ZABBIX

Introduction to Zabbix

The Universal Open Source Enterprise Level Monitoring Solution

Who am I?

Alexei Vladishev

Creator of Zabbix

CEO, Architect and Product Manager

Twitter: @avladishev

Email: alex@zabbix.com

About Zabbix team

Zabbix Team: 40 members working full-time

Offices in Riga (Headquarters), Tokyo and New-York

Business model is based on providing services:
technical support, trainings, development, turn-key solutions

My plan

- Introduction to Zabbix
- Problem detection
- Why Zabbix?

ZABBIX

Zabbix is a universal open source enterprise level monitoring solution

Zabbix is a universal open source enterprise level **monitoring solution**

ZABBIX

Zabbix is a **universal** open source enterprise level monitoring solution

ZABBIX

Zabbix is a universal **open source** enterprise level monitoring solution

ZABBIX

Zabbix is a universal open source **enterprise level** monitoring solution

Zabbix Is Open Source and Free!

Use all enterprise features of Zabbix

Monitor 100.000s of devices

Collect TBs of history data per day

Zabbix Architecture

Data collection

Analysis

Data collection

Zabbix server

History

Data collection

Analysis

Zabbix server

History

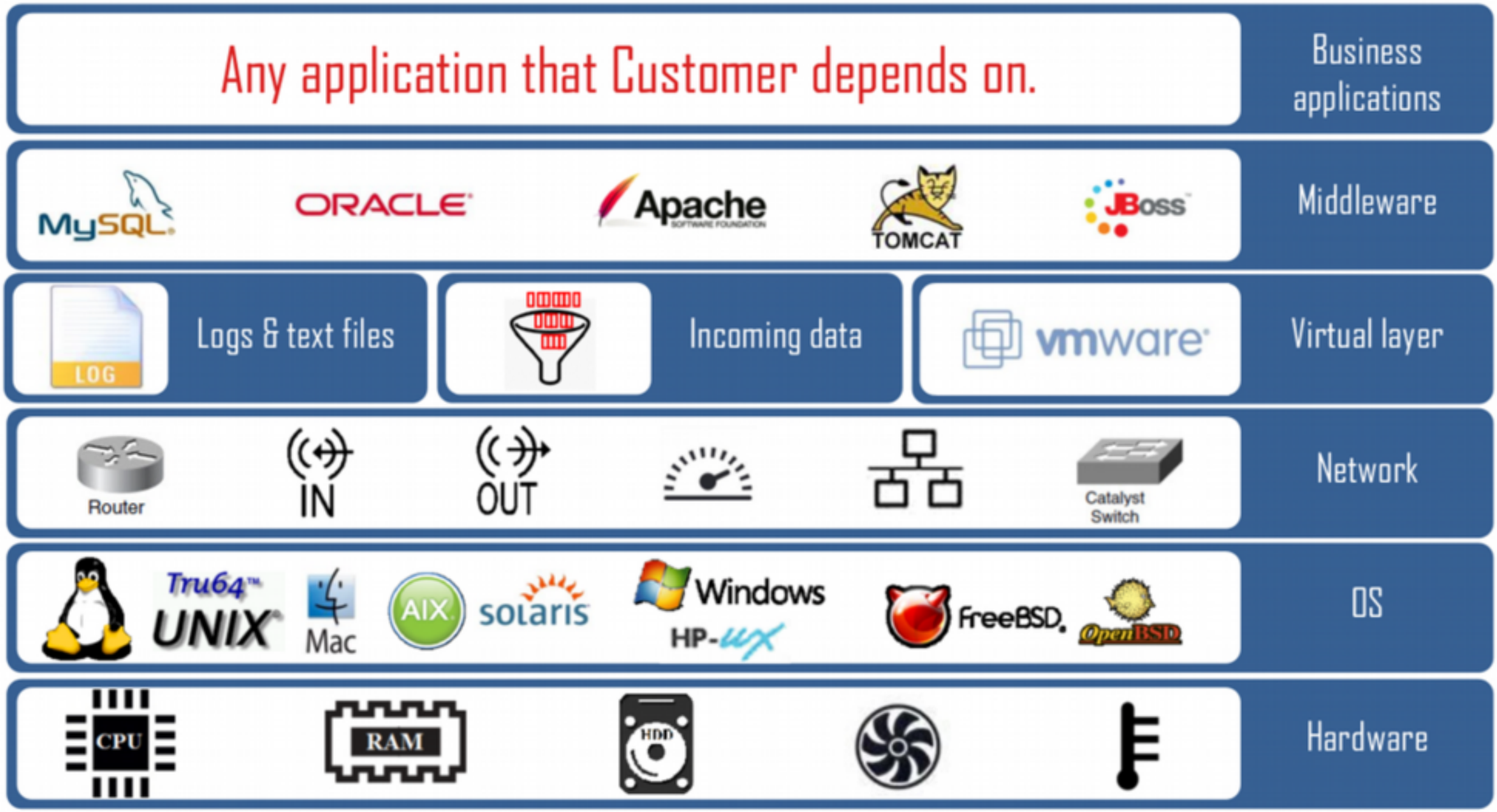
Alerts & Automatic actions

Visualization

ZABBIX

Data collection

All levels of infrastructure



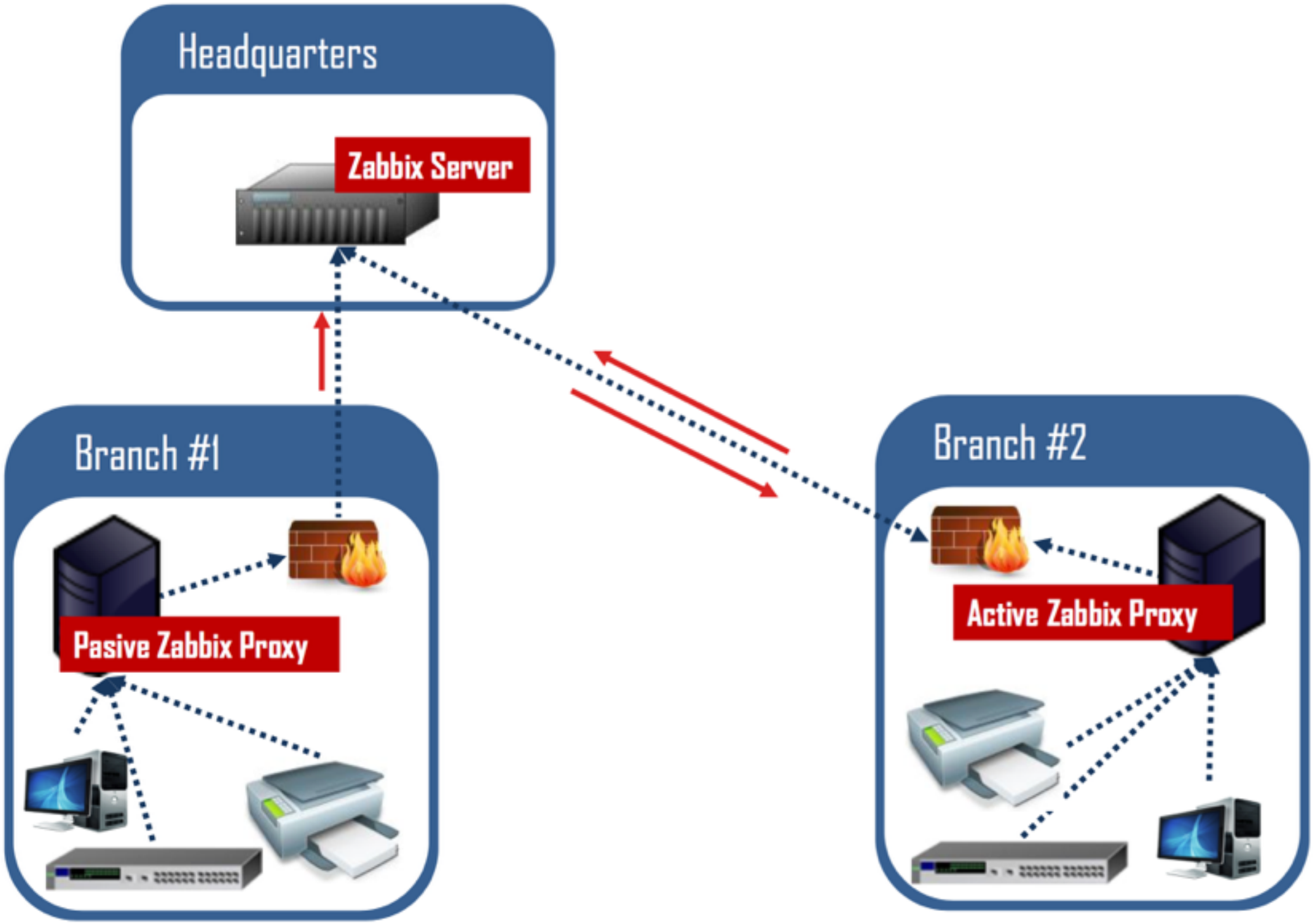
All platforms and OS

All platforms

Most of vendors

Thousands of devices

Distributed environment



Methods of data collection

Pull

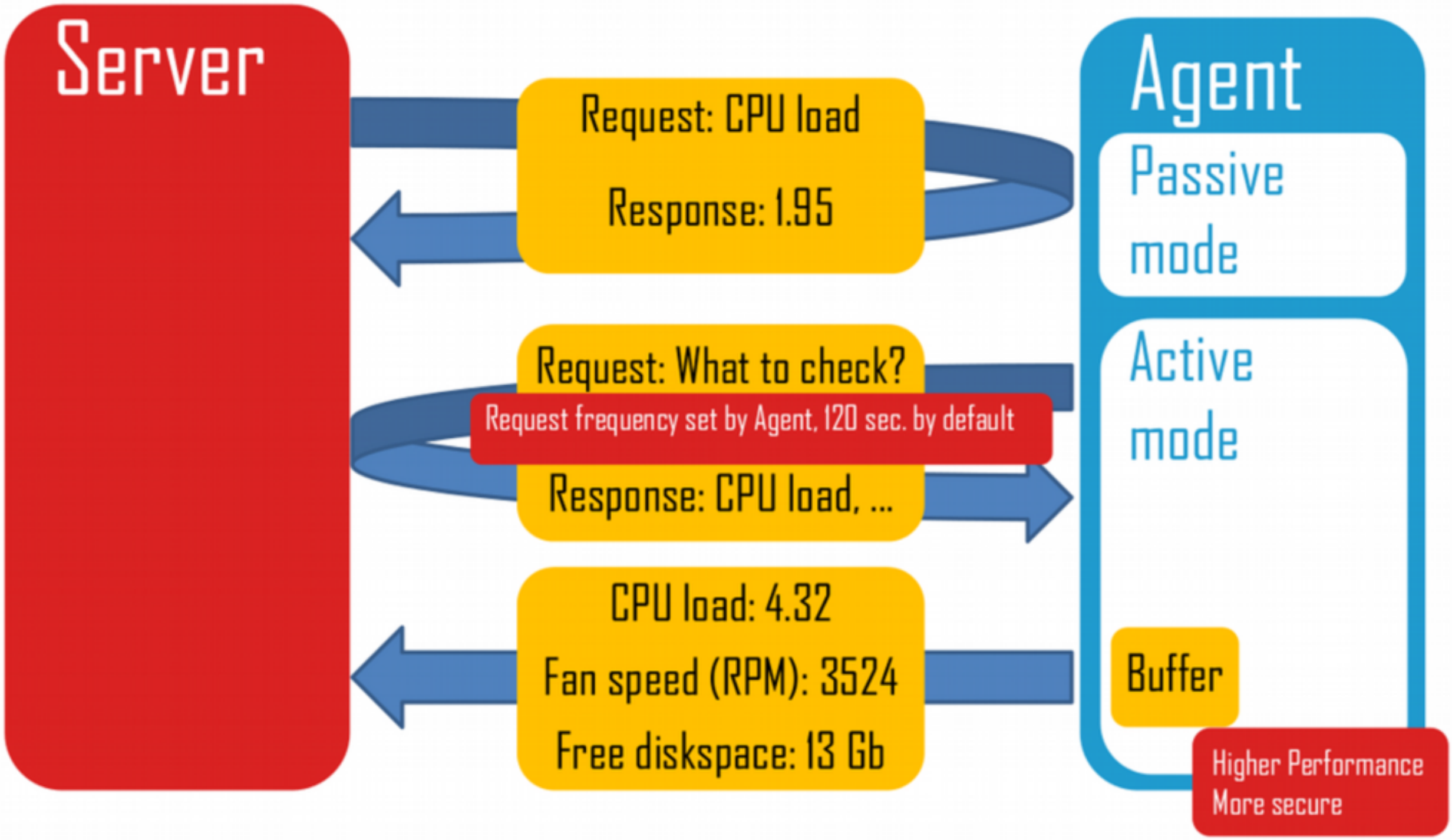
- Service checks: VMWare, HTTP, SSH, IMAP, NTP and other
- Passive agent
- Script execution via SSH and Telnet

Push

- Active agent
- Zabbix Trapper and SNMP Traps
- Monitoring of log files and event logs on Windows

And any other custom checks!

Active vs Passive



How often execute checks?

Every N seconds, down to 1 check per second

- Zabbix will evenly distribute checks

Different frequency in different time periods

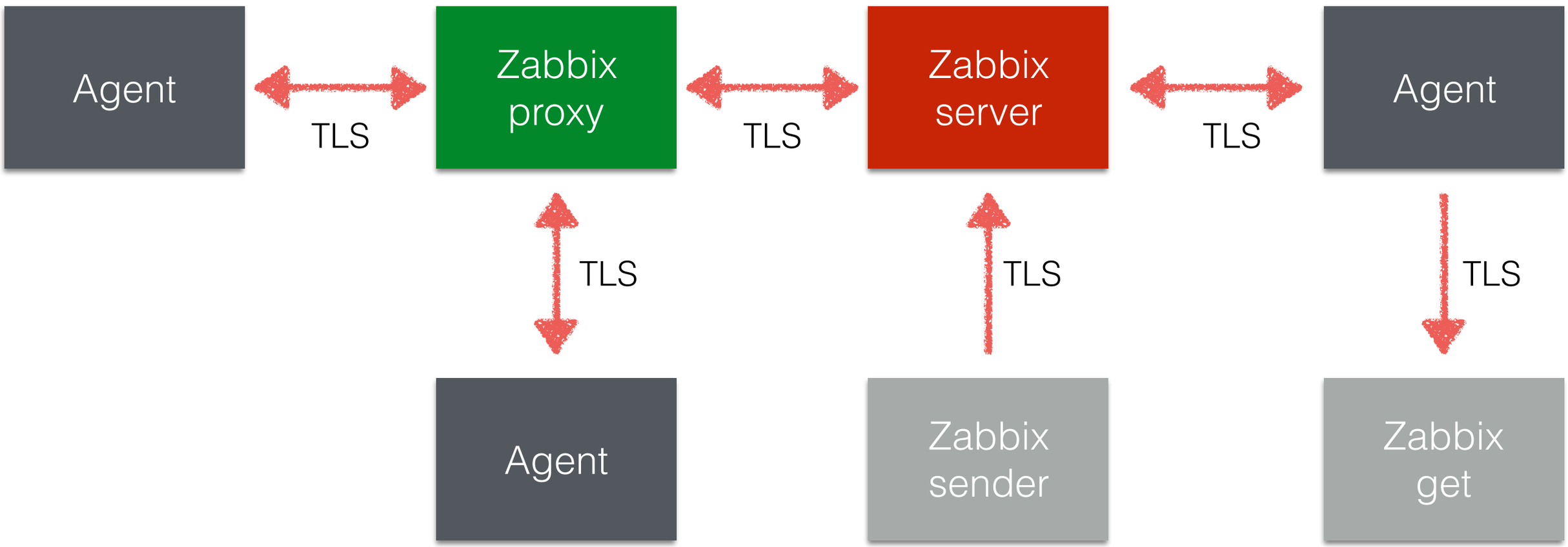
- Every X seconds in working time
- Every Y second in weekend

