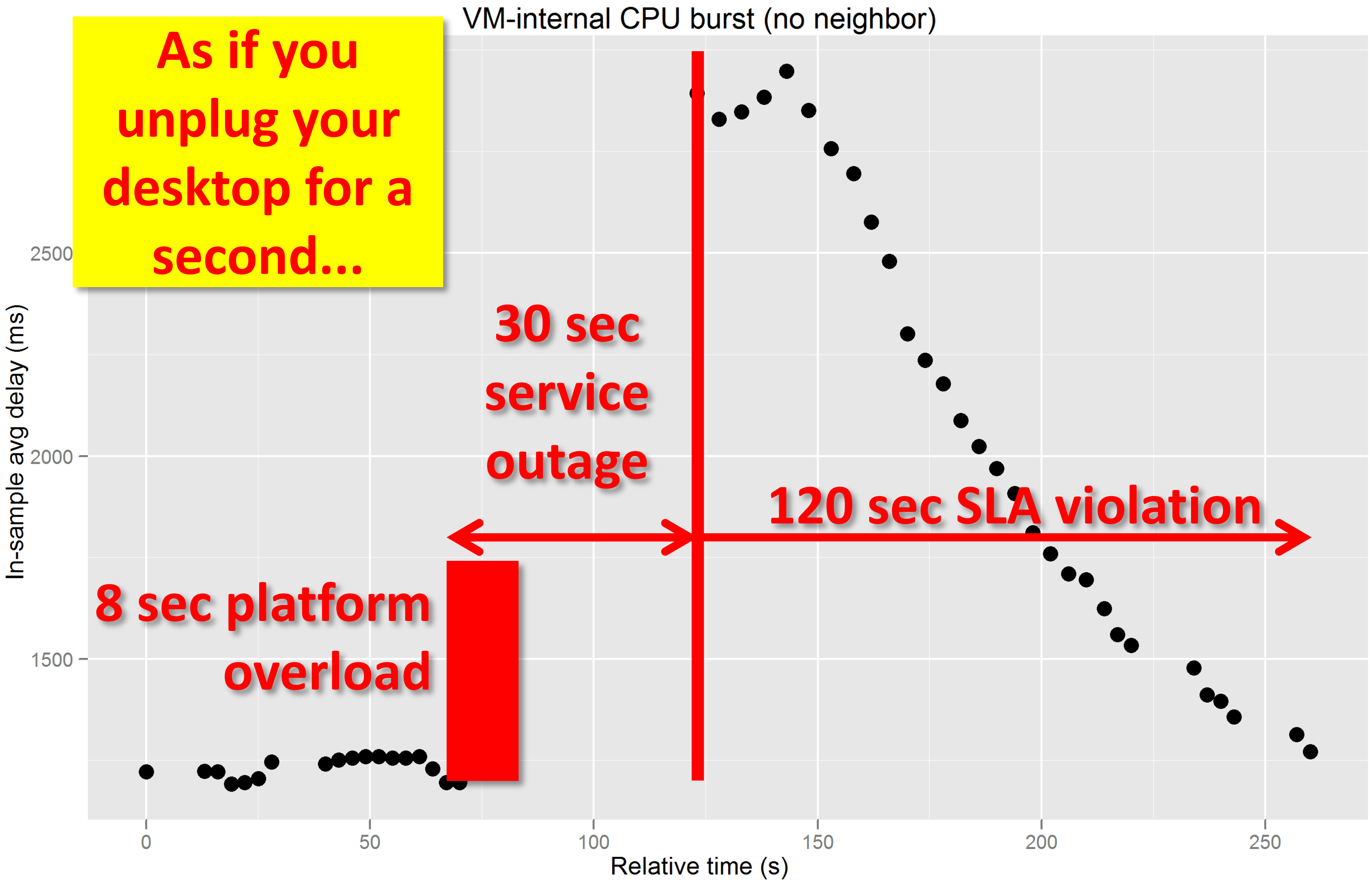
Self-induced

Parasitic influence





Short transient faults – long recovery



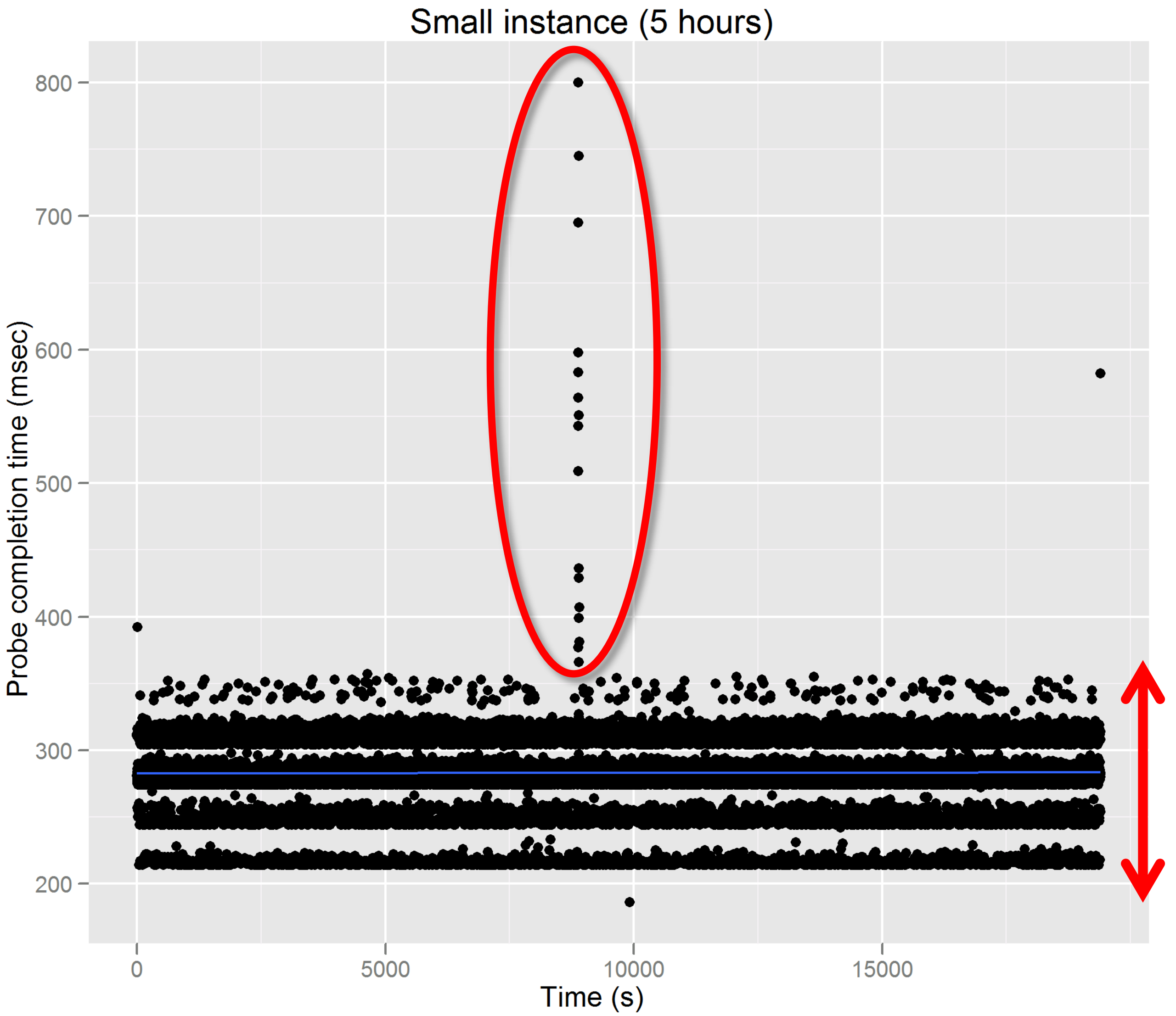
As if you unplug your desktop for a second...

