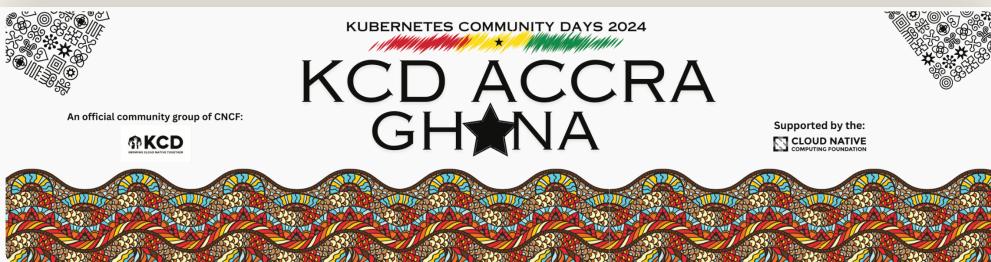


Open source observability in private cloud: mission impossible or not?

Virginia Diana Todea - Senior Site Reliability Engineer @ EQS Group

25th of November 2024



**SRE focused on o11y
OSS enthusiast
Support women in tech**



Stay in touch:
[LinkedIn](#) | [Github](#)



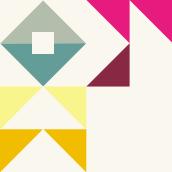
Agenda

Who are we?
Hybrid architecture
Private cloud

OSS observability
Challenges
Future plans

01

Who are we?

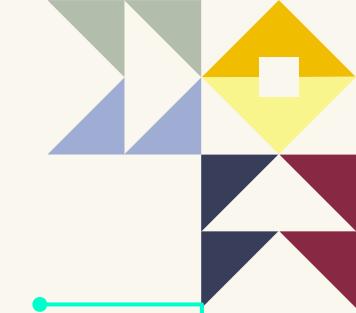


An international cloud software provider in the areas of corporate compliance, investor relations and sustainability reporting, headquartered in Munich, Germany.

Founded in 2000, present in DACH (DE, AUT, CH) and internationally in US, UK, France, Spain, Denmark, Hong Kong, Italy.

- Whistleblowing
- Data protection
- GDPR





Private cloud

Focused on the EU

Challenges in terms of monitoring and scaling

Sensitive to upgrades



Public cloud

Some companies that are not bound to very strict business regulations

Easier to handle and maintain

More costly



Developers

Plethora of frontend and backend applications connected to our databases. Lots of SDKs.

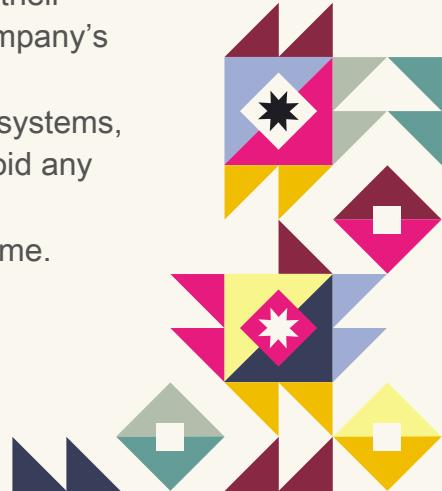


SREs

Support developers and align their infrastructure needs to the company's requirements.

Provide and maintain reliable systems, overlook the solutions and avoid any downtime.

Team of 5 SREs and limited time.





What?

- Type of cloud computing environment that is dedicated to a single organization.
- Designed to provide exclusive access and control
- Can be hosted on-premises or by a third-party service provider.



Who?

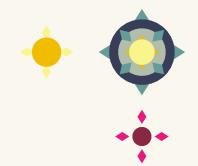
- T-systems: Germany
- Swisscom: Switzerland
- OVH: France



Why?

- For business reasons and state regulations
- Data stays local to that region, country





APM solution

Primarily focused off the shelf solutions where we ingested some logs and monitored data centers and applications.



OSS

Prometheus metrics, Grafana dashboards, ELK for logging become our observability pillars.



DevOps morphs into SRE

DevOps transitioned to SRE best practices. The crucial need for consistent and stable observability gets priority.