At a specific time

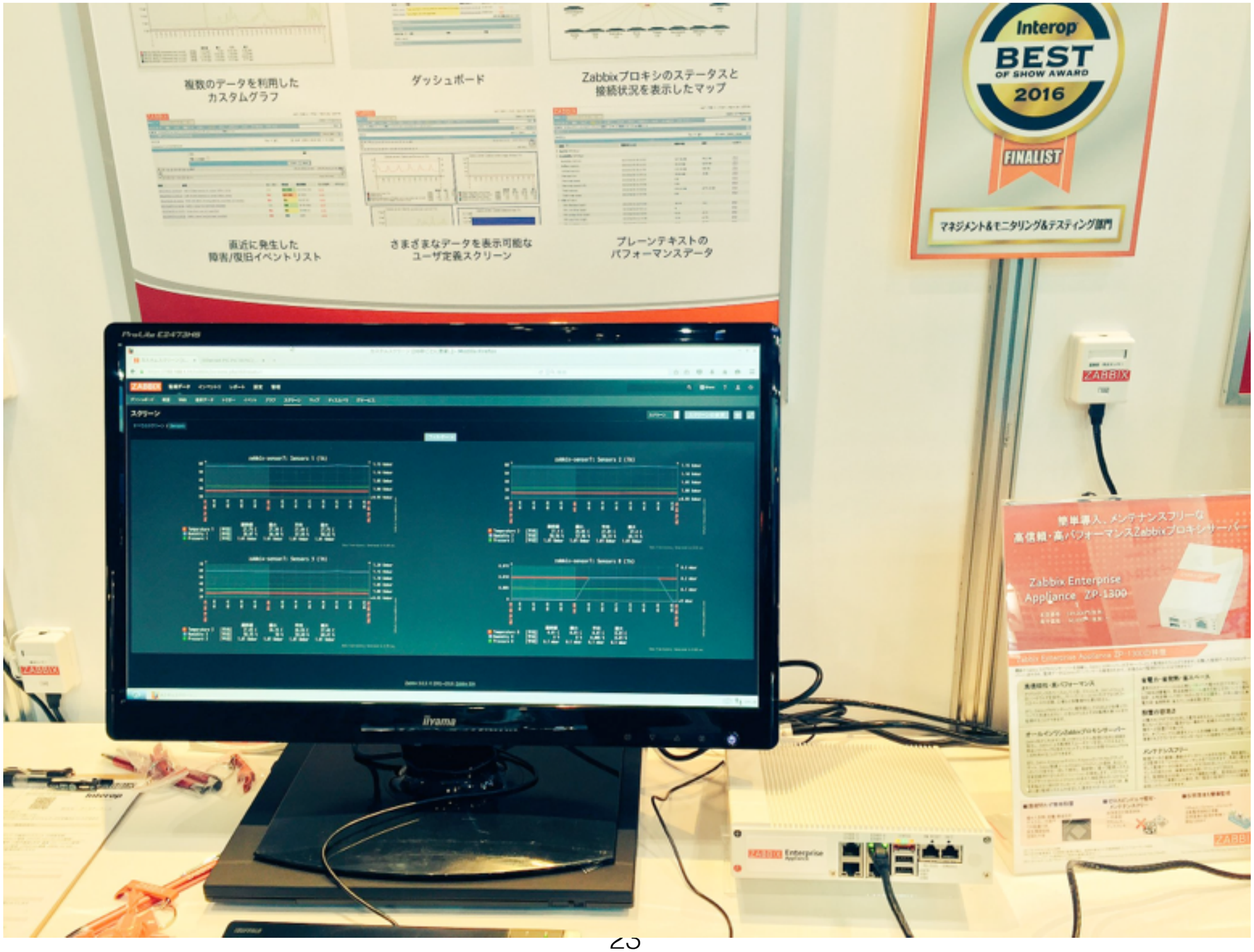
- Ready for business checks
- Every hour starting from 9:00 at working hours (9:00, 10:00, …, 18:00)

Encryption and authentication

Strong encryption and authentication for all components based on TLS v1.2

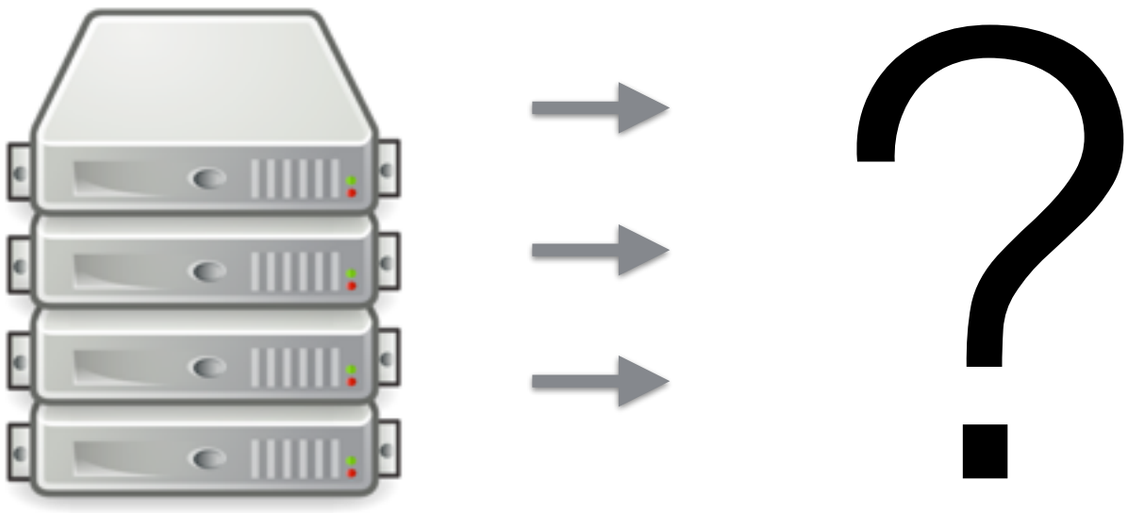


Sensors & small devices



Problem detection

How to detect problems in this data flow?



Data collection

ZABBIX

Triggers!

ZABBIX

Trigger is problem definition

ZABBIX

{server:system.cpu.load.last()} > 5

Triggers

Example

{server:system.cpu.load.last()} > 5

Operators

- + / * < > = <> <= >= or and not

Functions

min max avg last count date time diff regexp and much more!

Analyse everything: any metric and any hosts

{node1:system.cpu.load.avg(10m)} > 5 and
{node2:system.cpu.load.avg(10m)} > 5 and
{nodes:tps.min(30m)} > 5000

