



CHECK\_MK

# Event Console Integration

03.05.2018, Sven Panne  
Check\_MK Conference #4

**CONFERENCE**  
MUNICH 2018/5/2-4

**#4**

# Two Kinds of Monitoring



## Status-based

---

- Pushing/pulling states
- “Classic” Check\_MK



## Event-based

---

- Pushing of messages
- Independent of state
- Event Console

**Both kinds are useful!**

# Design Considerations



High frequency  
of incoming  
messages



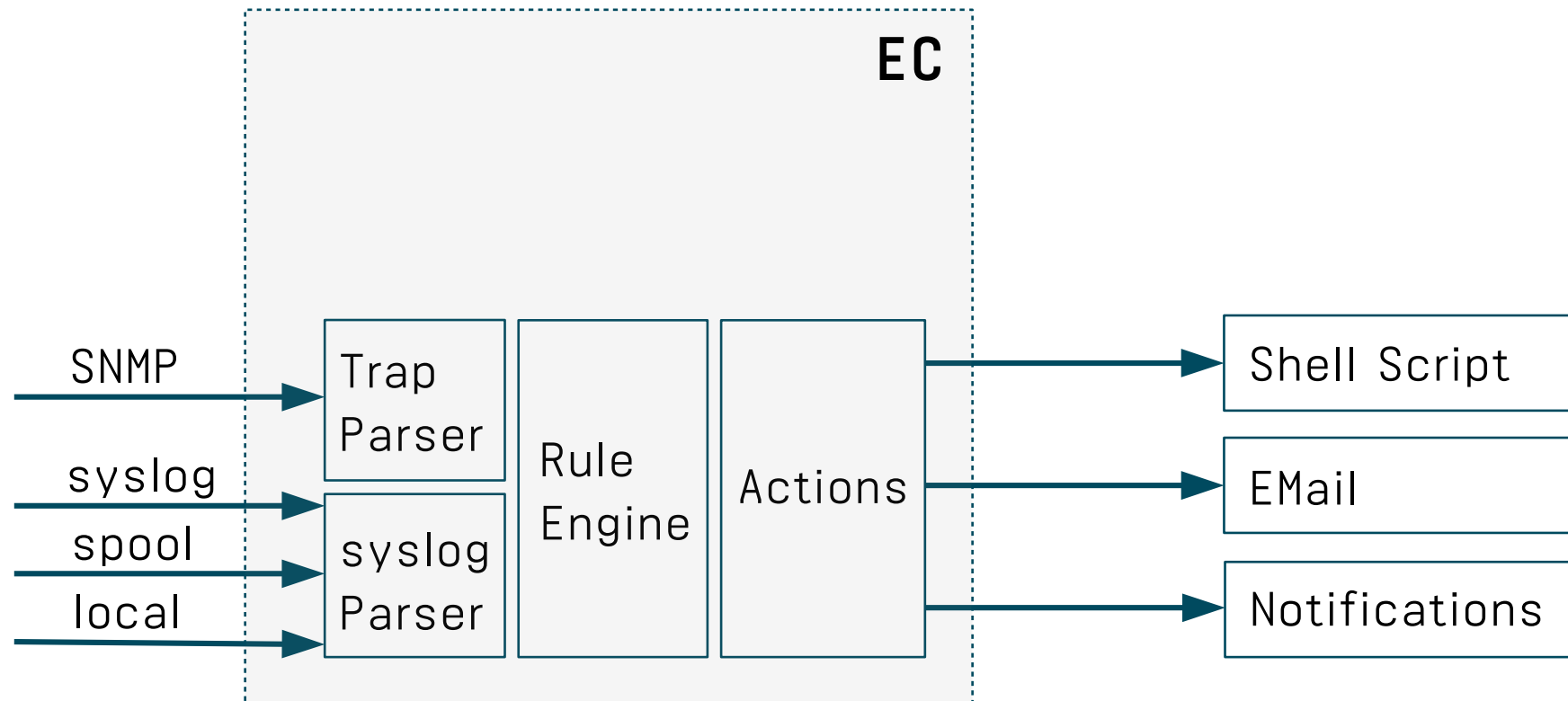
Human-manageable  
amount of  
actual events