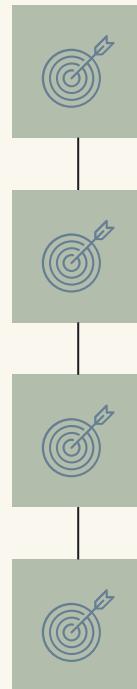
Logs, traces, metrics

Having multiple observability platforms for logs, metrics, traces leads to confusion and further lack of stability.





What private cloud taught us?



Security

Priority

Less is more

Are developers using it?



If it's trendy it doesn't mean it's good

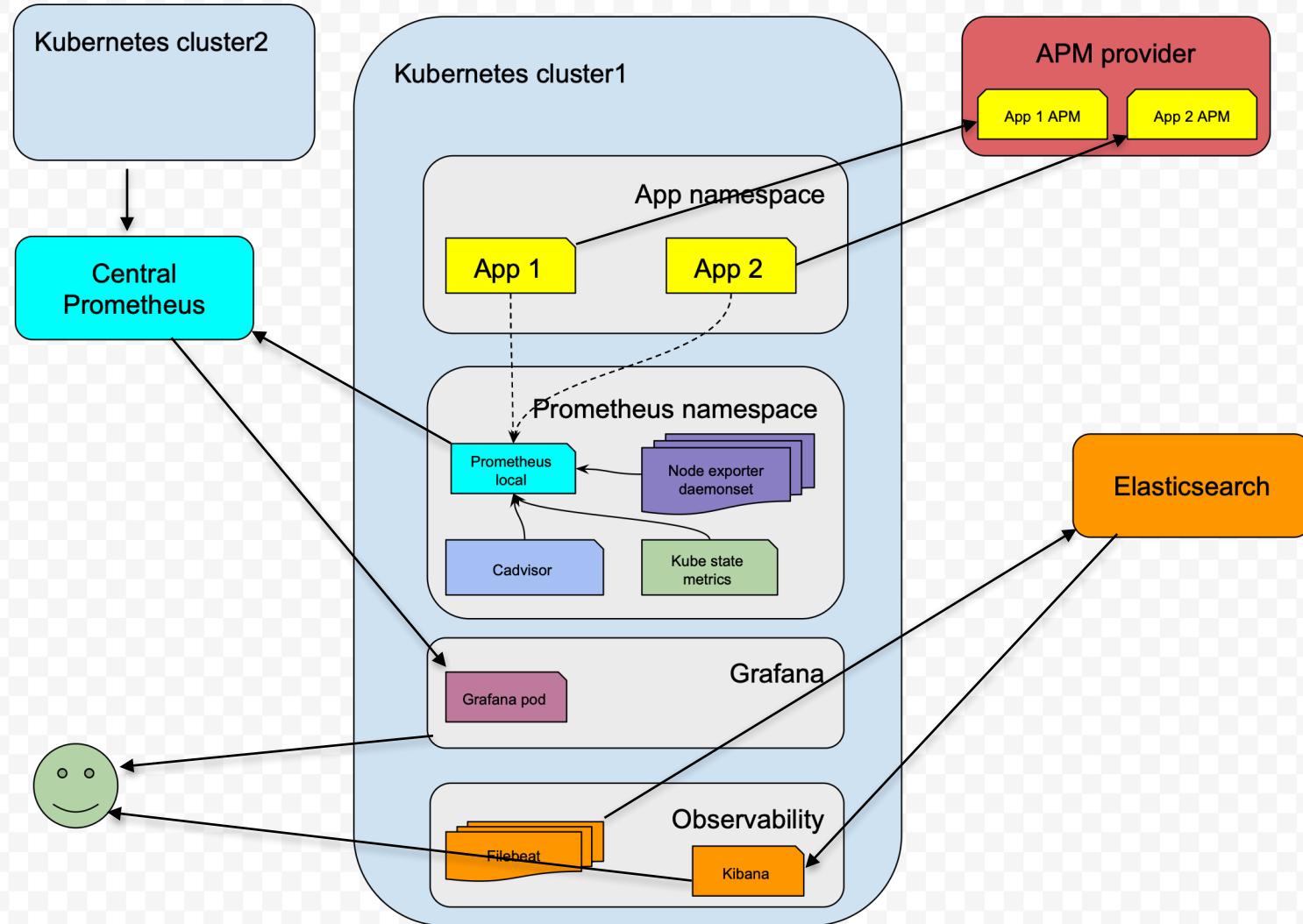


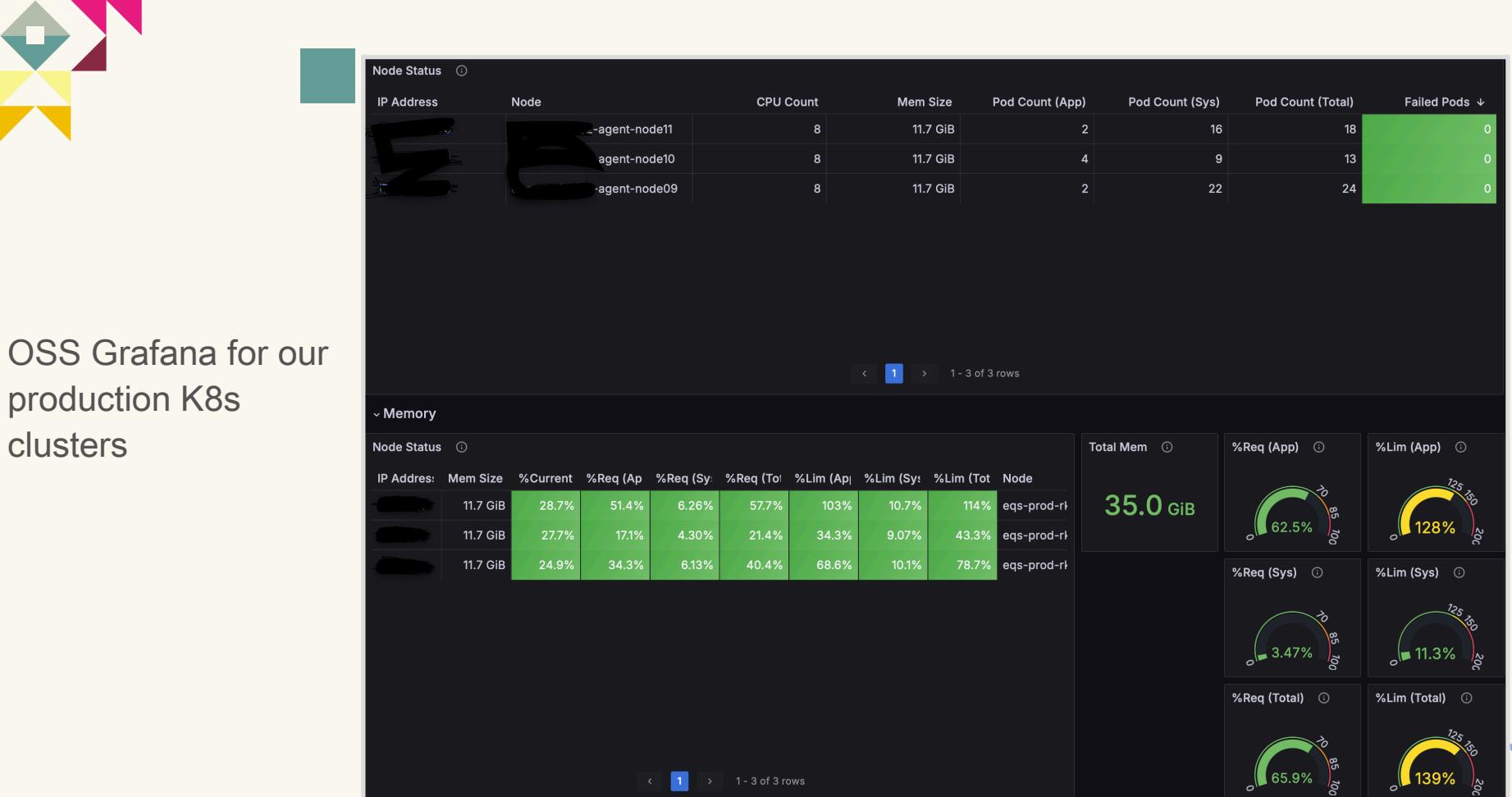
Think twice about the cost



02

Open source observability







Private cloud monitoring

Cluster: upstream-t-systems-prod | Namespace: kube-system | General K8s Object Status

