CLOUD NATIVE
COMPUTING FOUNDATION

Jaeger
Project Session

Pavol Loffay (Red Hat), Yuri Shkuro (Uber)

CloudNativeCon NA, Austin, Dec-8-2017

Agenda

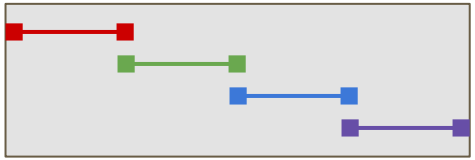
- Introduction to tracing
- Demo
- Zipkin drop-in replacement
- Istio Jaeger demo
- Roadmap
  - Path based dependency diagrams
  - Adaptive sampling
- Discussion

Distributed Tracing

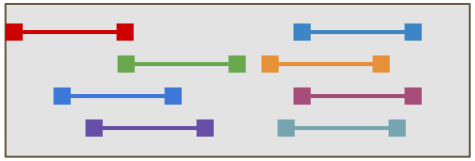
Concepts and terminology

Transaction Monitoring for Microservices

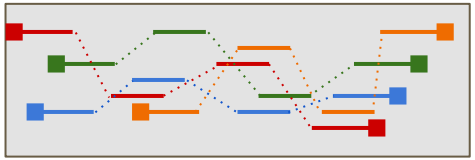
"The Simple [Inefficient] Thing"



