



Evaluating the integration of distributed tracing signals into the CERN Monitoring Infrastructure

Summer student: Maša Nešić

Supervisors: Borja Garrido Bear, Nikolay Tsvetkov, Ignacio Coterillo Coz

Contributors: Guillermo Facundo Colunga, Konstantinos Dalianis

Section: IT-DA-MMS (IT Department; Databases & Analytics Group; Messaging, Monitoring and Streaming Section)

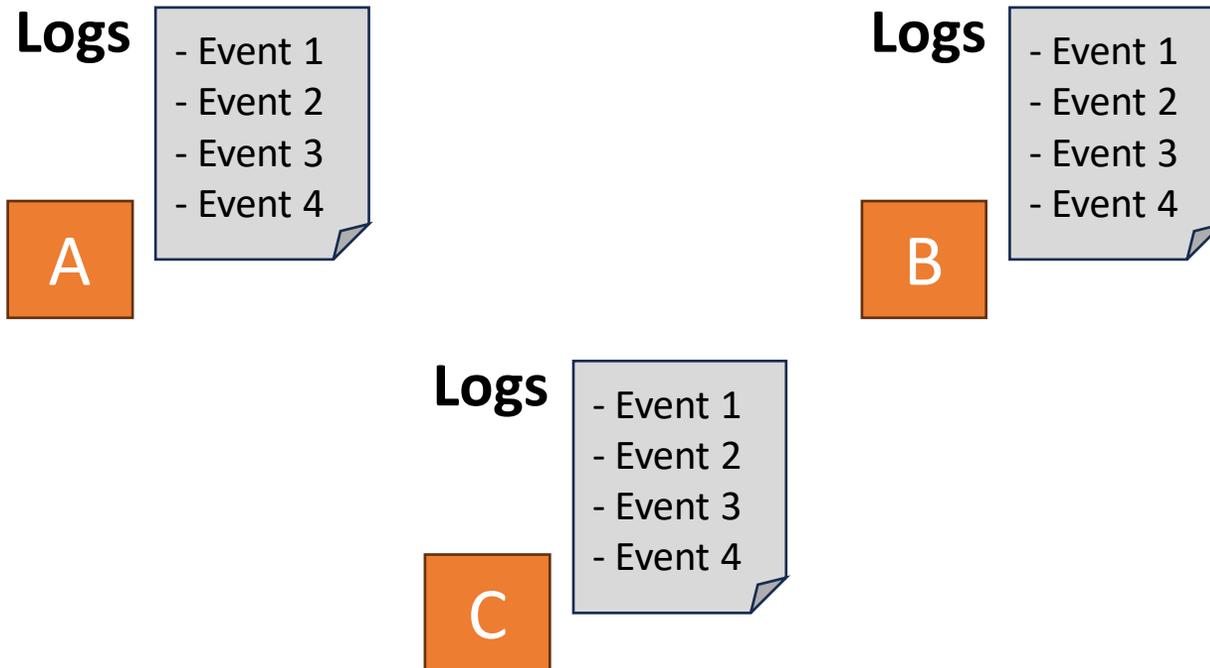
Motivation

- Idea – Make system monitoring easier and more elaborate by introducing distributed tracing
- So far, we have used **logs** and **metrics**



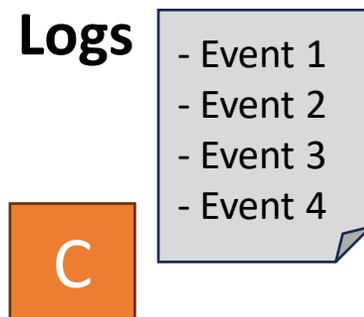
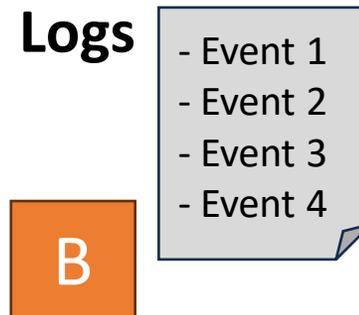
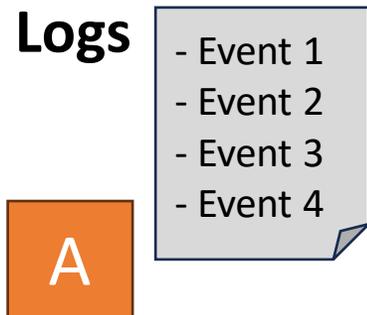
Motivation

- Idea – Make system monitoring easier and more elaborate by introducing distributed tracing
- So far, we have used **logs** and **metrics**

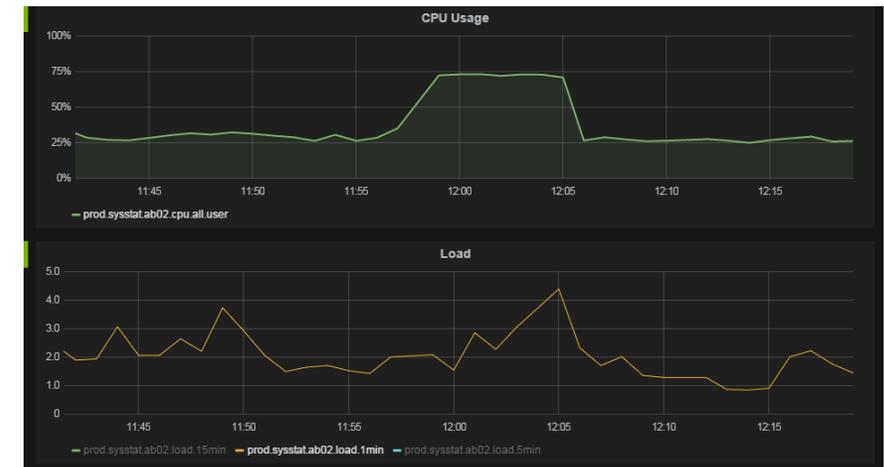


Motivation

- Idea – Make system monitoring easier and more elaborate by introducing distributed tracing
- So far, we have used **logs** and **metrics**



