

# MONITOR AND IDENTIFY ISSUES IN A MICROSERVICES ARCHITECTURE

**THOMAS HEUTE**

SENIOR ENGINEER MANAGER

RED HAT

# THE PROBLEM



**Honest Status Page**

@honest\_update

Follow



We replaced our monolith with micro services so that every outage could be more like a murder mystery.

1:10 AM - 8 Oct 2015

3,021 Retweets 2,465 Likes



20



3.0K

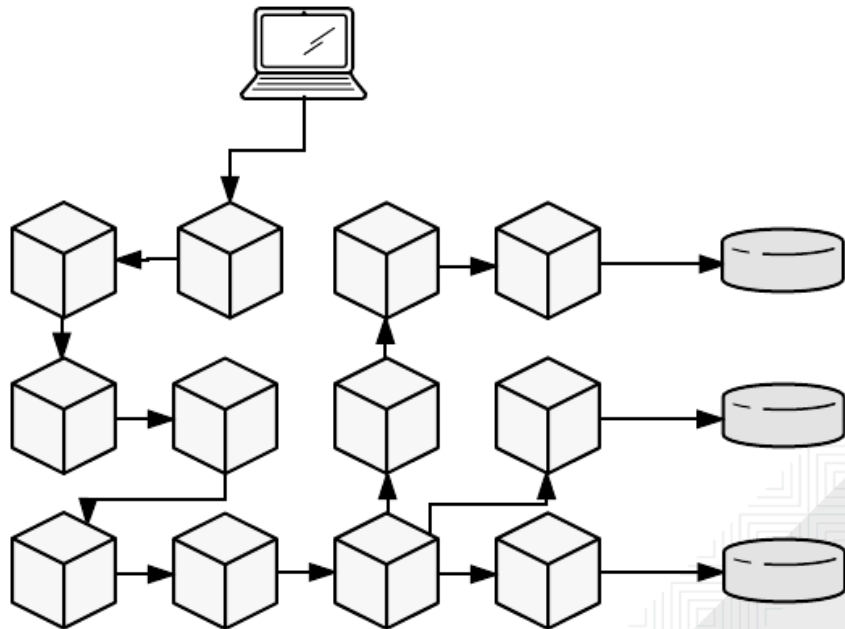
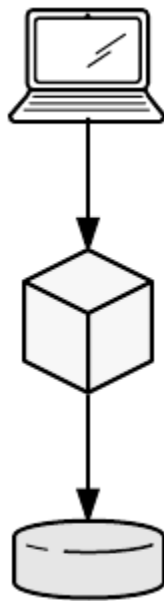


2.5K



# MONOLITH VS MICROSERVICES

Where is the problem ? / Where is it slow ?



# It's worse !

Along with Kubernetes those microservices are made to

- Be restarted
- Update frequently
- Scale up/down independently

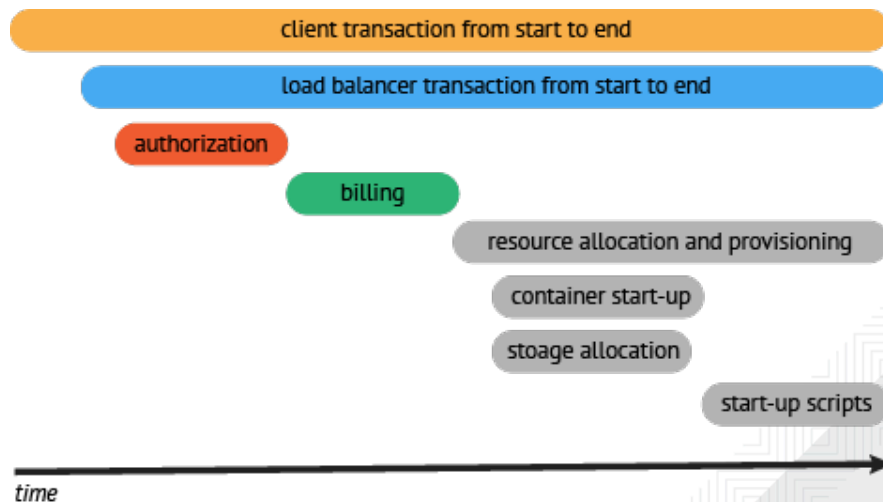
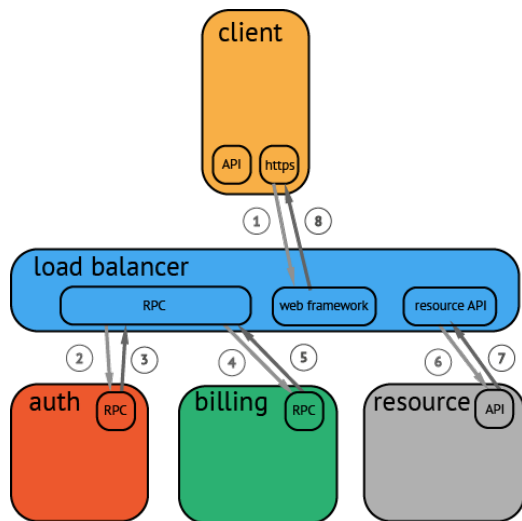
\* Not necessarily related, but often related



# WELCOME OPENTRACING !

# What is Distributed Tracing ?

- A story teller





# What is OpenTracing ?

- Vendor neutral distributed tracing API
- Associate logs relevant to a particular trace
- Have APIs for multiple languages
  - As of today: Go, Python, Javascript, Java, C#, Objective-C, C++, Ruby, PHP
- Part of CNCF: Cloud Native Computing Foundation
  - Like Kubernetes, Prometheus...

