



The Missing Link

Bridging the Gap Between Infrastructure and Applications



Sebastian Krahe
VP Product
Checkmk



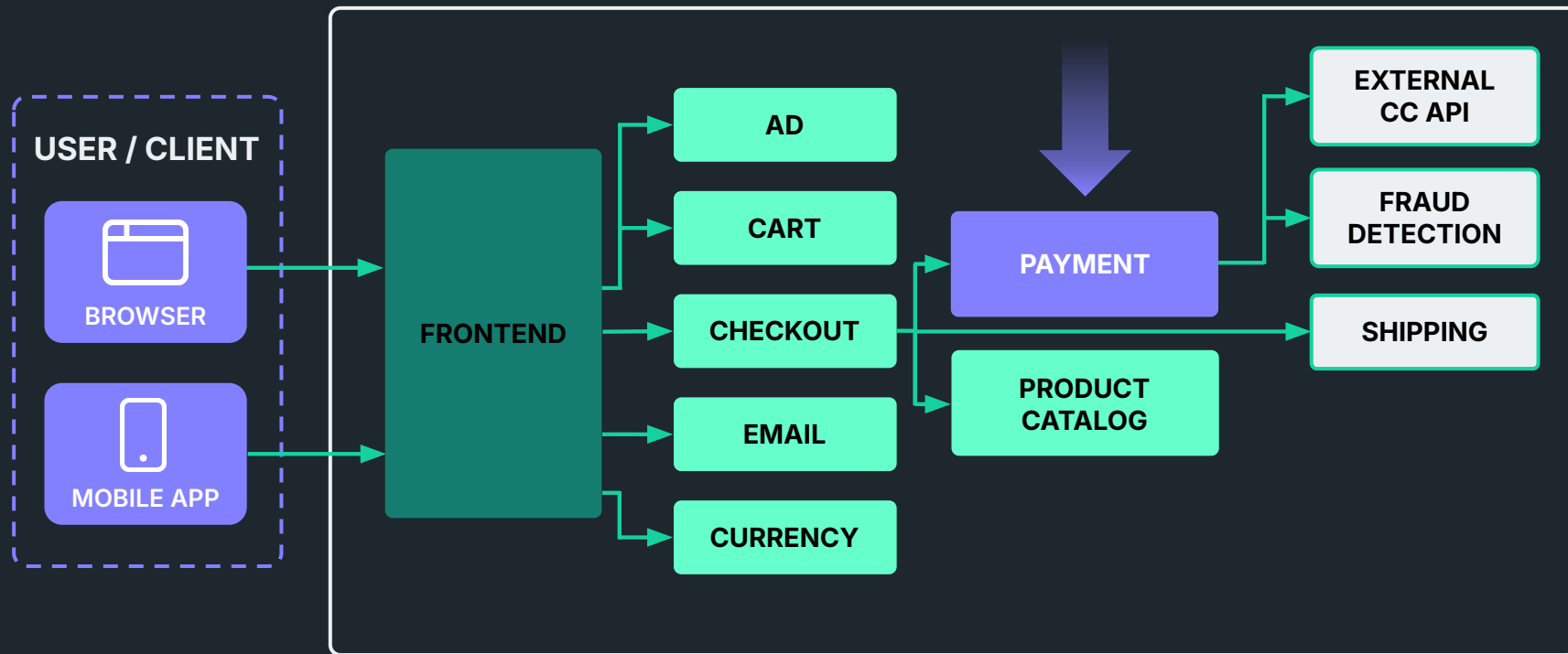
Jörg Herbel
Engineering Manager
Checkmk

Alert Notification

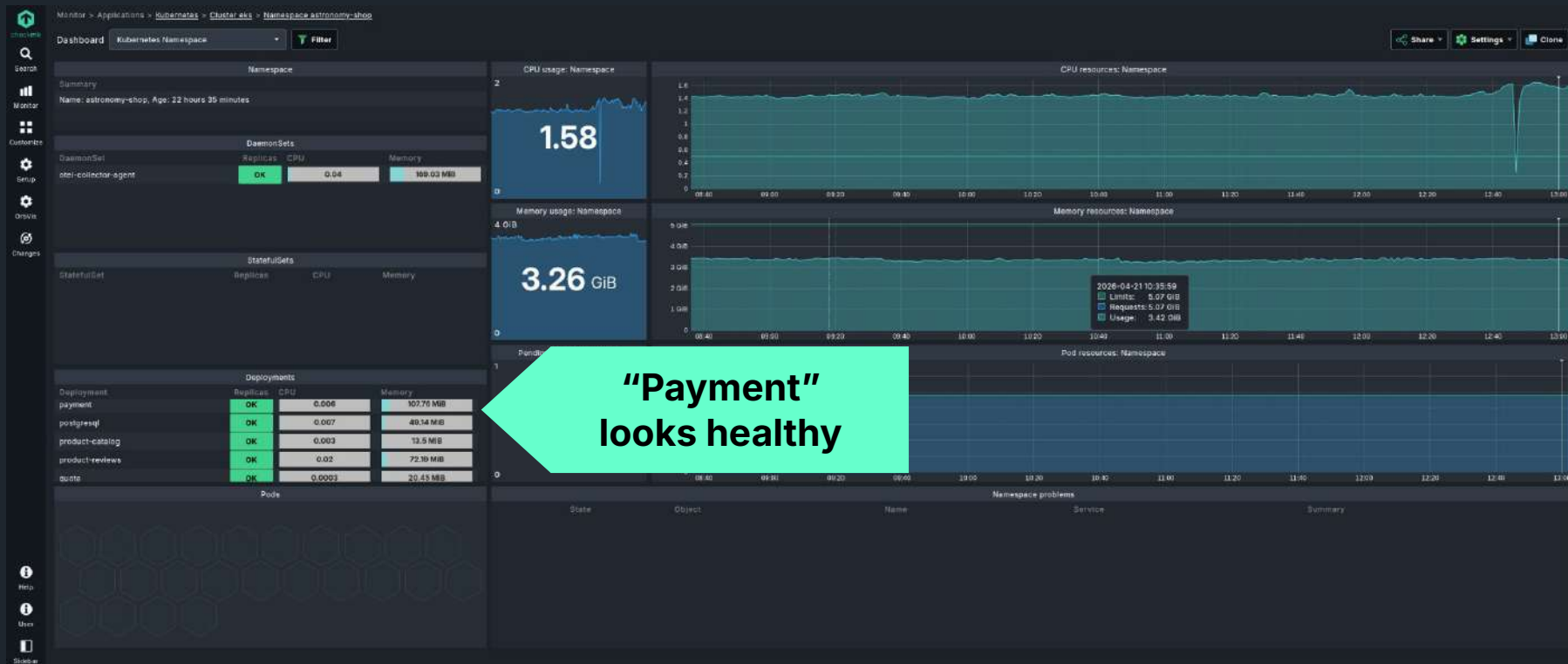


The Webshop

What's wrong with it?



K8s Namespace - infrastructure is green



K8s Deployment - even payment itself looks healthy





deployment_eks_astronomy-shop_payment						
State	Service	Icons	Summary	Age	Checked	Perf-O-Meter
OK	Check_MK	👉	[piggyback] Successfully processed from source 'kubernetes.company.com', execution time 0.0 sec	22 h	53.0 s	10 ms
OK	Check_MK Discovery	👉	Services: all up to date, Host labels: all up to date, execution time 0.1 sec	22 h	8 m	60 ms
OK	Check_MK HW/SW inventory	👉	Found 21 inventory entries, software changes, execution time 0.2 sec	22 h	22 h	0.23 s
OK	Condition	☰	All conditions OK.	22 h	64 s	
OK	CPU resources	👉	Usage: 0.006, Requests: 0.000 (0/1 containers with requests), Limits: 0.000 (0/1 containers with limits)	22 h	64 s	0.006
OK	info	☰	Name: payment, Namespace: astronomy-shop, Age: 22 hours 10 minutes	22 h	64 s	
OK	Memory resources	👉	Usage: 108 MiB, Requests utilization: 76.80% - 108 MiB of 140 MiB (1/1 containers with requests), Limits utilization: 76.80% - 108 MiB of 140 MiB (1/1 containers with limits)	22 h	64 s	107.52 MiB
OK	Pod resources	👉	Running: 1, Pending: 0, Succeeded: 0, Failed: 0, Unknown: 0	22 h	65 s	
OK	Replicas	👉	Ready: 1/1, Up-to-date: 1/1	22 h	65 s	

All green?!