30 sec service outage

120 sec SLA violation

8 sec platform overload

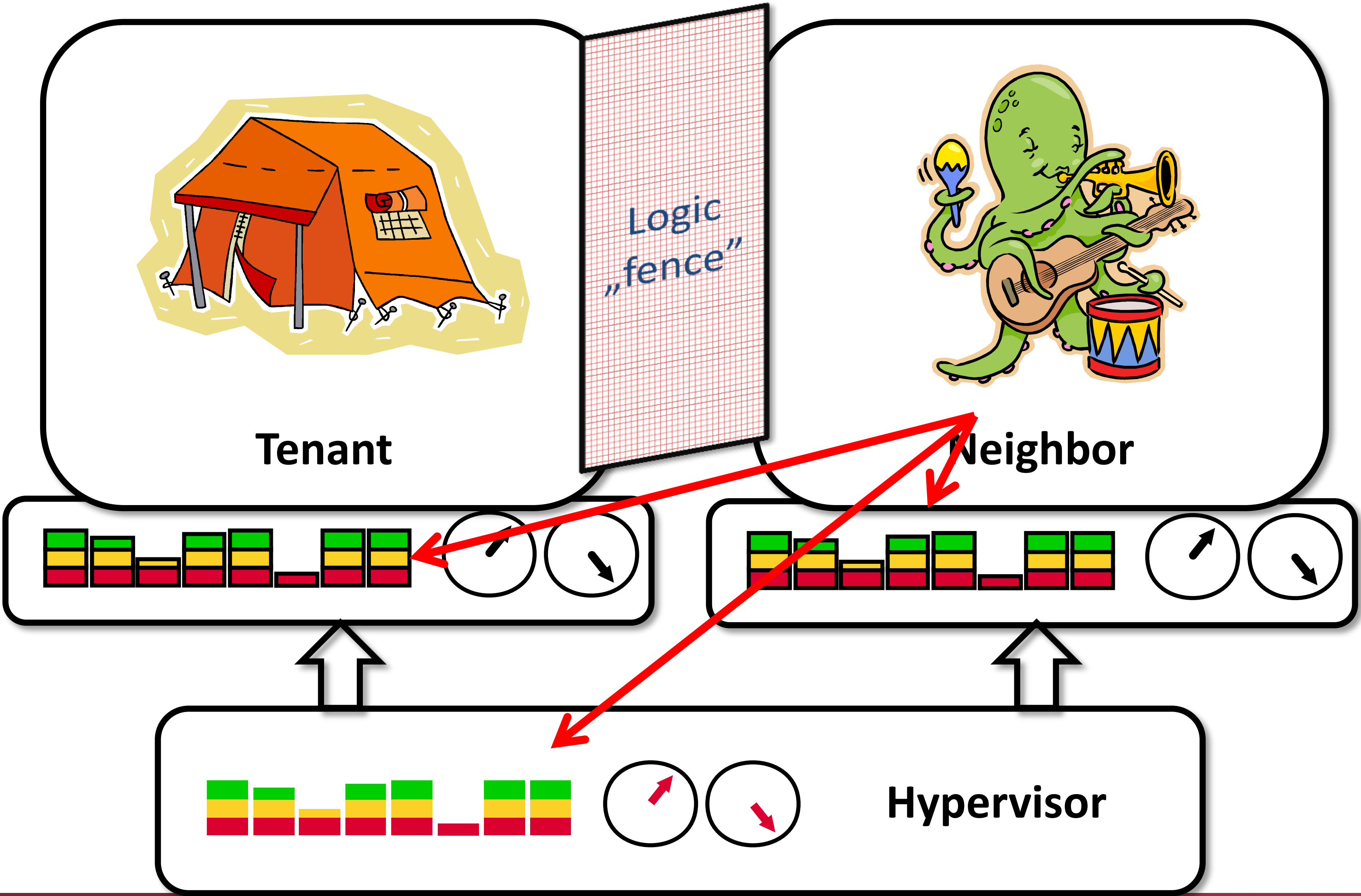
Deterministic (?!) run-time in the public cloud...



