Deploy, Scale and Extend Jaeger

CloudNativeCon Europe 2019 - Barcelona, Spain

Louis-Etienne Dorval

ticketmaster

Agenda

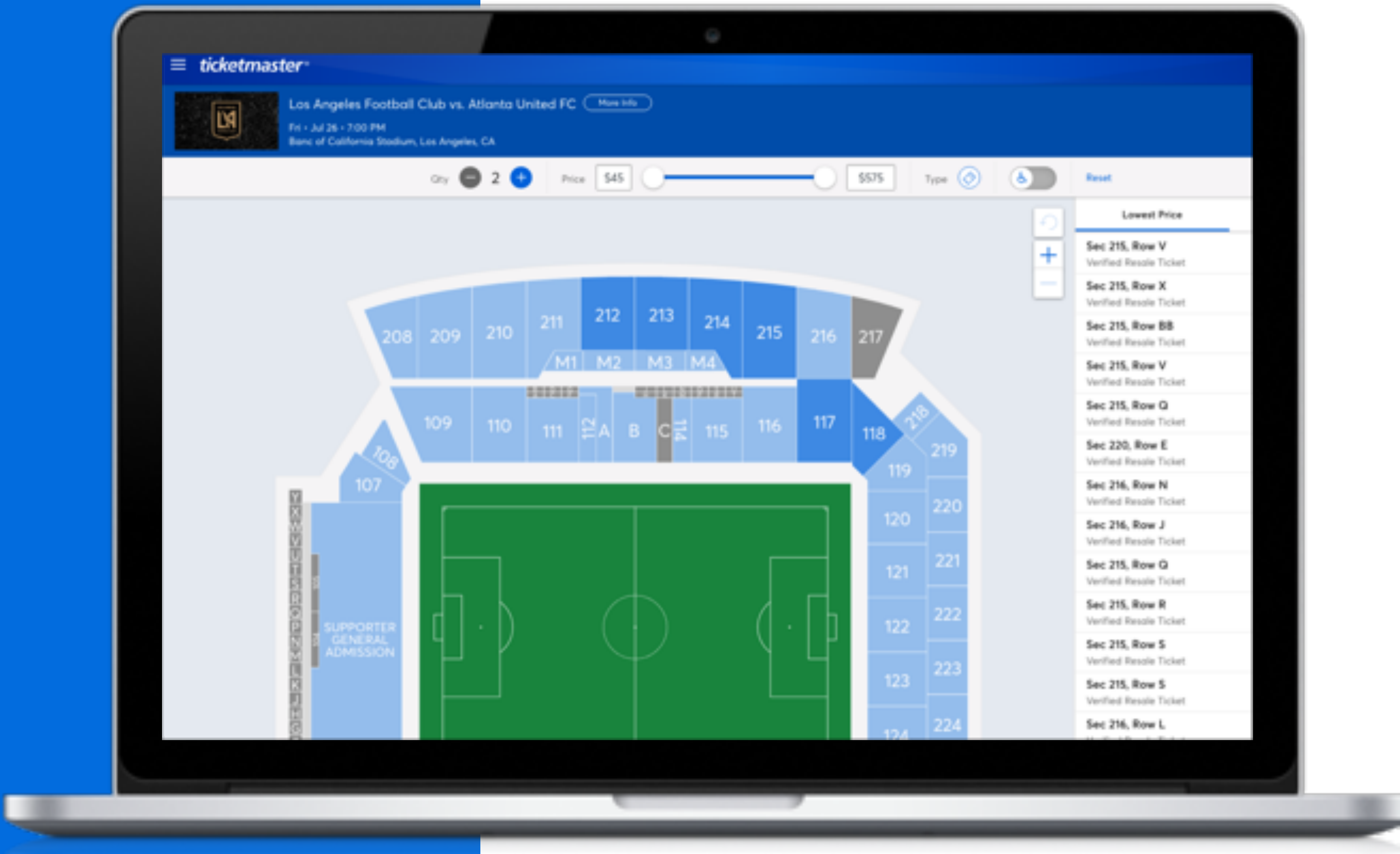
1. Introduction
2. Adoption
3. Deploy
4. Scale
5. Extend

What is Ticketmaster?

Power unforgettable moments of joy

What is Ticketmaster?

On sale



ticketmaster

Los Angeles Football Club vs. Atlanta United FC More Info

Fri • Jul 26 • 7:00 PM
Banc of California Stadium, Los Angeles, CA

Qty 2 Price $45 $575 Type Reset

Lowest Price

- Sec 215, Row V
  Verified Resale Ticket
- Sec 215, Row X
  Verified Resale Ticket
- Sec 215, Row BB
  Verified Resale Ticket
- Sec 215, Row V
  Verified Resale Ticket
- Sec 215, Row Q
  Verified Resale Ticket
- Sec 220, Row E
  Verified Resale Ticket
- Sec 216, Row N
  Verified Resale Ticket
- Sec 216, Row J
  Verified Resale Ticket
- Sec 215, Row Q
  Verified Resale Ticket
- Sec 215, Row R
  Verified Resale Ticket
- Sec 215, Row S
  Verified Resale Ticket
- Sec 215, Row S
  Verified Resale Ticket
- Sec 216, Row L

What is Ticketmaster?

Entry

What is Ticketmaster?

Behind the scene

40 years of innovation

27 Ticketing Systems and over 300 products

Hybrid Cloud

20,000+ VMs

7 data centers

15+ Kubernetes clusters

What is Ticketmaster?

Behind the scene

Java

Perl

C

NodeJS

Rust

Go

C#

And probably more!

What is Ticketmaster?

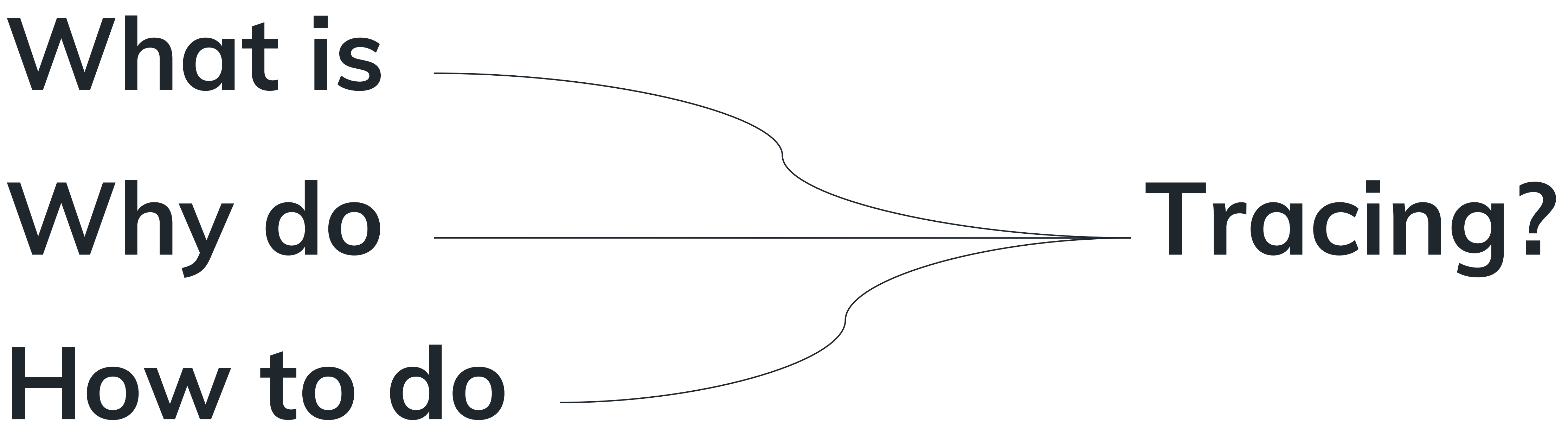
Technology

What is Ticketmaster?

Technology

VAX in Kubernetes (minikube)

Why Tracing?



What is

Why do

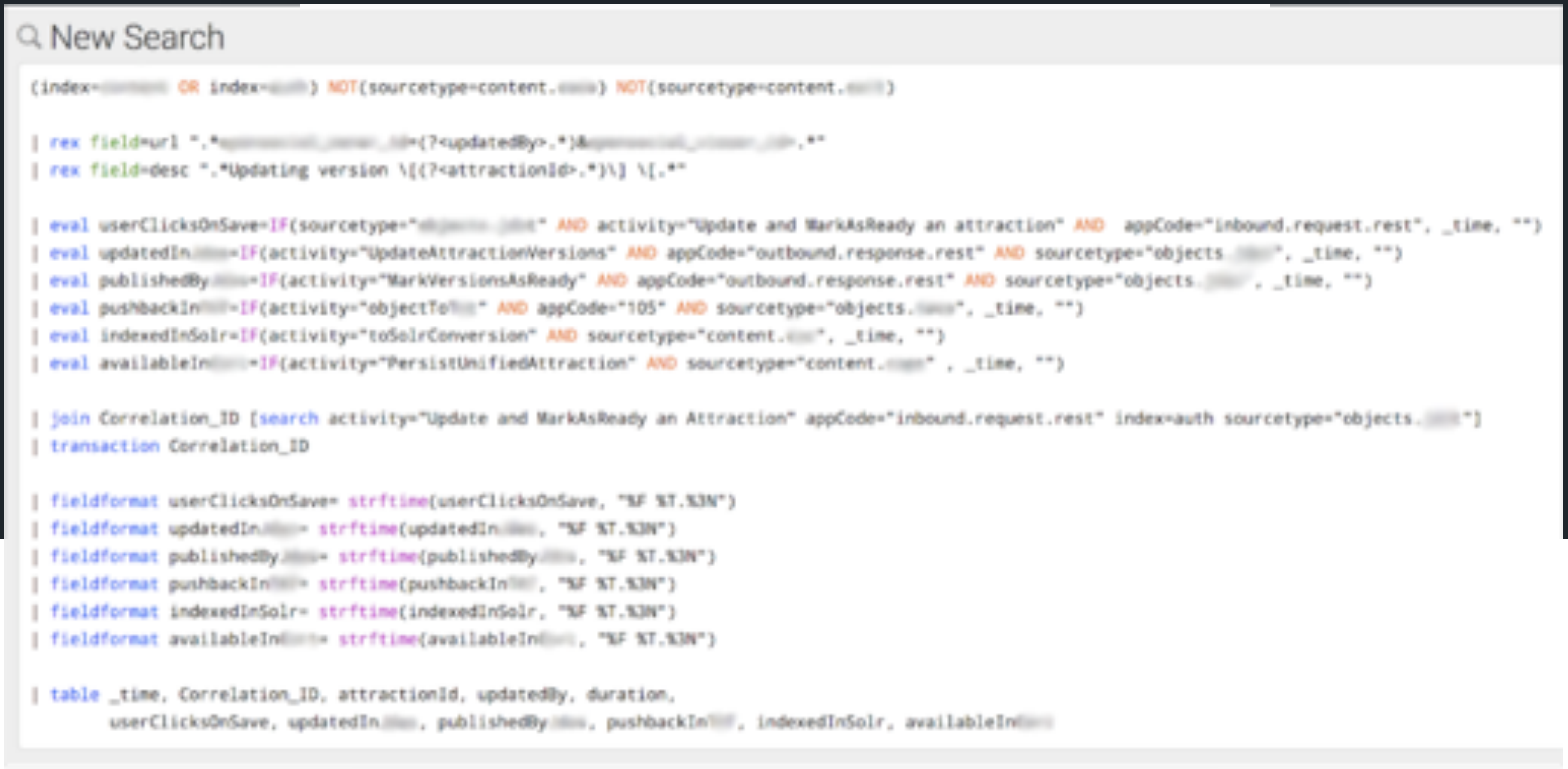
How to do

Tracing?

- Would You Like Some Tracing With Your Monitoring? - Yuri Shkuro, Uber Technologies
  - CloudNativeCon 2017 (https://sched.co/CU8f)
- Service Meshes and Observability - Ben Sigelman, Lightstep
  - CloudNativeCon 2017 (https://sched.co/CUCX)
- Distributed Tracing in Serverless Systems - Nitzan Shapira, Epsagon
  - CloudNativeCon 2018 (https://sched.co/GrXp)
- Understanding Microservices with Distributed Tracing - Lita Cho, Lyft
  - CloudNativeCon 2018 (https://sched.co/GrXj)

Introduction

Why Tracing?



New Search

```
(index=[redacted] OR index=[redacted]) NOT(sourcetype=content.[redacted]) NOT(sourcetype=content.[redacted])

| rex field=url ".*[redacted]=(?<updatedBy>.*)&[redacted].*"
| rex field=desc ".*Updating version \[(?<attractionId>.*)\] \[.*"

| eval userClicksOnSave=IF(sourcetype="[redacted]" AND activity="Update and MarkAsReady an attraction" AND  appCode="inbound.request.rest", _time, "")
| eval updatedIn[redacted]=IF(activity="UpdateAttractionVersions" AND appCode="outbound.response.rest" AND sourcetype="objects.[redacted]", _time, "")
| eval publishedBy[redacted]=IF(activity="MarkVersionsAsReady" AND appCode="outbound.response.rest" AND sourcetype="objects.[redacted]" , _time, "")
| eval pushbackIn[redacted]=IF(activity="objectTo[redacted]" AND appCode="105" AND sourcetype="objects.[redacted]", _time, "")
| eval indexedInSolr=IF(activity="toSolrConversion" AND sourcetype="content.[redacted]", _time, "")
| eval availableIn[redacted]=IF(activity="PersistUnifiedAttraction" AND sourcetype="content.[redacted]" , _time, "")

| join Correlation_ID [search activity="Update and MarkAsReady an Attraction" appCode="inbound.request.rest" index=auth sourcetype="objects.[redacted]"]
| transaction Correlation_ID

| fieldformat userClicksOnSave= strftime(userClicksOnSave, "%F %T.%3N")
| fieldformat updatedIn[redacted]= strftime(updatedIn[redacted], "%F %T.%3N")
| fieldformat publishedBy[redacted]= strftime(publishedBy[redacted], "%F %T.%3N")
| fieldformat pushbackIn[redacted]= strftime(pushbackIn[redacted], "%F %T.%3N")
| fieldformat indexedInSolr= strftime(indexedInSolr, "%F %T.%3N")
| fieldformat availableIn[redacted]= strftime(availableIn[redacted], "%F %T.%3N")

| table _time, Correlation_ID, attractionId, updatedBy, duration,
      userClicksOnSave, updatedIn[redacted], publishedBy[redacted], pushbackIn[redacted], indexedInSolr, availableIn[redacted]
```

JAEGER

Jaeger

Current Scale

55+ services instrumented

Java
NodeJS
Go
C#
Envoy (C++)

Up to 9,000 spans per second

2.1B spans in Elasticsearch

1.2TB of data in Elasticsearch

Adoption

Adoption

Work with development teams

Framework coverage

Main language used?

Common service template?

Common set of framework?

Documentation

How to instrument?

Where can I get help?

Best practices?

How to deploy?

Local development?