SLACK #incidents

Checkout page returning 500 errors

JIRA INC-4821

Users unable to complete transactions

The boss via SMS

The webshop is down. Fix it. NOW!

Customer Support

47 new tickets in the last 15 minutes

App is down!!

How many of you have had a lunch break, evening or even night like this?

**"You need to add monitoring
for the application. ASAP."**

- Your boss, later that day

When you think of the visibility gap, think about ...

... an iceberg!



Infrastructure layer

CPU · Memory · Disk
· Network · Processes

Infrastructure green

When you think of the visibility gap, think about ...

... an iceberg!



Infrastructure layer

CPU · Memory · Disk
· Network · Processes

Infrastructure green

Application layer

- + Request rate · Error rate · Latency (RED)
- + Per-endpoint latency breakdown
- + JVM heap & .NET runtime
- + Connection pool saturation
- + ...

Both layers would be great!



Dashboard A

UP

All hosts are green

Alert Engine A

Rules · Alerts · Notifications

IT Ops Team

"Not our domain"



Dashboard B



HTTP 500 errors spiking

Alert Engine B

Different rules · Different monitors

App Dev(Ops) Team

"Not our domain"



Dashboard A

UP

All hosts are green

Alert Engine A

Rules · Alerts · Notifications

IT Ops Team

"Not our domain"

The Divide



Dashboard B



HTTP 500 errors spiking

Alert Engine B

Different rules · Different monitors

App Dev(Ops) Team

"Not our domain"

20+ min per incident

"Is Checkmk REALLY going to do application monitoring?"