Not a  
high-performance  
syslog archive

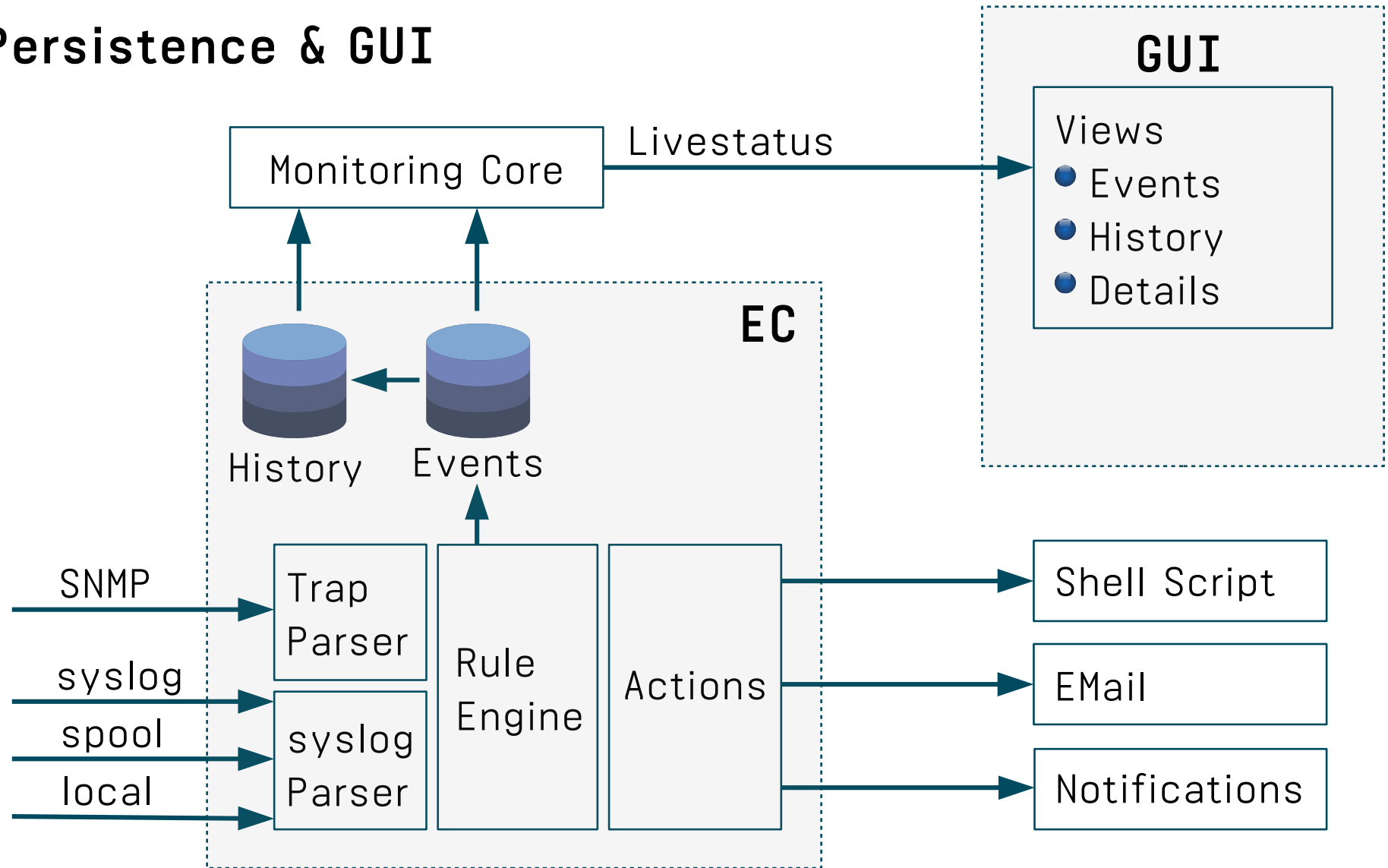
# Architectural Overview I

## Basic Pipeline



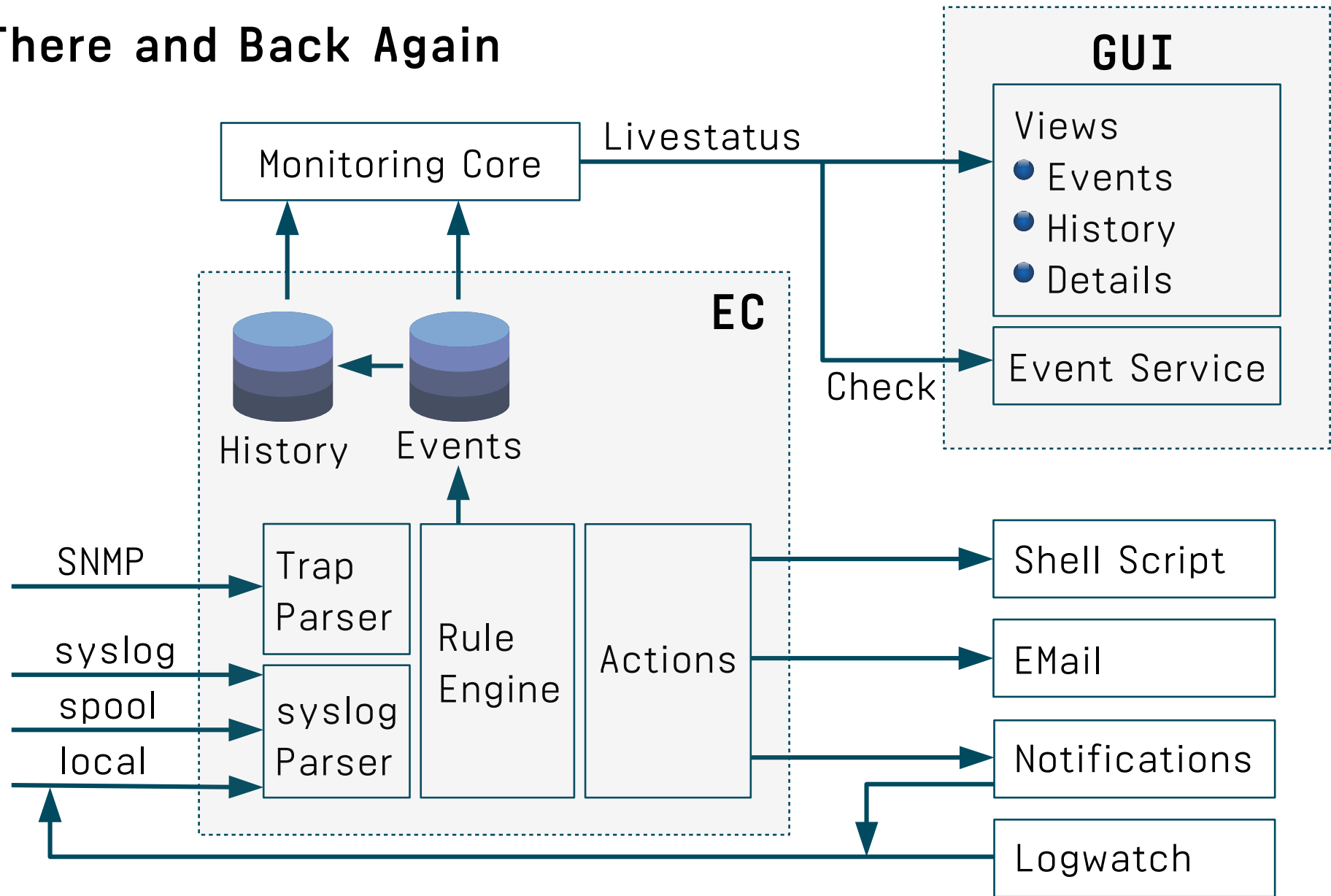
# Architectural Overview II

## Persistence & GUI



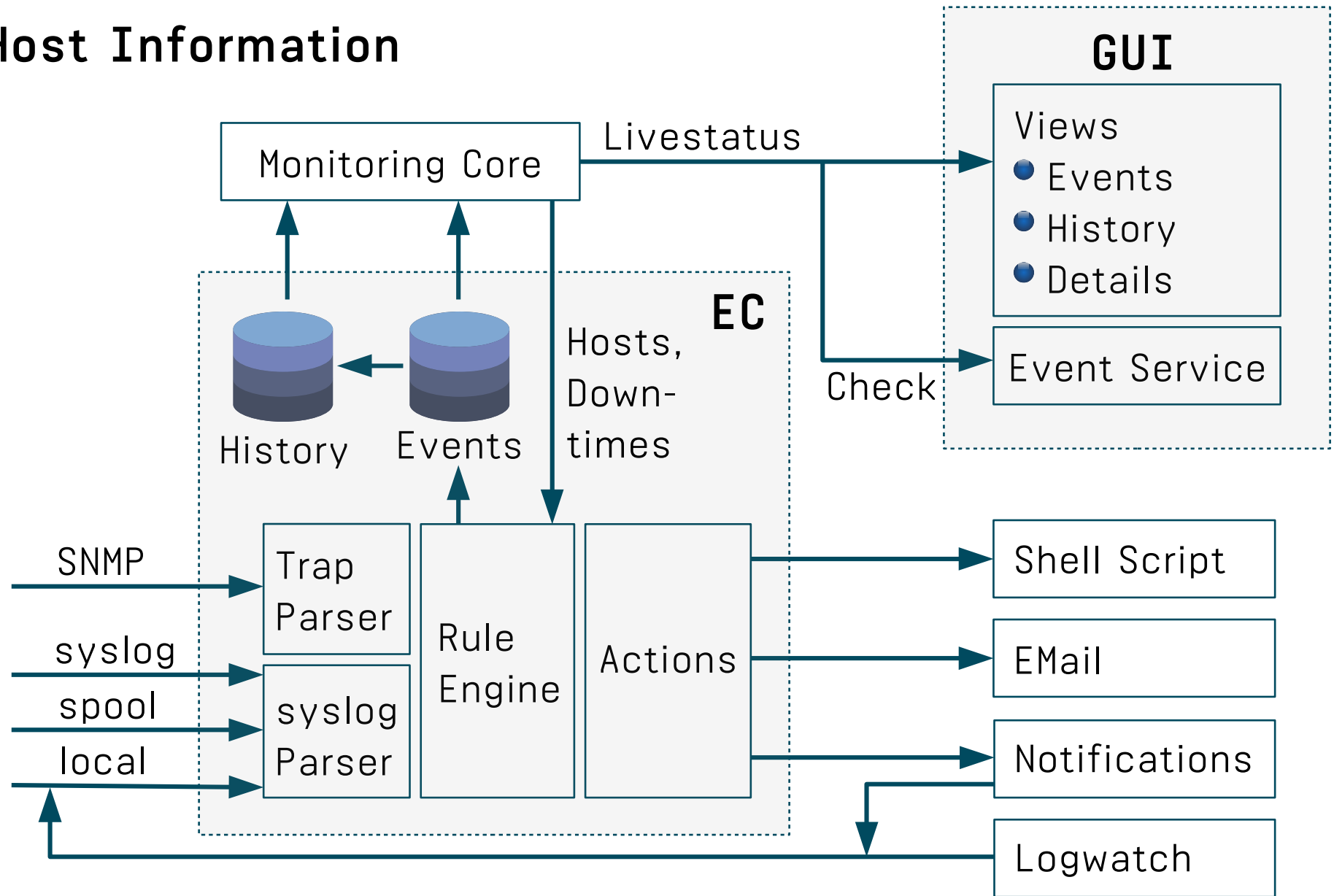
# Architectural Overview III

## There and Back Again



# Architectural Overview IV

## Host Information



# Events are useful, even during host downtimes



**Still generated**, but marked specially

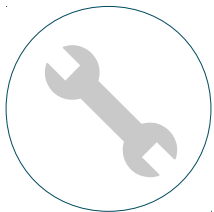
**Automatically archived** after downtime

**Tactical overview** considers them “handled”

**Actions** can be skipped



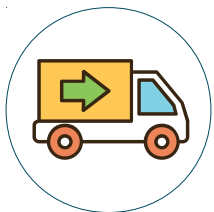
# Further Improvements



Finer configuration of limits



More powerful rewriting



Export of rule packs via MKPs





CHECK\_MK

**CONFERENCE**

MUNICH 2018/5/2-4

**#4**



CHECK\_MK

# Background jobs

03.05.2018, Tom Bärwinkel  
Check\_MK Conference #4

**CONFERENCE**  
MUNICH 2018/5/2-4

**#4**

# Examples of long running processes



Create Report



Bake Agents



Host renaming

# Problem – Apache process timeouts

**Internal error:** Your request timed out after 110 seconds. This issue may be related to a local configuration problem or a with a too large number of objects. But if you think this issue is a bug, please send a crash report.

An internal error occurred while processing your request. You can report this issue to the Check\_MK team to help fixing this is below for reporting.

## ▼ Crash Report

Name .....

tb

Email Address .....