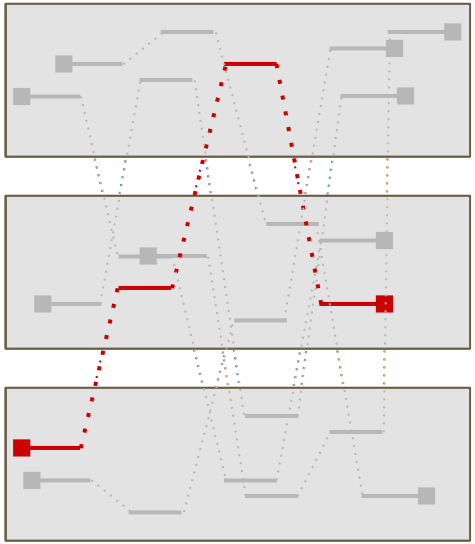
Basic Concurrency



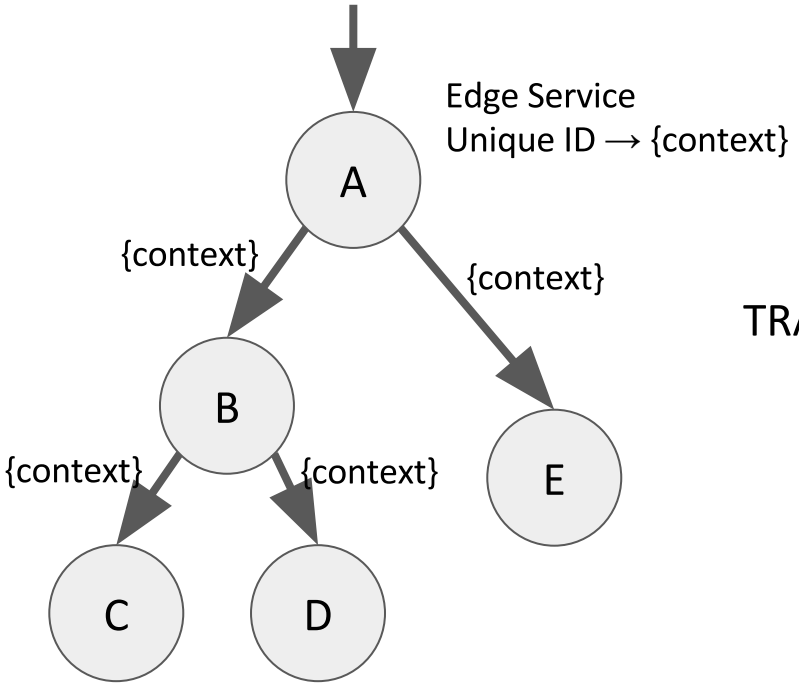
Async Concurrency

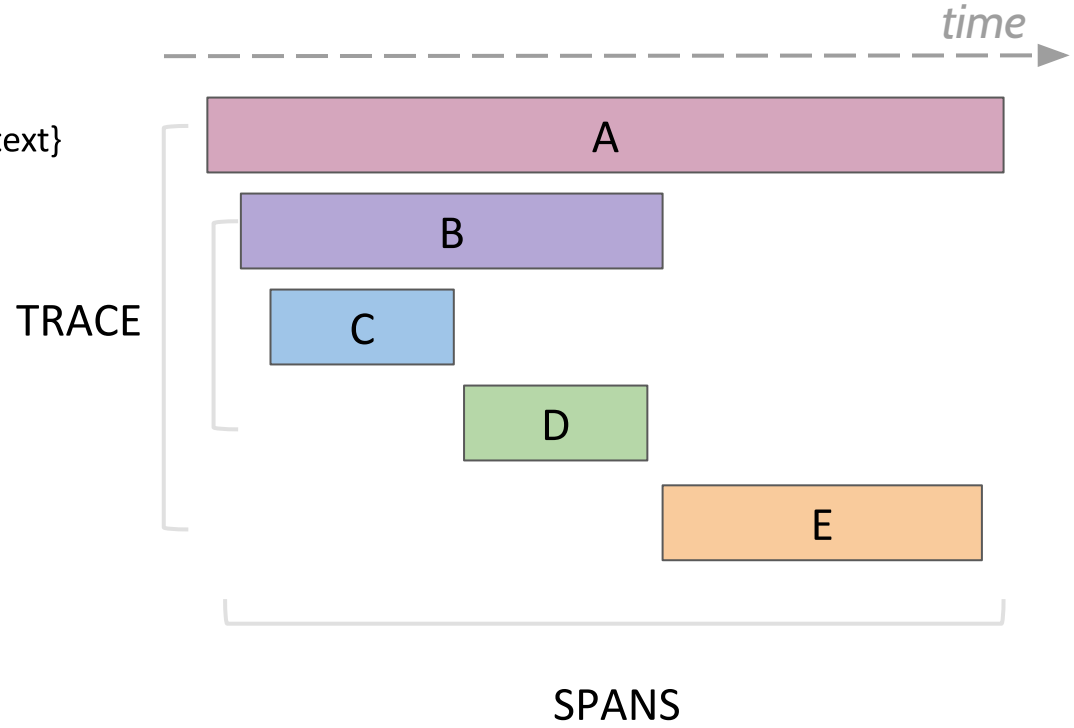


Distributed Concurrency



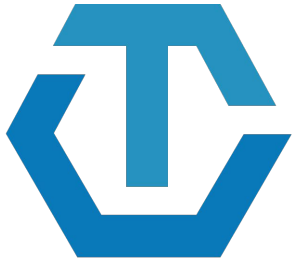
Context Propagation & Distributed Tracing





OpenTracing

- Instrumentation API
  - Context propagation
  - Distributed tracing
  - Contextualized logging
  - Contextualized metrics
- Vendor neutral
- Cross language
- CNCF member project



OPENTRACING

http://opentracing.io

CLOUD NATIVE
COMPUTING FOUNDATION

Let's look at some traces

demo time: http://bit.do/jaeger-hotrod

Jaeger, a Distributed Tracing System

https://jaegertracing.io

Jaeger - /ˈyāgər/, *noun*: hunter

- Inspired by Google's Dapper and OpenZipkin
- Started at Uber in August 2015
- Open sourced in April 2017
- Official CNCF project, Sep 2017
- Built-in OpenTracing support
- https://jaegertracing.io

UBER

Technology Stack

- Go backend
- Pluggable storage
  - Cassandra, Elasticsearch, memory, ...
- React/Javascript frontend
- OpenTracing Instrumentation libraries



Community

- 10 full time engineers at Uber and Red Hat
- 30+ contributors on GitHub
- Already used by many organizations
  - including Symantec, Red Hat, Base CRM, Uber, Massachusetts Open Cloud, Nets, FarmersEdge, GrafanaLabs, Northwestern Mutual, Zenly