Metrics



Motivation

- We are missing the connection between individual events and system-level state

Logs

- Event 1
- Event 2
- Event 3
- Event 4



Logs

- Event 1
- Event 2
- Event 3
- Event 4

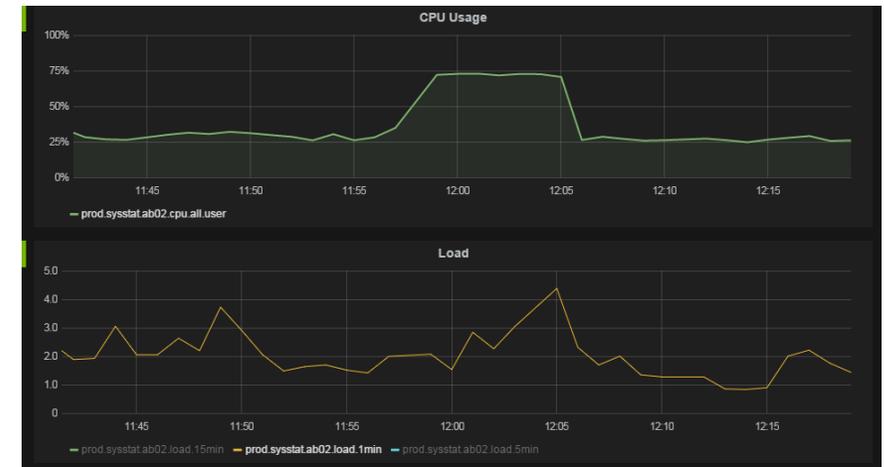


Logs

- Event 1
- Event 2
- Event 3
- Event 4

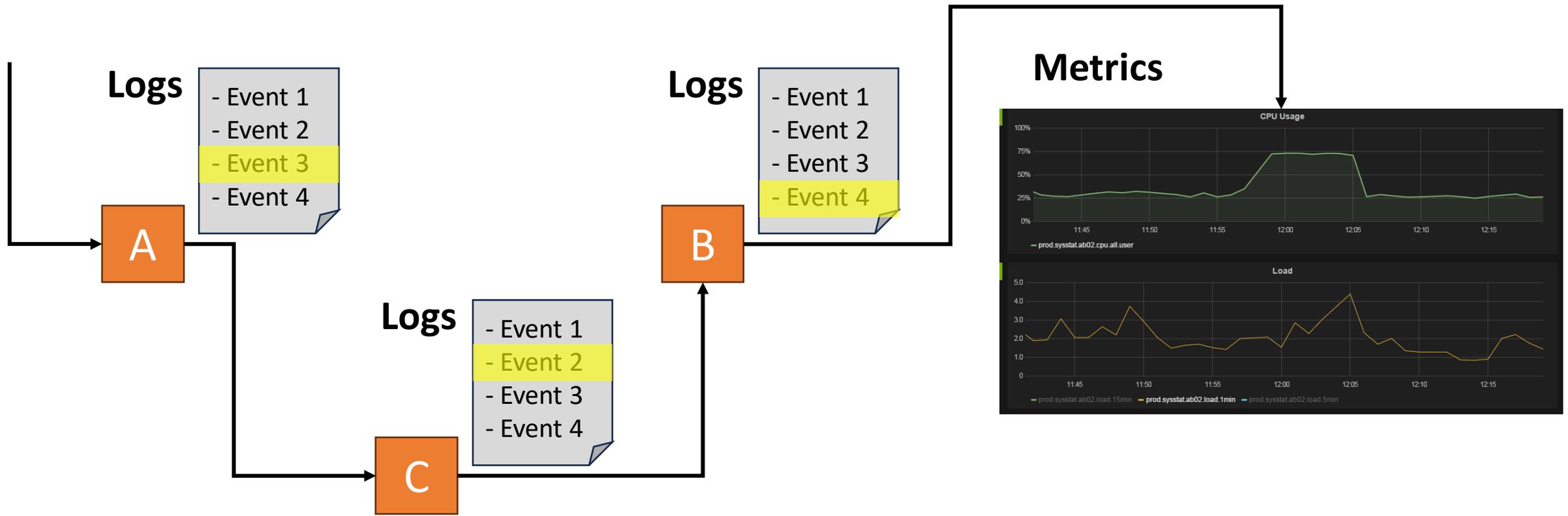