- Probably you, right now

The best of two worlds: unified visibility



Checkmk 2.5

Same alerting
& reporting

Same
dashboards

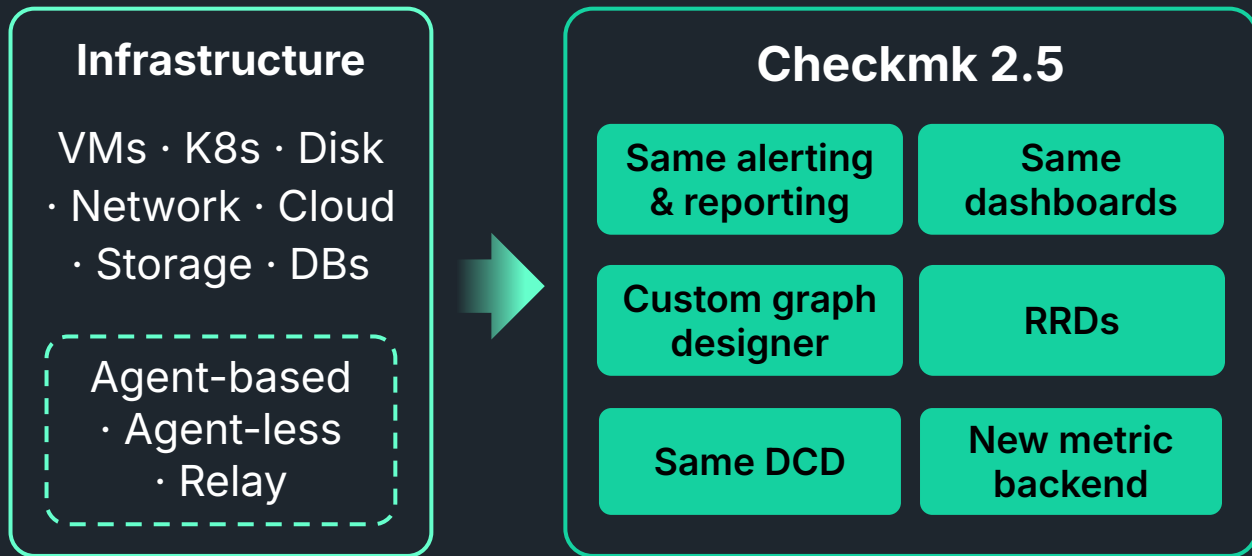
Custom graph
designer

RRDs

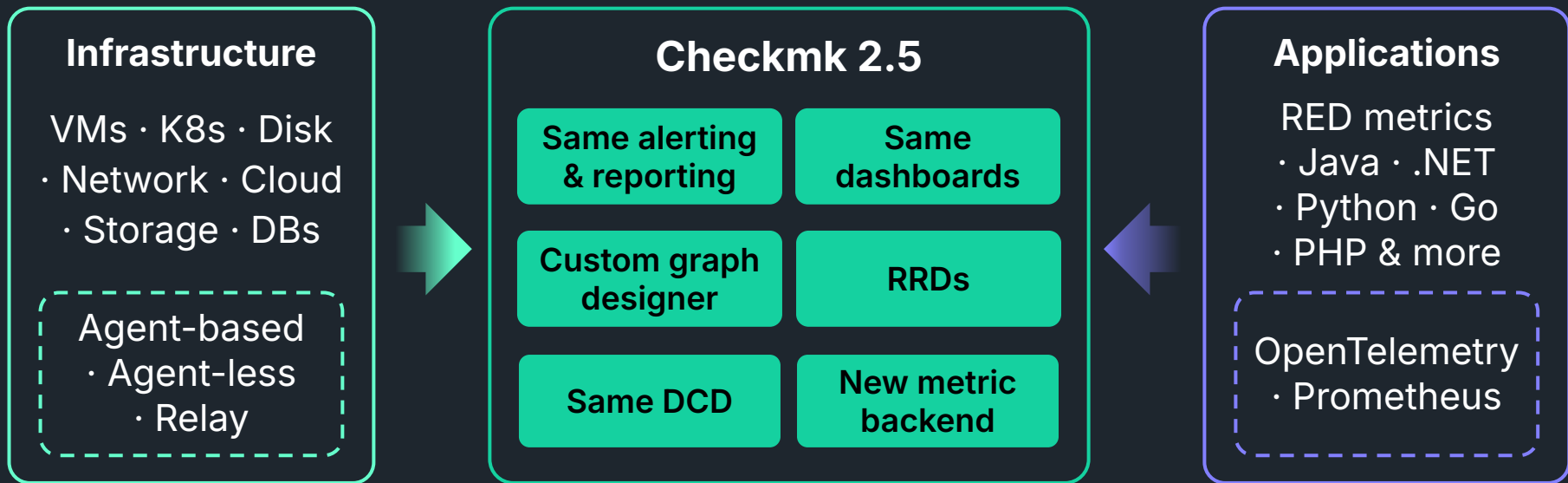
Same DCD

New metric
backend

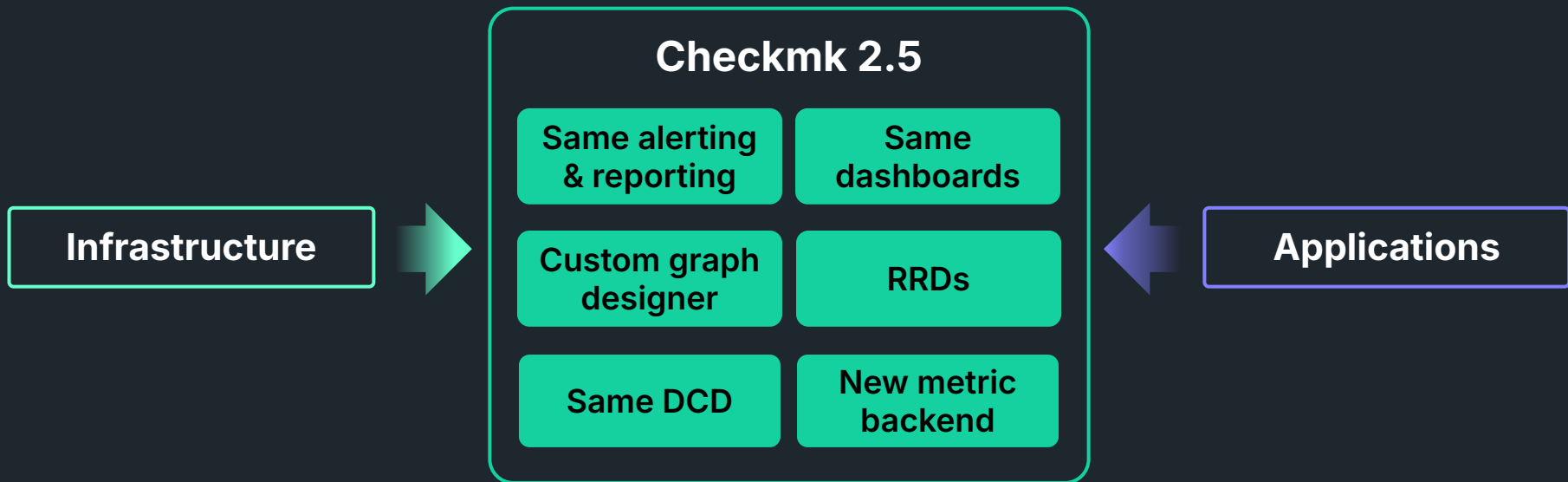
The best of two worlds: unified visibility



The best of two worlds: unified visibility



The best of two worlds: unified visibility



"Is it the code - or the infrastructure?"

- The question that needs to be answered **fast**

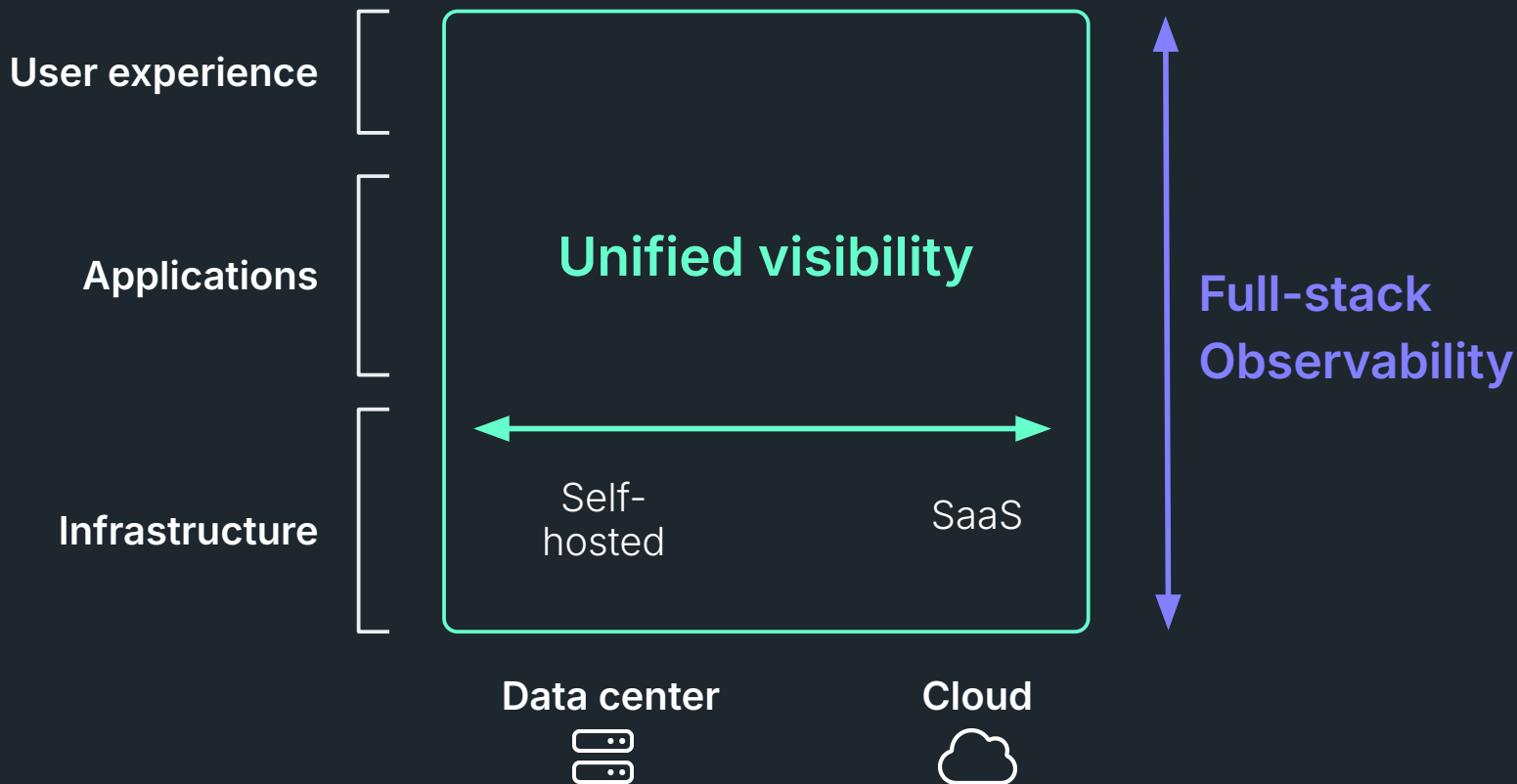
Your lunch break - re-imagined



Your lunch break - re-imagined

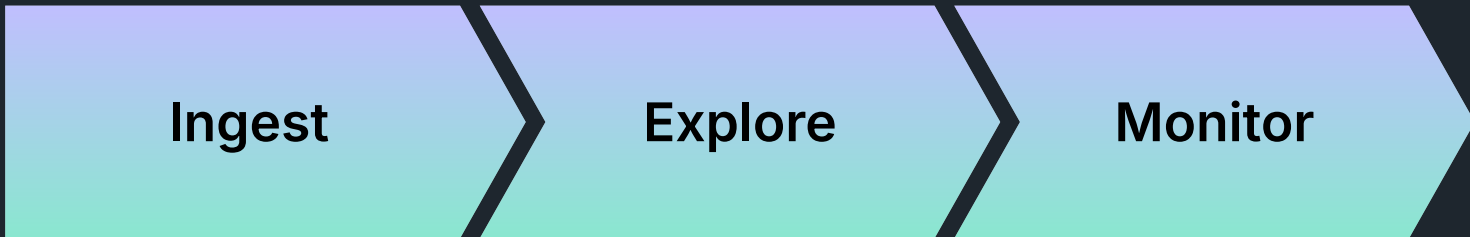


Checkmk 2.5 - unified visibility



So, (how) does this actually work?

Jörg, please show us ...



A horizontal process flow diagram consisting of three chevron-shaped boxes pointing from left to right. The boxes are filled with a light blue-to-green gradient and have a dark blue outline. The first box is labeled 'Ingest', the second 'Explore', and the third 'Monitor'.