UBER Service Dependency Graph

Jaeger 1.0

http://bit.do/jaeger-1-0

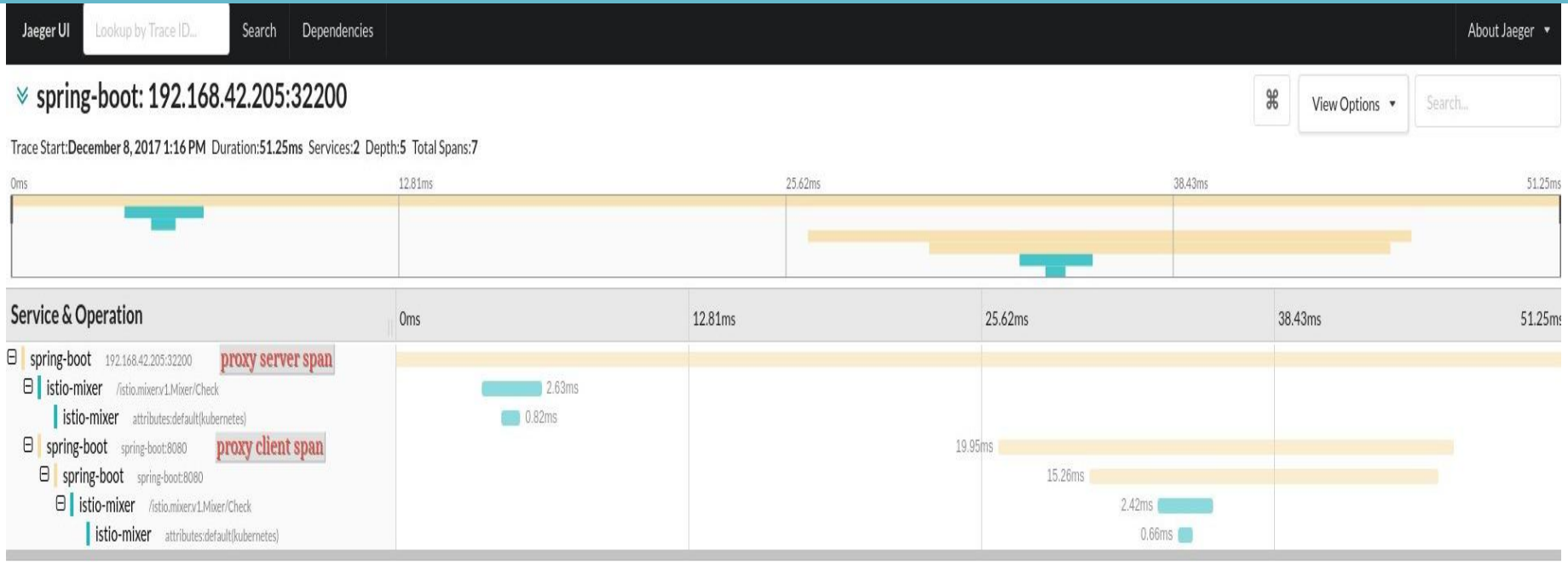
Announcing Release 1.0

- UI performance and usability improvements to view large traces
- Storage backends: Cassandra and ElasticSearch
- Spark job for building service dependencies diagram
- Client libraries: Go, Java, Python, Node.js, C++ (new)
- Integration with other CNCF projects
  - Templates for deploying Jaeger on Kubernetes
  - All Jaeger components expose Prometheus metrics by default
  - Integration with Envoy/Istio
- Drop-in replacement for Zipkin backend