Metrics

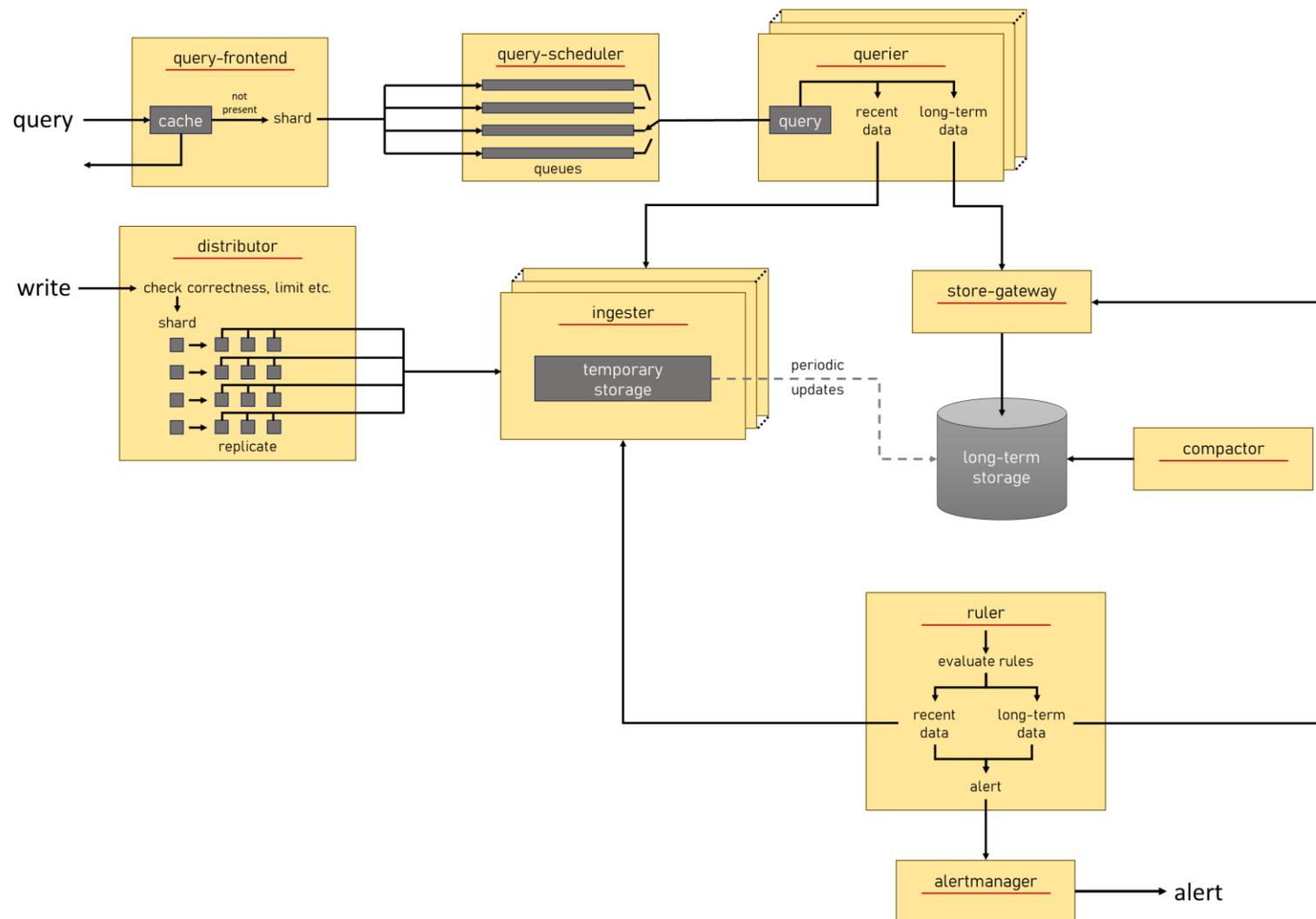


Motivation

- We are missing the connection between individual events and system-level state



Client system example



Results

Search Upload

Service (11)
mimir-query-frontend

Operation (27)
/frontendv2pb.FrontendForQuerier/QueryResult

Tags ⓘ
http.status_code=200 error=true

Lookback