Pods Totals

Pod Container Status ⓘ

Pod	Container	Owner	Node	Restarts	Req(CPU)	Lim
etcd-[REDACTED]-master-node01	etcd	Node	eqs-prod-rke2-maste	0	0.200	
etcd-[REDACTED]-master-node02	etcd	Node	eqs-prod-rke2-maste	0	0.200	
etcd-[REDACTED]-master-node03	etcd	Node	eqs-prod-rke2-maste	2	0.200	
hubble-relay-[REDACTED]-[REDACTED]	hubble-relay	ReplicaSet	eqs-prod-rke2-agent	0	No Req	
kube-apiserver-[REDACTED]-master-node01	kube-apiserver	Node	eqs-prod-rke2-maste	0	0.250	
kube-apiserver-[REDACTED]-master-node02	kube-apiserver	Node	eqs-prod-rke2-maste	0	0.250	
kube-apiserver-[REDACTED]-master-node03	kube-apiserver	Node	eqs-prod-rke2-maste	3	0.250	
kube-controller-manager-[REDACTED]-master-node01	kube-controller-man	Node	eqs-prod-rke2-maste	3	0.200	
kube-controller-manager-[REDACTED]-master-node02	kube-controller-man	Node	eqs-prod-rke2-maste	3	0.200	
kube-controller-manager-[REDACTED]-master-node03	kube-controller-man	Node	eqs-prod-rke2-maste	8	0.200	
kube-proxy-[REDACTED]-master-node01	kube-proxy	Node	eqs-prod-rke2-maste	0	0.250	
kube-proxy-[REDACTED]-master-node02	kube-proxy	Node	eqs-prod-rke2-maste	0	0.250	