Adoption

Start somewhere

1. Instrument a few frameworks

    Pro Tip: You might not have to do anything, look up opentracing.io/registry first

1. Add a little documentation

2. Try locally on a real project

3. Fix issues

4. Work with the team to have it deployed

5. Fix issues

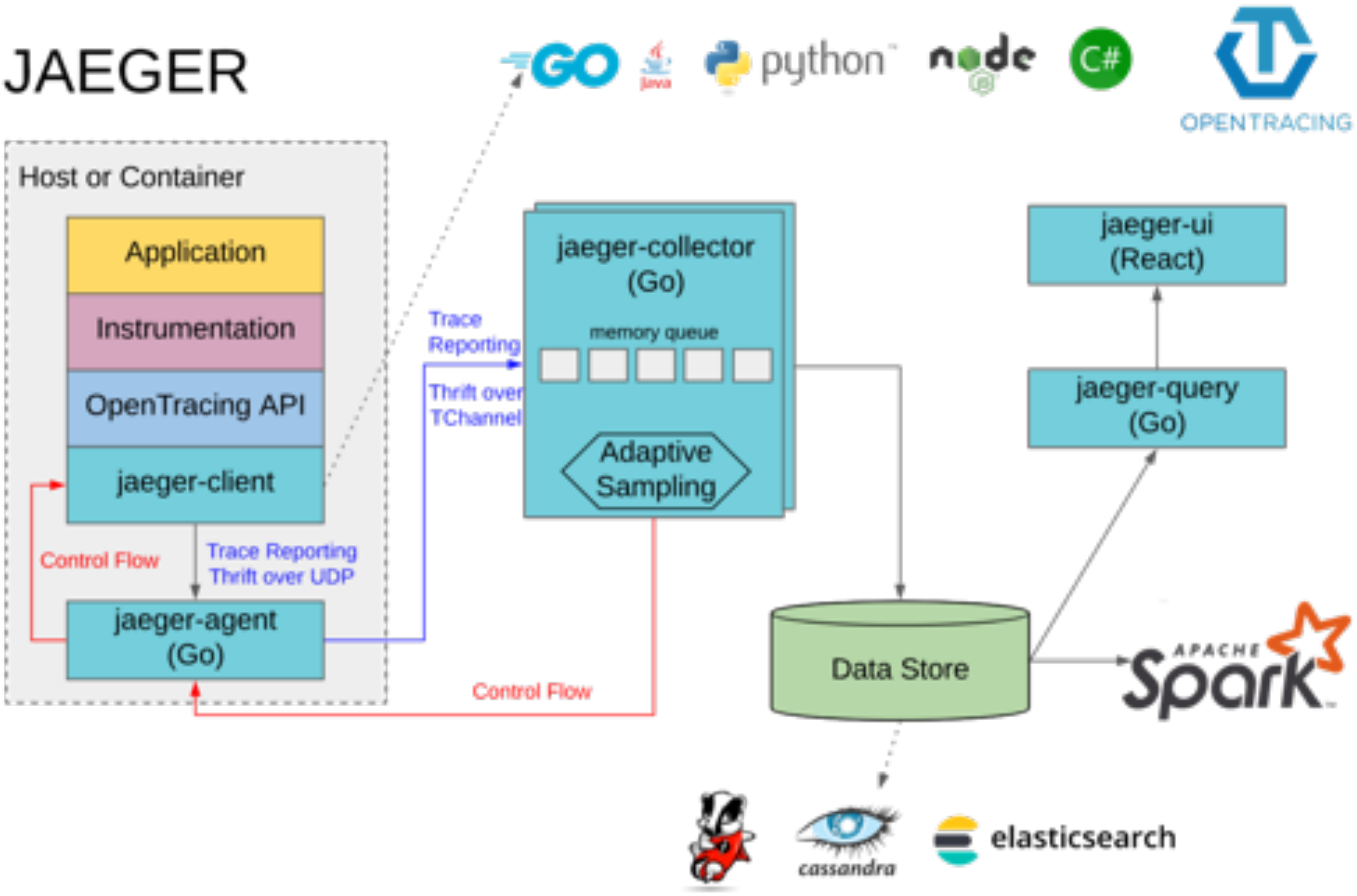
6. Go to #1

Adoption

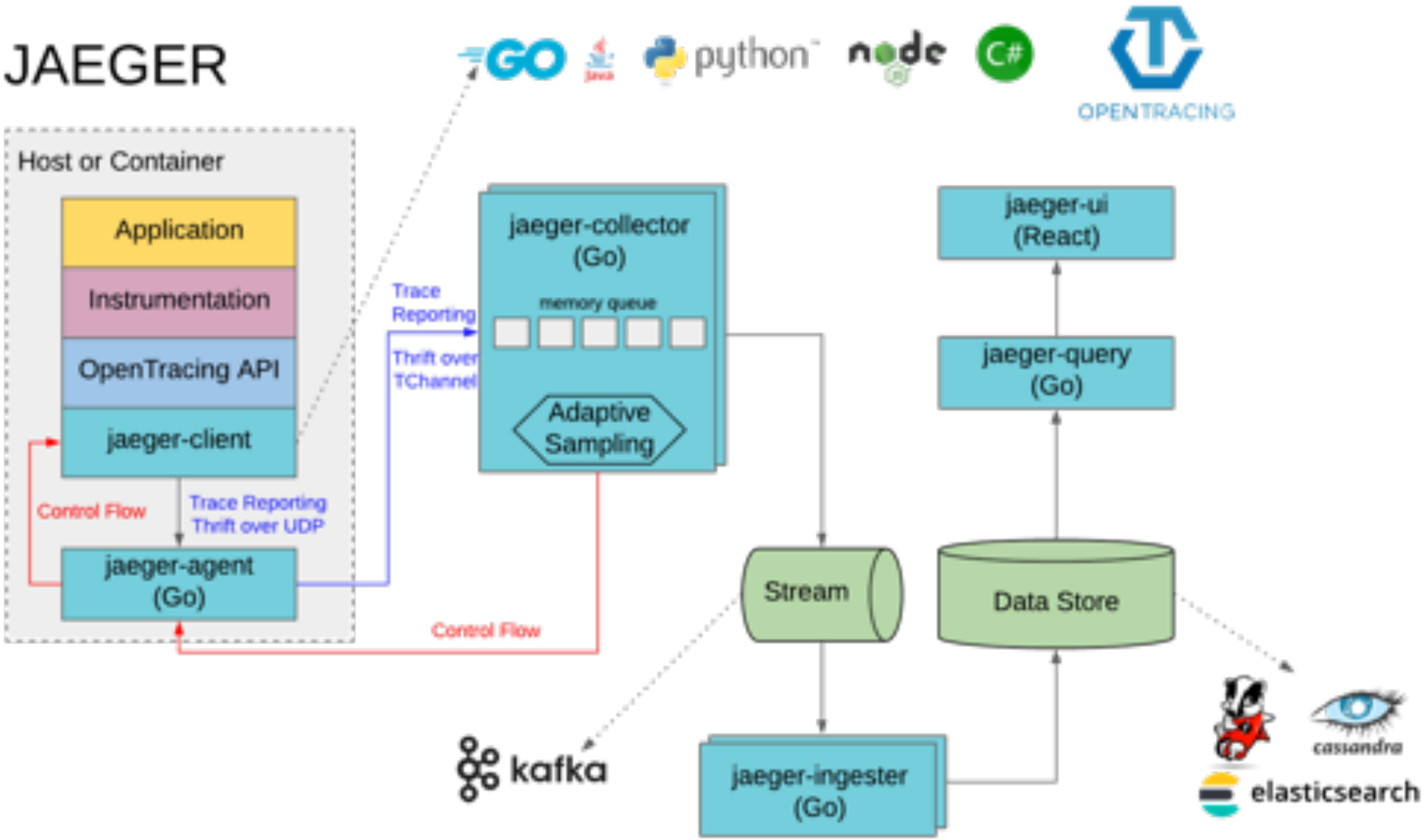
Snowball effect

Deploy Jaeger

JAEGER


GO
Java
python
node
C#
OPENTRACING
Host or Container
Application
Instrumentation
OpenTracing API
jaeger-client
Control Flow
Trace Reporting
Thrift over UDP
jaeger-agent
(Go)
Trace
Reporting
Thrift over
TChannel
jaeger-collector
(Go)
memory queue
Adaptive
Sampling
Control Flow
Data Store
jaeger-ui
(React)
jaeger-query
(Go)
APACHE
Spark
cassandra
elasticsearch

JAEGER



Host or Container

Application

Instrumentation

OpenTracing API

jaeger-client

Control Flow

Trace Reporting
Thrift over UDP

jaeger-agent
(Go)

Trace
Reporting

Thrift over
TChannel

jaeger-collector
(Go)

memory queue

Adaptive
Sampling

Control Flow

Stream

kafka

jaeger-ingester
(Go)

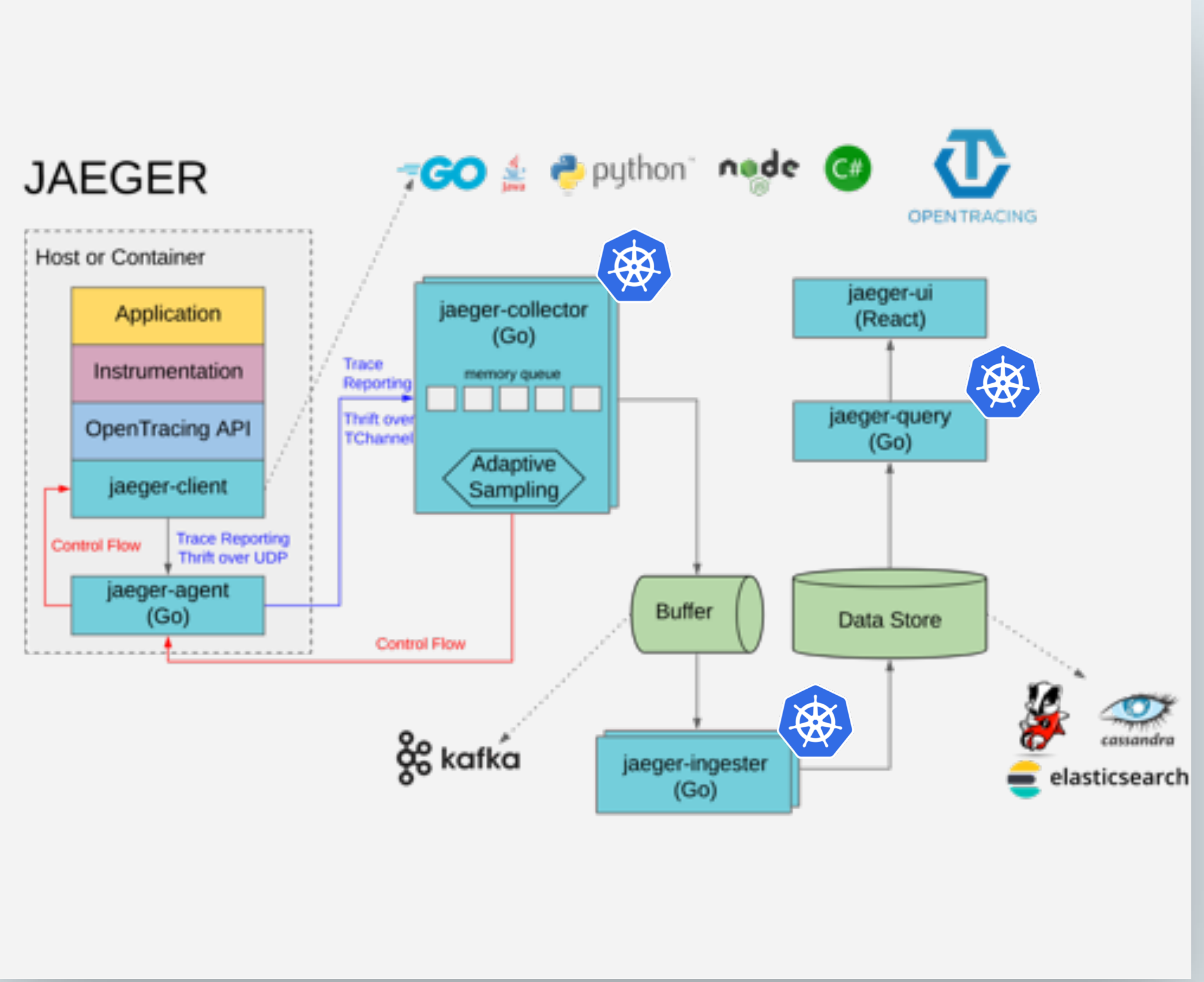
Data Store

jaeger-query
(Go)

jaeger-ui
(React)

cassandra

elasticsearch



Deployment

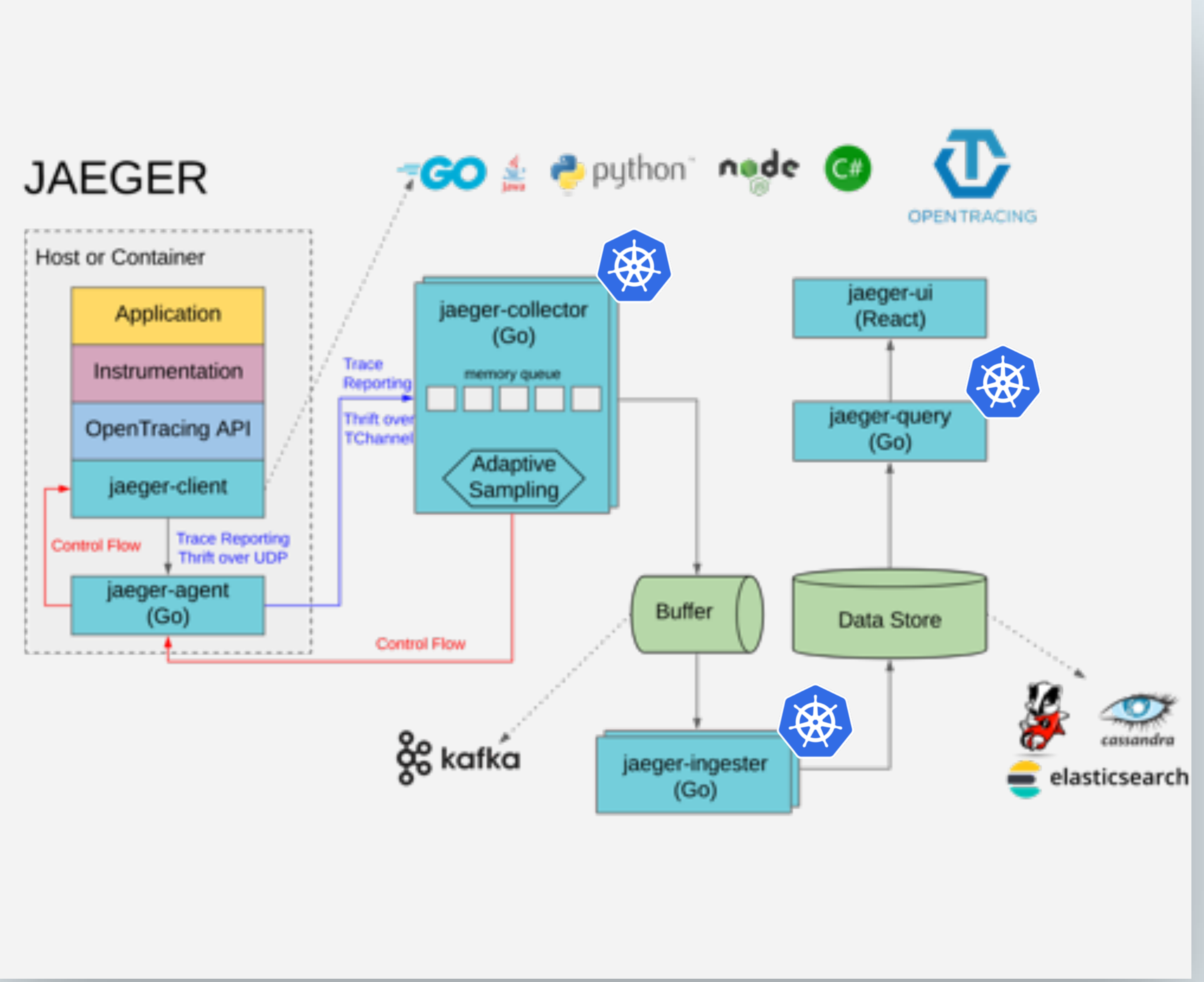
Kubernetes

Deployment (apps/v1)

- Jaeger Collector
- Jaeger Query
- Jaeger Ingester

CronJob (batch/v1beta1)

- es-index-cleaner
- spark-dependencies



Deployment

Kubernetes

Service (v1) - Type LoadBalancer

- Jaeger Collector
  - ELB in AWS

Ingress (extensions/v1beta1)

- Jaeger Query
  - ALB in AWS

Deployment

Jaeger Agent

Linux and Windows VM
- Binary → Daemon

Kubernetes
- Docker → Sidecar

AWS EC2
- Docker → Sidecar
  (docker run --link)

Sidecar

⊕ Isolation

Security

⊖ Resource utilization

Added complexity outside Kubernetes ⚠

Daemon

⊕ Resource utilization

Added complexity inside Kubernetes ⚠

⊖ Isolation

Security

Deployment

Jaeger Agent

⚠

1. If the the Jaeger Client can't reach the Jaeger-Agent on localhost, you need to set:
   a. JAEGER_AGENT_HOST
   b. JAEGER_SAMPLER_MANAGER_HOST_PORT

