JAVA DAY KYIV 2017

LATENCY TRACING IN DISTRIBUTED JAVA APPLICATIONS

KONSTANTIN SLISENKO
LEAD SOFTWARE ENGINEER

NOV 5, 2017

Konstantin Slisenko

Java Team Lead in EPAM

Financial services, trading solutions

Speaker at Java Meetups

github.com/kslisenko

kanstantsin_slisenka@epam.com

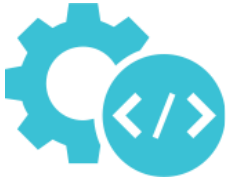
MY TALK IS ABOUT...


www

AGENDA

1. What is latency tracing?

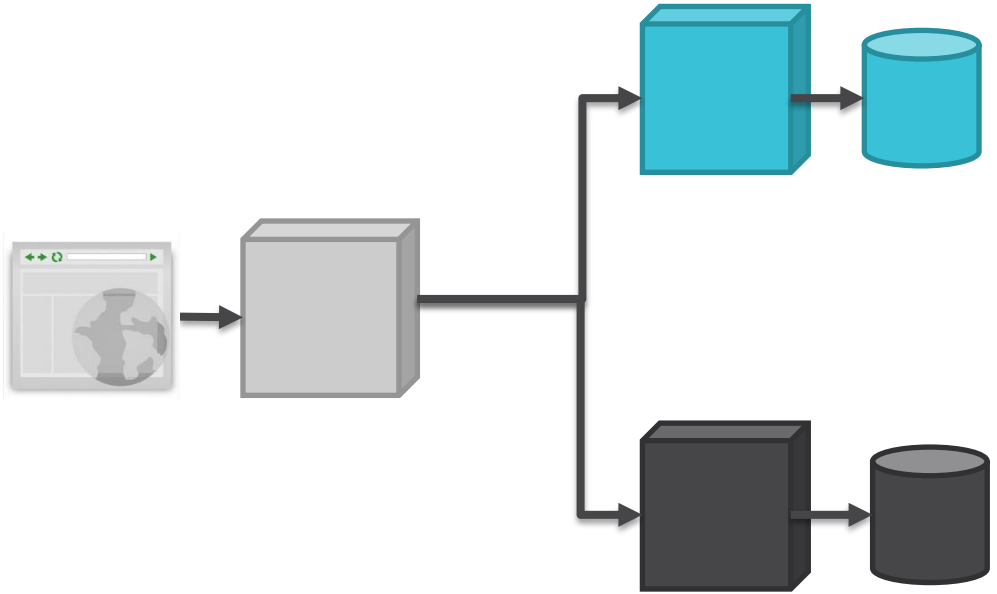
2. How it works?