Submit Report

## Crash Report

Crash Type	gui
Time	2018-04-20 16:15:18
Operating System	artful

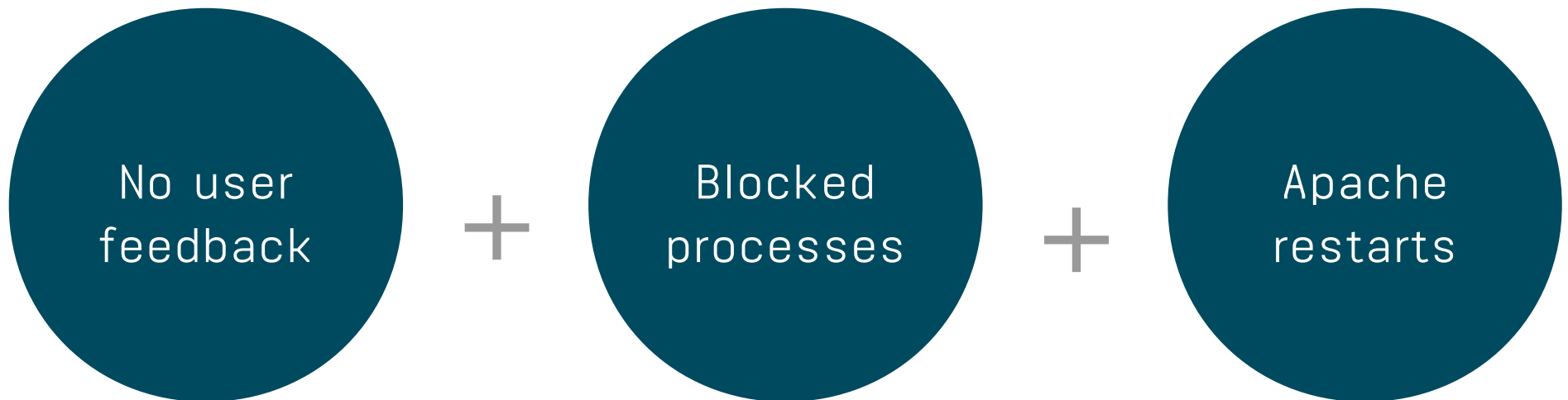


# The usual workaround

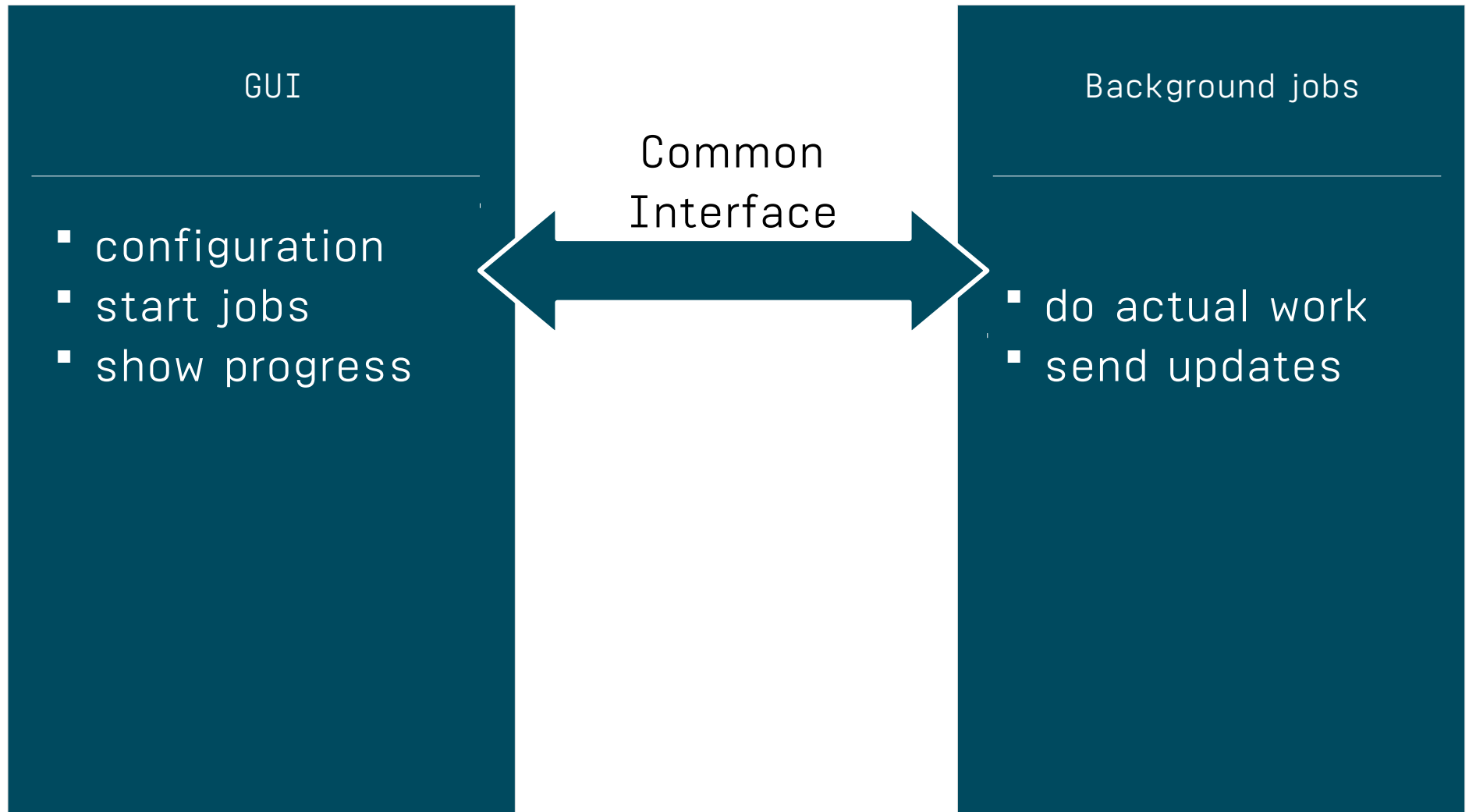
Let's just increase the timeout



# Still a problem - What's the status?



# Solution - separation of concerns





# How it looks in practice - Reports

Create new report schedule entry cmkadmin (admin) 11:08

▼ General Options

Unique ID .....