Jaeger in Istio demo

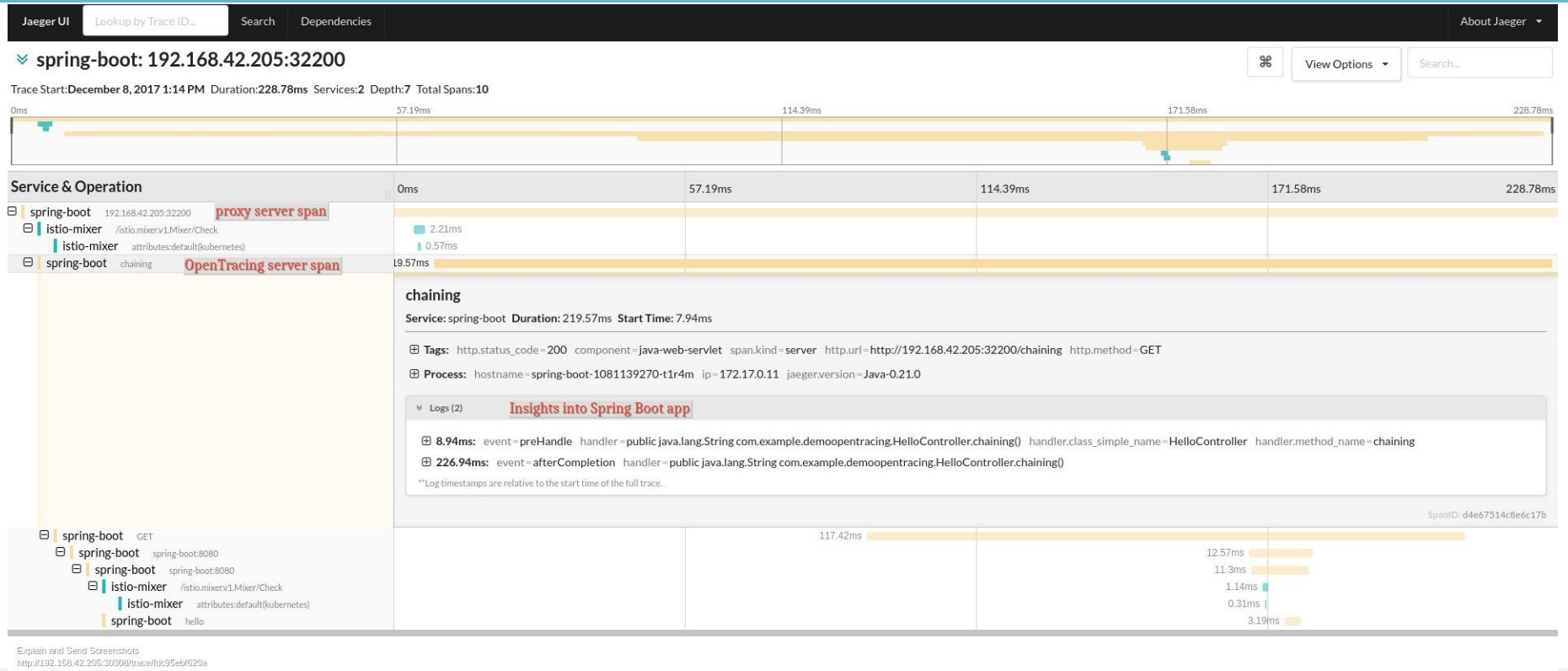
OpenTracing with Istio

Tracing via Istio Only



Explain and Send Screenshots
http://192.168.42.205:30308/trace/f9a4f887b348

Tracing With Istio And OpenTracing



Roadmap

http://bit.do/jaeger-roadmap

Adaptive Sampling

- APIs have endpoints with different QPS
- Service owners do not know the full impact of sampling probability

Adaptive Sampling is per service + endpoint, decided by Jaeger backend based on traffic

Data Pipeline

- Based on Kafka and Apache Flink
- Support aggregations and data mining
- Examples:
  - Pairwise dependencies diagram
  - Trace quality metrics by service
  - Path-based dependencies diagram
  - Latency histograms