2. The value of the 1b above vary from one language to another
   [jaeger-client-go #362]()

Sidecar

⊕ Isolation

Security

⊖ Resource utilization

Added complexity outside Kubernetes ⚠

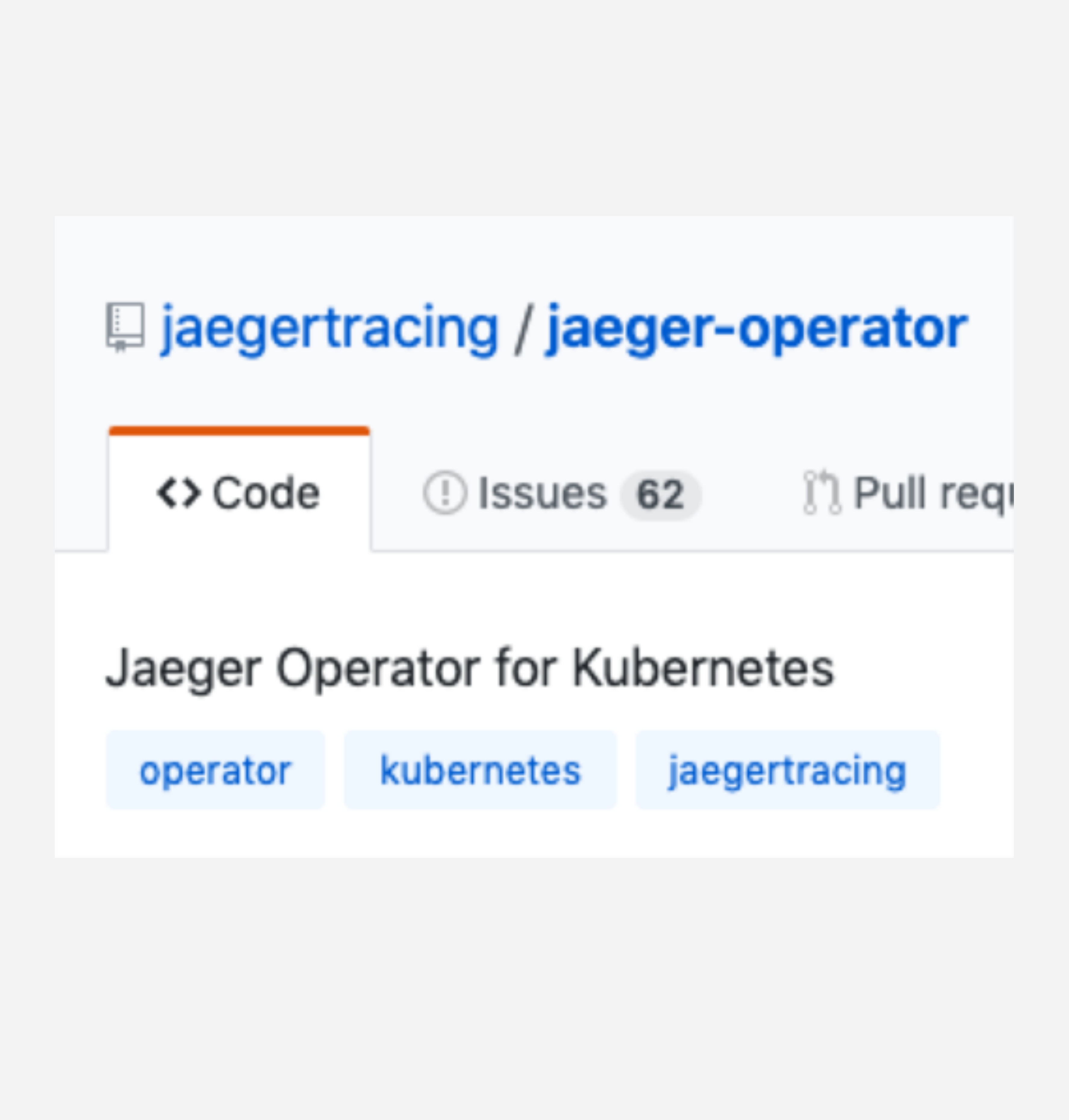
Daemon

⊕ Resource utilization

Added complexity inside Kubernetes ⚠

⊖ Isolation

Security



jaegertracing / jaeger-operator

Jaeger Operator for Kubernetes

Deployment

Jaeger Operator for Kubernetes

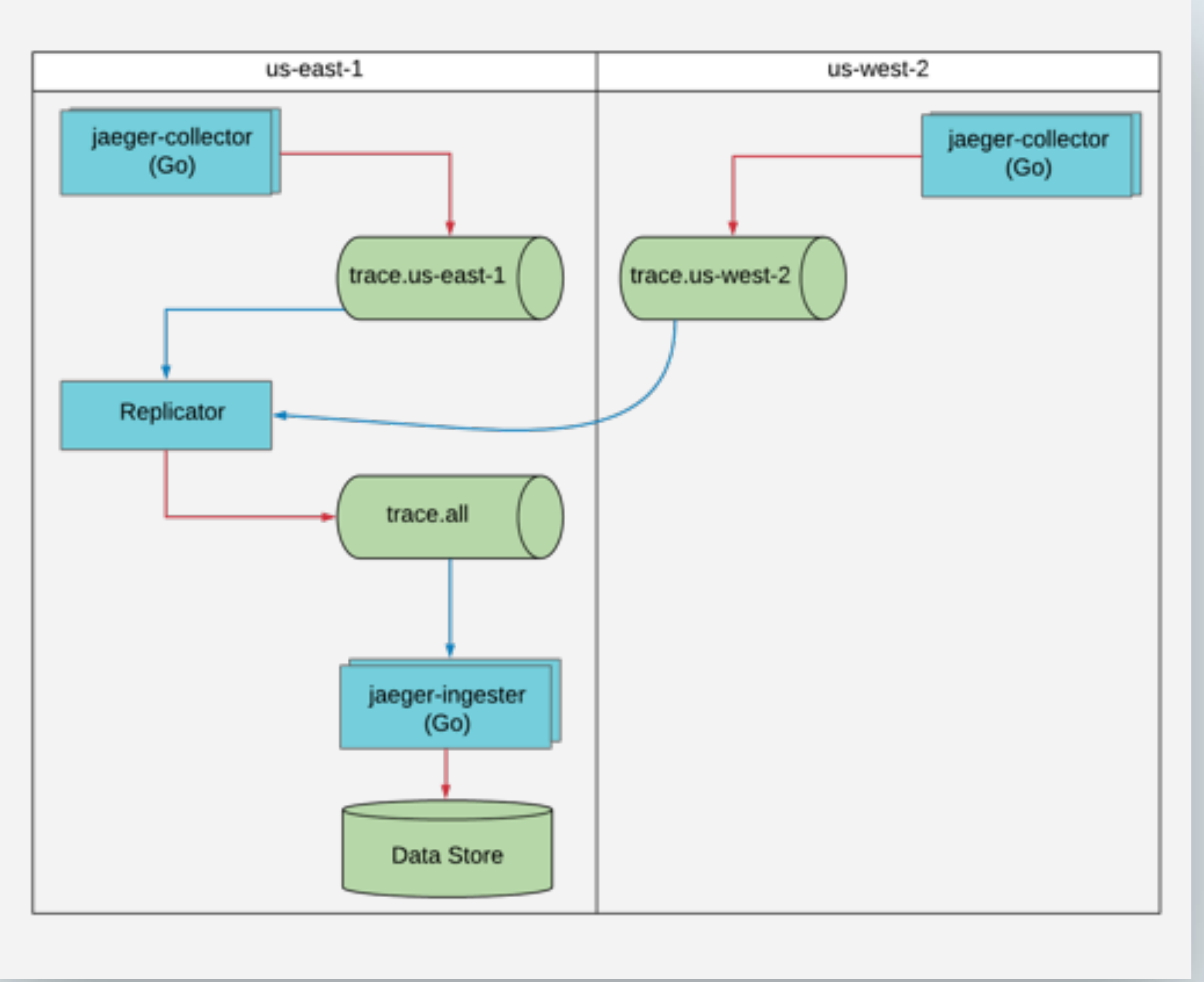
- Deployment strategies
  - All In One
  - Production
  - Streaming

- Jaeger Agent strategies
  - Sidecar (auto injection supported)
  - DaemonSet

- Version upgrades

Deployment

Multi-Region (single datastore)

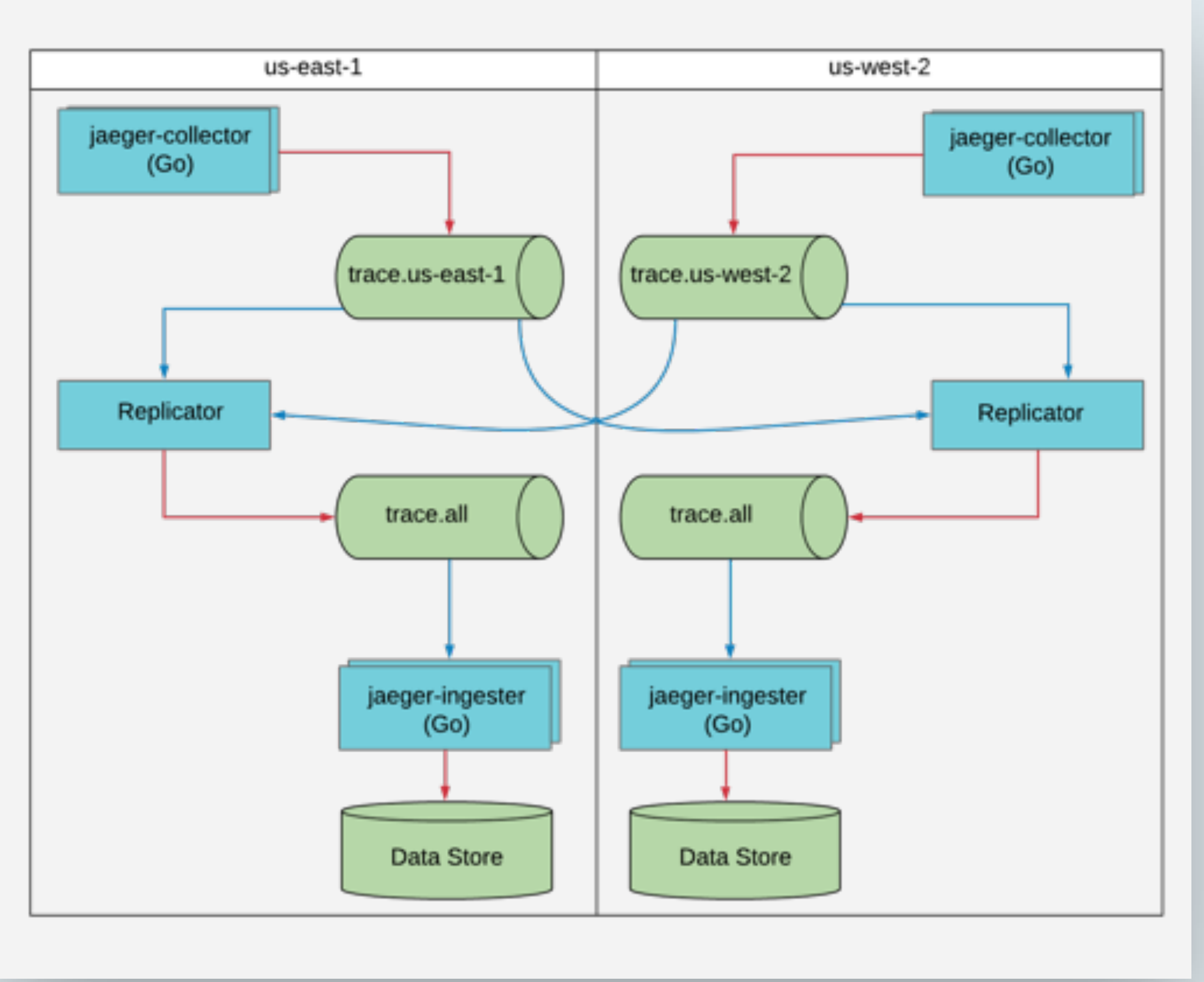


Write data to Kafka →

Read data from Kafka →

Deployment

Multi-region (multiple datastores)