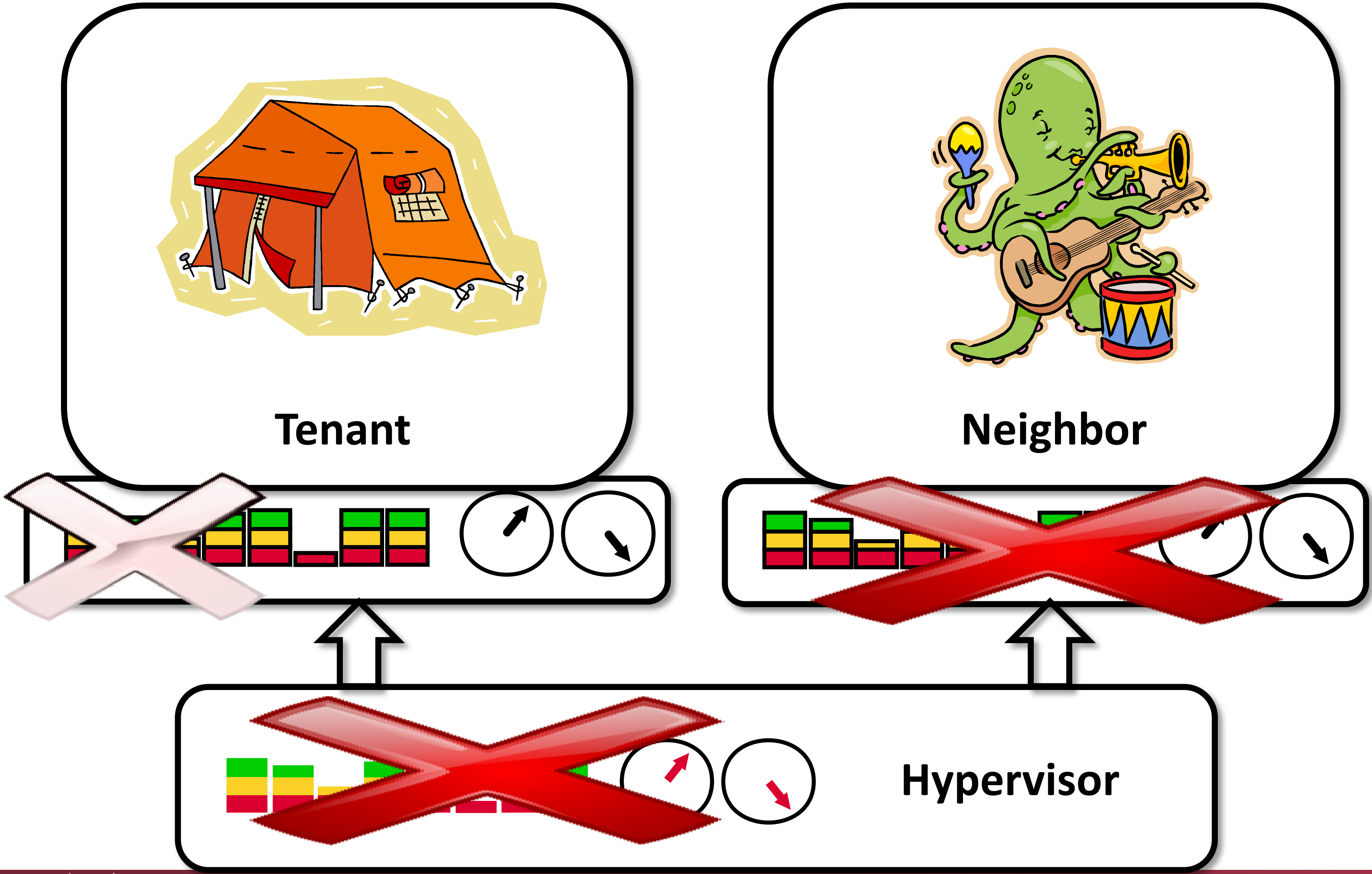
Performance outage intolerable by overcapacity

Variance tolerable by overcapacity

The noisy neighbour problem



Tenant-side measurability and observability



Let’s try it at user level

The mistery shopper concept

- Basic logic as with benchmarks, but...