Ingest

Explore

Monitor

Requirements for ingesting application metrics



**Compatible with
telemetry data**



**Scalable
performance**



**Multiple
deployment
options**

ClickHouse as new metric backend



Your systems



- ◆ Instrumented application
- ◆ Collector



Prometheus
endpoint

ClickHouse as new metric backend



Your systems



- Instrumented application
- Collector



Prometheus
endpoint

Checkmk site

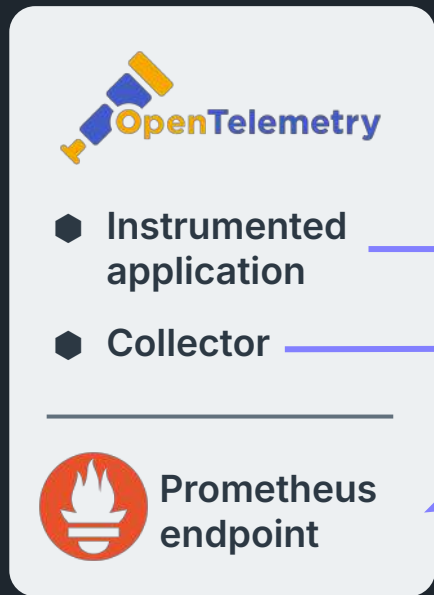


Collector

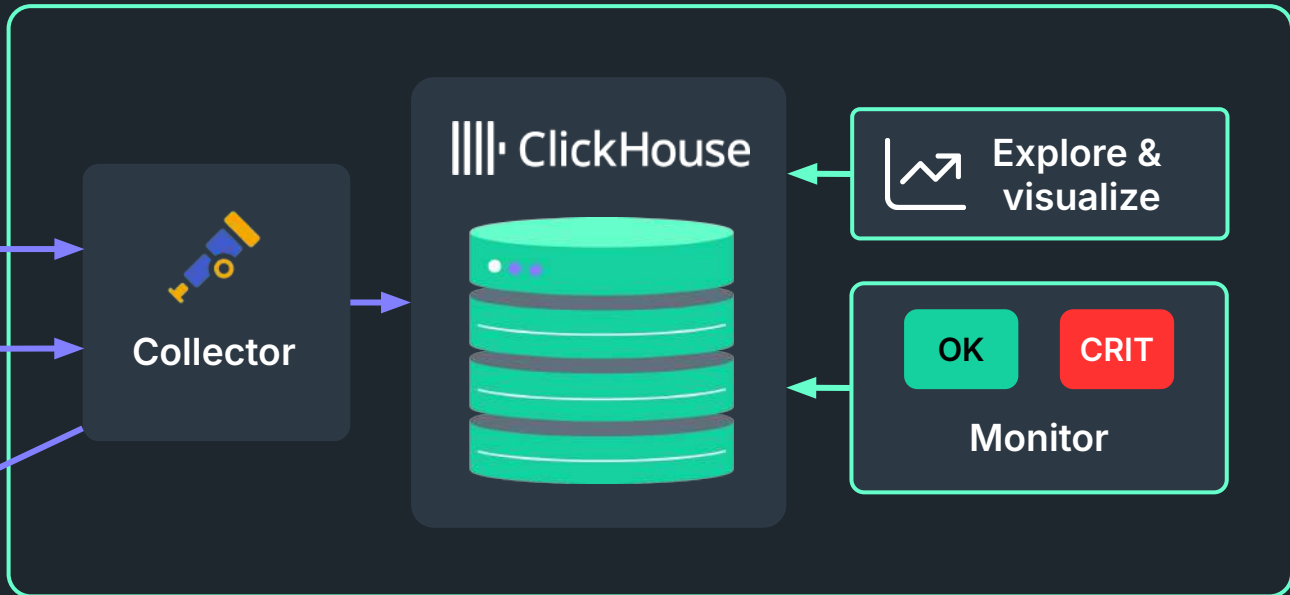
ClickHouse as new metric backend



Your systems



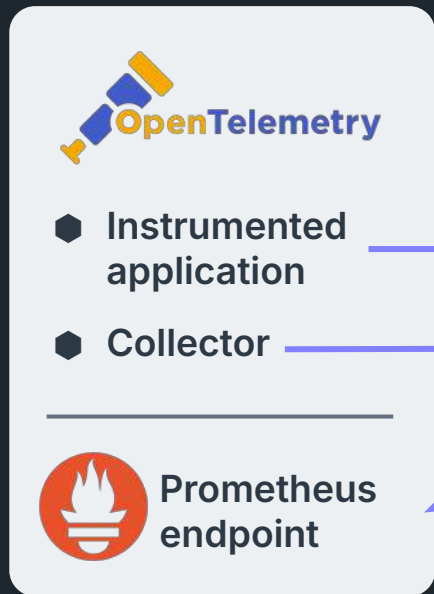
Checkmk site



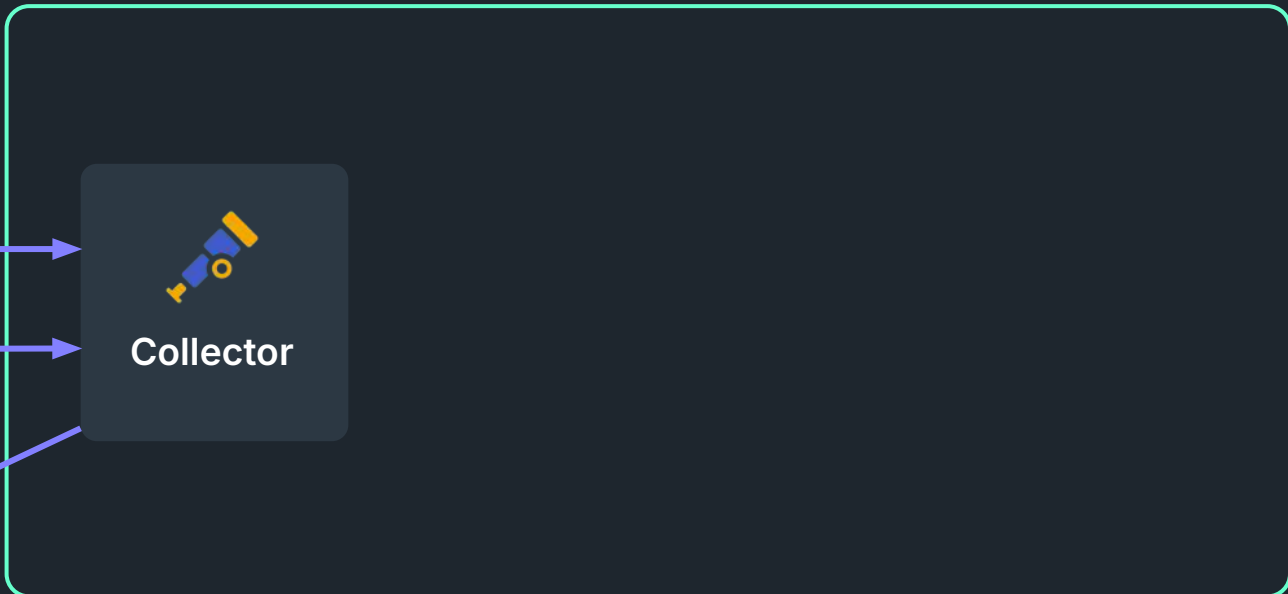
ClickHouse as new metric backend



Your systems



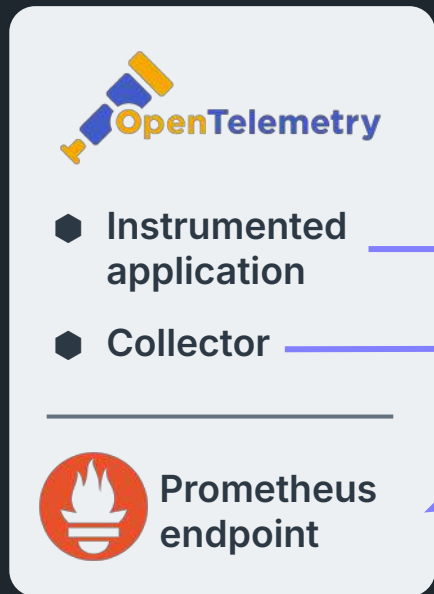
Checkmk site