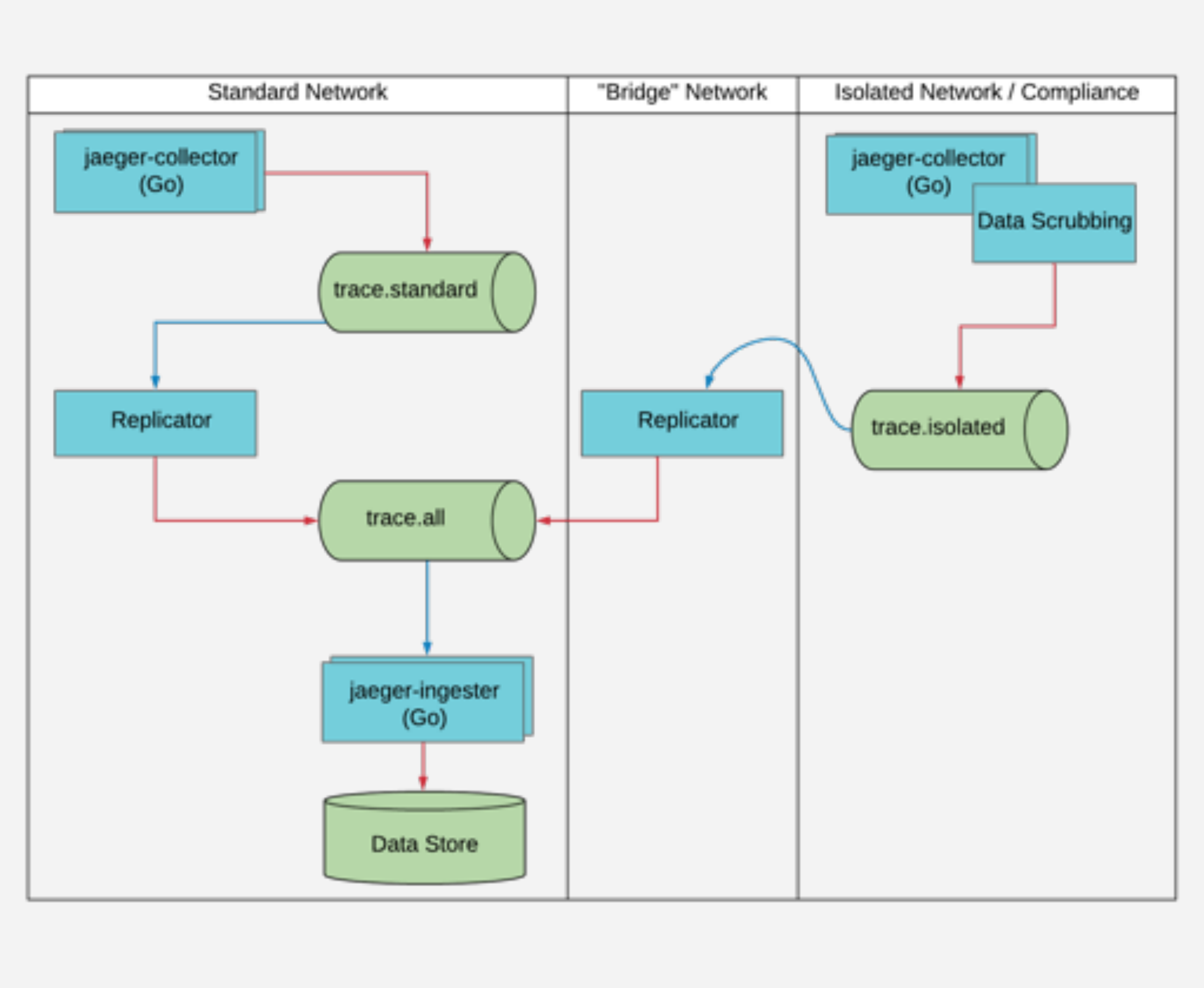
APACHE kafka®

Write data to Kafka 🟥

Read data from Kafka 🟦

Deployment

Isolated network / Compliance

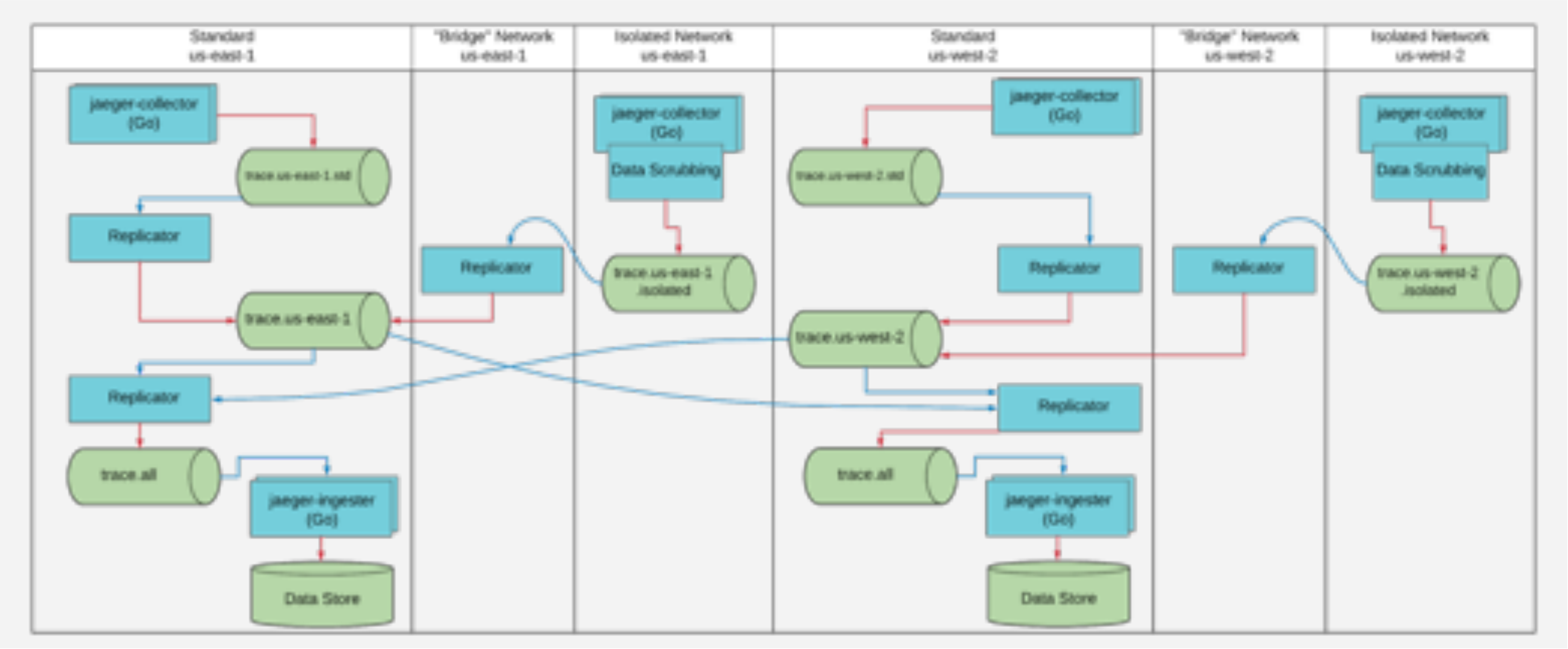


Write data to Kafka →

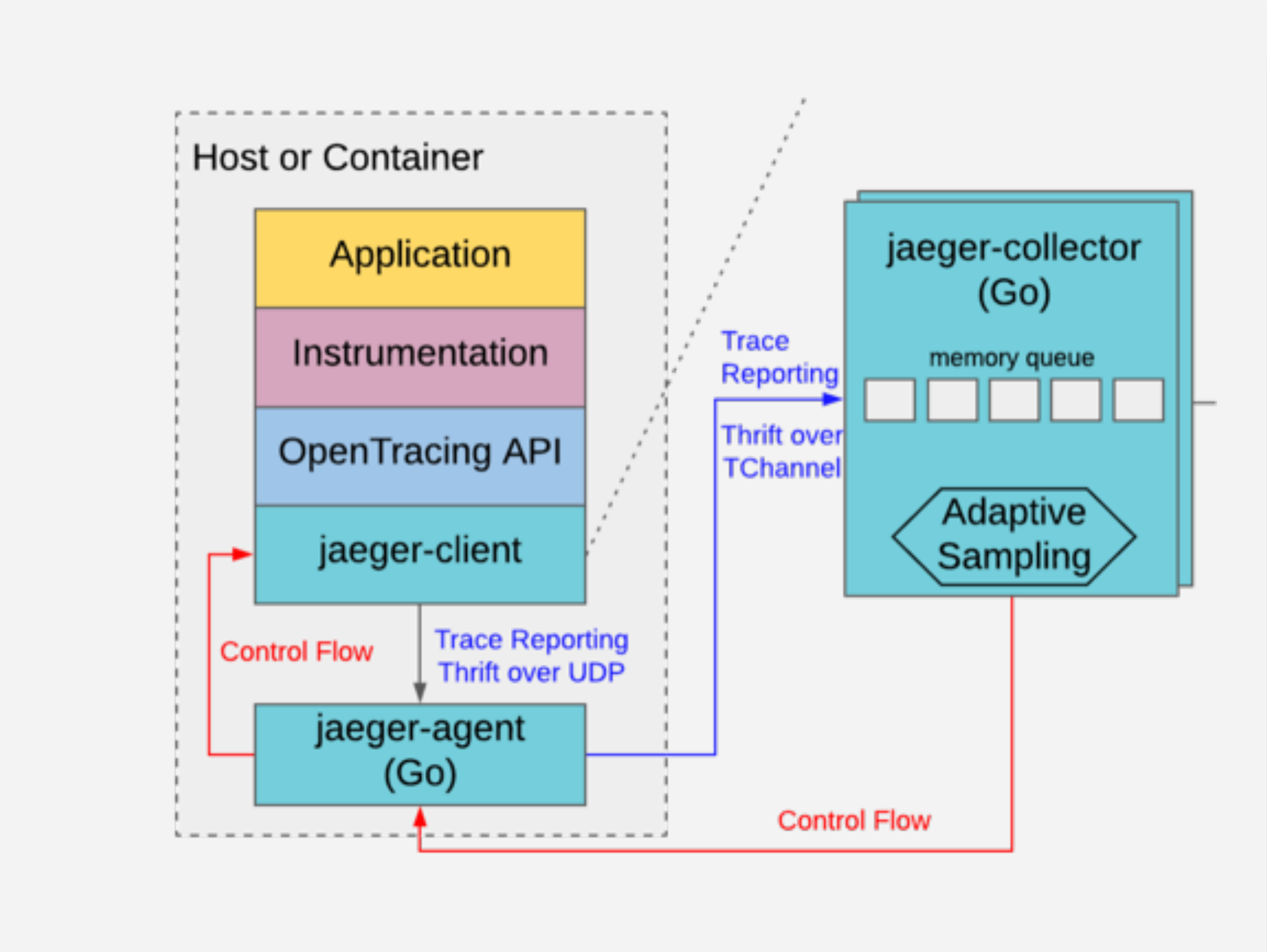
Read data from Kafka →

Deployment

Kafka



Scale Jaeger



Scaling

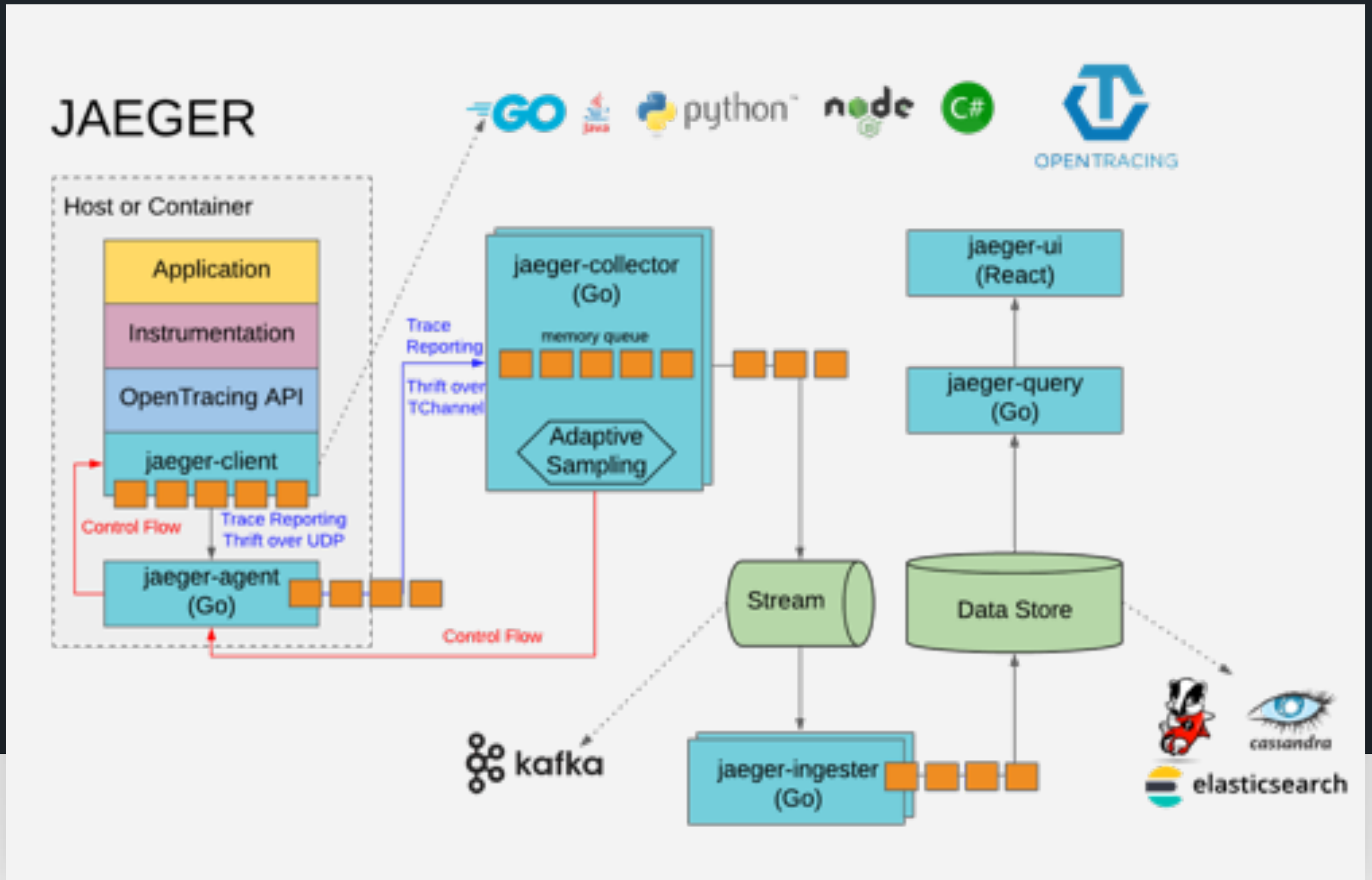
Start Small with Sampling

```
{
  "default_strategy": {
    "type": "ratelimiting",
    "param": 1
  }
}
```

Scale

Queues

https://medium.com/jaegertracing/tuning-jaegers-performance-7a60864cf3b1



Jaeger Agent

--processor.*.server-queue-size

Jaeger Collector

--collector.queue-size

--es.bulk.*

Kafka AsyncProducer
(based on max.message.bytes)

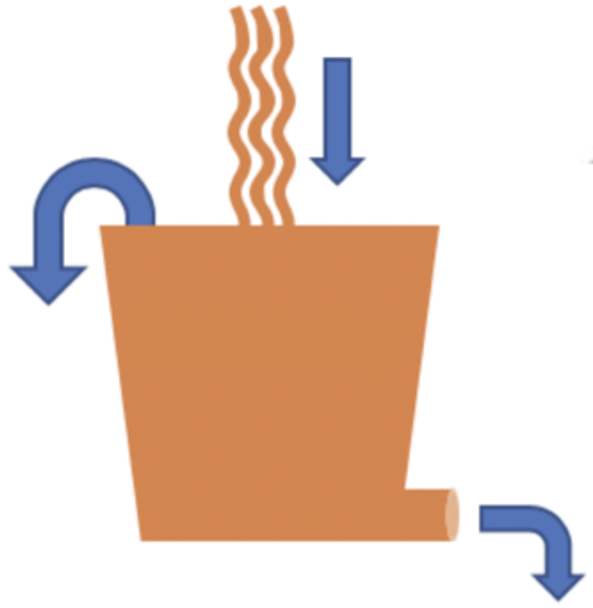
Jaeger Ingester

--collector.queue-size

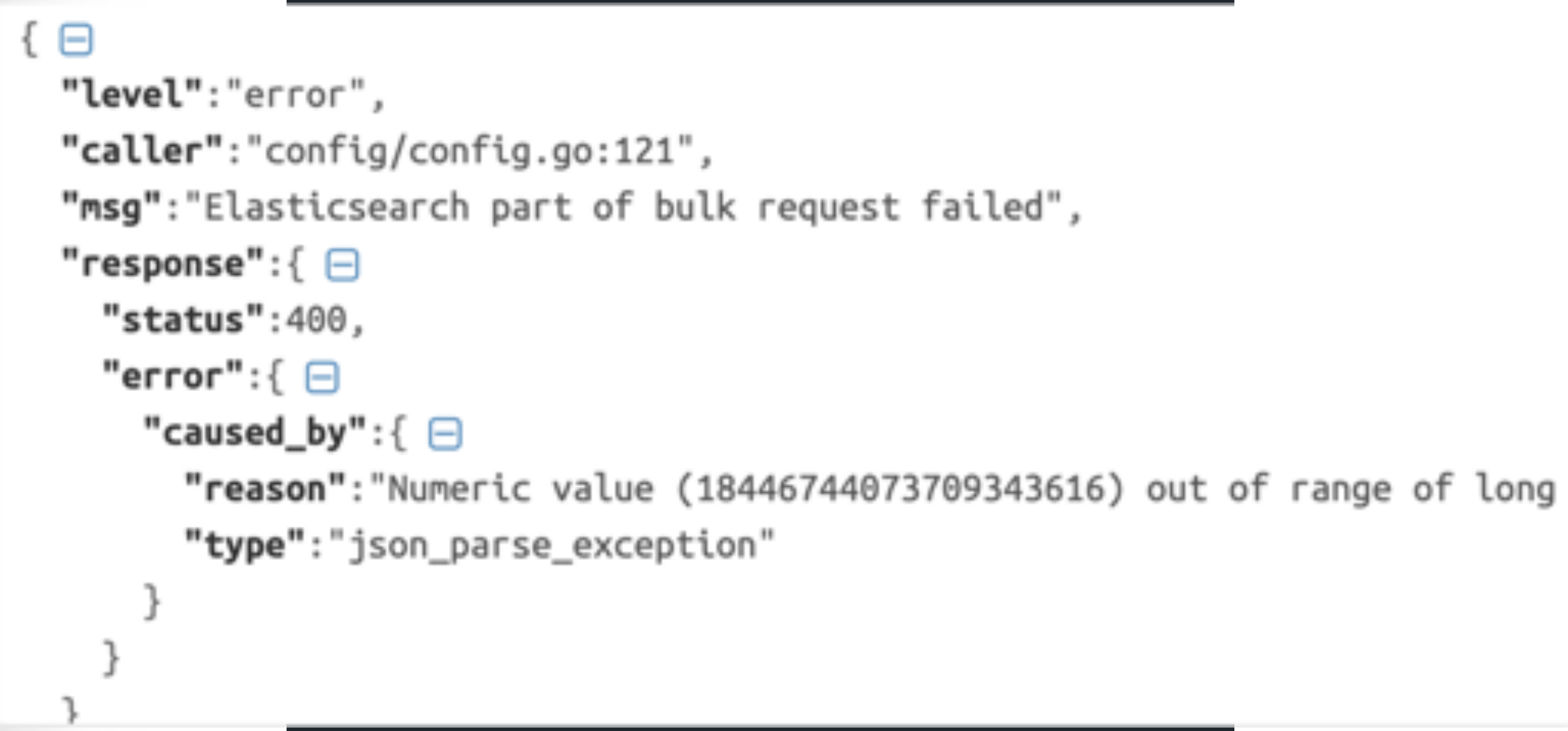
--es.bulk.*

Monitoring - Collector / Ingestor

jaeger_collector_spans_saved_by_svc_total{result="err”}



jaeger_collector_spans_dropped_total

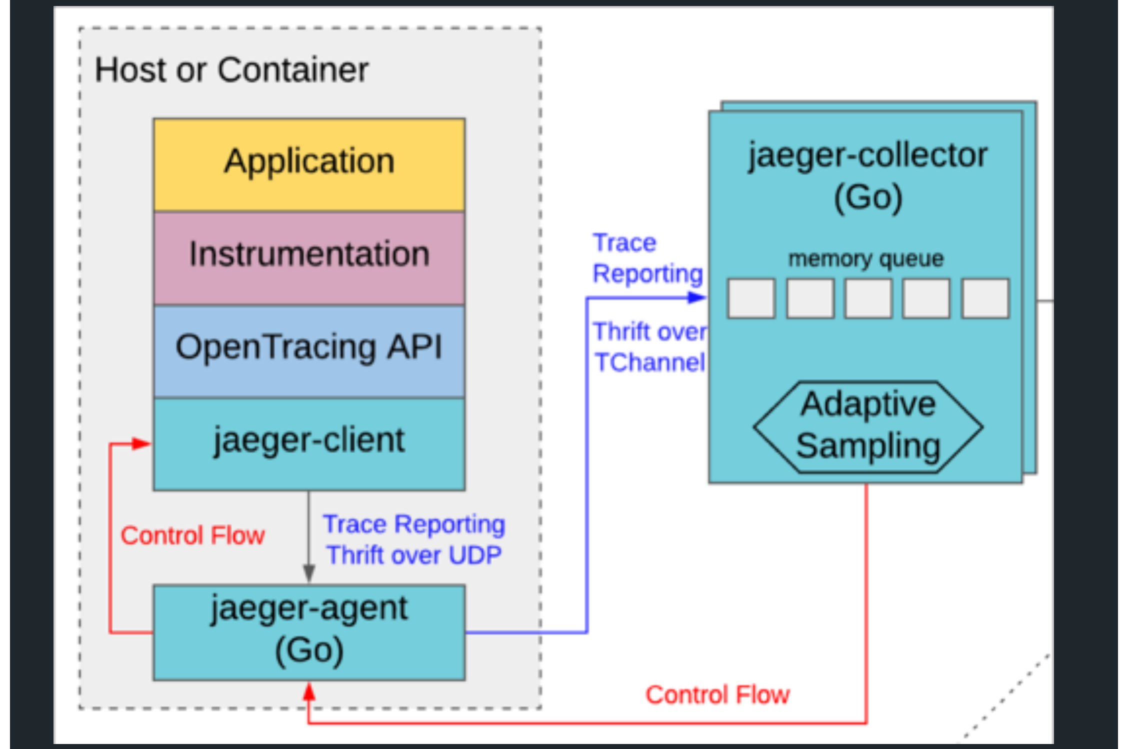


```
{ ⊟
  "level":"error",
  "caller":"config/config.go:121",
  "msg":"Elasticsearch part of bulk request failed",
  "response":{ ⊟
    "status":400,
    "error":{ ⊟
      "caused_by":{ ⊟
        "reason":"Numeric value (18446744073709343616) out of range of long
        "type":"json_parse_exception"
      }
    }
}
```