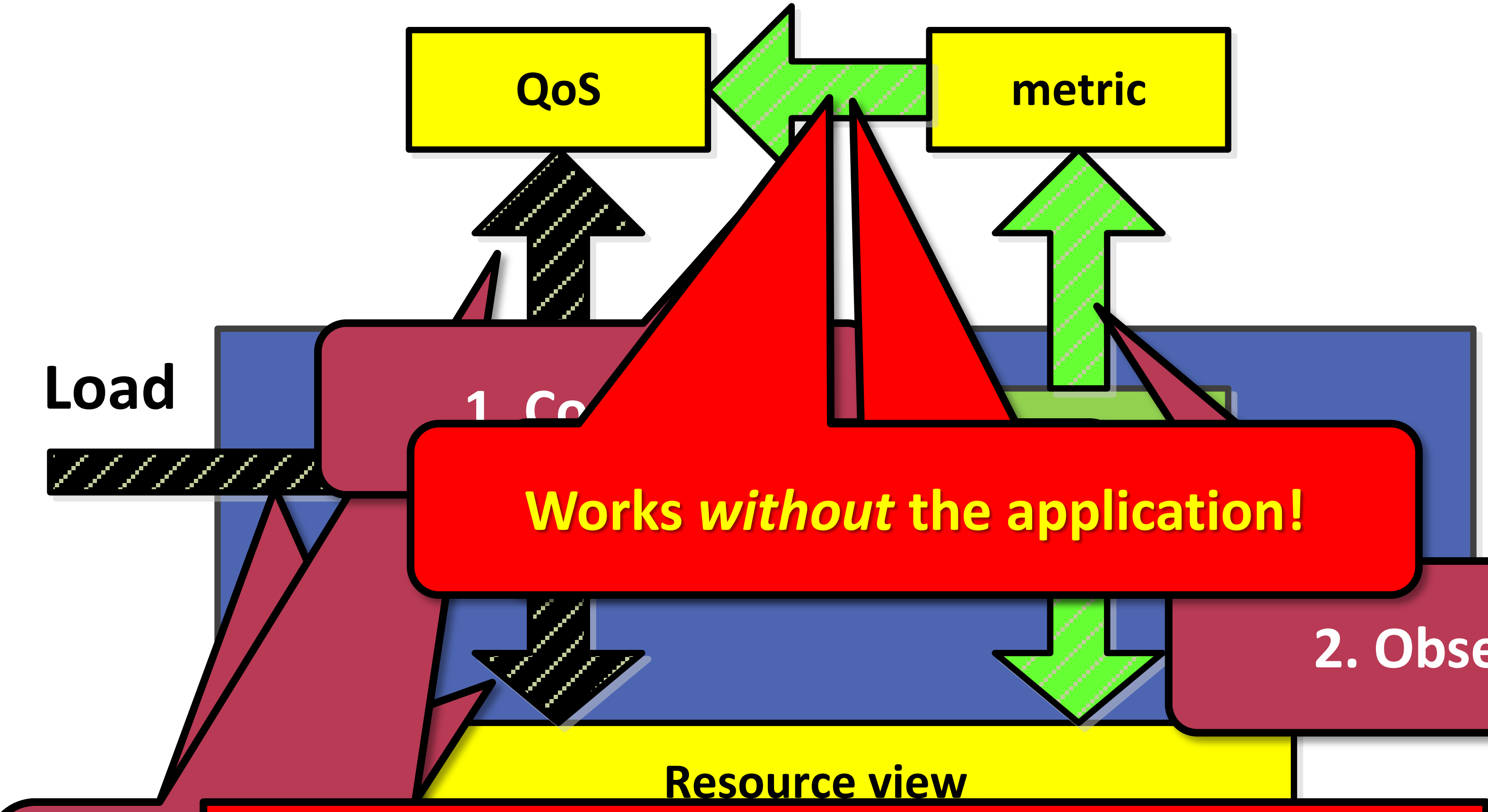
Not trivially feasible...
but everything else impossible

- Metric req:
  - same ***interference***-sensitivities as the service
  - same ***resource***-sens
  - representative for ***t***

Example: short computation bursts sampling available CPU for longer computation

- Runtime req:
  - ***Non-intrusiveness*** (instead of saturation)
  - ***Long running*** (rare events)
  - ***(Low specific impact on service)***