Custom Time Range

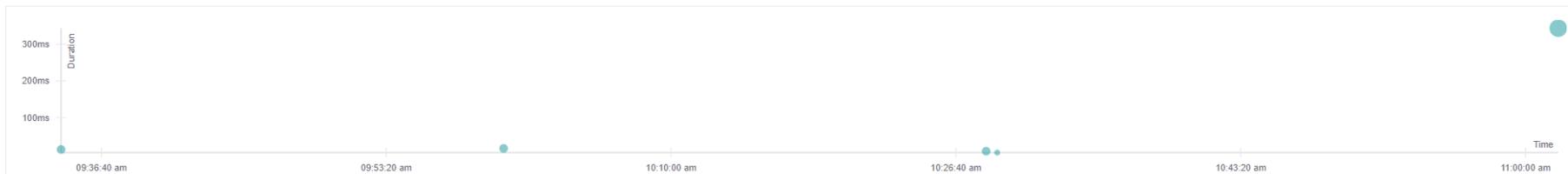
Start Time ⓘ
10-Aug-2023 09:00

End Time ⓘ
10-Aug-2023 11:15

Max Duration Min Duration
e.g. 1.2s, 100ms, 500us e.g. 1.2s, 100ms, 500us

Limit Results
20

Find Traces



5 Traces

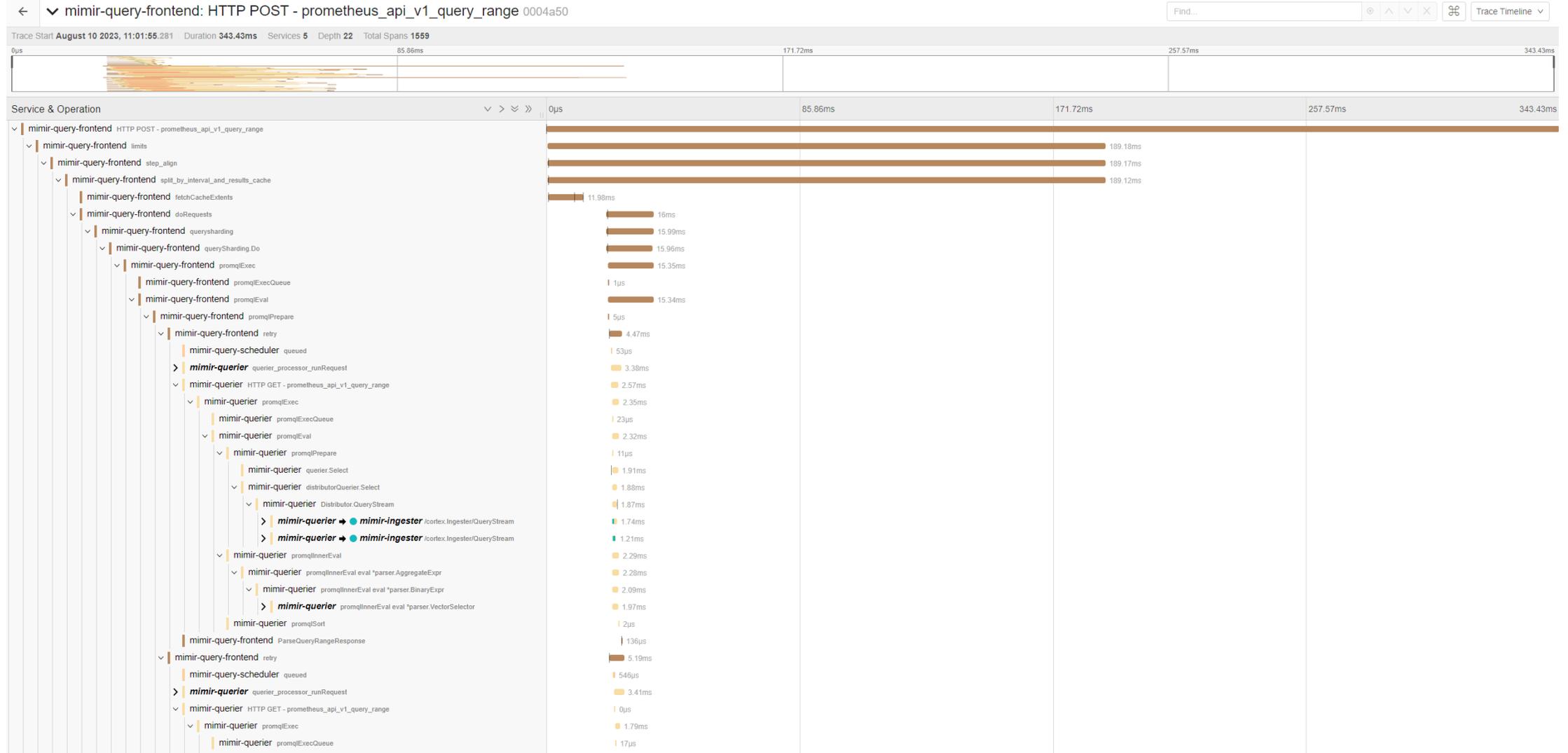
Sort: Most Recent Download Results Deep Dependency Graph