Grafana alerting

Prod > Application

1 firing | 3 normal | 0 1m | ⚙️ 🔍 ⚡

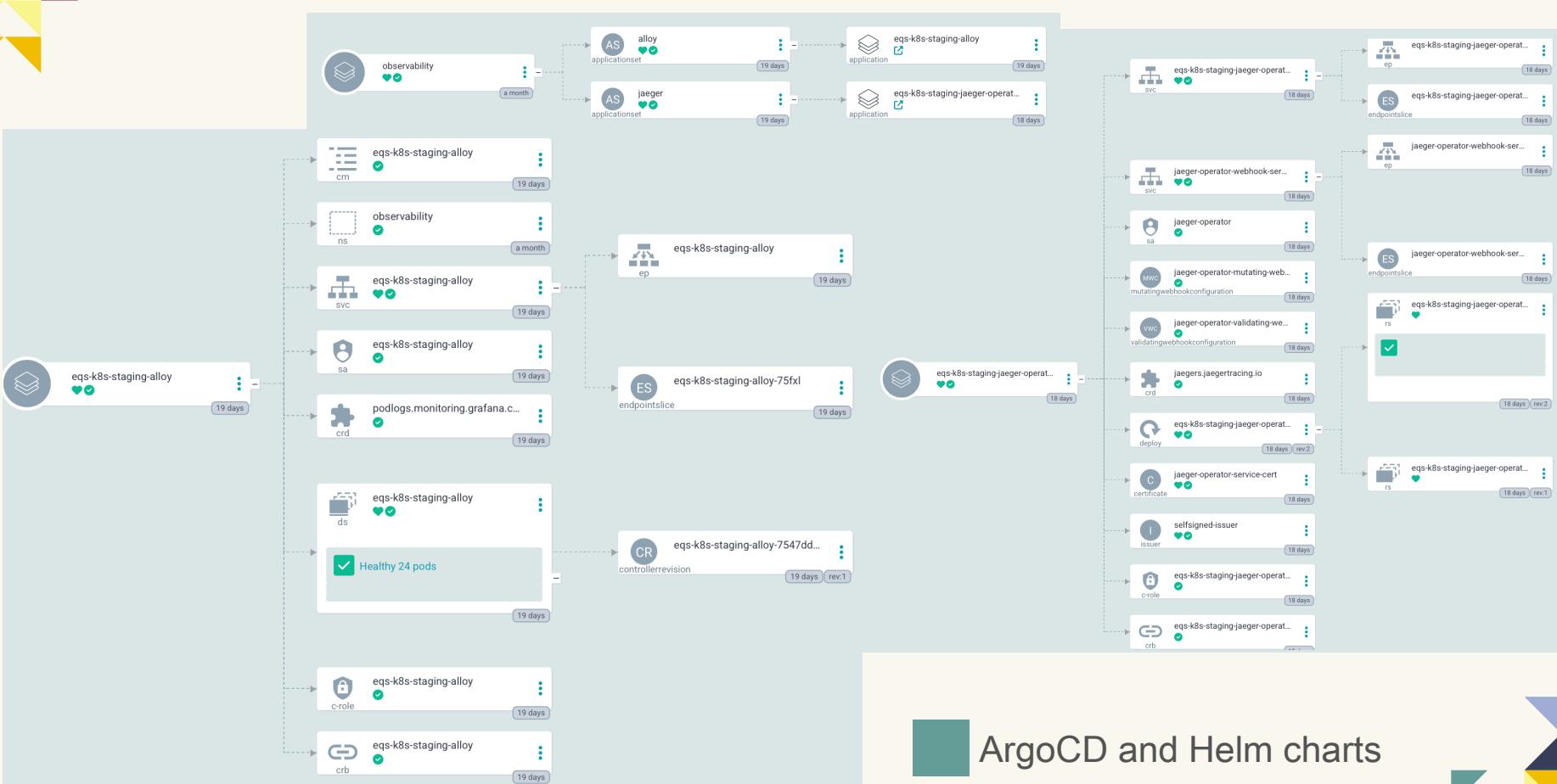
State	Name	Health	Summary	Next evaluation	Actions
> Normal	[K8s] Unschedulable pods	ok	Pod is pending scheduling	in a few seconds	🔗 ⚙️ More ▾
> Firing for 1d 18h 56m	[K8s] Pods in invalid status	ok	The pod is in invalid status (any status other than running/completed etc.).	in a few seconds	🔗 ⚙️ More ▾
> Normal	[K8s] [Private Cloud] Container throttling > 50%	ok	Pod has at least one container being throttled.	in a few seconds	🔗 ⚙️ More ▾
> Normal	[K8s] [AWS] Container throttling > 50%	ok	Pod has at least one container being throttled.	in a few seconds	🔗 ⚙️ More ▾

Prod > OS

3 normal | 0 1m | ⚙️ 🔍 ⚡

State	Name	Health	Summary	Next evaluation	Actions
> Normal	[OS K8s] Memory > 85%	ok	Used kubernetes node memory over 85%.	in a few seconds	🔗 ⚙️ More ▾
> Normal	[OS K8s] CPU > 85%	ok	Used kubernetes node cpu over 85%.	in a few seconds	🔗 ⚙️ More ▾
> Normal	[OS Gitlab] Service status	ok	One of the systemd gitlab services is down.	in a few seconds	🔗 ⚙️ More ▾





ArgoCD and Helm charts



Challenges: can we migrate our APM solution?