Report to create ..... BI Availability

Schedule title .....

Activation ..... ☐ do not create this report

**Generate report in background job ..... ☒ run as background job**

Period ..... Every day ▼

Time of day to create report at ..... 00:00

Time range to create report for ..... Last month ▼

Owner ..... cmkadmin - cmkadmin ▼

Action ..... Send via Email ▼

Email-Addresses to mail PDF reports to  
|

Options for report email

# How it looks in practice - Reports

Report Scheduler

1 row cmkadmin (admin) 11:54

New entry

Reports

Stored reports

Report generation started in background job [reporting-1524736450.2](#)

Scheduled reports

Actions	ID	Title	Owner	Report	Action	Time Range	Context	Period	Time	Last Run	Last
	background_job	Background Job	cmkadmin	Report of Host	Send via Email, Email-Addresses to mail PDF reports to: Options for report emails:	Last month	1 Filter settings	Every day	00:00		

Reporting

Actions	Job ID	Job Title	State	Started	Owner	PID	Runtime	Last progress info	Results
	reporting-1524736450.2	Report generation of background_job / Background Job	initialized	2018-04-26 11:54:10	cmkadmin	19474	0:00:00		
	reporting-1524736392.6	Report generation of background_job / Background Job	finished	2018-04-26 11:53:12	cmkadmin	18329	0:00:00	Report generation complete.	Report generation finished



# The Background jobs overview

WATO · Configuration

Main Menu

Monitoring Agents

Hosts

Host Tags

Global Settings

Host & Service Parameters

Manual Checks

Check Plugins

Host & Service Groups

Users

Roles & Permissions

Contact Groups

Notifications

Time Periods

Event Console

Business Intelligence

Distributed Monitoring

Backup

Passwords

Alert Handlers

Analyze configuration

**Background jobs**

Extension Packages

Logfile Pattern Analyzer

Custom Icons

Background jobs overview

cmkadmin (admin) 11:30

No changes

Main Menu

Agent baking

Actions	Job ID	Job Title	State	Started	Owner	PID	Runtime	Last progress info	Results
	agent_baking	Baking Agents...	finished	2018-04-26 11:03:27	cmkadmin	10165	0:00:02	Agent baking finished.	Baking successful

Fetch agent output

No entries

Host renaming

Actions	Job ID	Job Title	State	Started	Owner	PID	Runtime	Last progress info	Results
	rename-hosts	Renaming of heute_5 → heute_5_renamed, heute_4 → heute_4_renamed, heute_7 → heute_7_renamed, heute_6 → heute_6_renamed, heute_3 → heute_3_renamed, heute_2 → heute_2_renamed, heute_8 → heute_8_renamed	finished	2018-04-26 11:30:27	cmkadmin	22680	0:00:11	Calling final hooks	Renamed 7 hosts at the following places: <ul style="list-style-type: none"><li>• Auto-discovered services of the host (6 times)</li><li>• Cached output of monitoring agent (7 times)</li><li>• File with performance counter (6 times)</li><li>• WATO folder (7 times)</li><li>• Recent hardware/software inventory (7 times)</li><li>• The current monitoring state (including acknowledgements and downtimes) (7 times)</li></ul>

# Special rules for critical jobs

- Affected jobs:
  - ♦ bake agents
  - ♦ rename host
- Only one job allowed
- Error page with details
- Require acknowledgement

Monitoring Agents - Agent Bakery

[Main Menu](#) [Rules](#) [Agent files](#) [Automatic updates](#)

The last bake operation ran into an exception. Please acknowledge this to proceed.

[Acknowledge last bake result](#)

Result of last baking process

ID	agent_baking
Title	Baking Agents...
Started	2018-04-26 13:55:27
Owner	cmkadmin
Actions	
State	exception
Acknowledged by	
Runtime	0:00:00
PID	30132
Result	
Exceptions	Exception while preparing background function environment: Traceback File "/omd/sites/heute/share/check_mk/web/htdocs/background_job.py" raise Exception('Unexpected Error') Exception: Unexpected Error More information can be found in ~/var/log/web.log
Progress Info	