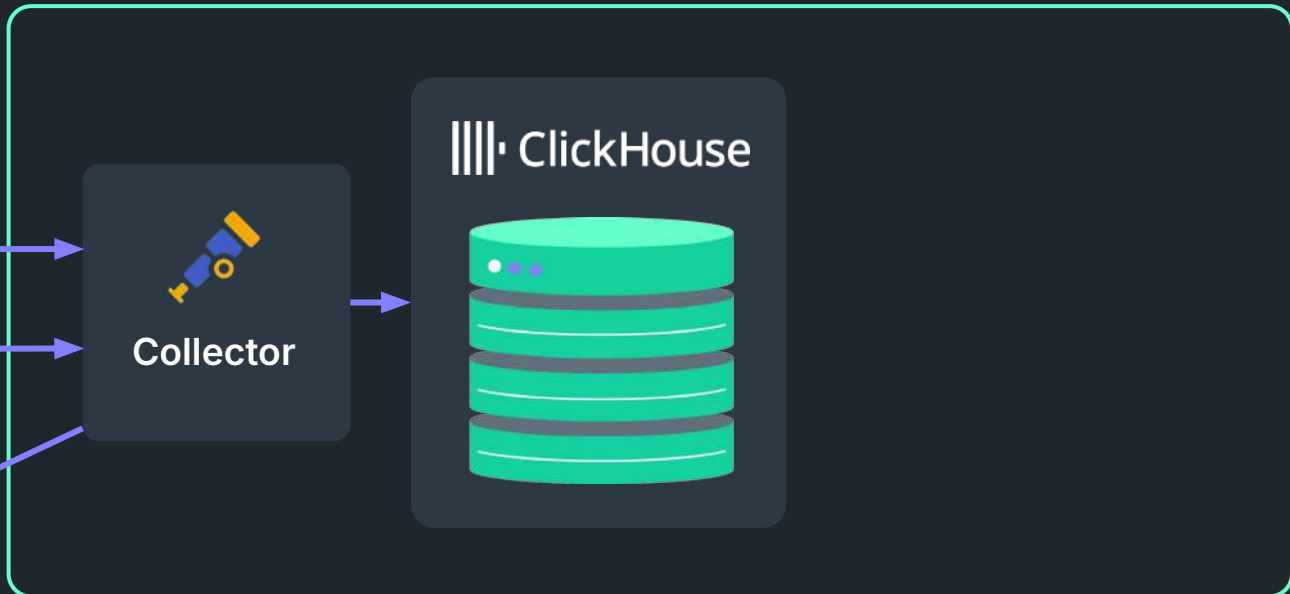
ClickHouse as new metric backend



Your systems



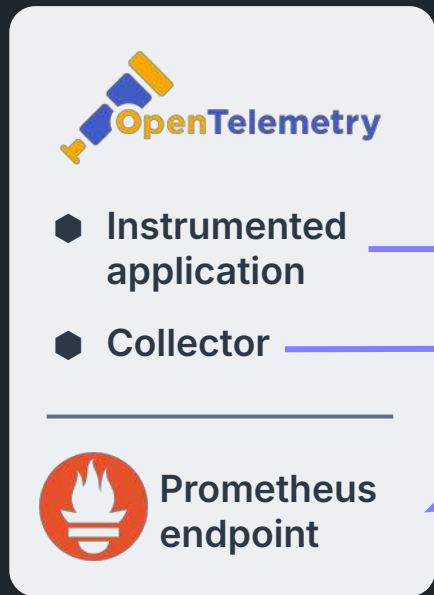
Checkmk site



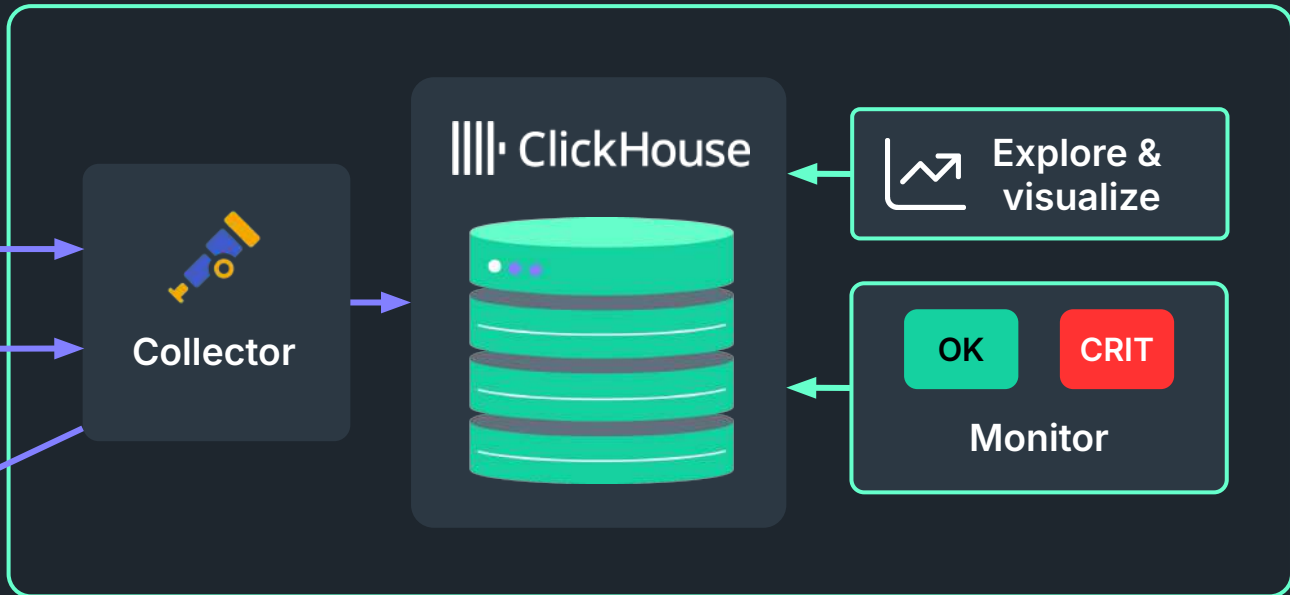
ClickHouse as new metric backend



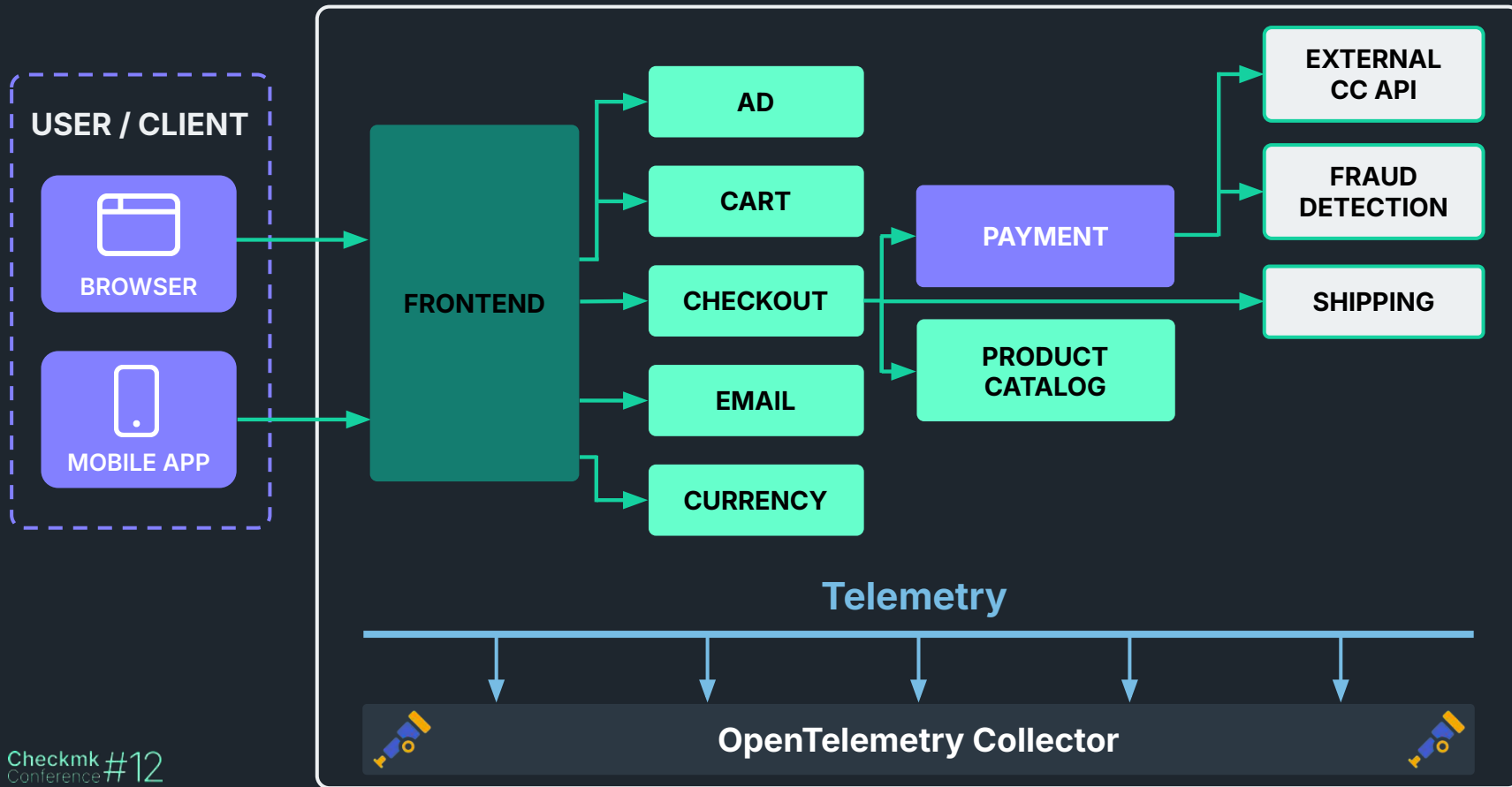
Your systems



Checkmk site



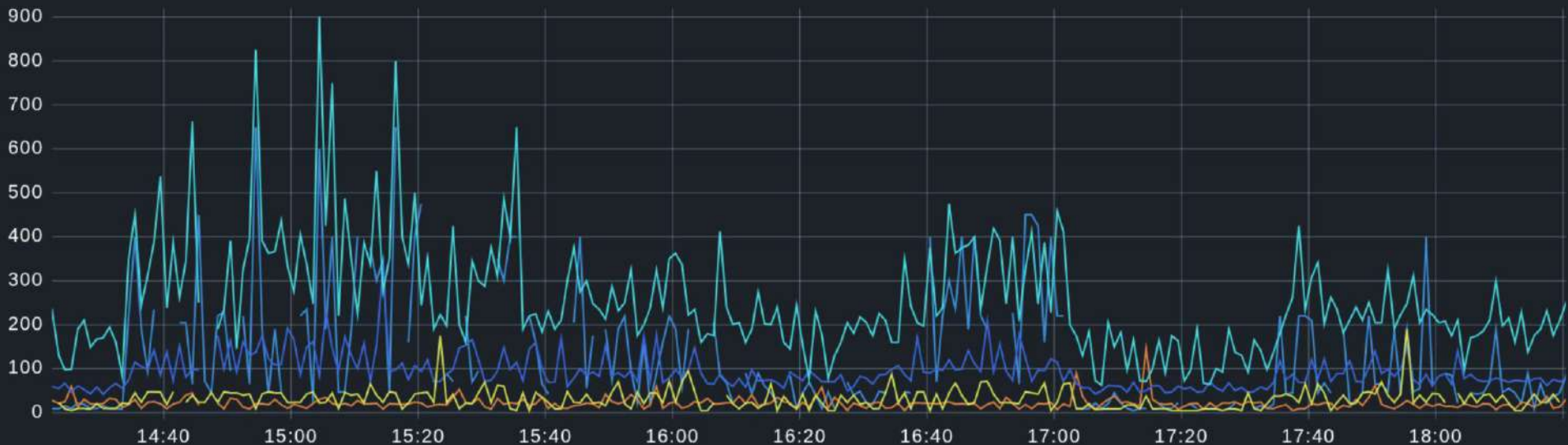
What's wrong with the Webshop?





HTTP Request Duration

2026-04-23 @ 1m



	Minimum	Maximum	Average	Last
http.client.duration - load-generator	62.5	900	255.66	195
http.client.duration - load-generator	n/a	n/a	n/a	n/a
http.client.duration - load-generator	n/a	n/a	n/a	n/a
http.client.duration - payment	n/a	n/a	n/a	n/a
http.client.duration - frontend	n/a	n/a	n/a	n/a
http.client.duration - load-generator	n/a	n/a	n/a	n/a
http.client.duration - payment	n/a	n/a	n/a	n/a
http.client.duration - payment	n/a	n/a	n/a	n/a
http.client.duration - load-generator	n/a	n/a	n/a	n/a
http.client.duration - frontend	n/a	n/a	n/a	n/a
http.client.duration - frontend	n/a	n/a	n/a	n/a
http.client.duration - product-reviews	n/a	n/a	n/a	n/a

Query:

Metric (required) http.client.duration ▼

OTel metrics exploration in custom graph designer



Graph lines (OpenTelemetry)

Query	Metric(required)	http.client.duration ▼	
	Resource attributes	Attribute key ▼	Attribute value ▼
	Scope attributes	Attribute key ▼	Attribute value ▼
	Data point attributes	Attribute key ▼	Attribute value ▼

✓ Configure what to render via queries

✓ Dropdowns offer auto-completion

OTel metrics exploration in custom graph designer



Graph lines (OpenTelemetry)