Really useful for
developers

Only in APM vendor at the
moment

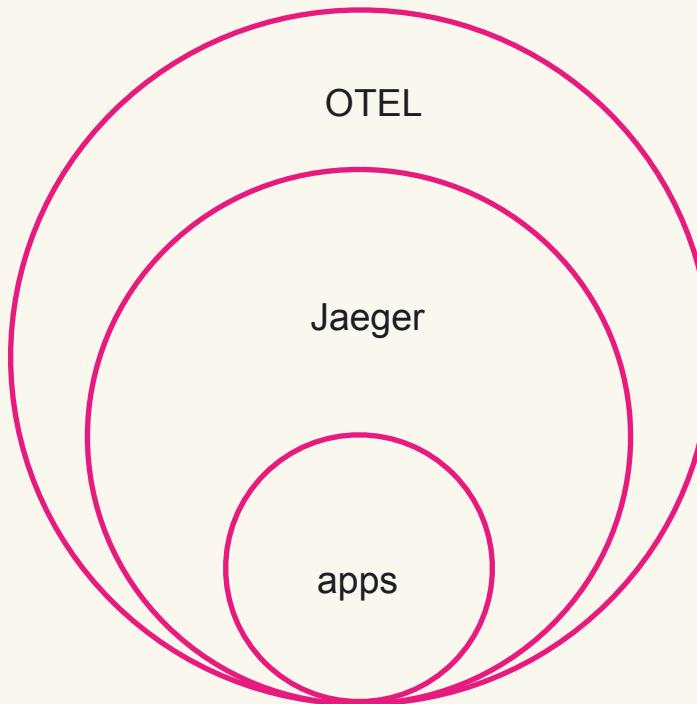
Synthetic monitoring only in
Grafana Cloud

Can ELK be an alternative?

Private cloud limitations

Distributed tracing: OSS





Open Telemetry

Further OTEL integration and customization within our apps is needed for Jaeger operator to work properly and for traces to be sent and seen from and within the applications.

Jaeger

Easy to setup and test, Jaeger operator with K8s clusters was our choice for distributed tracing.

Many programming languages

The necessity to follow the behavior of the apps and account for their traces will play a huge role in our observability.





Lessons learned



OTEL insufficient docs

Testing OTEL with simple apps lead us in the first instance to drop it due to insufficient explanation of official/non-official docs.



Jaeger POC with 3 SDKs

Successful integration of Jaeger Operator with K8s.



Not going to prod

Why? Profiling signals in OTEL



Dashboards

Great, but we want to have lots of visualizations. Can we try Grafana Alloy?



Standby

Documentation and all integrations to become more mature and stable. Still work in progress. Keeping an eye on the OTEL community and further developments.



Stay open

Open to further OSS projects and POCs. Each SRE should POC their own favorite project.



Where to?

We will continue developing our OSS strategy and maintain our OSS tools with Grafana, Prometheus. Further integrating apps with OpenTelemetry considering that it brings a lot value to our monitoring and infrastructure. Achieve consistency on all platforms. Stay open to further OSS POCs and tools.

Improve:

- Alerts, dashboards, customization
- Substitute public cloud APM solutions with OSS Jaeger tracing
- Plan and solve any capacity, scaling or data ingestion questions: where to centralize the data, how many metrics to pull and ingest, which data to choose from
- Communication between developers and SREs
- Incident management: provide uptime status of internal tools, better reaction time to handling incidents
- Culture: accountability and ownership



OpenTelemetry Certified Associate (OTCA)

As cloud native systems grow more complex, the demand for professionals who can leverage telemetry data is growing rapidly. Open new career paths – prove your expertise in OpenTelemetry – the industry standard for tracing, metrics & logs.

OTCA includes:

- ✓ 12-months to schedule & take the exam
- ✓ Two exam attempts

IMPORTANT: The OTCA certification exam is now available for purchase, but the ability to schedule the OTCA exam will not be available until January 2025.

<https://training.linuxfoundation.org/certification/opentelemetry-certified-associate-otca/>





Open Source projects

<https://opentelemetry.io/docs/languages/>

<https://www.jaegertracing.io/docs/1.24/operator/>

<https://prometheus.io/>

<https://grafana.com/docs/grafana/latest/alerting/>

Community

OTEL-SIG: <https://github.com/open-telemetry/community>

Thanks!

Any questions?

https://github.com/didiViking/Conferences_Talks



<https://www.linkedin.com/in/diana-todea-b2a79968>

CREDITS: This presentation template was created by [Slidesgo](#), and includes icons by [Flaticon](#), and infographics & images by [Freepik](#)

Please keep this slide for attribution