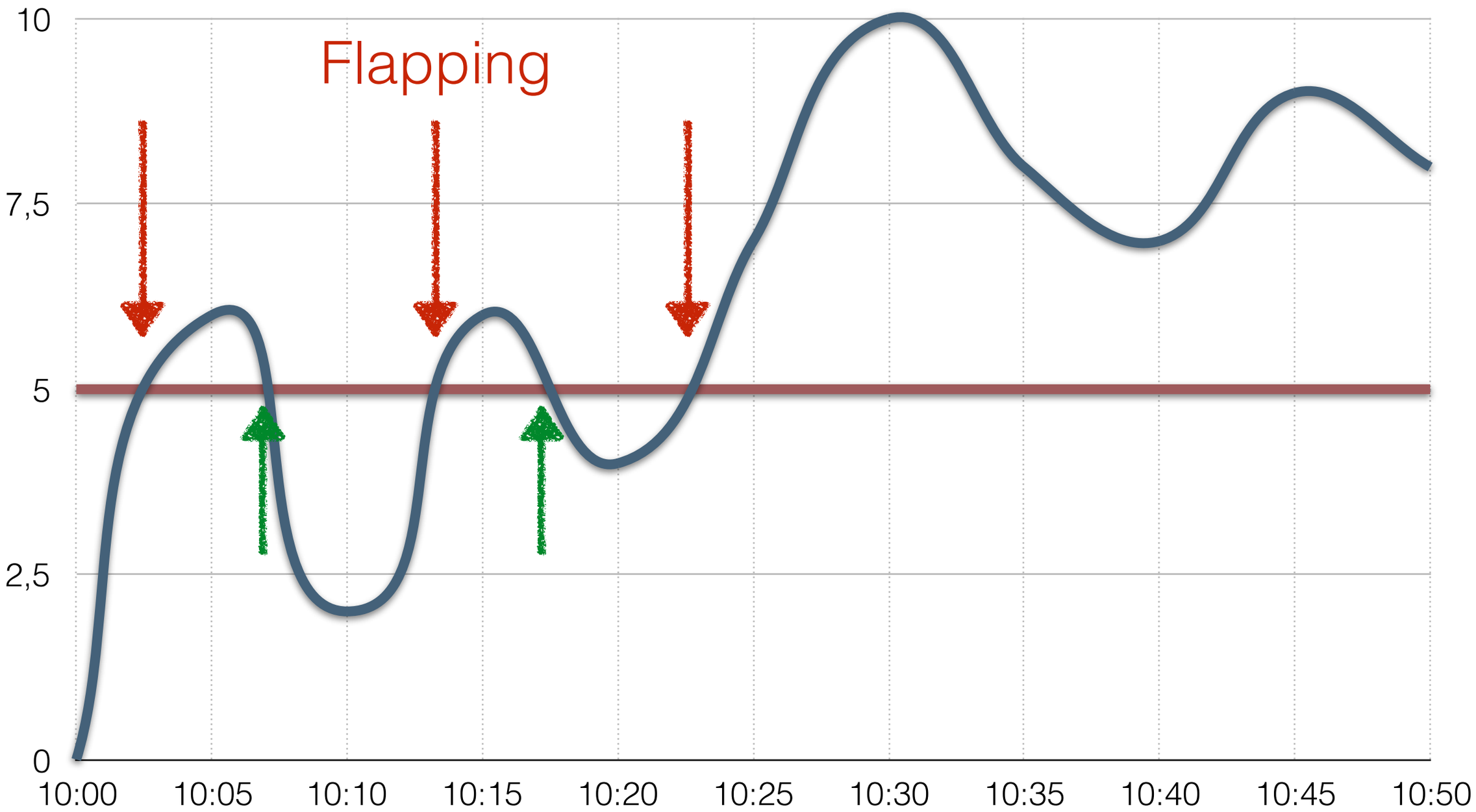
Art of problem detection

Junior level

Performance

{server:system.cpu.load.last()} > 5

False positives



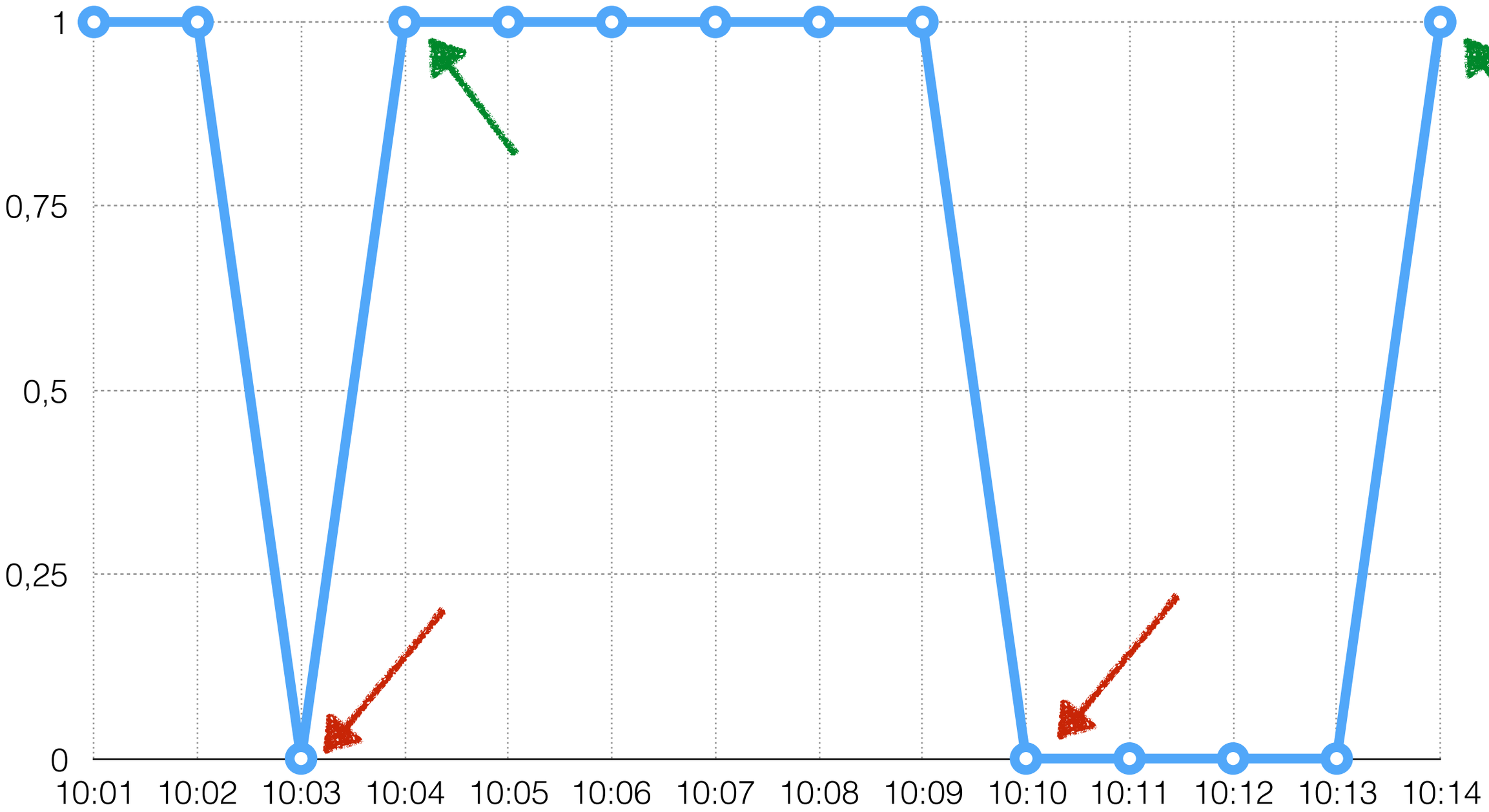
{server:system.cpu.load.last()} > 5

Junior level

Availability

{server:net.tcp.service[http].last()} = 0

Too sensitive



{server:net.tcp.service[http].last()} = 0

Too sensitive leads to false positives

How to get rid of false positives?

Properly define problem conditions and think carefully!

What really means

system is overloaded
running out of disk space
a service is not available

?