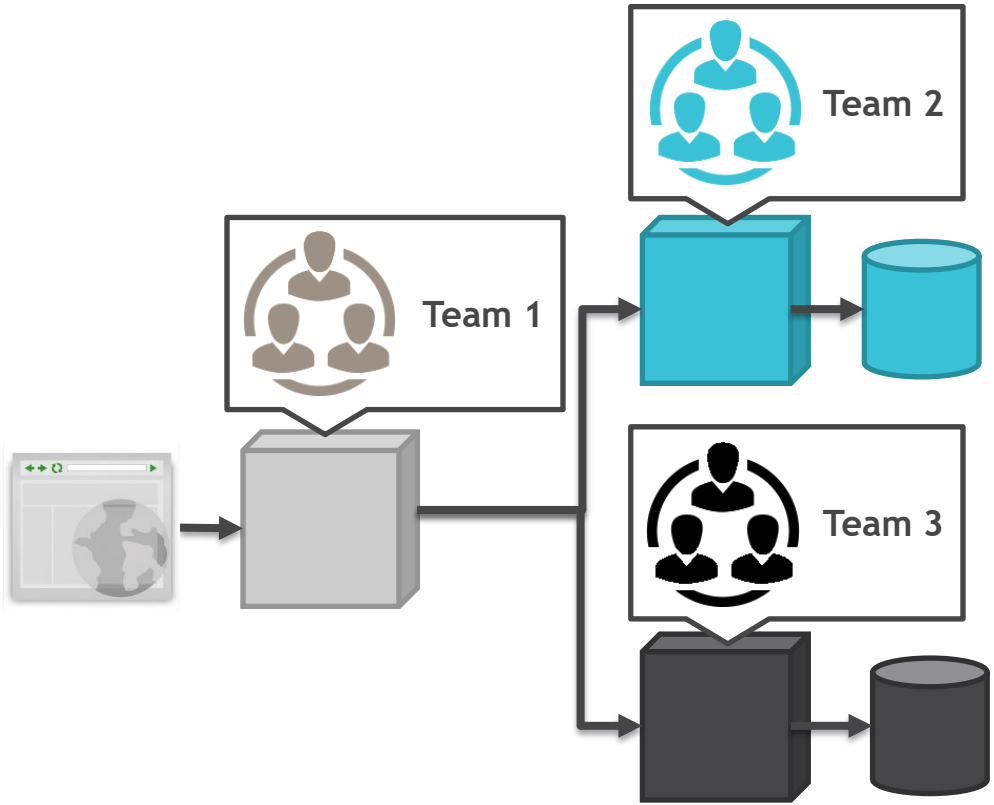


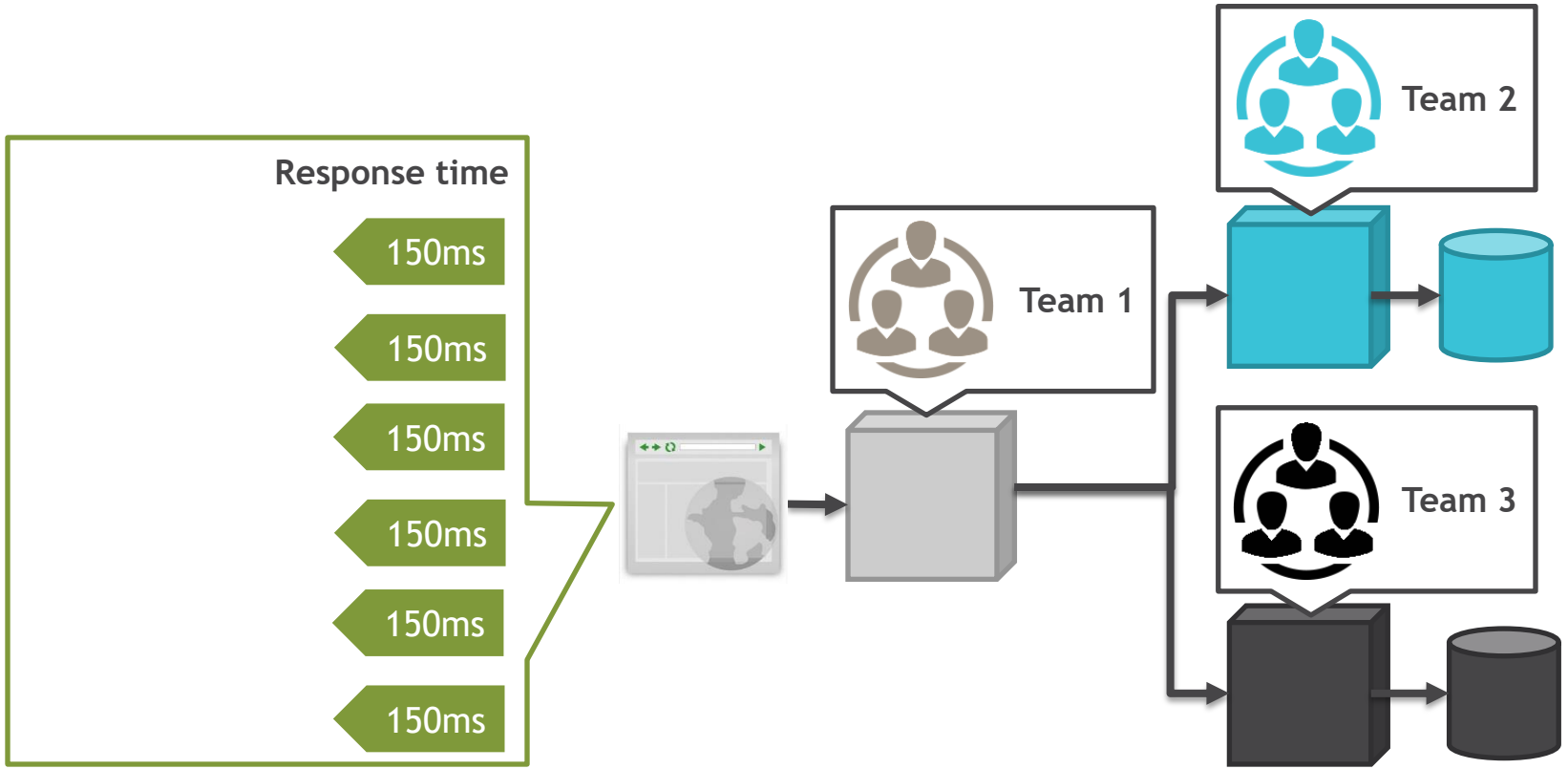