Indirect platform & QoS observability



1. Co

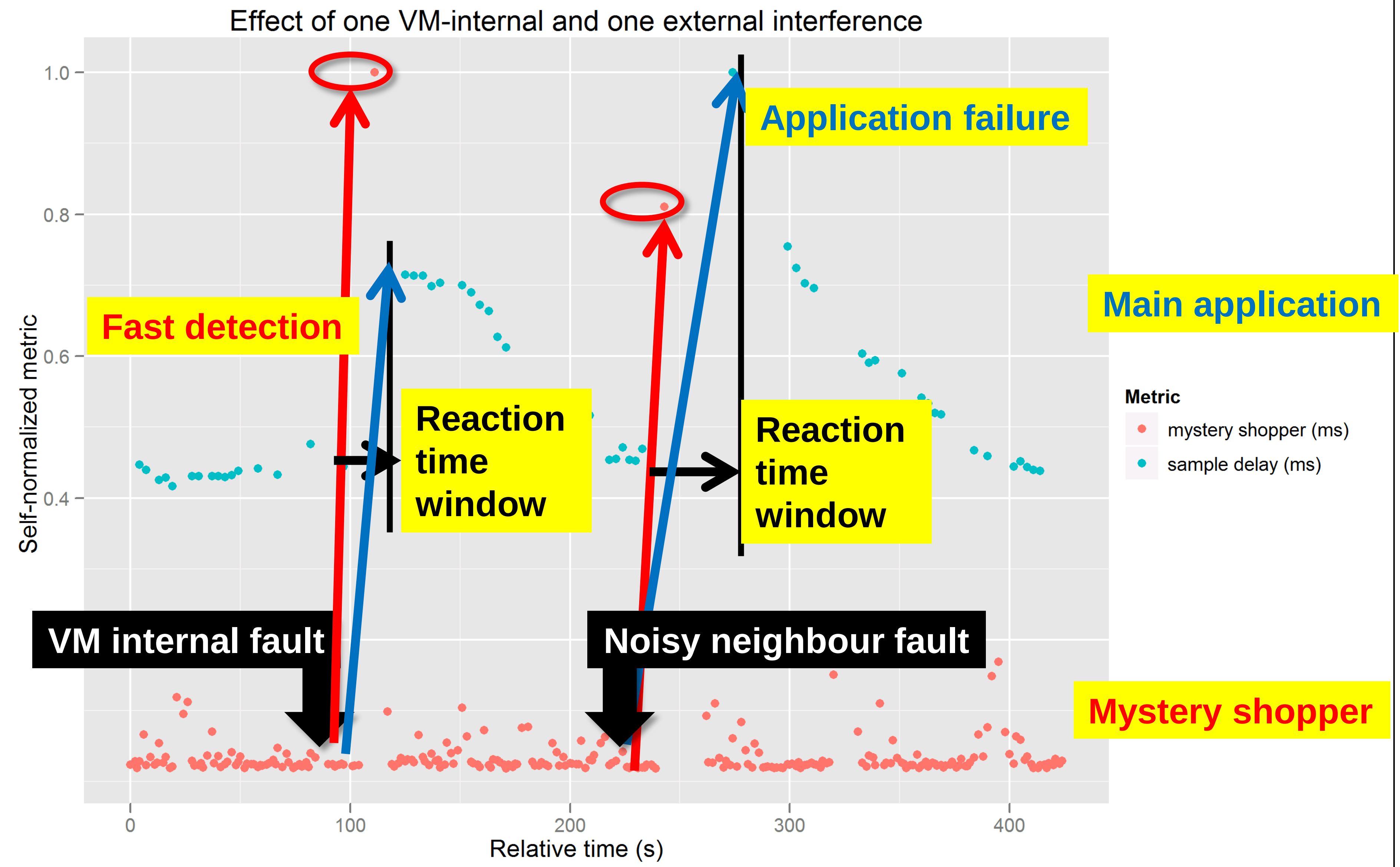
Works without the application!

2. Observe

Observability problems (if present) bypassed

deploy, run/test, observe, analyze

Mystery shopper & service QoS



Fast detection

Application failure

Main application

Reaction time window

Reaction time window

VM internal fault

Noisy neighbour fault

Mystery shopper

Summary

- Technical
  - SLA coverage needed for all aspects
  - Missing guarantees can be (somewhat) compensated
    - Cheap computing power -> redundancy
    - „Double” autonomic computing
      - Cloud level – provider
      - Application level – user
- Methodology
  - Visual exploratory data analysis for insight
  - Algorithmic analysis for proofs and evaluation
  - Fault-tolerance design patterns revisited
    - Cheap redundancy in the cloud