Take advantage of history

System performance

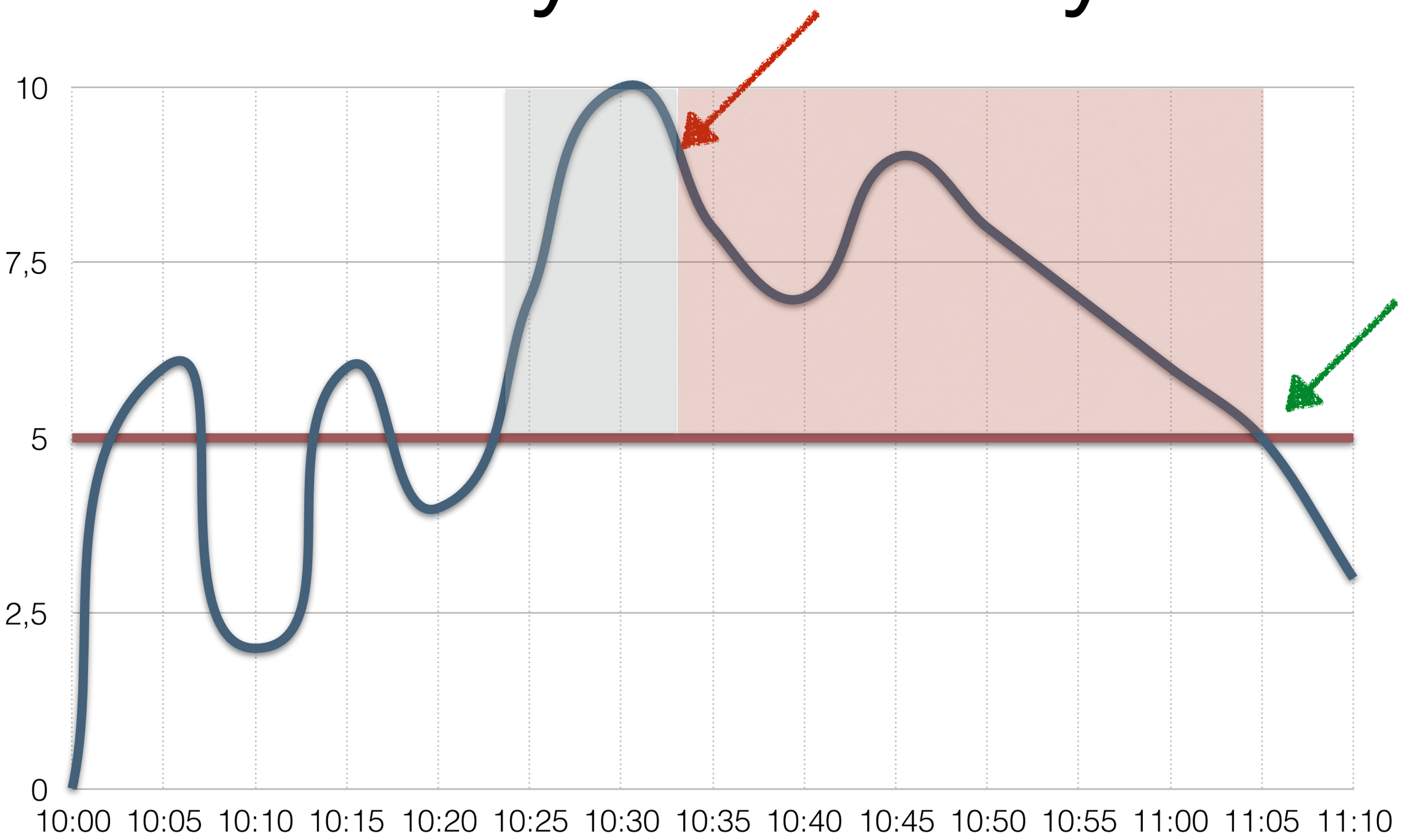
{server:system.cpu.load.min(10m)} > 5

Service availability

{server:net.tcp.service[http].max(5m)} = 0

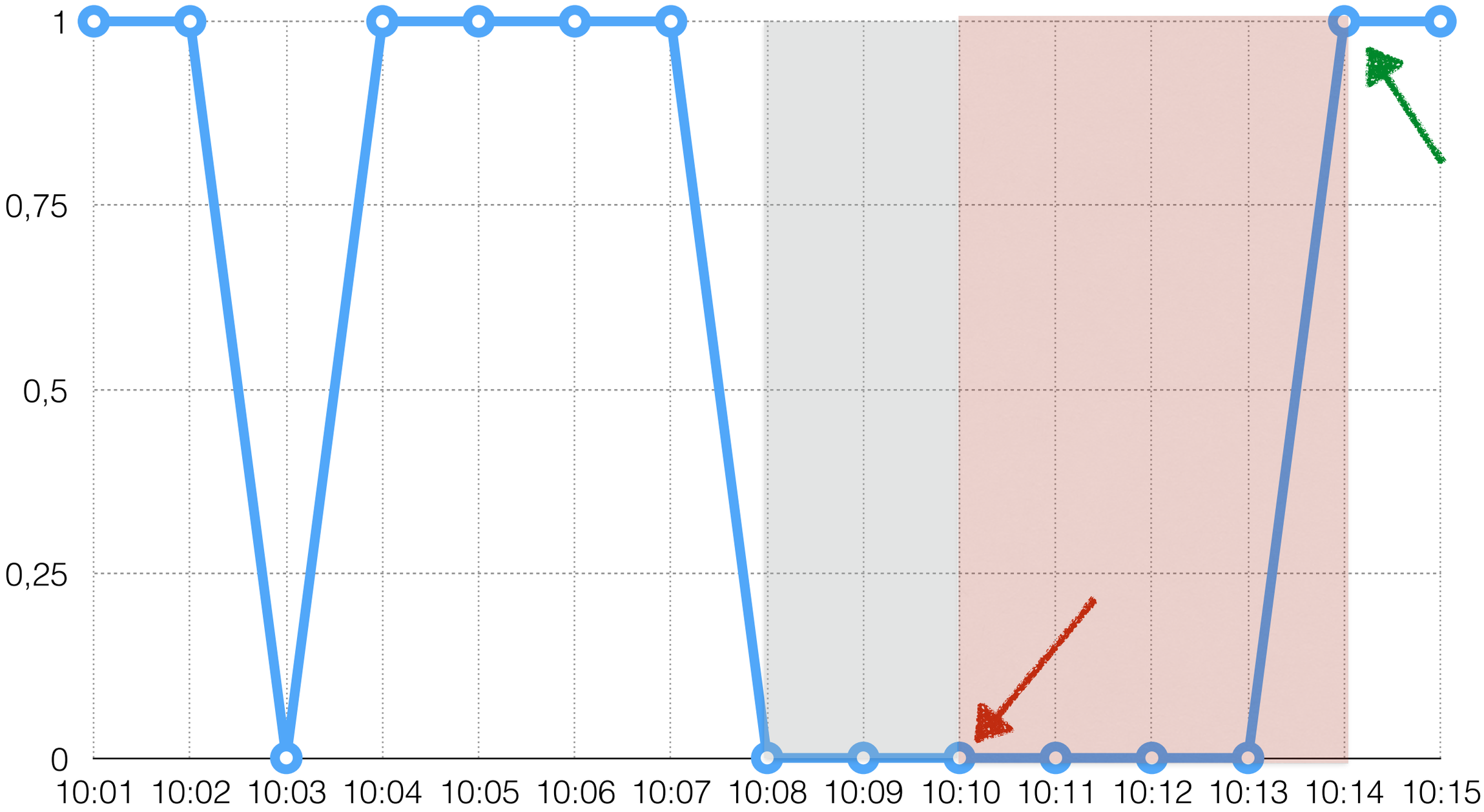
{server:net.tcp.service[http].max(#3)} = 0

Analyse history



{server:system.cpu.load.min(10m)} > 5

Analyse history



{server:net.tcp.service[http].max(#3)} = 0

ZABBIX

Problem disappeared

!=

problem is resolved

A few examples

Problem: free disk space < 10%

No problem: free disk space = 10.001% Resolved?

A few examples

Problem: free disk space < 10%
No problem: free disk space = 10.001% Resolved?

Problem: CPU load > 5
No problem: CPU load = 4.99 Resolved?

A few examples

Problem: free disk space < 10%
No problem: free disk space = 10.001% Resolved?

Problem: CPU load > 5
No problem: CPU load = 4.99 Resolved?

Problem: SSH check failed
No problem: SSH is up Resolved?

Different conditions for problem and recovery

Before

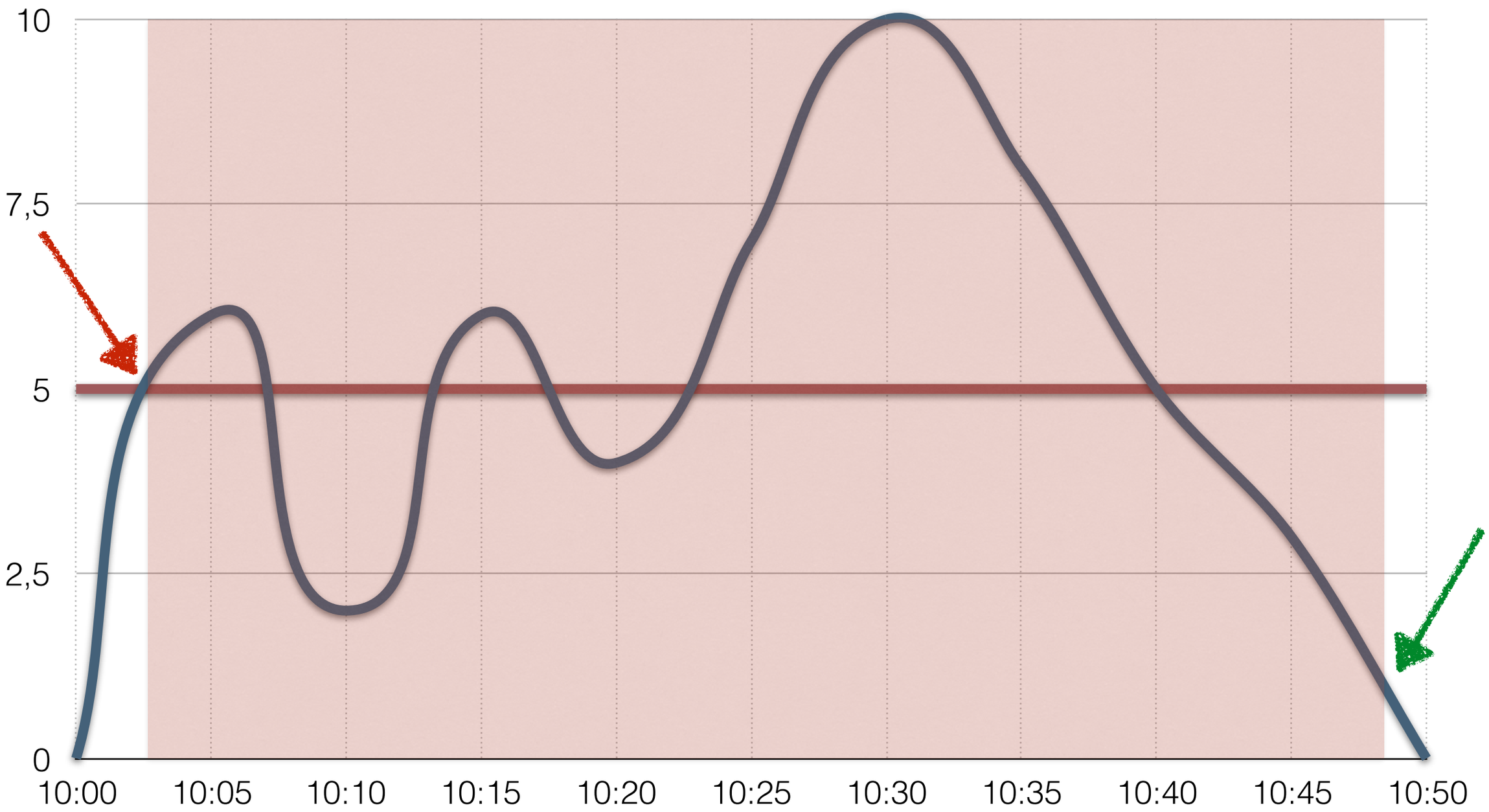
{server:system.cpu.load.last()} > 5

Now

Problem: {server:system.cpu.load.last()}>5)

Recovery: {server:system.cpu.load.last()}<1)

Hysteresis



{server:system.cpu.load.last()} > 5 . {server:system.cpu.load.last()} < 1
47

ZABBIX

No flapping!