Tracing Quality Metrics by Service

Tracing Score for my-service-x

Completeness: **0.63** out of 1.0

Quality: **1.00** out of 1.0

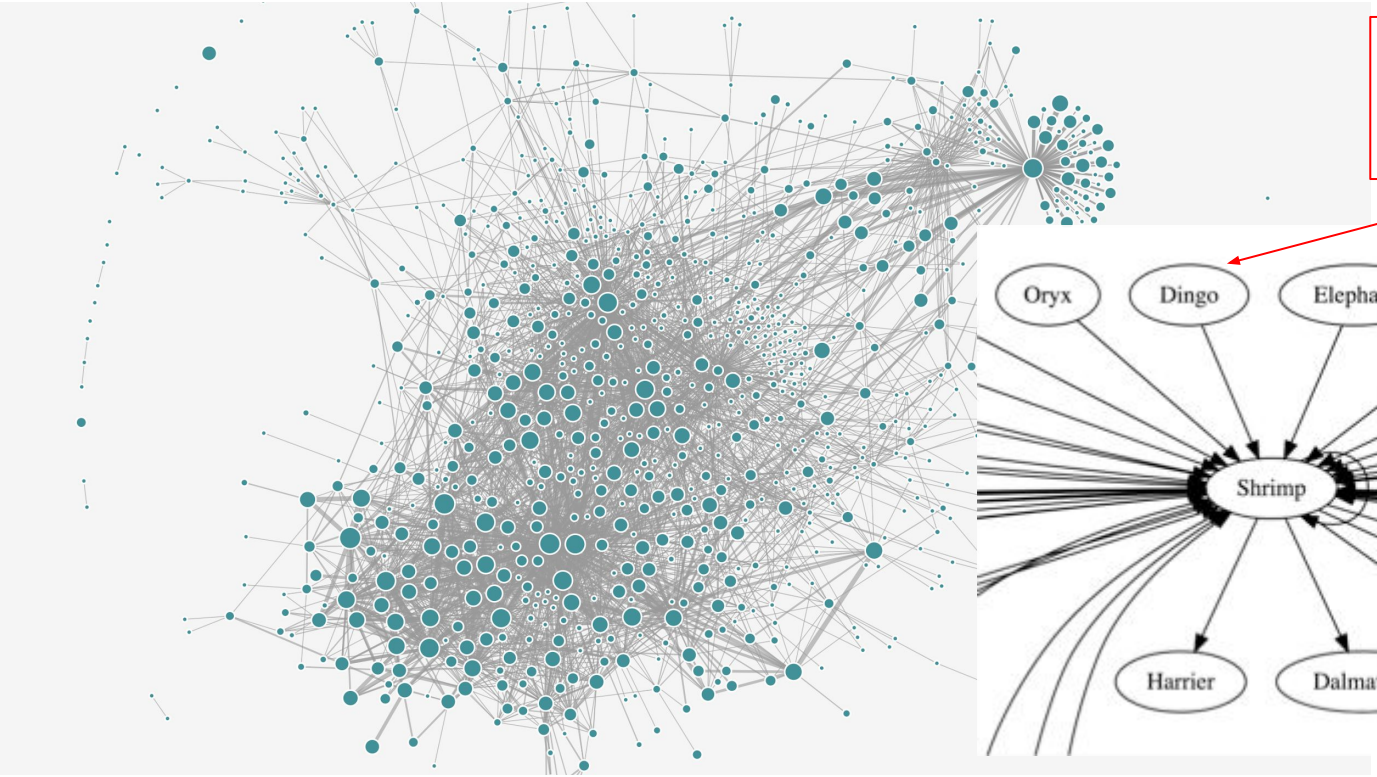
How do I improve my score?

Tracing Quality Metrics for my-service-x

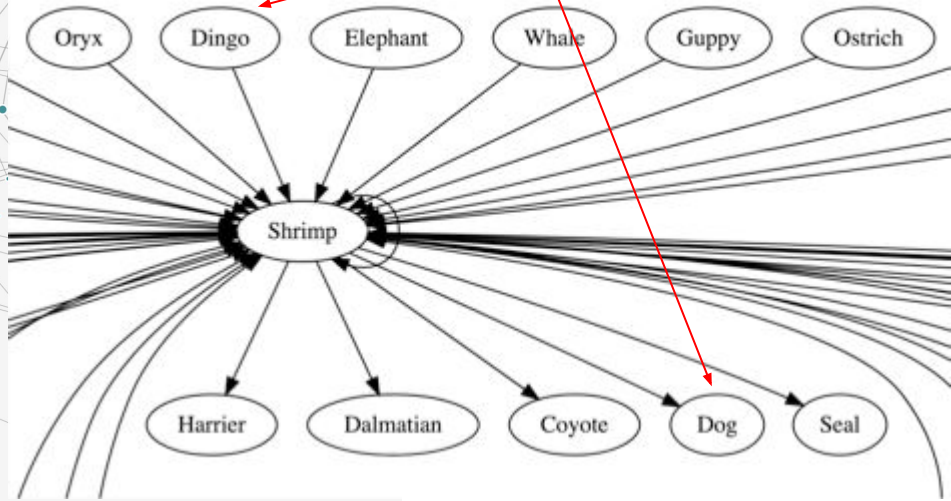
Click on pass or fail numbers to see example traces that exhibit that behavior