Compare traces by selecting result items

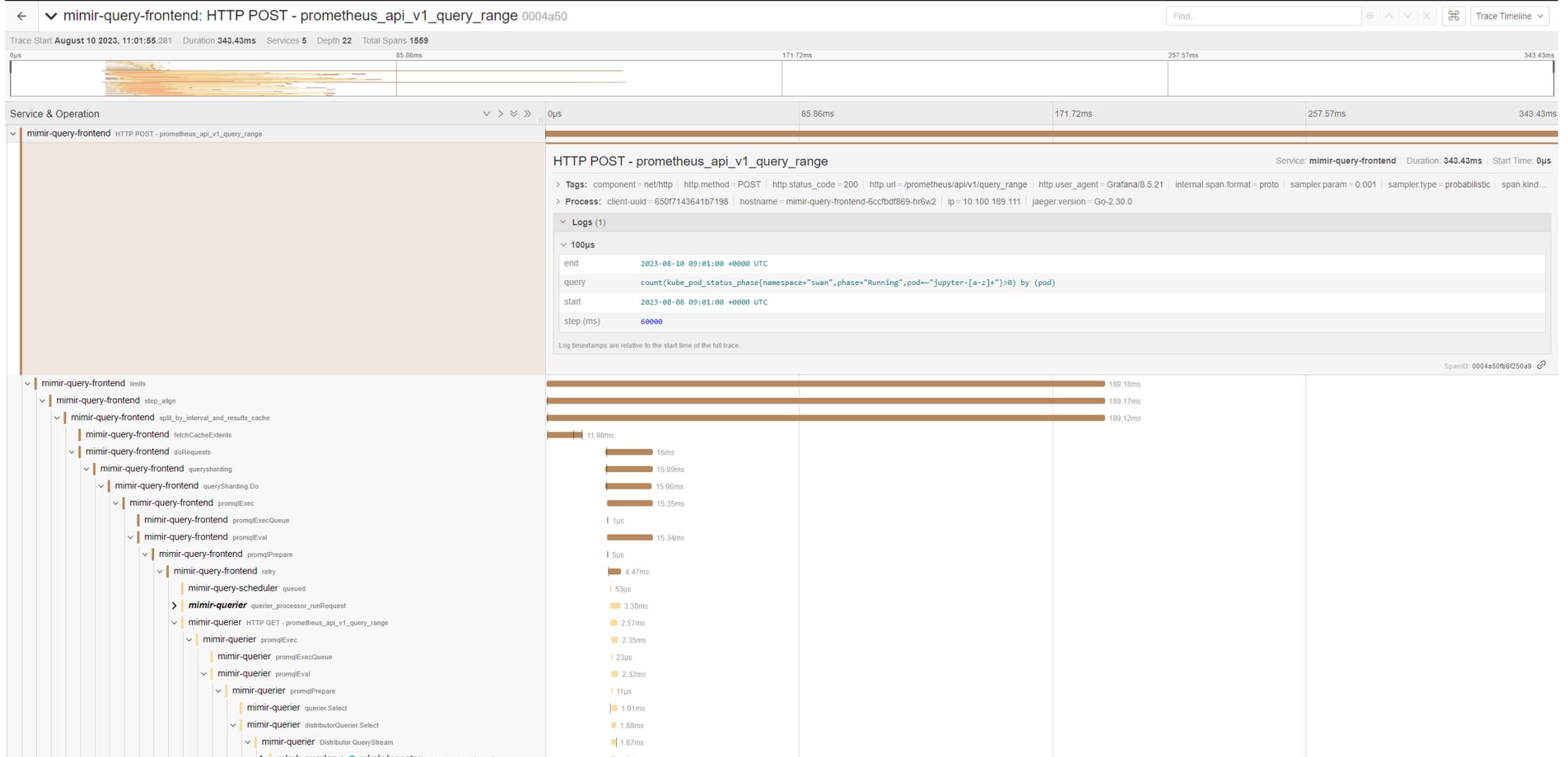
<input type="checkbox"/>	mimir-query-frontend: HTTP POST - prometheus_api_v1_query_range 0004a50	343.43ms
	1559 Spans	mimir-ingester (128) mimir-querier (912) mimir-query-frontend (183) mimir-query-scheduler (48) mimir-store-gateway (288)
		Aug 10 11:01:55 am 4 days ago
<input type="checkbox"/>	mimir-query-frontend: HTTP POST - prometheus_api_v1_query_range 0004d48	5.28ms
	44 Spans	mimir-ingester (8) mimir-querier (23) mimir-query-frontend (12) mimir-query-scheduler (1)
		Aug 10 10:29:05 am 4 days ago
<input type="checkbox"/>	mimir-query-frontend: HTTP POST - prometheus_api_v1_query_range 0005f7b	9.06ms
	401 Spans	mimir-ingester (64) mimir-querier (256) mimir-query-frontend (65) mimir-query-scheduler (16)
		Aug 10 10:28:26 am 4 days ago
<input type="checkbox"/>	mimir-query-frontend: HTTP POST - prometheus_api_v1_query_range 00176c9	16.62ms
	404 Spans	mimir-ingester (64) mimir-querier (256) mimir-query-frontend (68) mimir-query-scheduler (16)
		Aug 10 10:00:12 am 4 days ago
<input type="checkbox"/>	mimir-query-frontend: HTTP POST - prometheus_api_v1_query_range 00041bd	13.71ms
	404 Spans	mimir-ingester (64) mimir-querier (256) mimir-query-frontend (68) mimir-query-scheduler (16)
		Aug 10 9:34:18 am 4 days ago