Query

Metric **(required)**

http.client.duration ▼

Resource attributes

X service.name ▼ payment ▼

Attribute key ▼ Attribute value ▼

Scope attributes

Attribute key ▼ Attribute value ▼

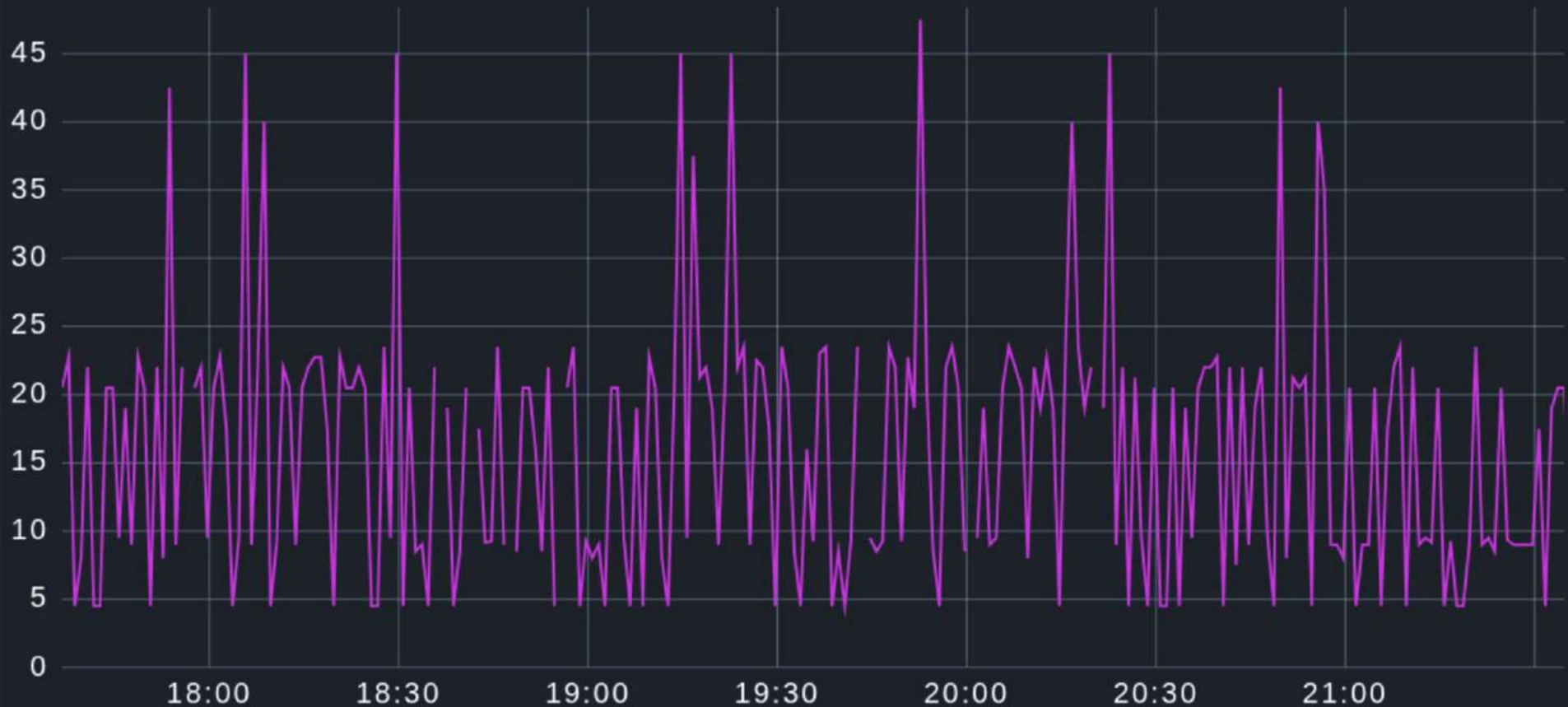
Data point attributes

X http.method ▼ POST ▼

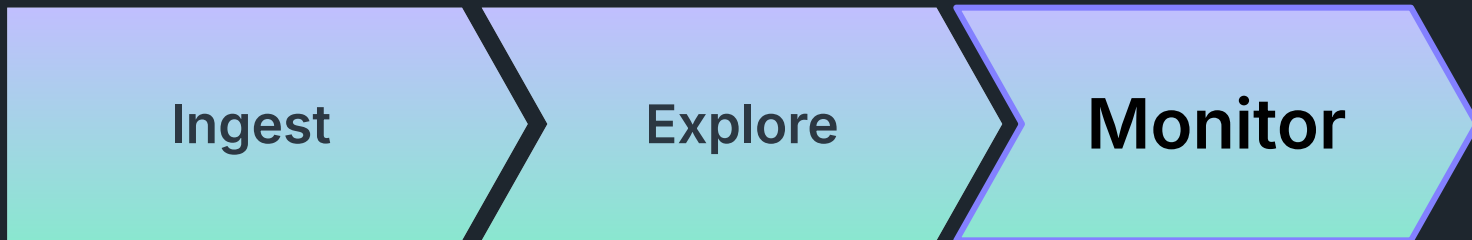
Attribute key ▼ Attribute value ▼

HTTP Request Duration

2026-05-19 @ 1m



	Minimum	Maximum	Average	Last
http.client.duration - payment	4.5	47.5	15.98	9.5

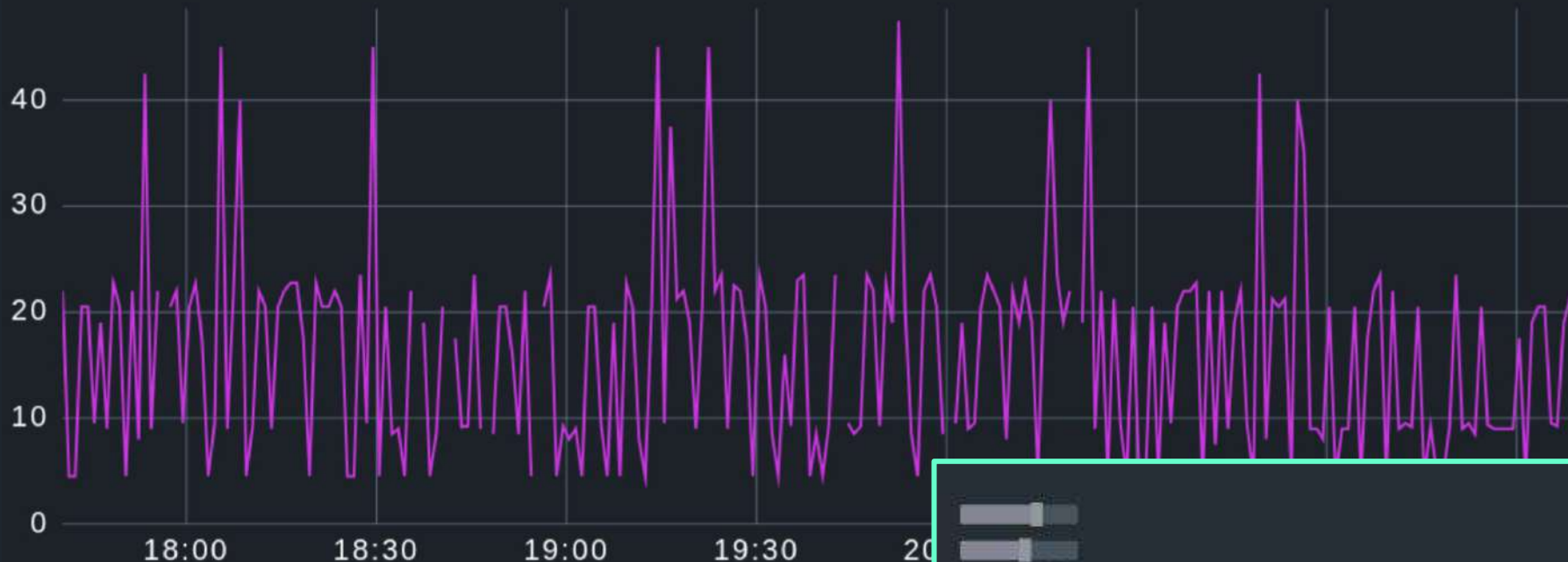


Custom query monitoring — add rule



HTTP Request Duration

2026-05-19 @ 1m



Minimum Maximum

http.client.duration - payment

4.5

Add rule: Metric backend (Custom query)

Custom query monitoring — add rule



Add rule: Metric backend (Custom query)

Save Cancel

> Rule properties

▼ Metric backend (custom query)



At this point, rule values cannot be changed. Either go back and change the custom graph as needed or save the rule and edit later via Setup > Agents > Other integrations > Metric backend (custom query).

```
Metric: http.client.duration
Resource Attributes: service.name:payment
Data Point Attributes: http.method:POST
Aggregation lookback: 2 Minutes
Percentile (histograms): 90 %
Service name template: $METRIC_NAMES$ - $RESOURCE_ATTR.service.name$
```