Several examples

System is overloaded

Problem: {server:system.cpu.load.min(5m)}>3

Recovery: {server:system.cpu.load.min(2m)}<1

No free disk space on /

Problem: {server:vfs.fs.size[/,pfree].last()}<10

Recovery: {server:vfs.fs.size[/,pfree].min(15m)}>30

SSH server is not available

Problem: {server:net.tcp.service[ssh].max(#3)}=0

Recovery: {server:net.tcp.service[ssh].min(#10)}=1

Typical Use: Problem Detection

… but Zabbix also offers:

Anomaly detection

Problem forecasting

Trend prediction

ZABBIX

Anomalies

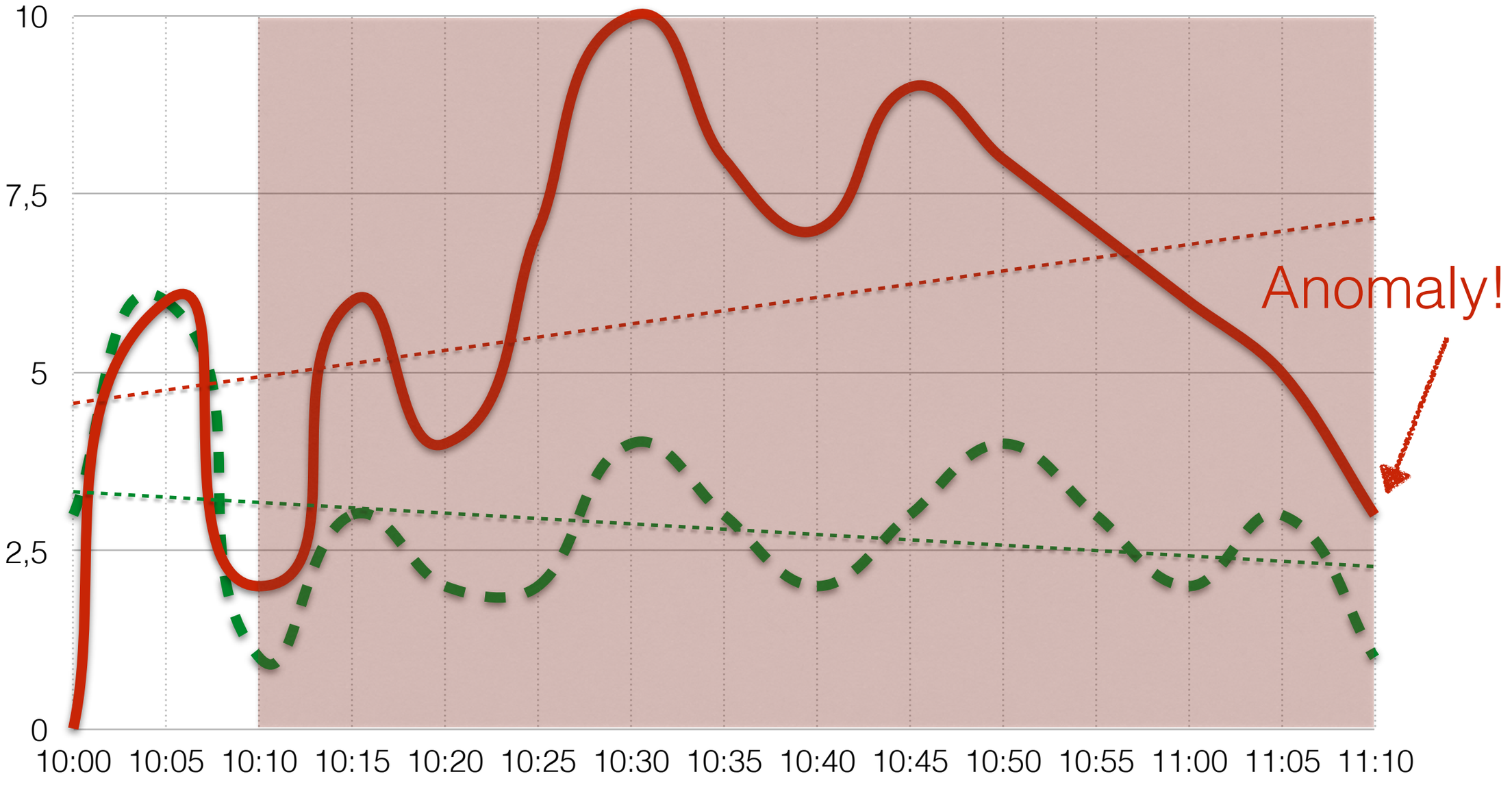
How to detect?

Compare with a norm, where norm is system state in the past.

Average number of transactions per second for the last hour is 2x less than number of transactions per second for the same period week ago