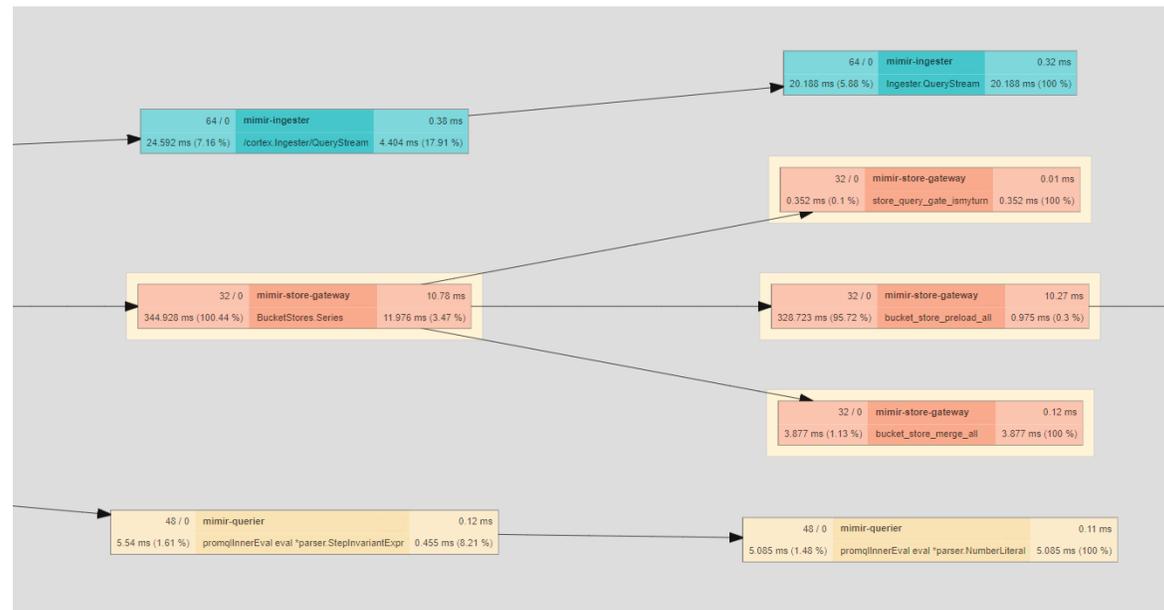
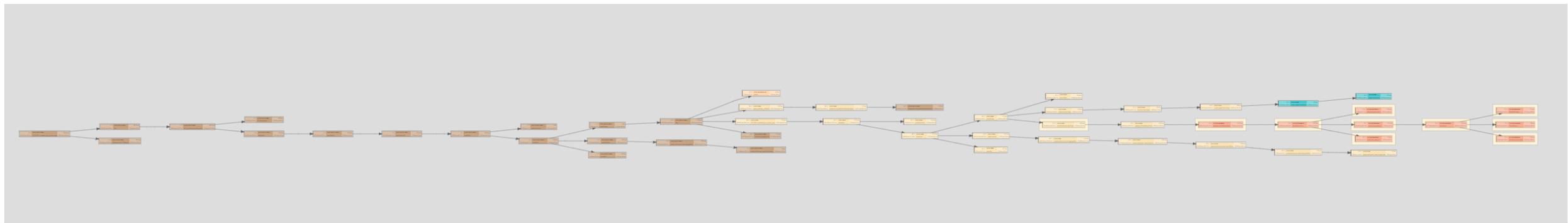
Results