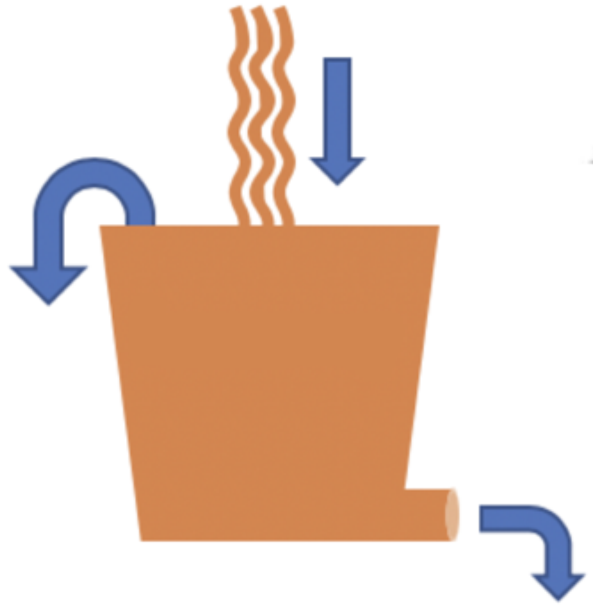
Logs!

Scaling

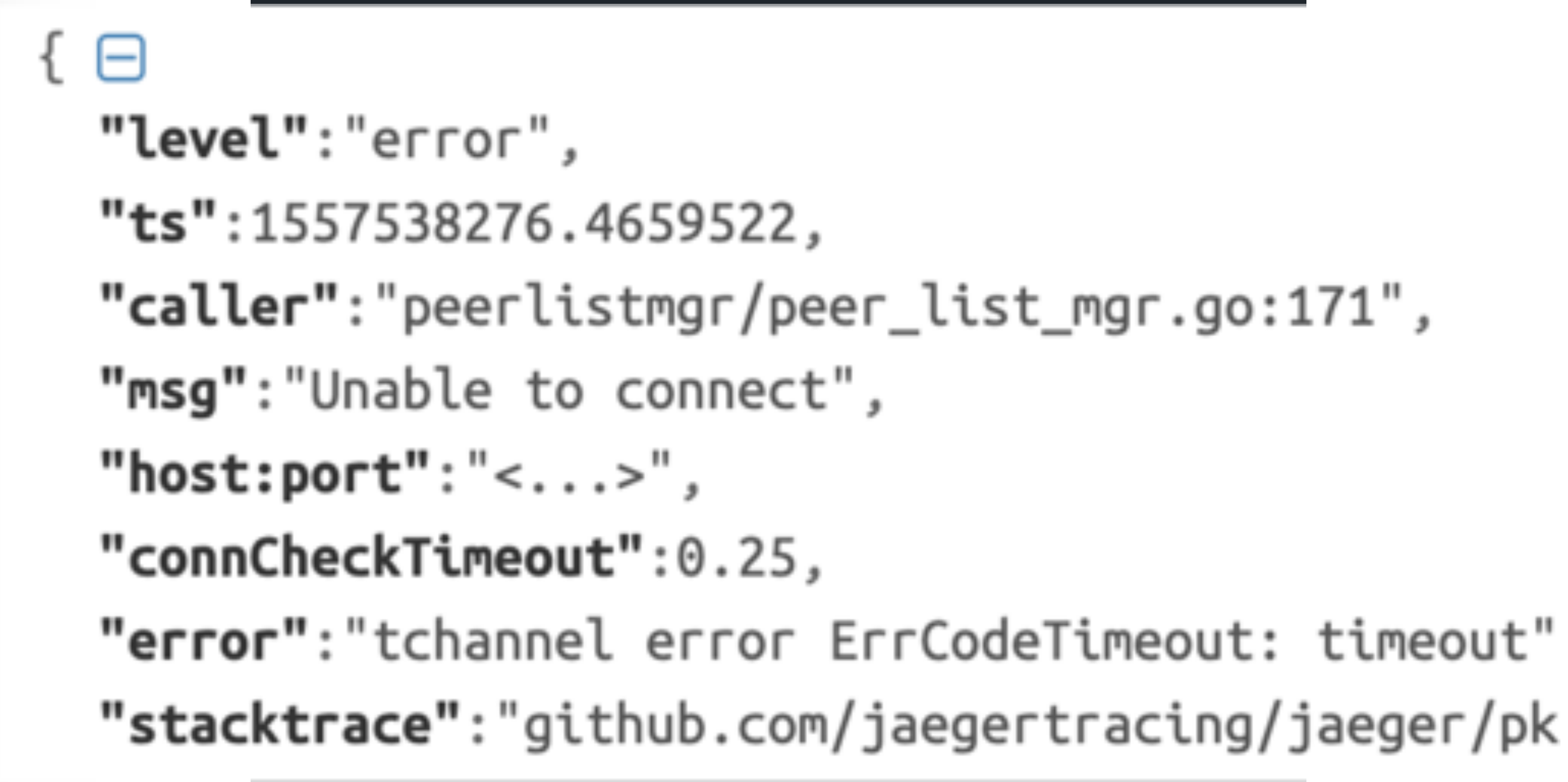
Monitoring - Agent



jaeger_agent_
collector_proxy_total
{result="err"}



jaeger_agent_*_
dropped_total



```
{ ⊟
  "level":"error",
  "ts":1557538276.4659522,
  "caller":"peerlistmgr/peer_list_mgr.go:171",
  "msg":"Unable to connect",
  "host:port":"<...>",
  "connCheckTimeout":0.25,
  "error":"tchannel error ErrCodeTimeout: timeout"
  "stacktrace":"github.com/jaegertracing/jaeger/pk
```

Logs!

Scaling

And...

elastic
cassandra
kafka

Scaling

Current infrastructure

Elasticsearch

3x i3.xlarge

➔4 CPUs

➔30.5 GiB

➔950 GB (NVMe SSD)

Jaeger Collector

4x Pods

➔2 CPUs

➔4 GiB

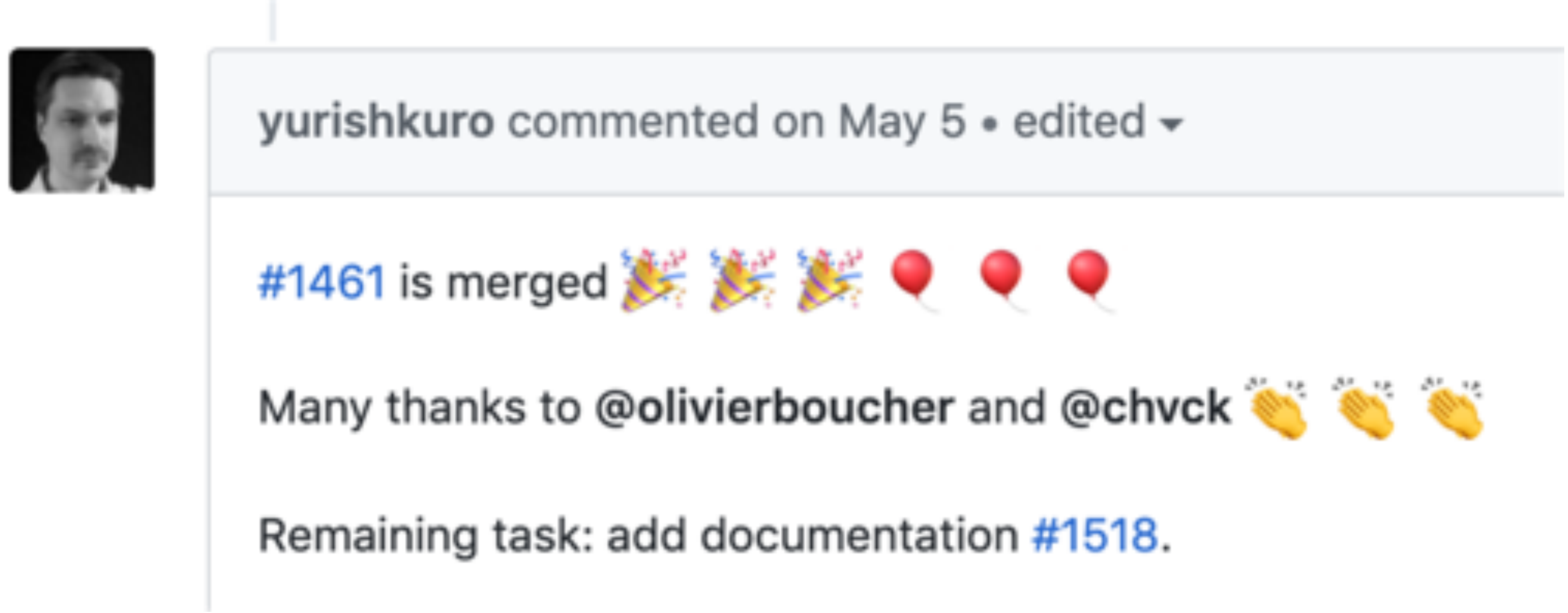
➔Queue Size: 300,000

➔Workers: 50 (default)

Extend Jaeger

Extend

Storage plugin



yurishkuro commented on May 5 • edited ▾

#1461 is merged 🎉 🎉 🎉 🎈 🎈 🎈

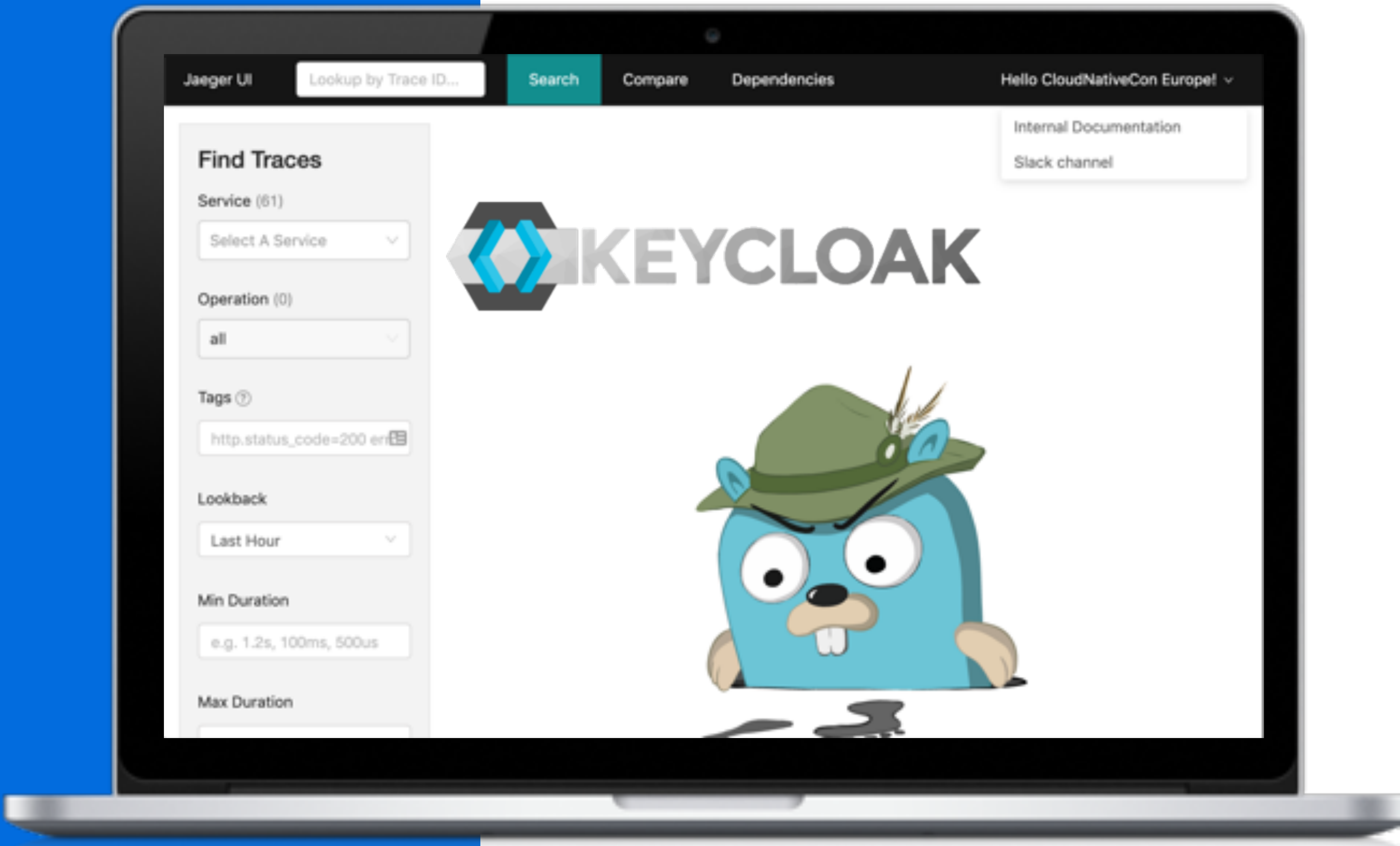
Many thanks to **@olivierboucher** and **@chvck** 👏 👏 👏

Remaining task: add documentation #1518.