# Upcoming background jobs



Download of agent output



Service discovery



SLA calculations





**CHECK\_MK**

**CONFERENCE**

MUNICH 2018/5/2-4

**#4**



CHECK\_MK

# Time-specific check parameters

03.05.2018, Konstantin Büttner  
Check\_MK Conference #4

**CONFERENCE**  
MUNICH 2018/5/2-4

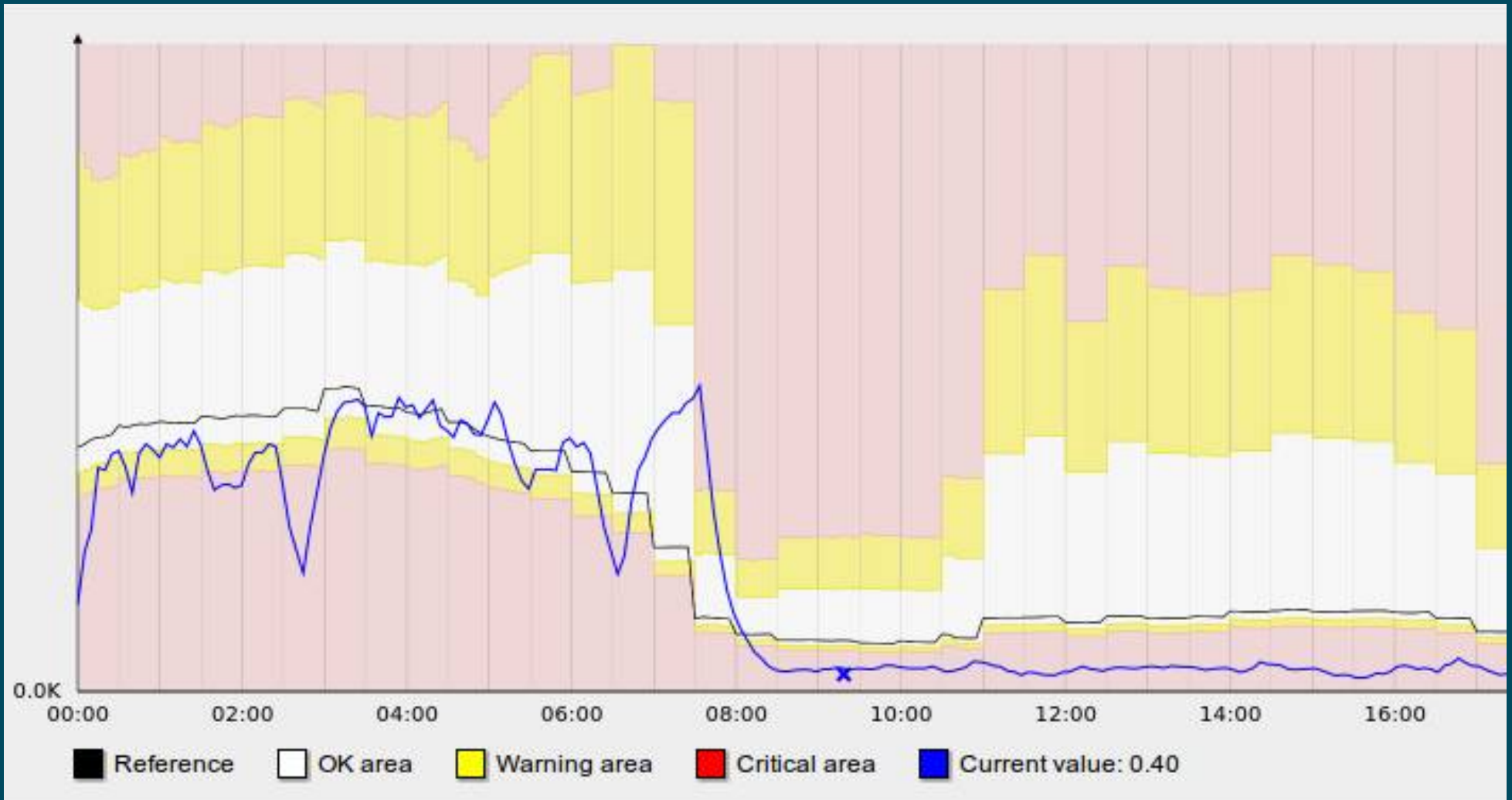
**#4**

# Use case

- Recurring events lead to conditions that would be abnormal during other times
- Eg. backup jobs etc.



# Predictive Levels?



# Solution – Timespecific parameters

- Define parameters for time periods
- Generic solution – works for **all** check plugins

# How does it work?

▼ Parameters

Fixed Levels ▼

Enable timespecific parameters

Warning at 5.00 per core

Critical at 10.00 per core

# How does it work?

▼ Parameters

Configured timeperiod parameters [Disable timespecific parameters](#)

[Add new element](#)

Default parameters when no timeperiod matches

Fixed Levels ▼

Warning at  per core

Critical at  per core

▼ Parameters

Configured timeperiod parameters Disable timespecific parameters

Match only during timeperiod Select a timeperiod ▼

Fixed Levels ▼

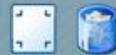


Warning at 5.00 per core

Critical at 10.00 per core

Match only during timeperiod Select a timeperiod ▼

Fixed Levels ▼



Warning at 5.00 per core

Critical at 10.00 per core

Add new element

Default parameters when no timeperiod matches

Fixed Levels ▼

Warning at 5.00 per core

Critical at 10.00 per core



**CHECK\_MK**

**CONFERENCE**

MUNICH 2018/5/2-4

**#4**





CHECK\_MK

# Customizable graph layouts

03.05.2018, Konstantin Büttner  
Check\_MK Conference #4

**CONFERENCE**  
MUNICH 2018/5/2-4

**#4**

# Our objective in graphing

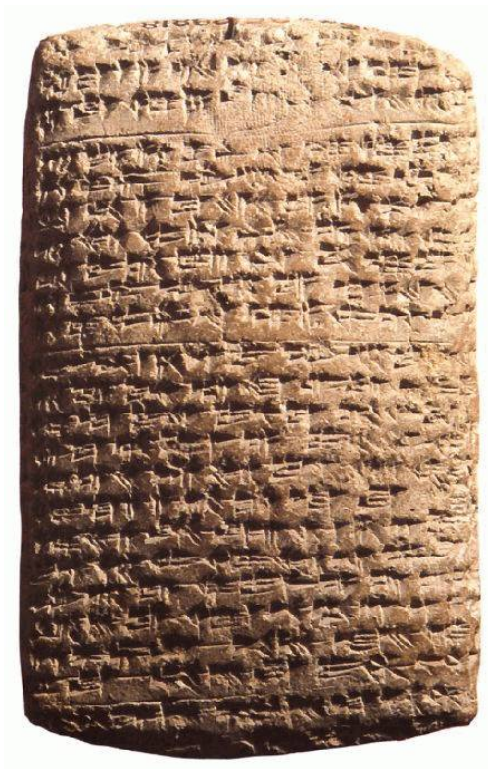
- Traditional graphing system are for customizing graphs
- But: day-to-day, we'd rather just **have** graphs  
Tweaking graphs is fun, but time consuming
- Our objective: Make our data useful out of the box



# A bit of history - PNP4Nagios

## PNP graph

---



## In the Olden Days

---

- Graphs were defined for each check type
- Templates in PHP ( ノ 0 益 0) ノ 彡 — — — — —
- No semantics, no uniformity\*
- No customizability\*\*

\*except for checks sharing templates

\*\*unless you wrote your own templates

# A bit of history - Metrics

## Metrics graph



## A new dawn

- Most visible: New look
- Most important: Semantics
- Checks no longer have graphs...
- ...but values have semantics
  - Automatic graphs
  - Custom graphs

# Now: More customizability

- Unified options for dashlets, views, reports (Werk #5569)

## ▼ Display Options

Graph rendering options .....