# Why should I care ?

- Popular frameworks have better chance to be instrumented
  - JAX-RS, Spring, JDBC, Kafka client, CDI, Mongo, ElasticSearch, Redis...
  - Useful data out of the box
- Instrument once for any tracer
  - Jaeger, Zipkin, Lightstep and a few more
- Can build a trace from (web) client, down to the database

# Jaeger

# Jaeger

- Built for OpenTracing
  - Perfect mapping of the models
- Born with real needs
  - A tracer implementation built by Uber, used for ~2000 microservices architecture
- Opensourced in April 2017
  - Contributions from Red Hat, in particular on frameworks instrumentation (OpenTracing), support for OpenShift...
- Support for multiple languages
  - Go, Java, Python, JS/Node.js

## frontend: HTTP GET /dispatch

View Options ▾

Search...

Trace Start: April 10, 2017 1:59 PM Duration: 736.564ms Services: 6 Depth: 5 Total Spans: 50



Span Name	Timeline	184.14ms	368.28ms	552.42ms	736.56ms
frontend HTTP GET /dispatch					
frontend HTTP GET: /customer			338.65ms		
frontend HTTP GET			338.56ms		
customer HTTP GET: /c...			338.06ms		
mysql SQL SELECT			337.86ms		
frontend Driver::findNearest			175.26ms		
redis GetDriver			9.62ms		
driver Driver::findNear...			174.61ms		
redis FindDriverIDs			19.31ms		
redis GetDriver			7.29ms		
redis GetDriver			9ms		
redis GetDriver			29.24ms		
redis GetDriver			11.57ms		
redis GetDriver			10.08ms		
redis GetDriver			11.26ms		
redis GetDriver			7.74ms		
redis GetDriver			30.06ms		
redis GetDriver			5.67ms		
redis GetDriver			12.71ms		
redis GetDriver			0.01ms		

# Jaeger

frontend HTTP GET /dispatch

## HTTP GET /dispatch

Service: frontend Duration: 774.85ms Start Time: 0ms

Tags: sampler.type=const sampler.param=true http.method=GET http.url=/dispatch?customer=123&nonce=0.8534872559455979 component=net/http h..  
Process: hostname=ys-C02PQB6FG8WM ip=192.168.1.4 jaeger.version=Go-2.6.0

### Logs (17)

0.04ms: event=HTTP request received level=info method=GET url=/dispatch?customer=123&nonce=0.8534872559455979  
0.09ms: event=Getting customer customer\_id=123 level=info  
308.12ms: event=Found customer level=info  
308.18ms: event=Finding nearest drivers level=info location=115,277  
566.85ms: event=Found drivers level=info  
566.99ms: event=Finding route dropoff=115,277 level=info pickup=232,869  
566.99ms: event=Finding route dropoff=115,277 level=info pickup=619,253  
567.02ms: event=Finding route dropoff=115,277 level=info pickup=921,217  
612.76ms: event=Finding route dropoff=115,277 level=info pickup=350,45  
631.34ms: event=Finding route dropoff=115,277 level=info pickup=616,351  
636.24ms: event=Finding route dropoff=115,277 level=info pickup=122,79  
667.41ms: event=Finding route dropoff=115,277 level=info pickup=204,996  
682.43ms: event=Finding route dropoff=115,277 level=info pickup=411,489  
698.28ms: event=Finding route dropoff=115,277 level=info pickup=194,47  
711.68ms: event=Finding route dropoff=115,277 level=info pickup=43,664  
774.71ms: event=Found routes level=info  
774.79ms: event=Dispatch successful driver=T746494C eta=2m0s level=info

# Jaeger

- Docker ready
- Kubernetes ready
- OpenShift ready
  - (But not “Red Hat supported” yet)
- Very likely joining the CNCF (Incubation project, skipping “inception”)
  - <https://lists.cncf.io/pipermail/cncf-toc/2017-September/001149.html>
  - So OpenTracing API + Jaeger server would be part of CNCF

# Made for high scale - Sample technique





Instrument once - use twice

# Metrics

- Expose metrics as Prometheus endpoints
  - Error rates
  - Response times
  - Business metrics
    - Span Baggage
- Metrics about all transactions not just samples

# DEMO



# RED HAT **FORUM**

Europe, Middle East & Africa