Extend

Protect Jaeger UI

https://medium.com/jaegertracing/protecting-jaeger-ui-with-an-oauth-sidecar-proxy-34205cca4bb1



Find Traces

Service (61)
Select A Service

Operation (0)
all

Tags
http.status_code=200 er

Lookback
Last Hour

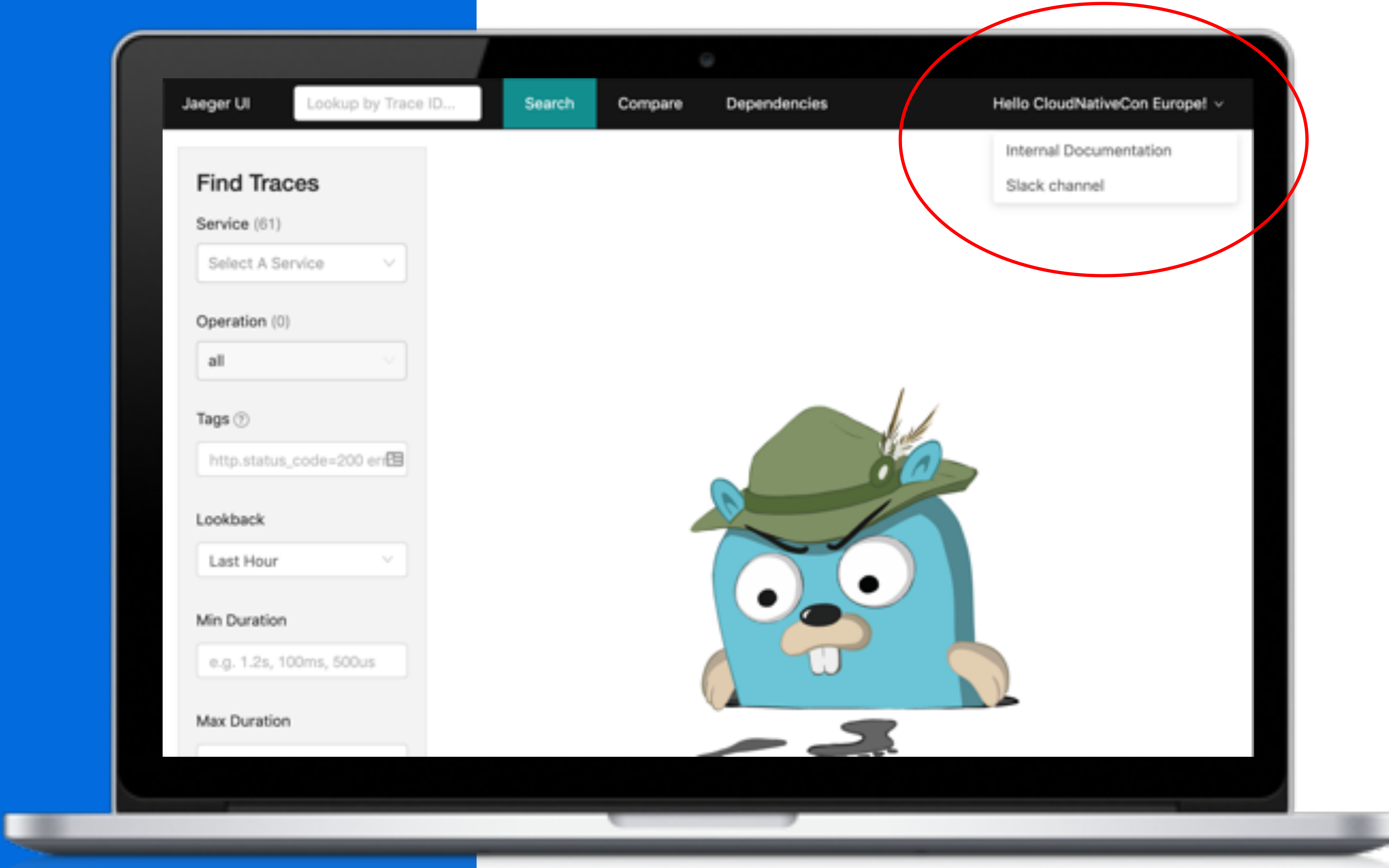
Min Duration
e.g. 1.2s, 100ms, 500us

Max Duration

Extend

Custom Menu

- → Internal documentation
- → Support Slack channel
- → Add a Logout link
  - ◆ jaegertracing/jaeger-ui/pull/223
- → Architecture diagram
- → PagerDuty information



Find Traces

Service (61)
Select A Service

Operation (0)
all

Tags
http.status_code=200 err

Lookback
Last Hour

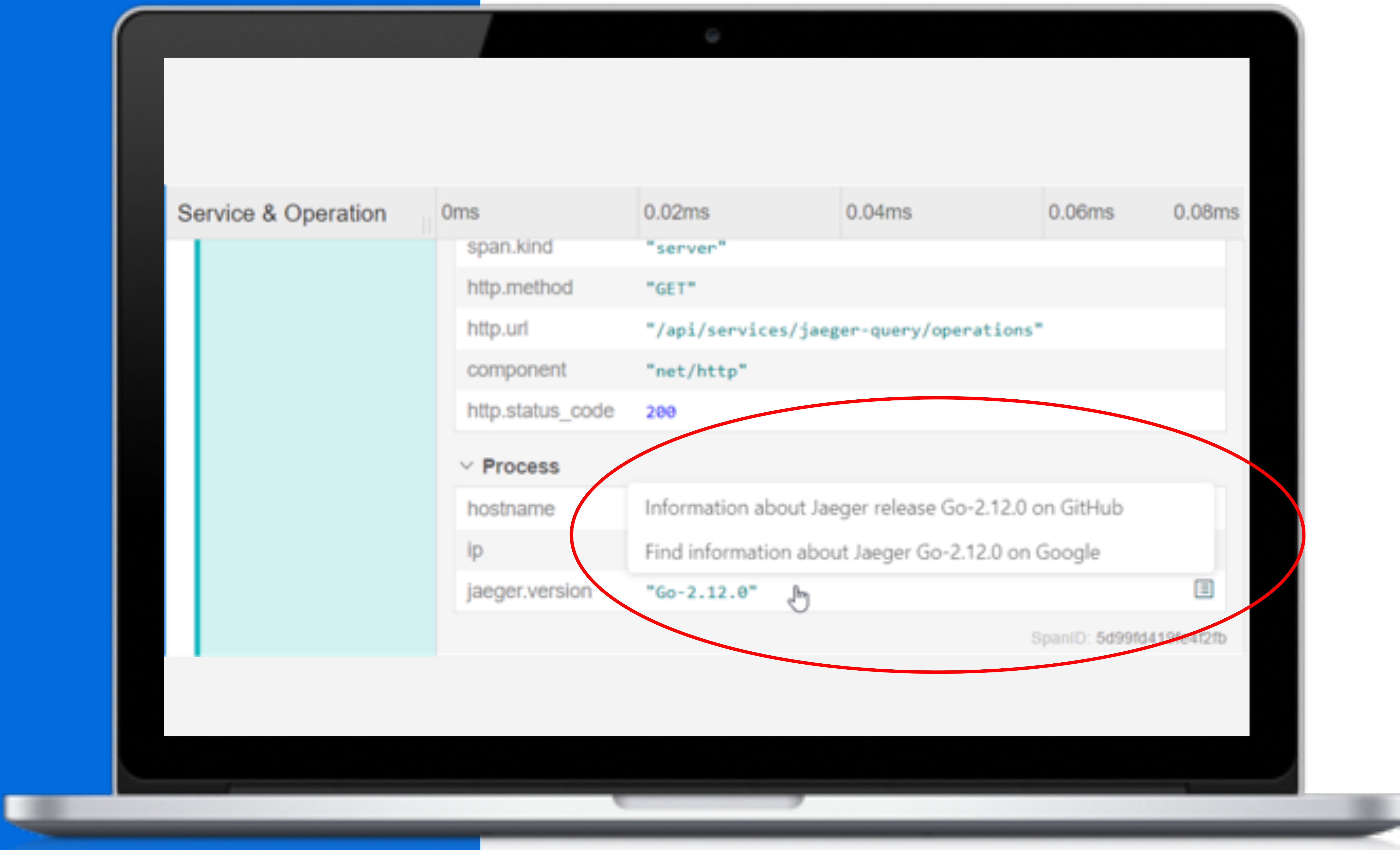
Min Duration
e.g. 1.2s, 100ms, 500us

Max Duration

Extend

Links on Tag

- ➔ Inventory Asset Management
  - ◆ OnCall information
  - ◆ Slack channel
  - ◆ Source code



span.kind	"server"
http.method	"GET"
http.url	"/api/services/jaeger-query/operations"
component	"net/http"
http.status_code	200

⌄ Process

hostname	Information about Jaeger release Go-2.12.0 on GitHub
ip	Find information about Jaeger Go-2.12.0 on Google
jaeger.version	"Go-2.12.0"

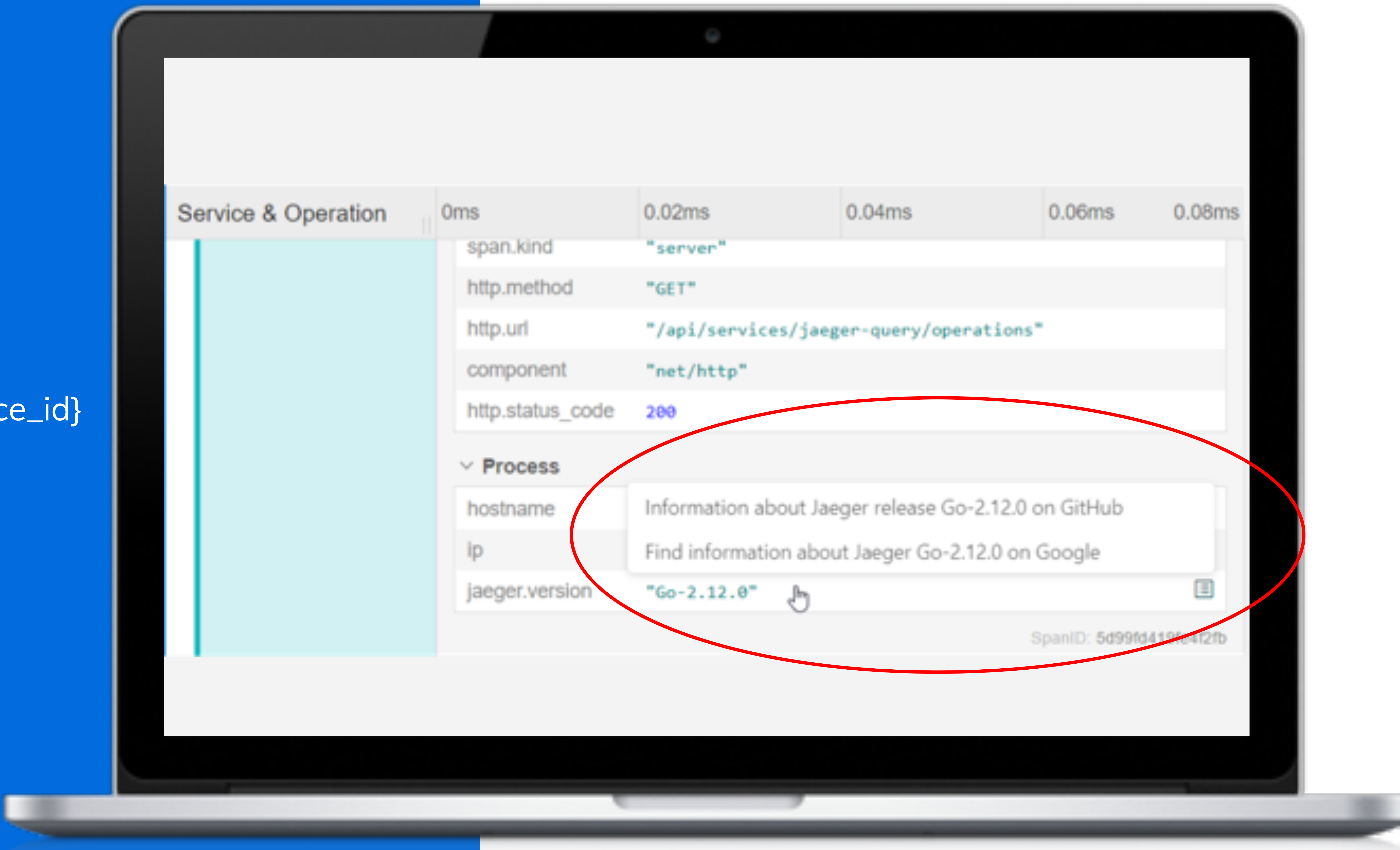
SpanID: 5d99fd419fe4f2fb

Extend

Links on Tag

➔ AWS Console

https://console.aws.amazon.com/ec2/v2
/home?#Instances:search=#{aws.instance_id}



Service & Operation | 0ms | 0.02ms | 0.04ms | 0.06ms | 0.08ms

span.kind	"server"
http.method	"GET"
http.url	"/api/services/jaeger-query/operations"
component	"net/http"
http.status_code	200

Process

hostname	
ip	
jaeger.version	"Go-2.12.0"

Information about Jaeger release Go-2.12.0 on GitHub

Find information about Jaeger Go-2.12.0 on Google

SpanID: 5d99fd419fe4f2fb

Extend

Links on Tag

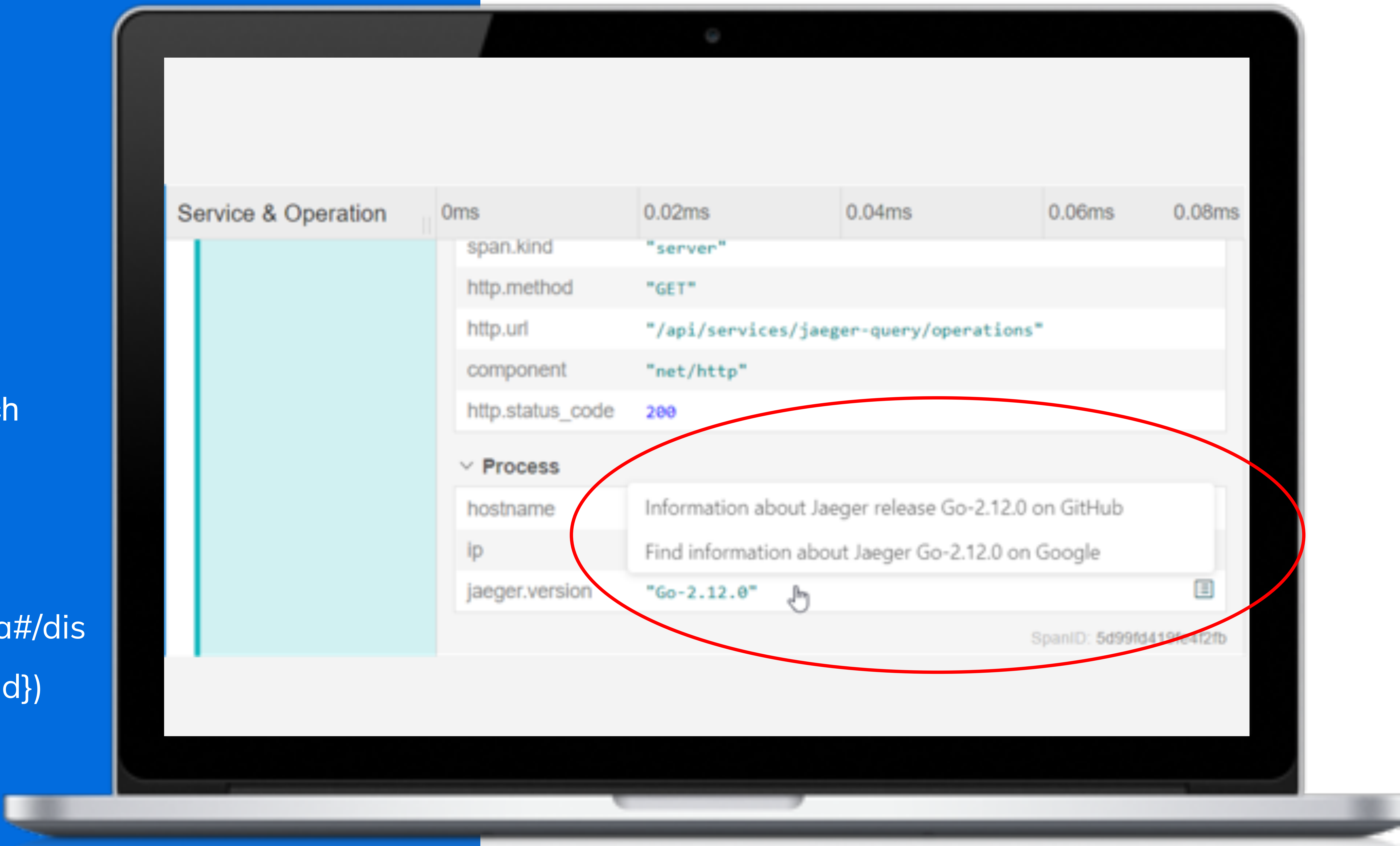
➔ Find the associated logs

- ◆ Splunk:

https://splunk/app/search/search?q=search index=* session_id=#{session_id}

- ◆ Kibana:

https://kibana/_plugin/kibana/app/kibana#/discover?_a=(query:'session_id=#{session_id})



Service & Operation | 0ms | 0.02ms | 0.04ms | 0.06ms | 0.08ms

span.kind	"server"
http.method	"GET"
http.url	"/api/services/jaeger-query/operations"
component	"net/http"
http.status_code	200

Process

hostname	
ip	
jaeger.version	"Go-2.12.0"