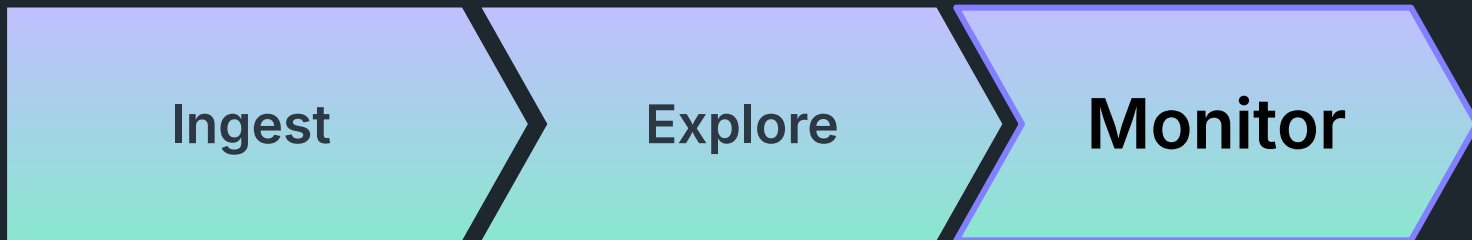
```
Metric: http.client.duration
Resource Attributes: service.name:payment
Data Point Attributes: http.method:POST
Service name template: $METRIC_NAMES$ - $RESOURCE_ATTR.service.name$
```

Custom query monitoring

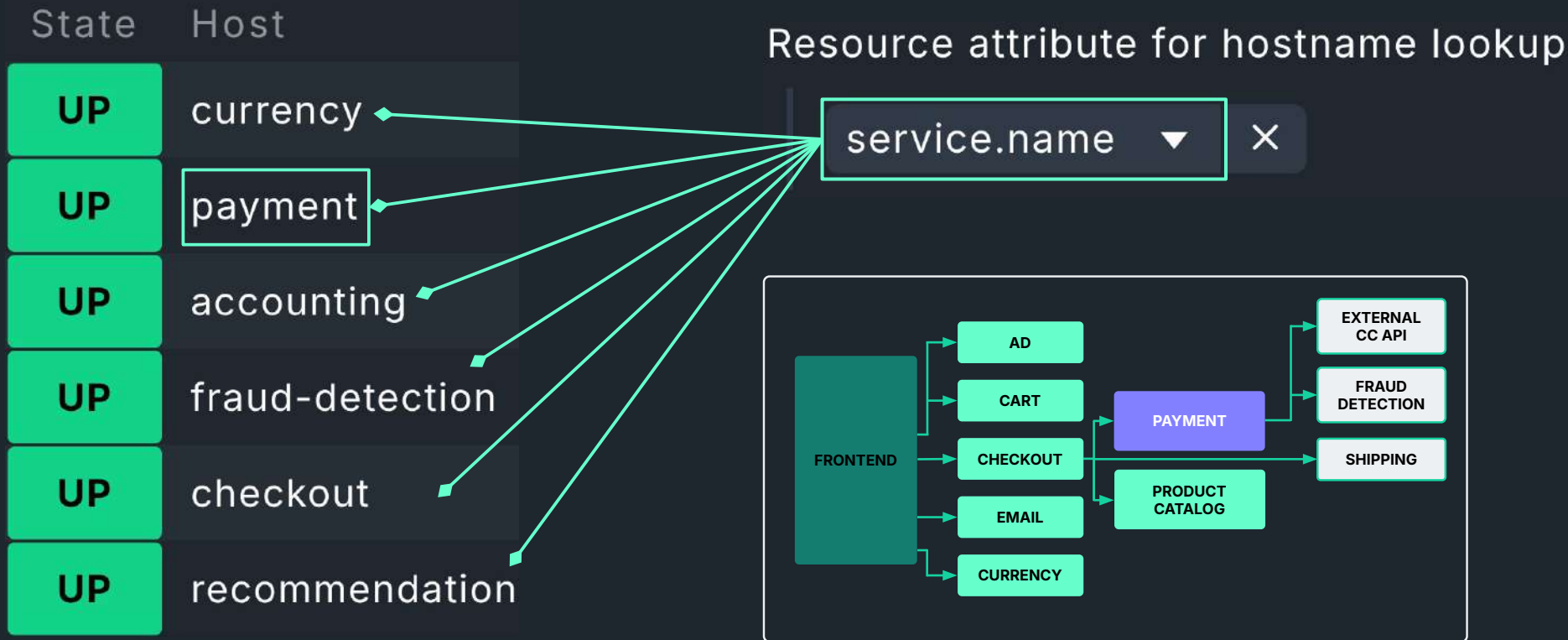


State	Service	Summary
OK	http.client.duration - payment	http.client.duration: 19.0 ms

- ✓ Custom queries are monitored as standard Checkmk services.
- ✓ Upper and lower thresholds are configurable.
- ✓ Can monitor multiple queries per host.



OTel DCD connector — created hosts



OTel DCD connector — discovered services



OK	OTel metric go.memory.allocations	502.55 {allocation}/s
OK	OTel metric go.memory.allocated	267770.27 By/s
OK	OTel metric go.processor.limit	2.00 {thread}
OK	OTel metric go.memory.gc.goal	7458122.00 By



OTel DCD connector — RED metrics

RED = Requests, Errors, Duration

OK

Requests
(HTTP)

Last 5 minutes total requests: 99, Last 5 minutes request rate: 0.33 req/s

OK

Errors
(HTTP)

Last 5 minutes total errors: 0, Last 5 minutes error rate: 0%, Last 5 minutes total requests: 99

OK

Latency
(HTTP)

Last 5 minutes latency (p90): 0.01 ms, Last 5 minutes latency (p95): 0.01 ms, Last 5 minutes latency (p99): 0.02 ms, Last 5 minutes latency (avg): 0.00 ms



OTel DCD connector — RED metrics

RED = **Requests** Errors, Duration

OK

Requests
(HTTP)

Last 5 minutes total requests: 99, Last 5 minutes request rate: 0.33 req/s

OK

Errors
(HTTP)

Last 5 minutes total errors: 0, Last 5 minutes error rate: 0%, Last 5 minutes total requests: 99

OK

Latency
(HTTP)

Last 5 minutes latency (p90): 0.01 ms, Last 5 minutes latency (p95): 0.01 ms, Last 5 minutes latency (p99): 0.02 ms, Last 5 minutes latency (avg): 0.00 ms

OTel DCD connector — RED metrics

RED = Requests, **Errors** Duration

OK

Requests
(HTTP)

Last 5 minutes total requests: 99, Last 5 minutes request rate: 0.33 req/s

OK

Errors
(HTTP)

Last 5 minutes total errors: 0, Last 5 minutes error rate: 0%, Last 5 minutes total requests: 99

OK

Latency
(HTTP)

Last 5 minutes latency (p90): 0.01 ms, Last 5 minutes latency (p95): 0.01 ms, Last 5 minutes latency (p99): 0.02 ms, Last 5 minutes latency (avg): 0.00 ms



OTel DCD connector — RED metrics

RED = Requests, Errors, **Duration**

OK

Requests
(HTTP)

Last 5 minutes total requests: 99, Last 5 minutes request rate: 0.33 req/s

OK

Errors
(HTTP)

Last 5 minutes total errors: 0, Last 5 minutes error rate: 0%, Last 5 minutes total requests: 99

OK

Latency
(HTTP)

Last 5 minutes latency (p90): 0.01 ms, Last 5 minutes latency (p95): 0.01 ms, Last 5 minutes latency (p99): 0.02 ms, Last 5 minutes latency (avg): 0.00 ms

Checkmk 2.5: unified view on infra and application



Checkmk #12

Conference