{server:tps.avg(1h)} < 2 * {server:tps.avg(1h,**7d**)}

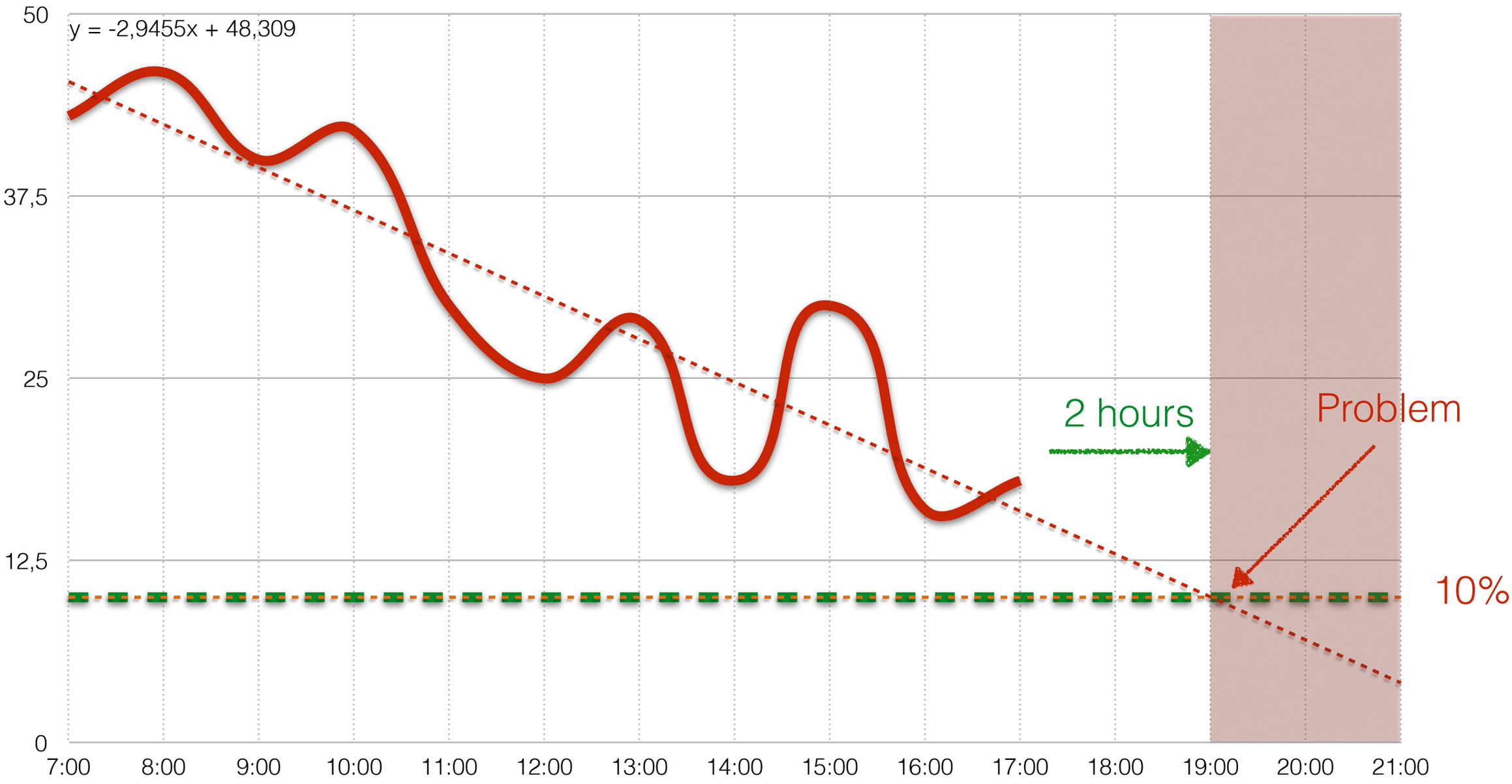
Anomaly detection



Compare current system state with the past

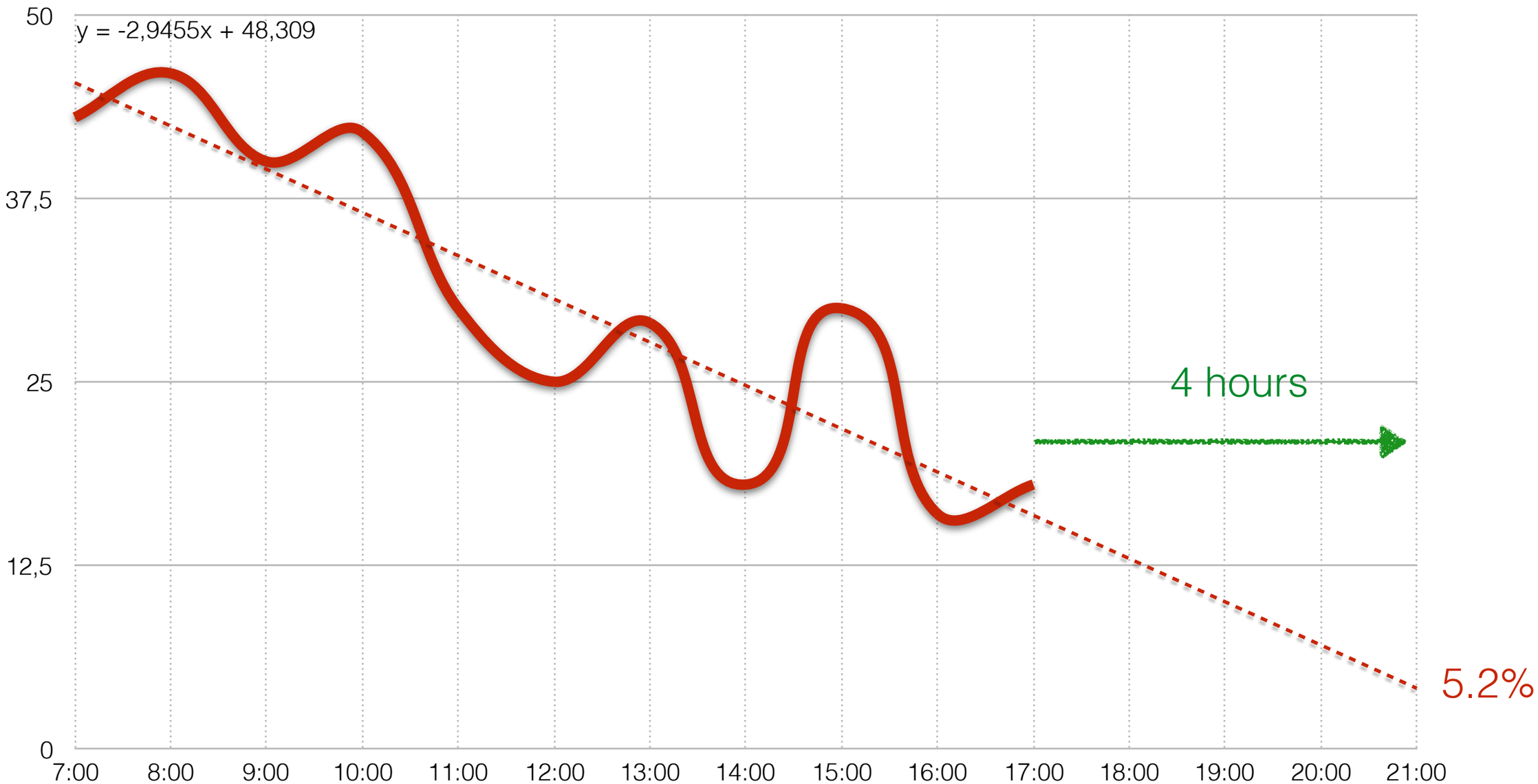
Problem forecasting

Problem Forecasting



Trigger function timeleft()

Trend Prediction



Trigger function forecast()

How Zabbix reacts on problems?

Possible reactions

- Automatic problem resolution
- Sending alerts to user and user group
- Opening tickets in Helpdesk systems
- Unlimited number of possible reactions

Escalate!

- Immediate reaction
- Delayed reaction
- Notification if automatic action failed
- Repeated notifications
- Escalation to a new level

Escalate!

- Immediate reaction
- Delayed reaction
- Notification if automatic action failed
- Repeated notifications
- Escalation to a new level

Example

0 min — Repeated Email

5 min — SMS and ticket

10 min — Service restart

15 min — SMS to manager

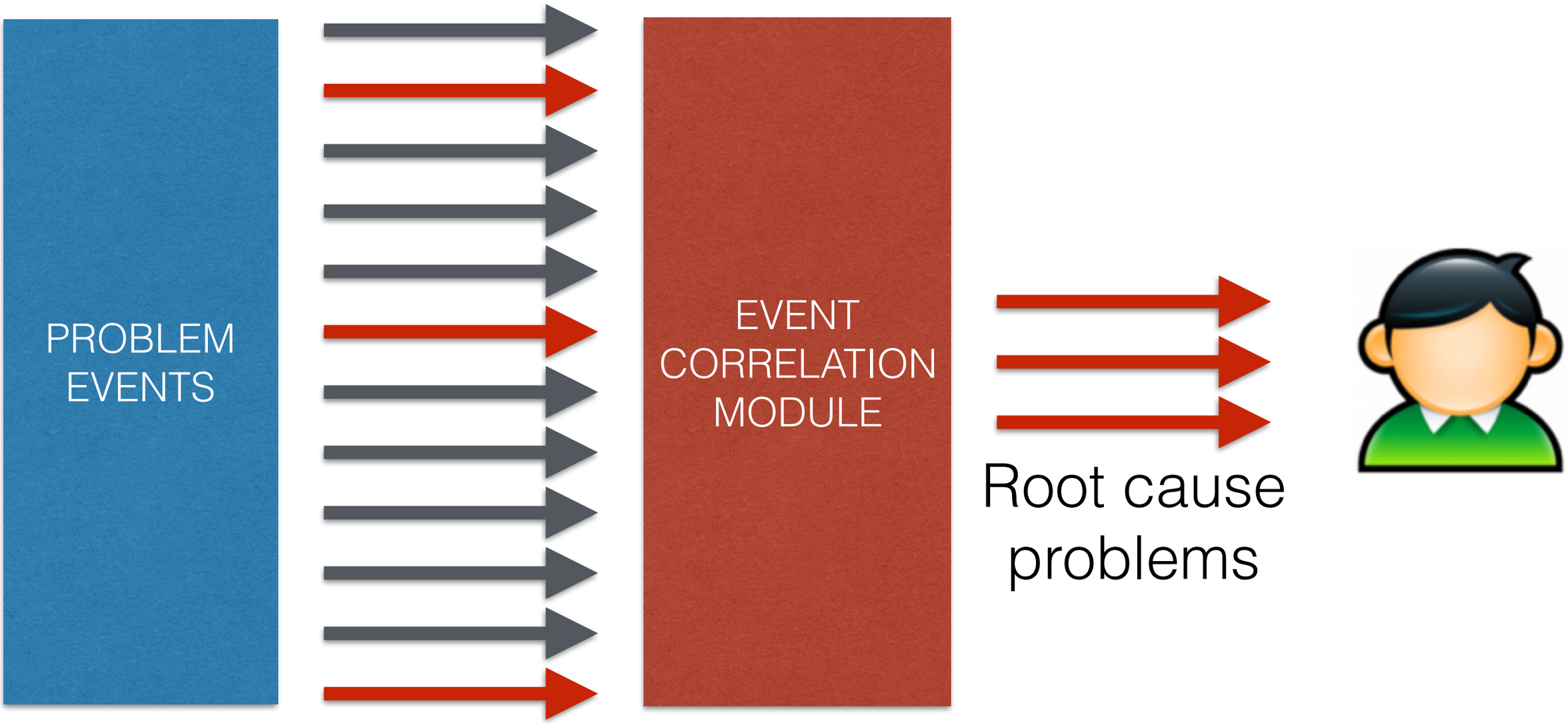
20 min …

Critical problem

ZABBIX

What if there are too many problems?