Information about Jaeger release Go-2.12.0 on GitHub

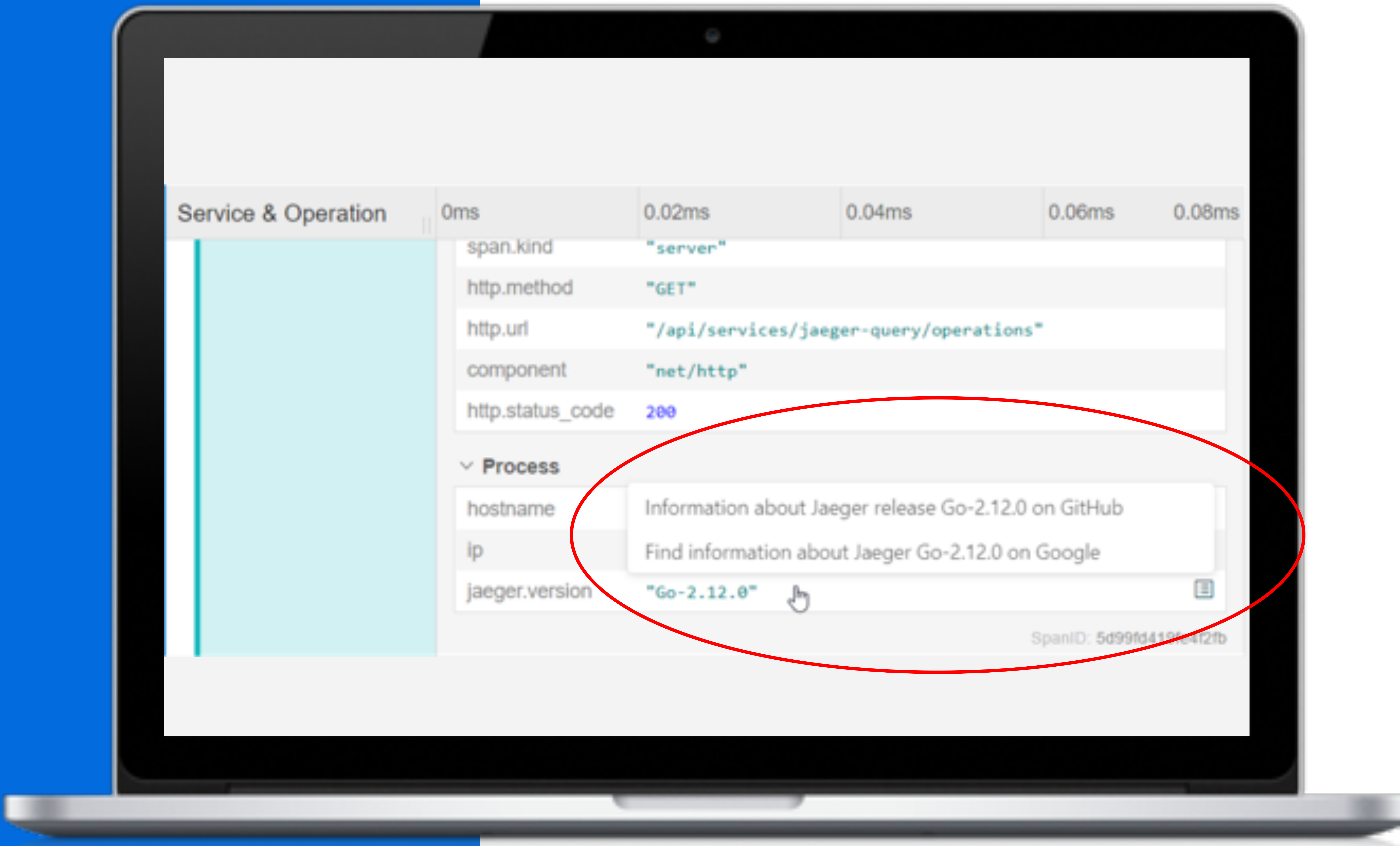
Find information about Jaeger Go-2.12.0 on Google

SpanID: 5d99fd419fe412fb

Extend

Links on Tag

- → Monitoring Dashboards
- → Git
- → Runbook
- → etc.



Service & Operation | 0ms | 0.02ms | 0.04ms | 0.06ms | 0.08ms

span.kind	"server"
http.method	"GET"
http.url	"/api/services/jaeger-query/operations"
component	"net/http"
http.status_code	200

Process

hostname	
ip	
jaeger.version	"Go-2.12.0"

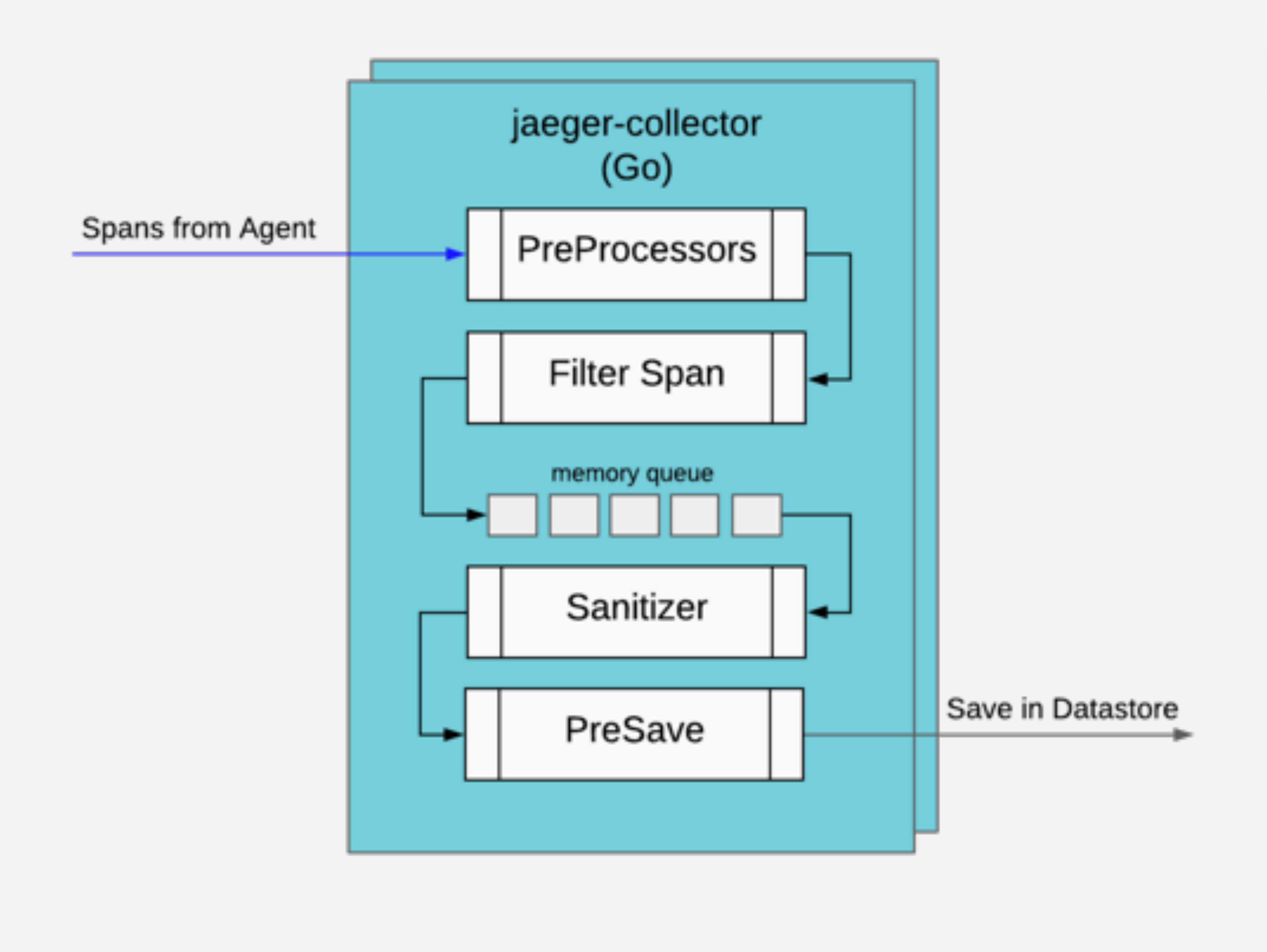
Information about Jaeger release Go-2.12.0 on GitHub