Font size

8.00 pt

Title

Show graph title ▼

Title format

Add additional information ▼

☐

Add host name

☐

Add host alias

☐

Add service description

Show graph time range

☒

Show the graph time range on top of the graph

Show margin round the graph

☒

Show a margin round the graph

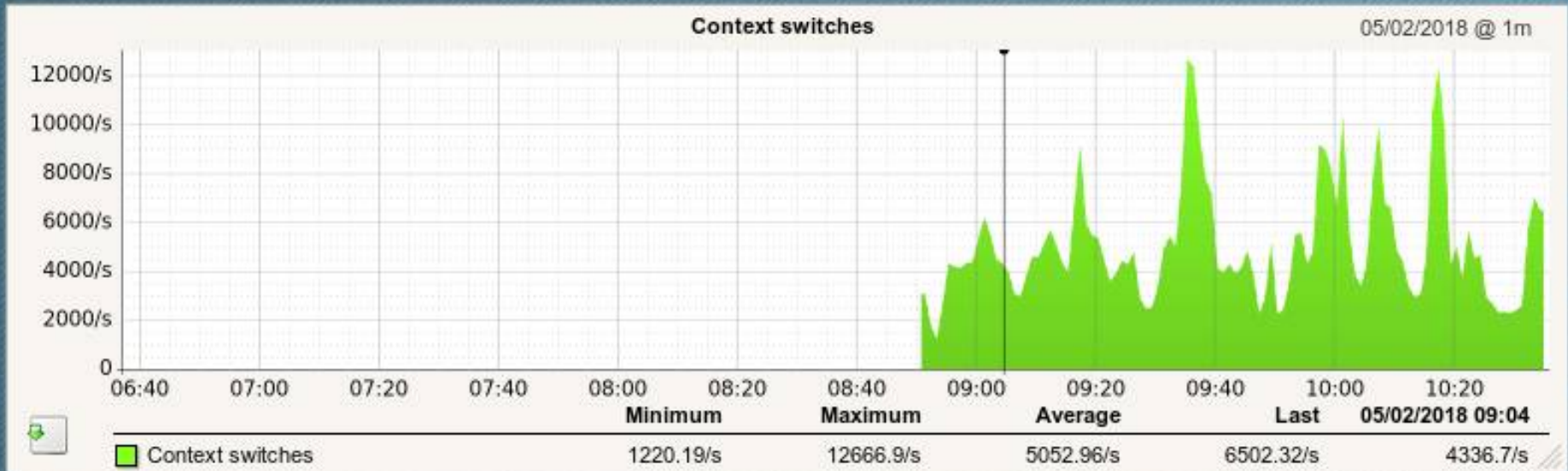
Show legend

☒

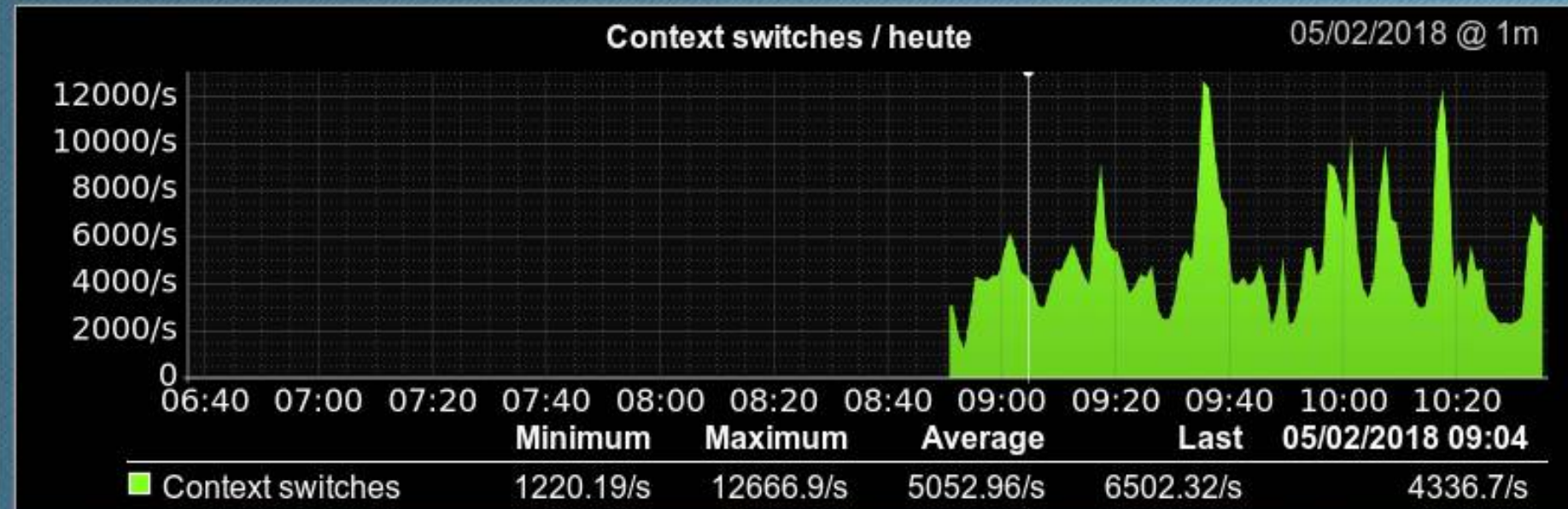
Show the graph legend



# Der Master, heute, Kernel Context Switches



# Der Master, heute, Kernel Context Switches



# Now: More customizability

- Unified options for dashlets, views, reports (Werk #5569)
- Vertical scaling and mirroring customizable (Werk #3920)



▼ Tunings

Graph ..... Heap and non-heap memory ▼

Tunings .....