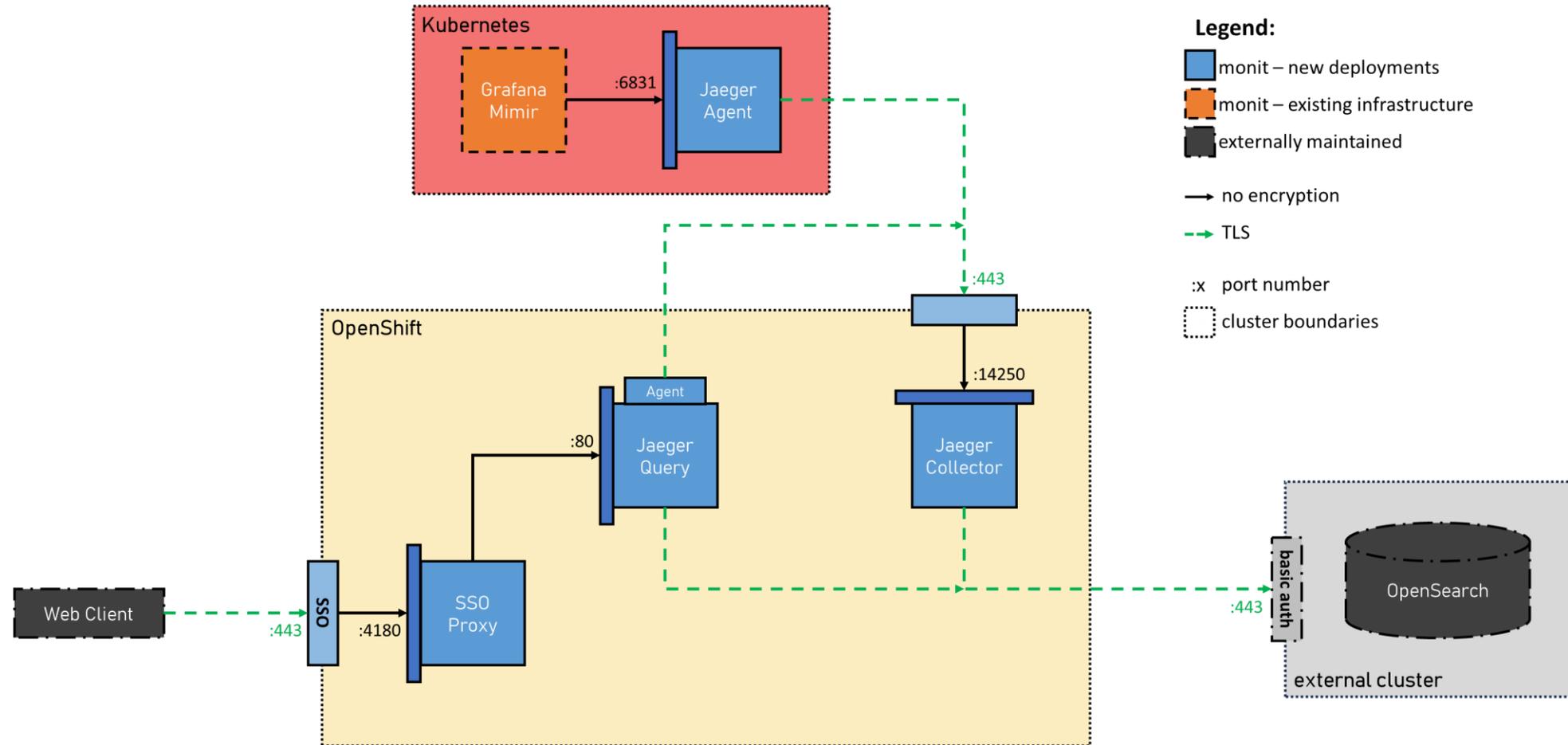
Results



Results

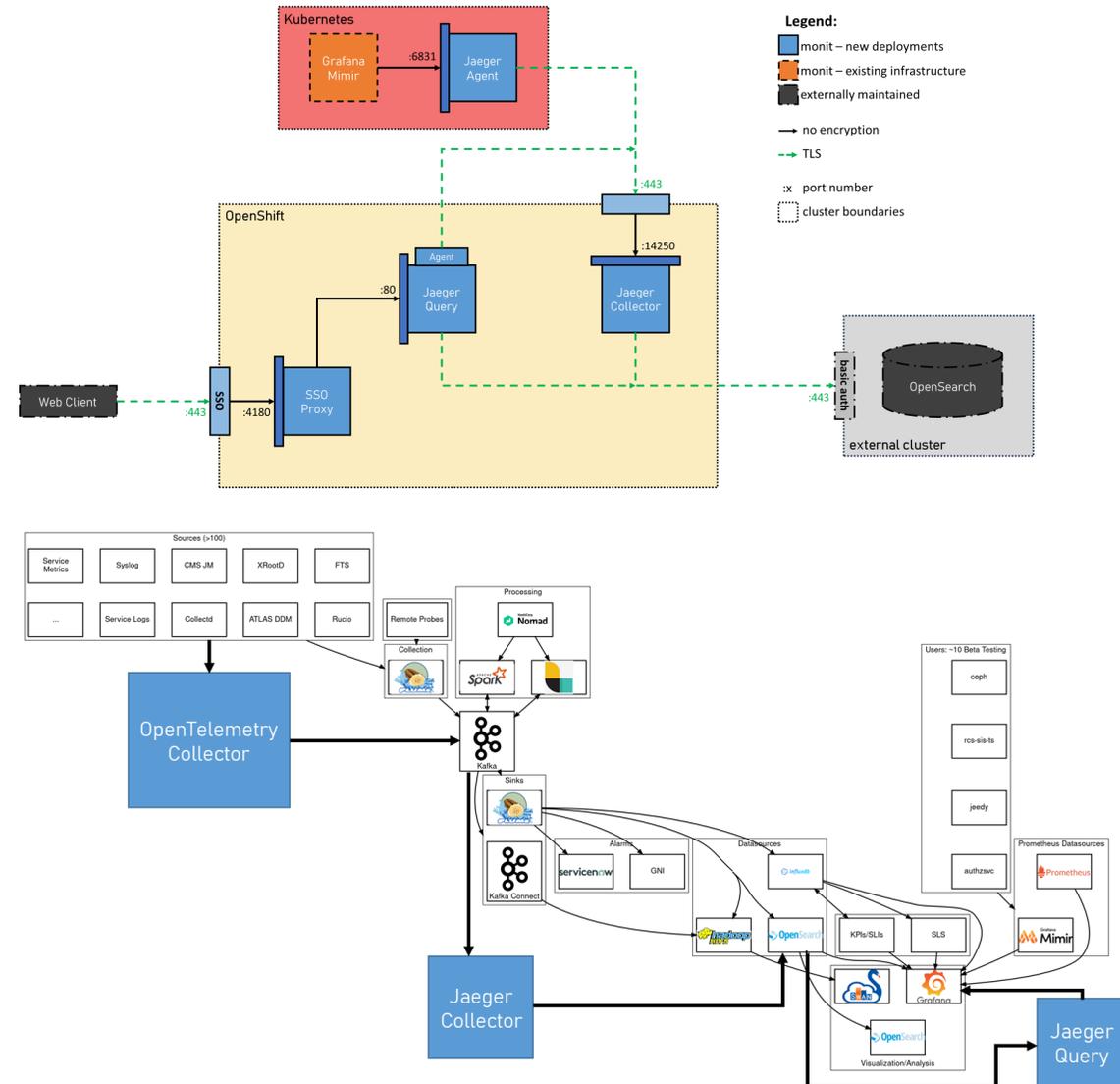


Implementation – system architecture



Next steps

- Explore multitenancy
 - collection side
 - presentation side
- Integration with different client types
- Unified handling of logs, metrics and traces