Find information about Jaeger Go-2.12.0 on Google

SpanID: 5d99fd419fe4f2fb

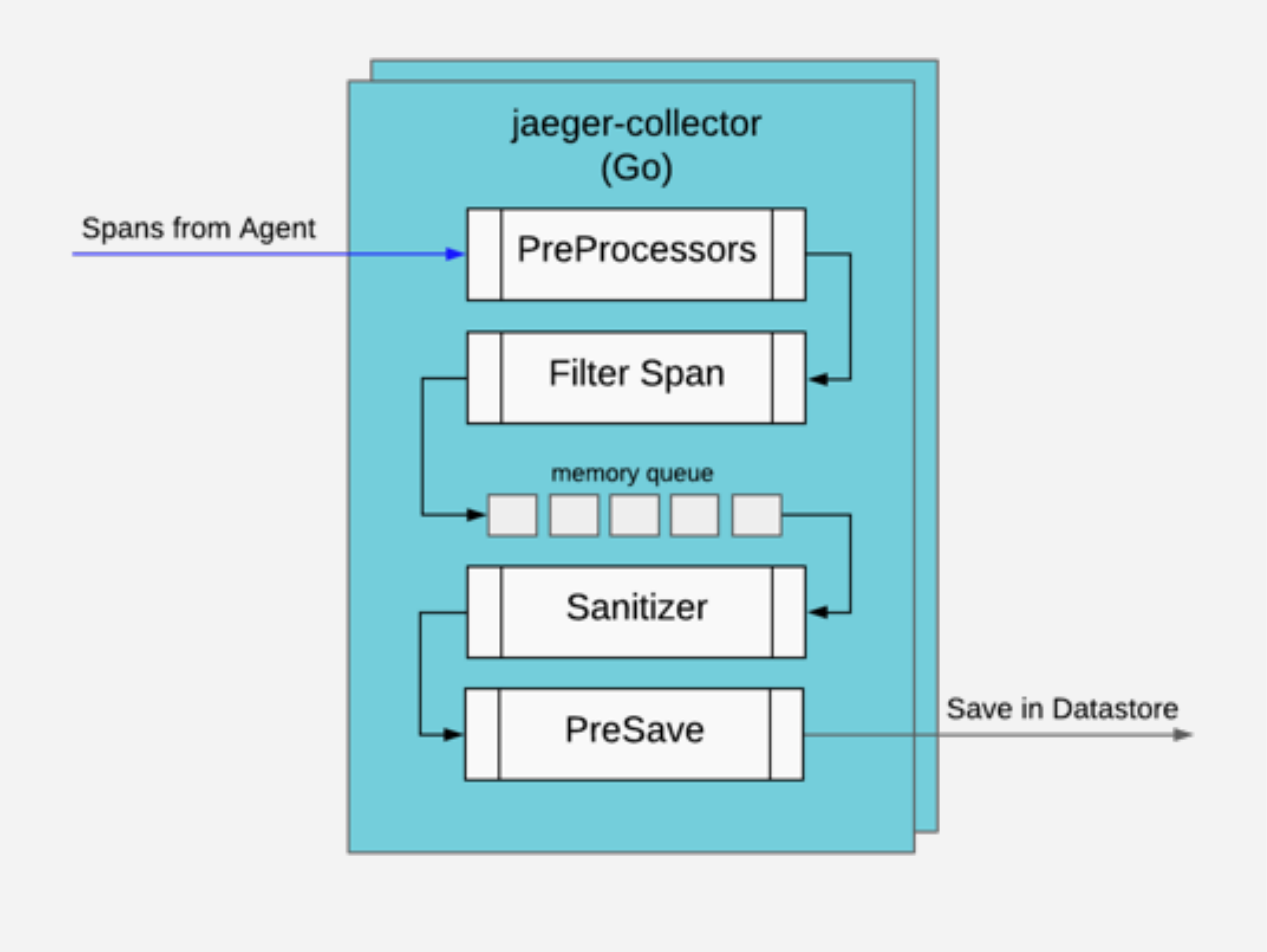
Extend

Processors



Currently not configurable

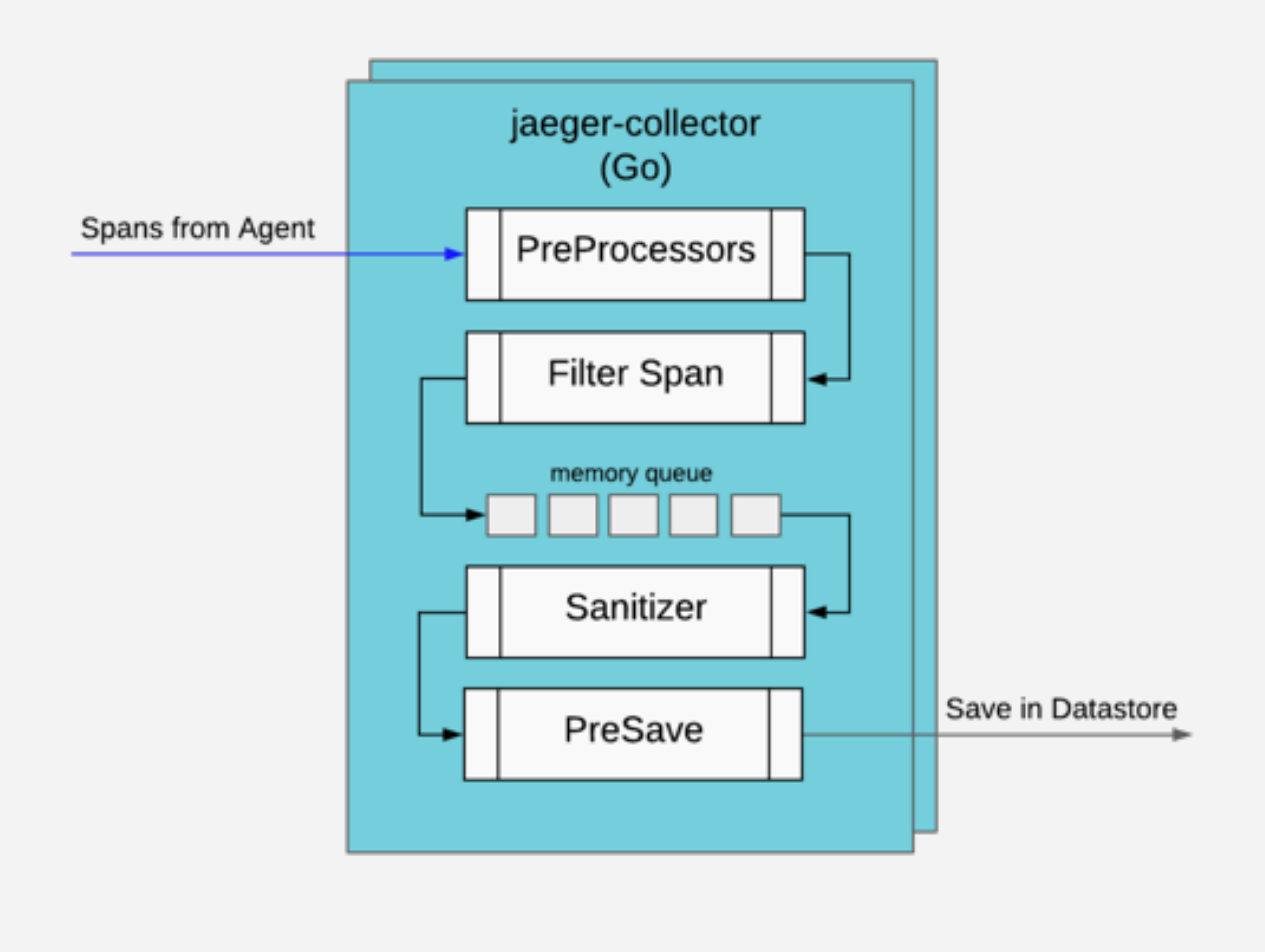
jaegertracing/jaeger/issues/1530



Extend

Spans PreProcessors

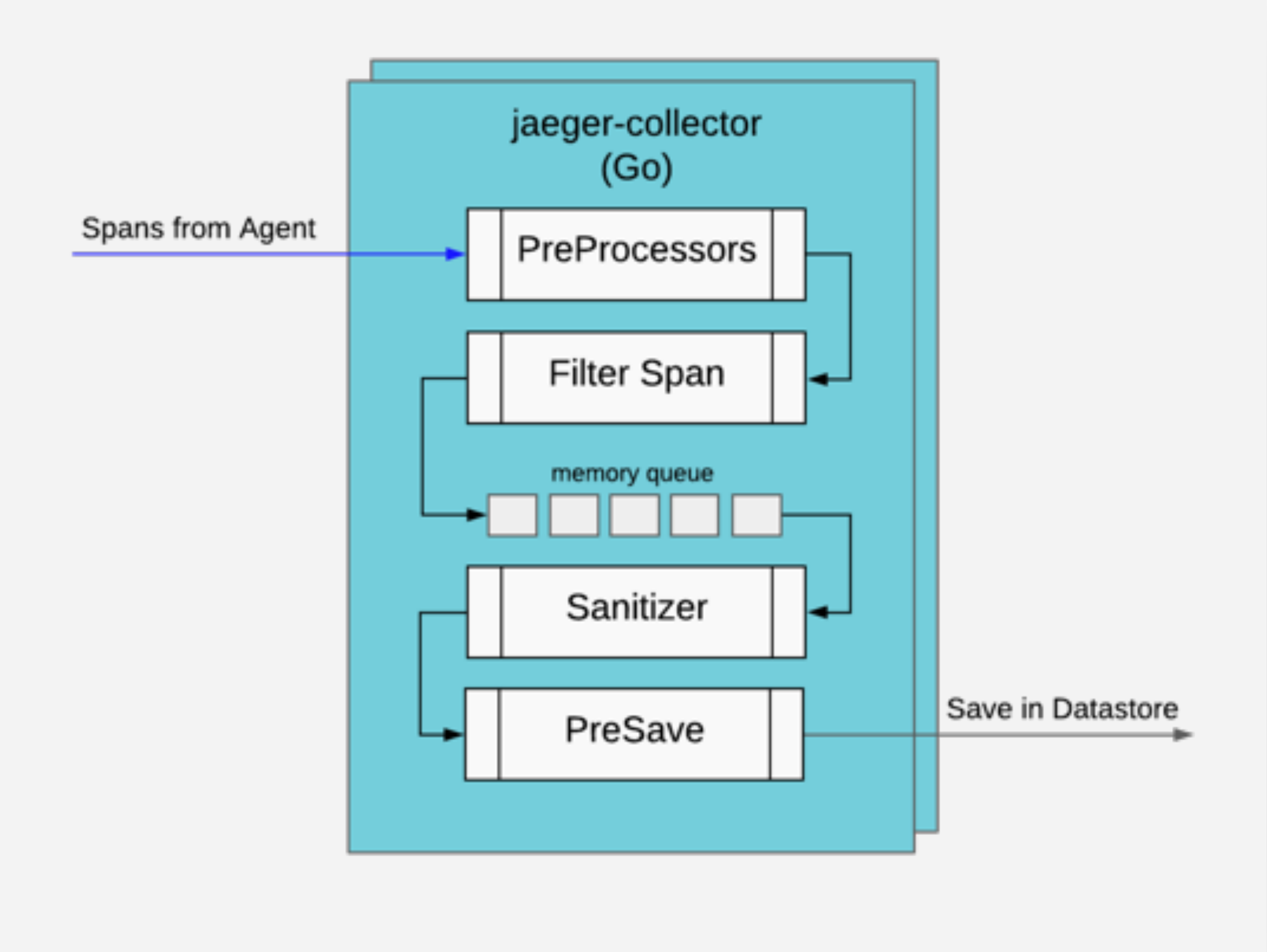
- ➔Extract Metrics per batch
- ➔Normalize Tags that are used in Filter Span
- ➔Best practice
  - ◆ No Tags
  - ◆ Minimum client version



Extend

Filter Span

- ➔Blacklist / whitelist service
- ➔Enforce mandatory tags
- ➔Enforce Remote Sampler
    - ◆ jaegertracing/jaeger/issues/1287



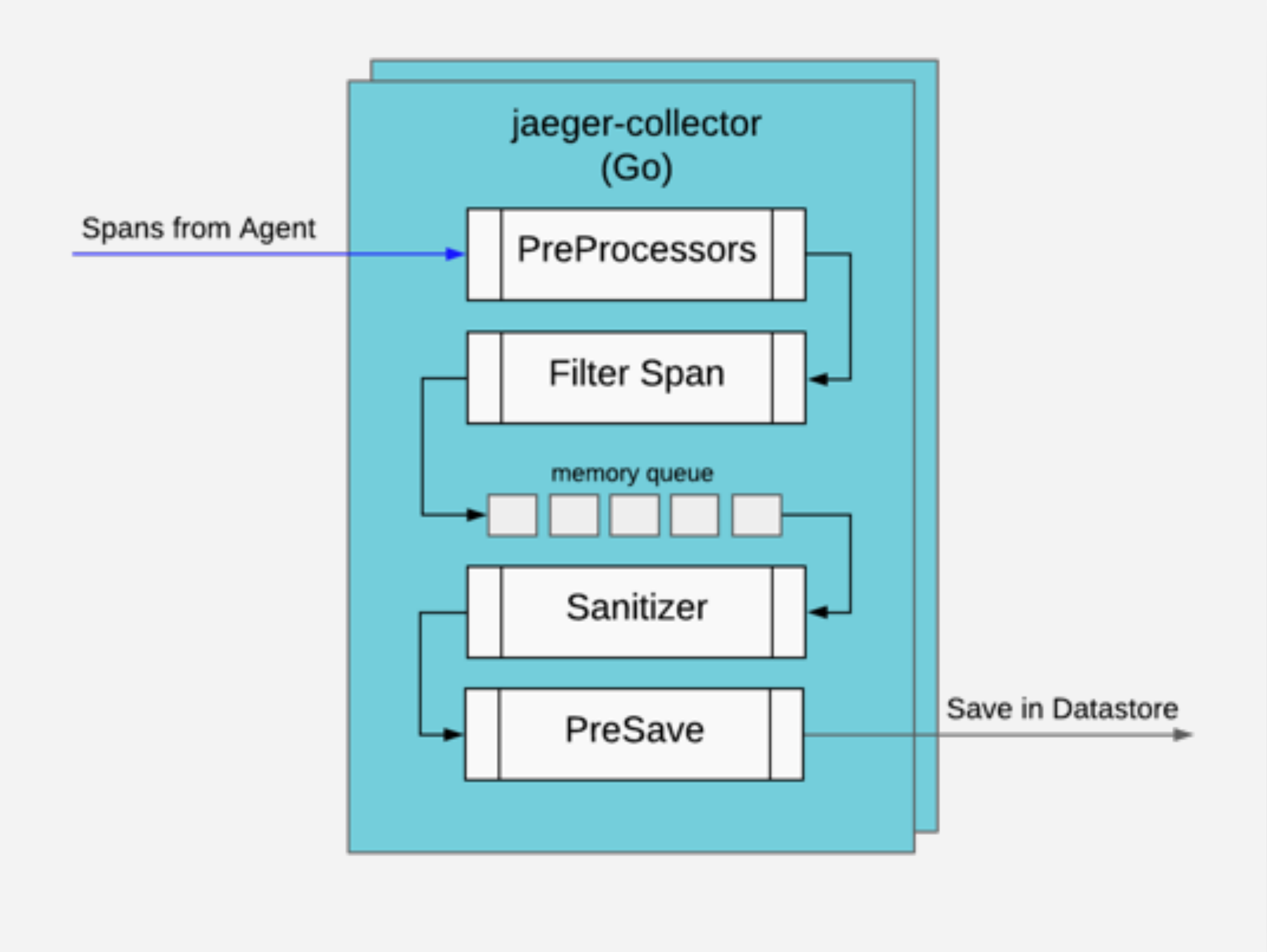
Extend

Span Sanitizer

- ➔Normalize Tags format
  - ◆ http_status, http.status, http.status_code
- ➔Data scrubbing
- ➔Add information from other system
  - ◆ Inventory Asset Management

Extend

Pre Save



?!

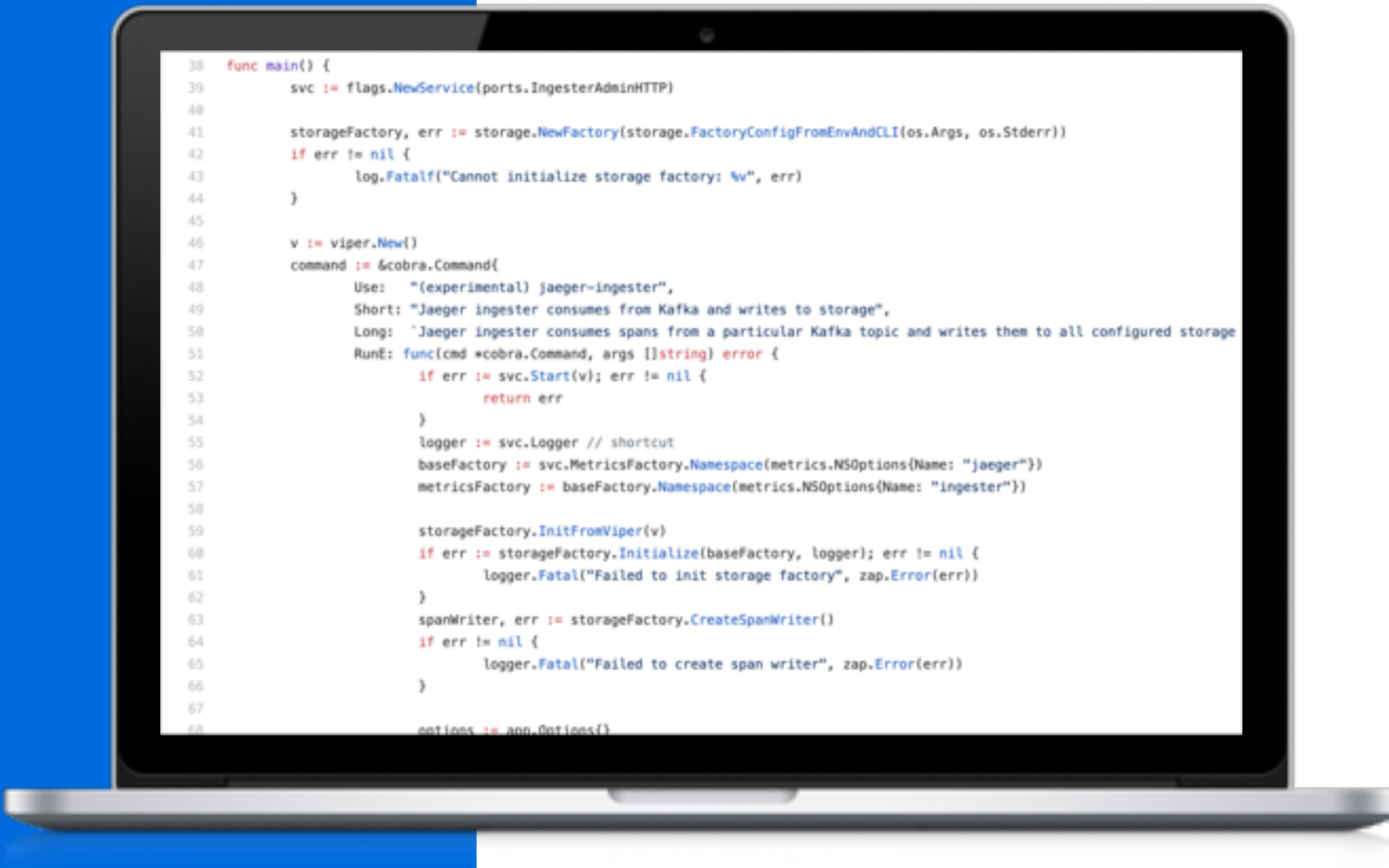
Any ideas? Comment here:

jaegertracing/jaeger/issues/1530

Extend

Don’t fork

Implement your own main()



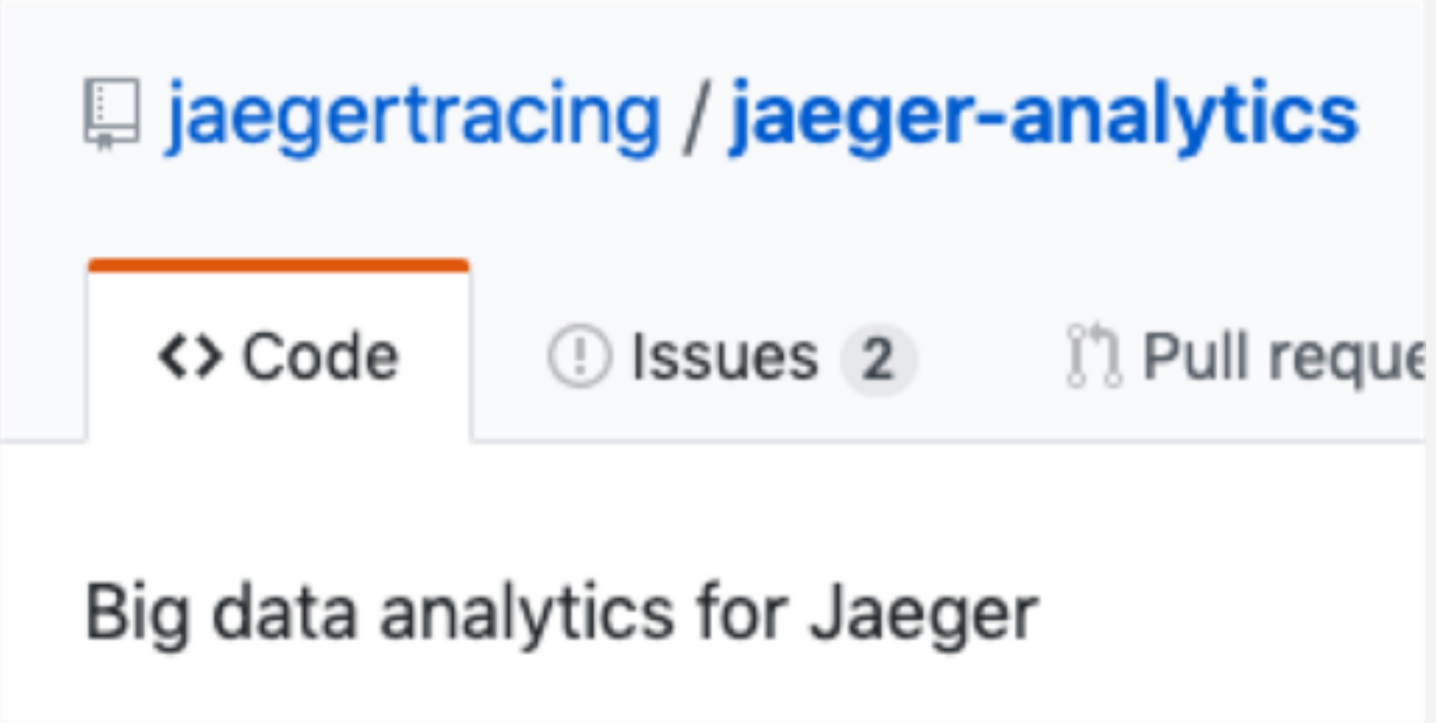
```go
func main() {
	svc := flags.NewService(ports.IngesterAdminHTTP)

	storageFactory, err := storage.NewFactory(storage.FactoryConfigFromEnvAndCLI(os.Args, os.Stderr))
	if err != nil {
		log.Fatalf("Cannot initialize storage factory: %v", err)
	}

	v := viper.New()
	command := &cobra.Command{
		Use:   "(experimental) jaeger-ingester",
		Short: "Jaeger ingester consumes from Kafka and writes to storage",
		Long:  `Jaeger ingester consumes spans from a particular Kafka topic and writes them to all configured storage
		RunE: func(cmd *cobra.Command, args []string) error {
			if err := svc.Start(v); err != nil {
				return err
			}
			logger := svc.Logger // shortcut
			baseFactory := svc.MetricsFactory.Namespace(metrics.NSOptions{Name: "jaeger"})
			metricsFactory := baseFactory.Namespace(metrics.NSOptions{Name: "ingester"})

			storageFactory.InitFromViper(v)
			if err := storageFactory.Initialize(baseFactory, logger); err != nil {
				logger.Fatal("Failed to init storage factory", zap.Error(err))
			}
			spanWriter, err := storageFactory.CreateSpanWriter()
			if err != nil {
				logger.Fatal("Failed to create span writer", zap.Error(err))
			}

			options := app.Options{}
```


jaegertracing / jaeger-analytics

Big data analytics for Jaeger

Extend

Analytics

- ➔Dependency Map
  - ◆ Show services
    - ● AWS API
    - ● Google Maps API
  - ◆ External component
    - ● Redis
    - ● SQL/NoSQL
- ➔Trace Quality
- ➔Latency Histogram

Agenda

1. ~~Introduction~~
2. ~~Adoption~~
3. ~~Deploy~~
4. ~~Scale~~
5. ~~Extend~~

Thank you!

Louis-Etienne Dorval

github.com/ledor473

twitter.com/ledor473

References

- https://medium.com/jaegertracing/deployment-strategies-for-the-jaeger-agent-1d6f91796d09
- https://medium.com/jaegertracing/tuning-jaegers-performance-7a60864cf3b1
- https://medium.com/jaegertracing/running-jaeger-agent-on-bare-metal-d1fc47d31fab
- https://www.jaegertracing.io/docs/latest/architecture/
- https://github.com/jaegertracing/jaeger
- https://github.com/jaegertracing/jaeger-analytics