Apply to

- ☐ Specific destination
- ☒ Template graphs
  - ☐ Site
  - ☐ Hostname
  - ☐ Service description
- ☒ Combined graph
  - Apply to combined graphs

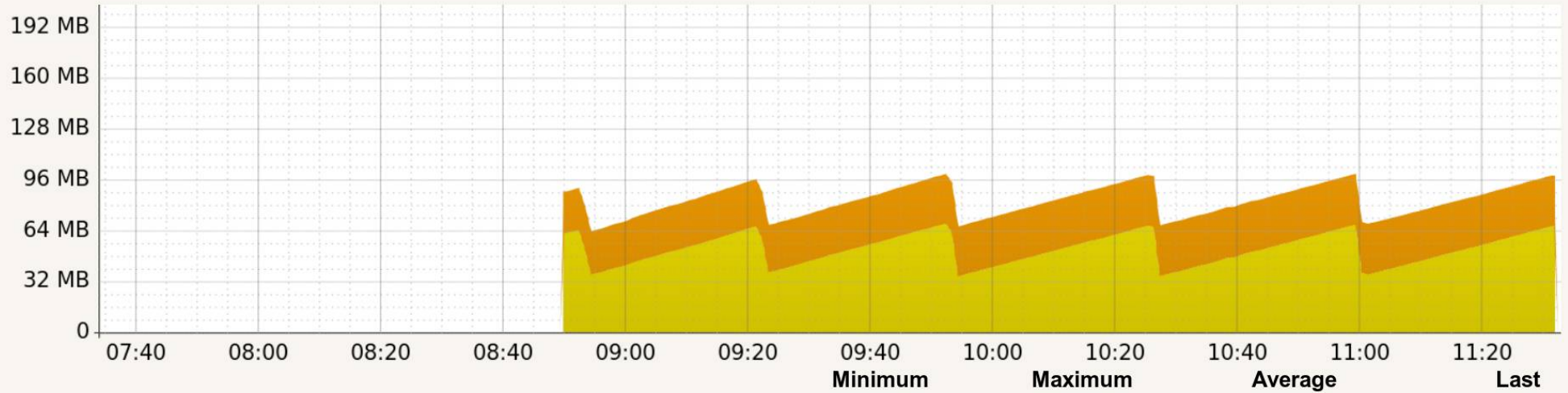
☒ Vertical axis scaling

- Explicit range ▼
- Lower:  Upper:
- ☐ Mirror the vertical axis

Add new element

# Heap and non-heap memory

05/02/2018 @ 1m



Non-heap memory usage  
Heap memory usage

	Minimum	Maximum	Average	Last
Non-heap memory usage	26.06 MB	31.90 MB	30.75 MB	31.79 MB
Heap memory usage	35.54 MB	68.75 MB	52.33 MB	67.05 MB





**CHECK\_MK**

**CONFERENCE**

MUNICH 2018/5/2-4

**#4**



CHECK\_MK

# Performance improvements

03.05.2018, Konstantin Büttner  
Check\_MK Conference #4

**CONFERENCE**  
MUNICH 2018/5/2-4

**#4**

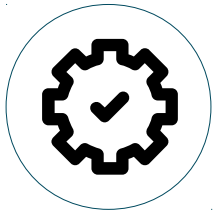
# Why is a performance focus critical?

- The typical Check\_MK setup keeps growing – more services, more sites, more users
- Better scalability makes new things possible

# How to make things go faster



Avoid **unnecessary** work



**Optimize**



Make things **feel** faster



# Avoid unnecessary work - Tabs

- A typical browser: 50 open tabs
- You don't use them all... but they all make regular livestatus queries, though
- New: Only update active tabs/windows (Werk #4753)
- Benefit: No more wasted data transfers

# Avoid unnecessary work – Emails

- Graphs can make emails reasonably large
- Multiply that by the number of recipients ...
- What's a mail server for? (Werk #4813)
- Benefit: Notification system becomes more scalable

# Avoid unnecessary work – Sites

- Frequent scenario: multi-site setup
- For many users, only few sites are actually relevant
- Yet, GUI in the past contacted all sites for every user
- No more: Now configureable for each user  
(Werk #4921)



# Optimize – BI

- Large aggregations now compile faster (Werk #5142)
- Magic - “it’s the algorithm, stupid”
- Further reworking already planned



# Optimize – liveproxyd

- Previously: `# The main loop of the daemon goes here`
- One python process – GIL limits scalability
- Now what?

# Optimize – liveproxyd

- Now: One process per site, master process to manage
- Process view:

```
OMD[heute]:~$ ps -ef | grep liveproxyd
UID      PID  PPID  C  STIME TTY      TIME  CMD
heute    9261    1   0  11:40 ?        00:00:00 liveproxyd[master]
heute    9262  9261   0  11:40 ?        00:00:00 liveproxyd[heute_slave_1]
heute    9263  9261   0  11:40 ?        00:00:00 liveproxyd[heute_slave_2]
```

- More resilient, more performant through use of multiple CPUs (cf. Werk #4901)

# Make things feel faster - Graphs

- Previously: View with many Graphs? Better wait for **all** the data
- Now: Load the view with placeholders, update graphs asynchronously

# Take-home message



Attention to detail pays off



Continuously improving



If you have lots of sites and lots of users, Werk #4921 may help you





**CHECK\_MK**

**CONFERENCE**

MUNICH 2018/5/2-4

**#4**