Event correlation



Deduplication, event filtering, root cause analysis, etc

ZABBIX

Scalability

Core internet services: DNS, DNSSEC, WHOIS

Management of TLDs, domain registration and registrars

Billions of users

ZABBIX

60.000 hosts

ZABBIX

60.000 hosts

2.000.000 metrics
20.000.000 triggers
6TB history
40 locations, proxy per location

Zabbix performance (avg):
21.000 checks per second

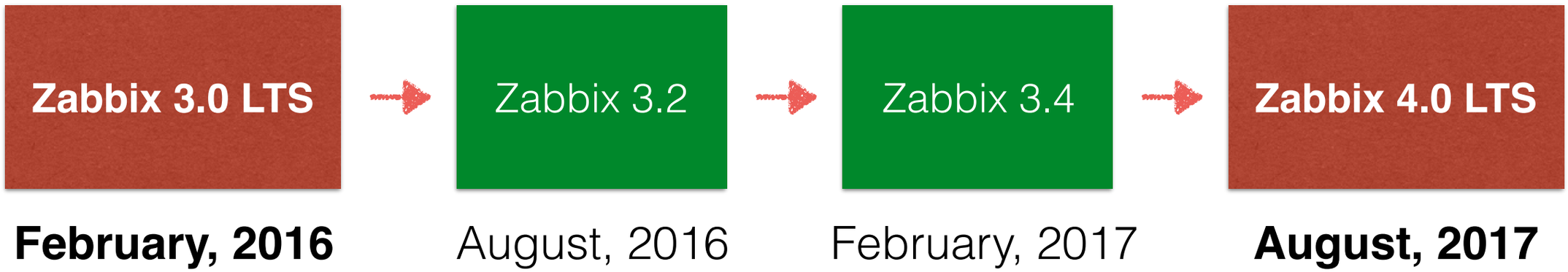
ICANN

Zabbix Lifecycle

Lifecycle

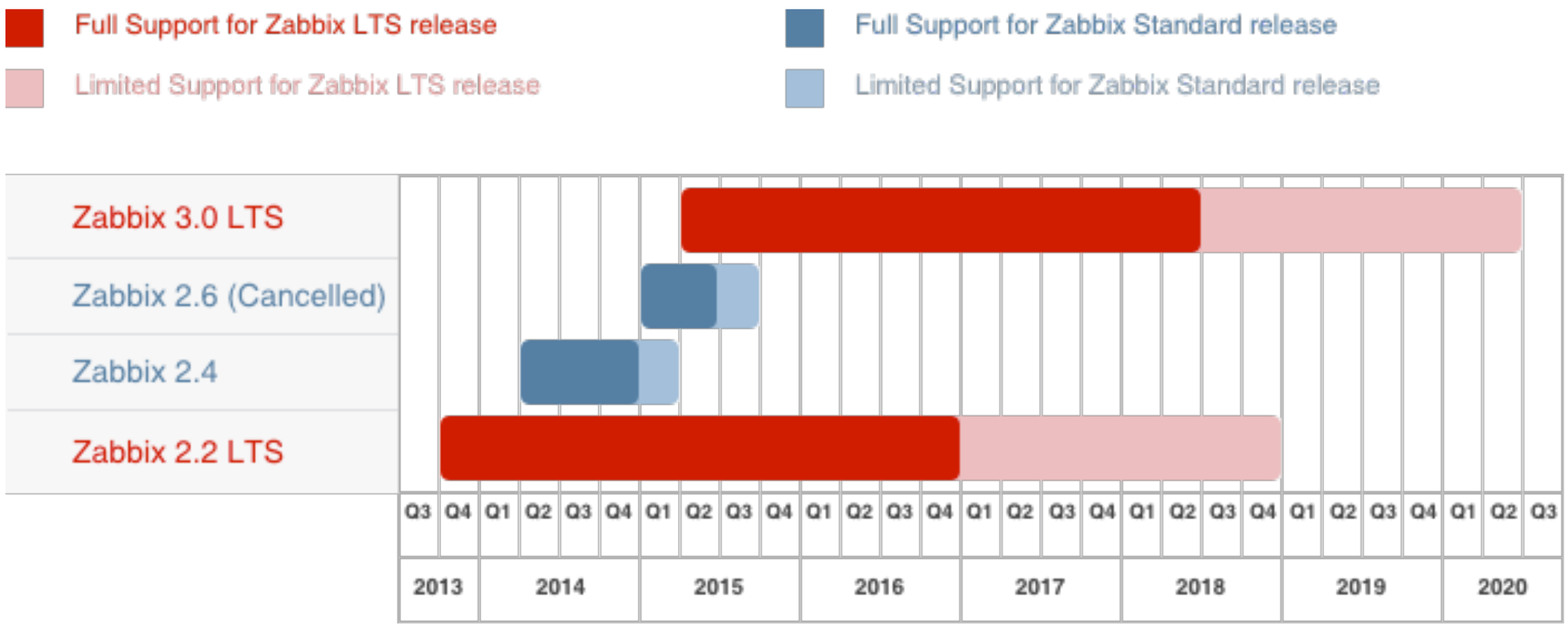
One major release every 6 months

One **LTS** release every 1.5 years



Support policy

LTS releases: 5 years
Normal releases: till next release + 1 month



What users like in Zabbix?

All-in-one solution

Data collection	Agent based monitoring	Scalability	Centralized management	User permissions
Trend prediction	Service checks	Visualization	Maintenance	Integration with AD, OpenLDAP
Problem detection	Encryption	Discovery	Trigger dependencies	WEB front-end
Automatic actions	Anomaly detection	Event correlation	IoT and embedded	All OSes
Distributed monitoring	Zabbix API	Alerting	Escalations	… and more

ZABBIX

Easy to maintain

```
alex@dev:~/zabbix$ ls -lh
total 2.9M
drwxrwxr-x 14 alex alex 4.0K Jun 22 16:55 ui
-rwxrwxr-x  1 alex alex 339K Jun 22 16:55 zabbix_agentd
-rwxrwxr-x  1 alex alex 1.3M Jun 22 16:55 zabbix_proxy
-rwxrwxr-x  1 alex alex 1.4M Jun 22 16:55 zabbix_server
alex@dev:~/zabbix$ du -sh ui
28M     ui
alex@dev:~/zabbix$
```

All components are compatible within one major release

Zabbix Agents are backward compatible since Zabbix 1.0!

Benefits of Zabbix

All in one universal monitoring platform

True Free Software

Easy to adopt, use commercial services if needed

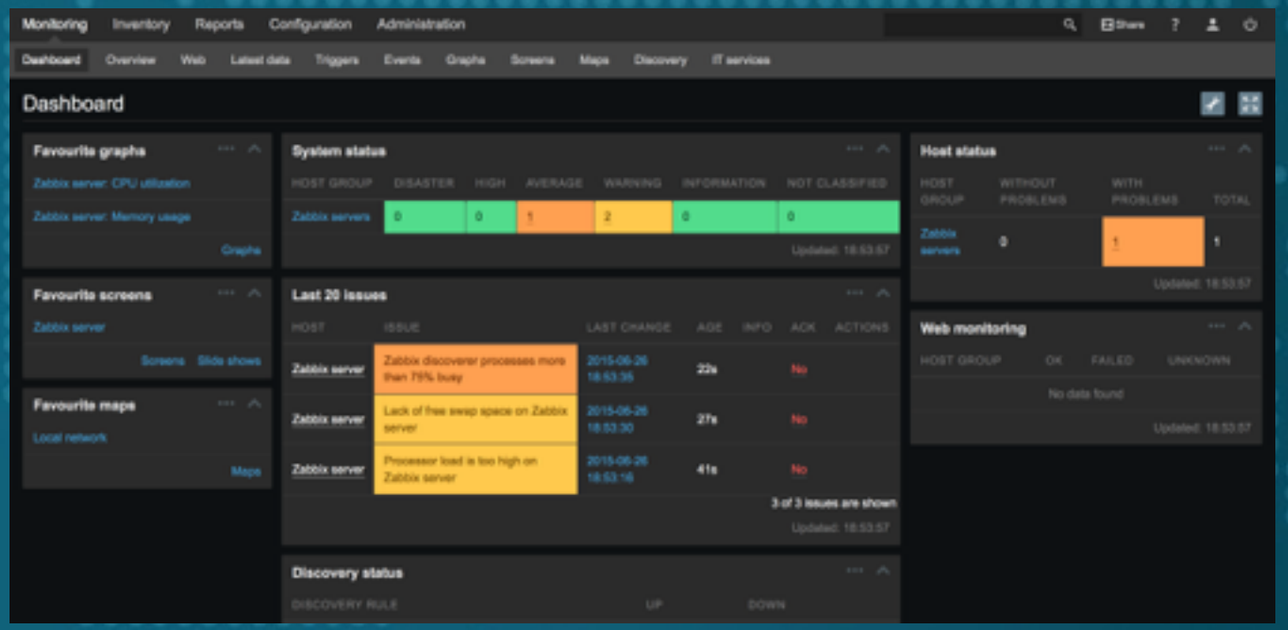
No License Fees

Extremely low TCO

No vendor lock in

ZABBIX

Thank you!



Twitter: **@avladishev**

Email: **alex@zabbix.com**

Learn more at **www.zabbix.com**

The Universal Open Source Enterprise Level Monitoring Solution