Metric	Type	Pass %	Num Passes	Num Failures	Last Failure	Description
HasClientAddress	Other	0	0	962939	0 hours ago	The server span emitted by this server had a good Client Address annotation saying where the request was coming from
HasClientSpans	Completeness	100	4521692	0		The service emitted a client-side span with good client annotations
HasSamplerType	Other	0	0	109862	0 hours ago	This service initiated the trace and emitted the `sampler.type` annotation
HasServerAddress	Other	100	4521692	0		The client span emitted by this server had a good Server Address annotation saying where the request was going
HasServerSpans	Completeness	89	962939	109862	0 hours ago	The service emitted a servier-side span with good server annotations
MeaningfulEndpointName	Other	100	5484631	0		The name of the endpoint being called had a meaningful name, e.g. not GET or POST
MinimumClientVersionCheck	Completeness	0	0	1072801	0 hours ago	This service emittted a span that has an acceptable client version. Minimum versions: go 2.6.0, node 3.0.0, python 3.3.1, java 0.17.0
ParentSpanExists	Other	93	2341174	172936	0 hours ago	The service (the Parent) that called this service emitted a span
TracedRootService	Other	100	109862	0		This service was the root service (i.e. it initiated the trace) and it correctly emitted a span
UniqueClientSpanID	Quality	100	4521692	0		Multiple client spans in this trace share the same span ID
UniqueServerSpanID	Quality	100	962939	0		Multiple server spans in this trace share the same span ID