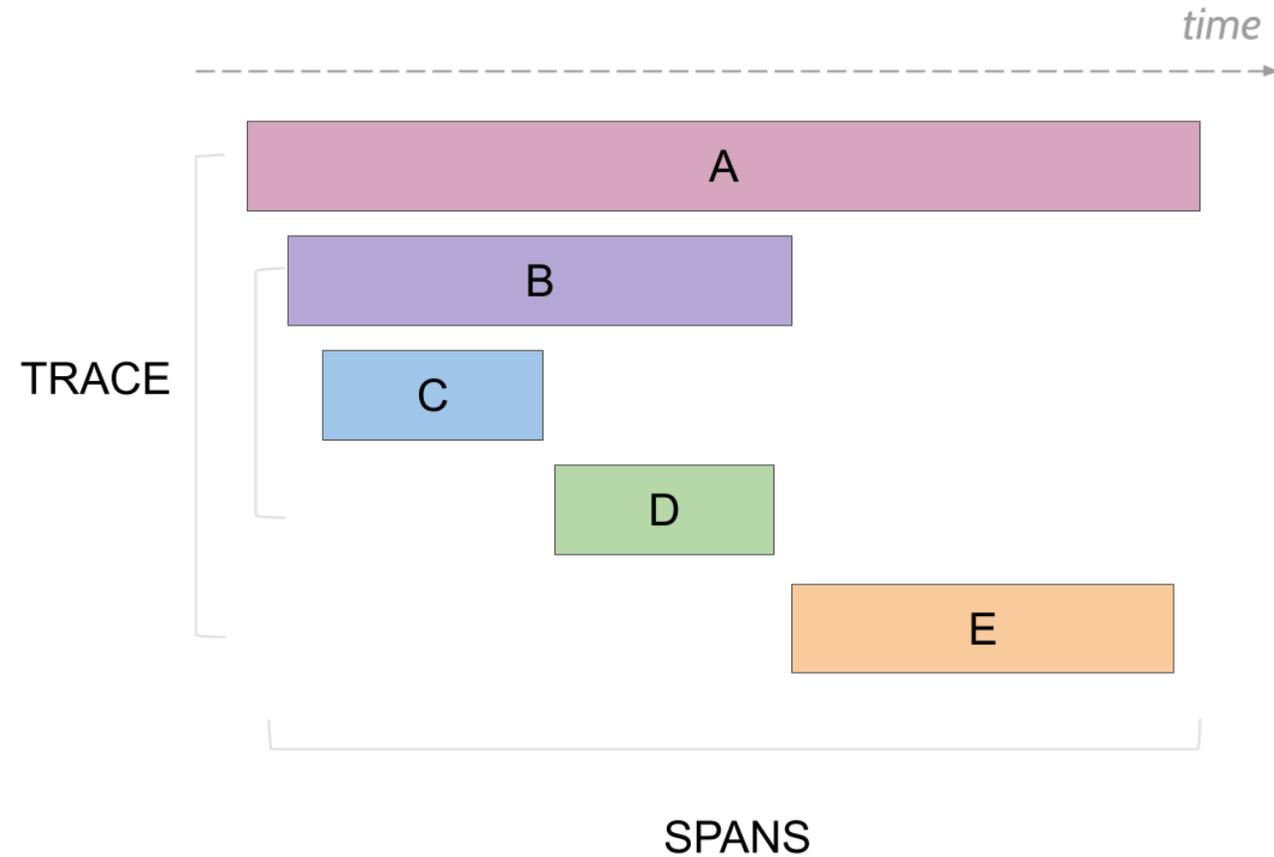
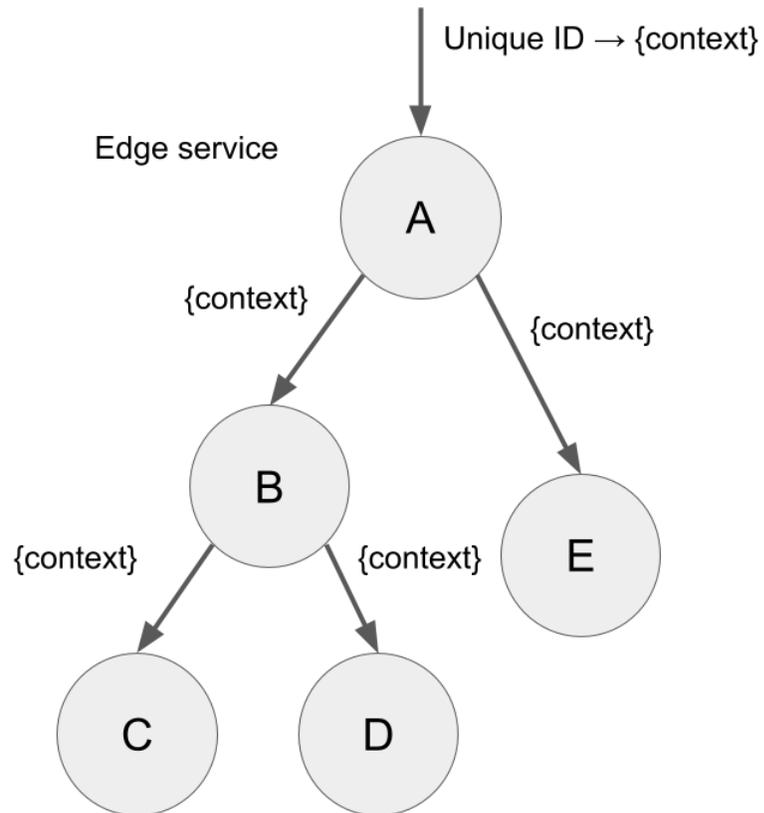
Summary

- Idea:
 - Make system monitoring easier and more elaborate by introducing distributed tracing
- Technical goals:
 - Create a proof-of-concept deployment of the distributed tracing infrastructure
 - Assess the advantages it brings
 - Evaluate possibilities for its integration with the existing monitoring infrastructure
- Further steps:
 - Exploring multitenancy
 - Intuitive integration with different client types
 - Unified handling of logs, metrics and traces



Appendix

Distributed tracing



Integration into the Monitoring infrastructure