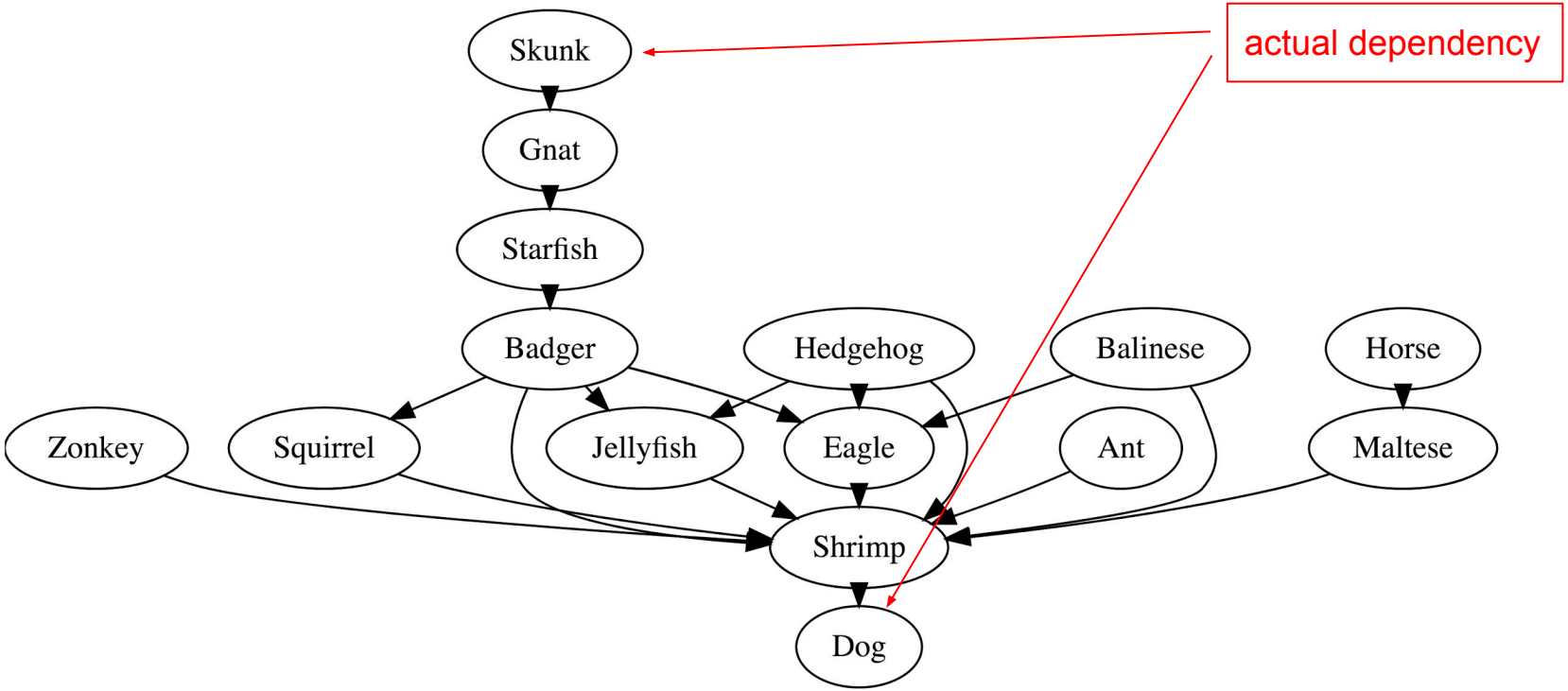
Pairwise Service Dependency Diagram



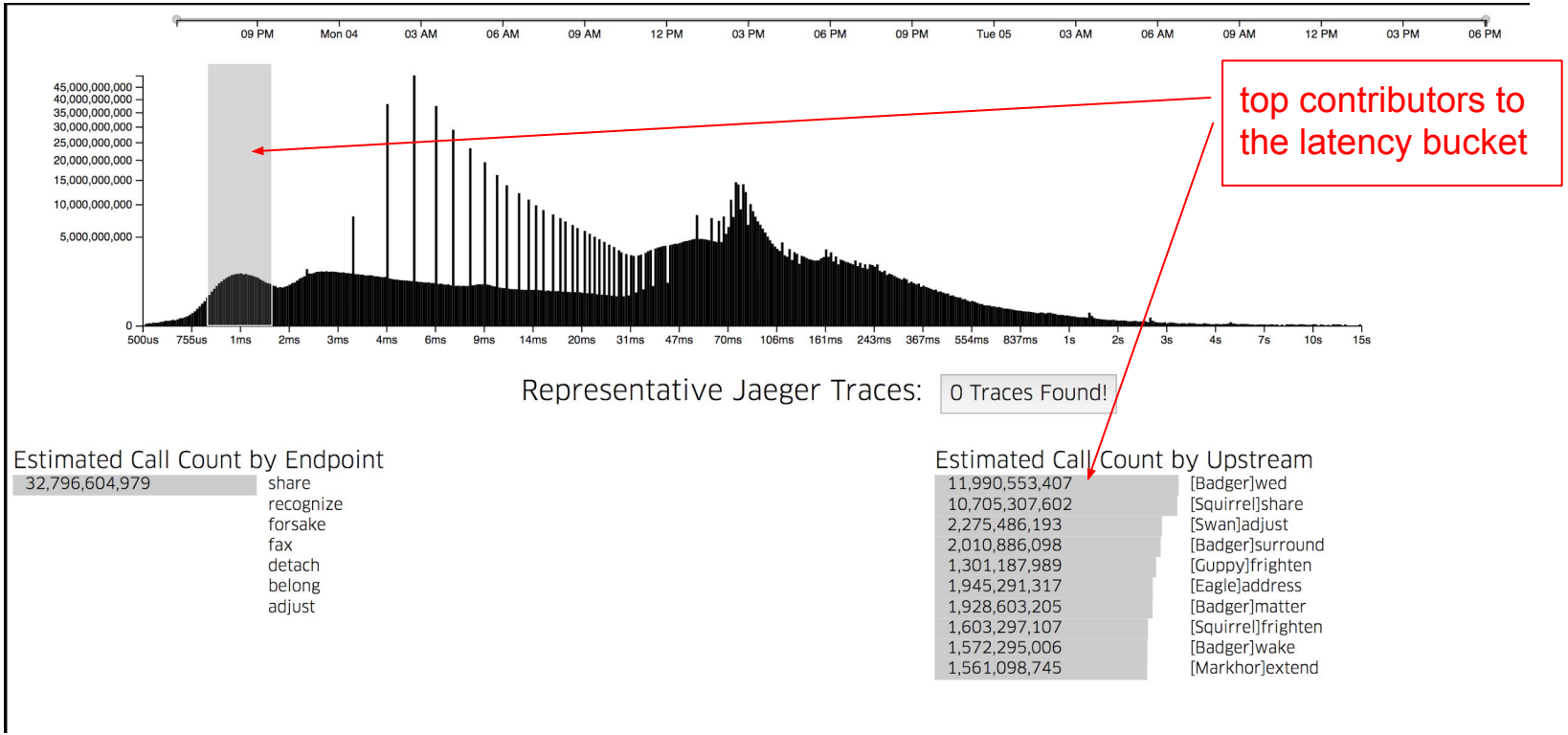
Dingo → Shrimp → Dog.
Does it mean Dingo depends on Dog?



Path-Based Service Dependency Diagrams



Latency Histograms



Q & A

Open Discussion

Getting in Touch

- GitHub: https://github.com/jaegertracing
- Chat: https://gitter.im/jaegertracing/
- Mailing List - jaeger-tracing@googlegroups.com
- Blog: https://medium.com/jaegertracing
- Twitter: https://twitter.com/JaegerTracing
- Bi-Weekly Community Meetings