3. Live demo

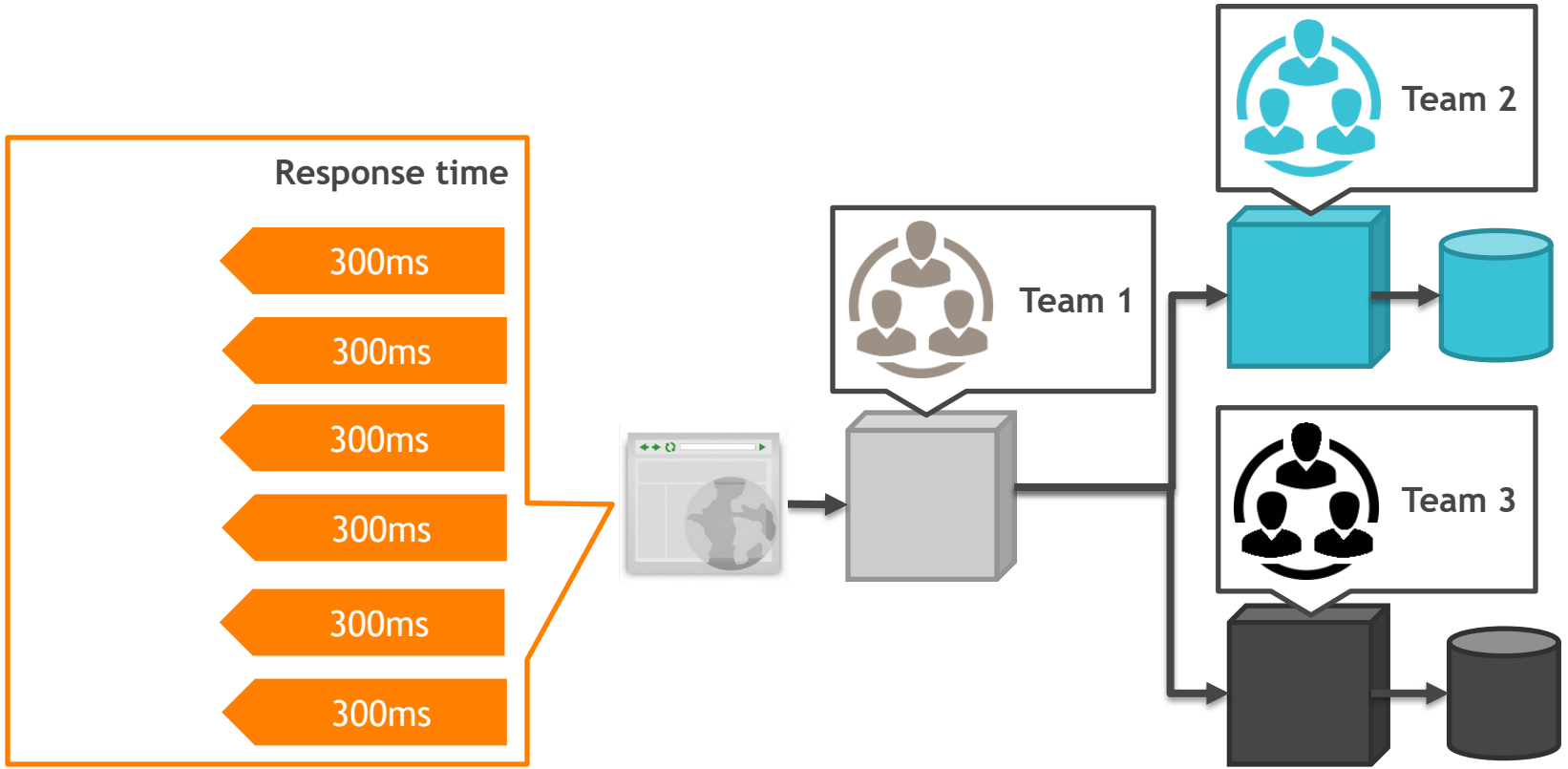
LITTLE STORY ABOUT

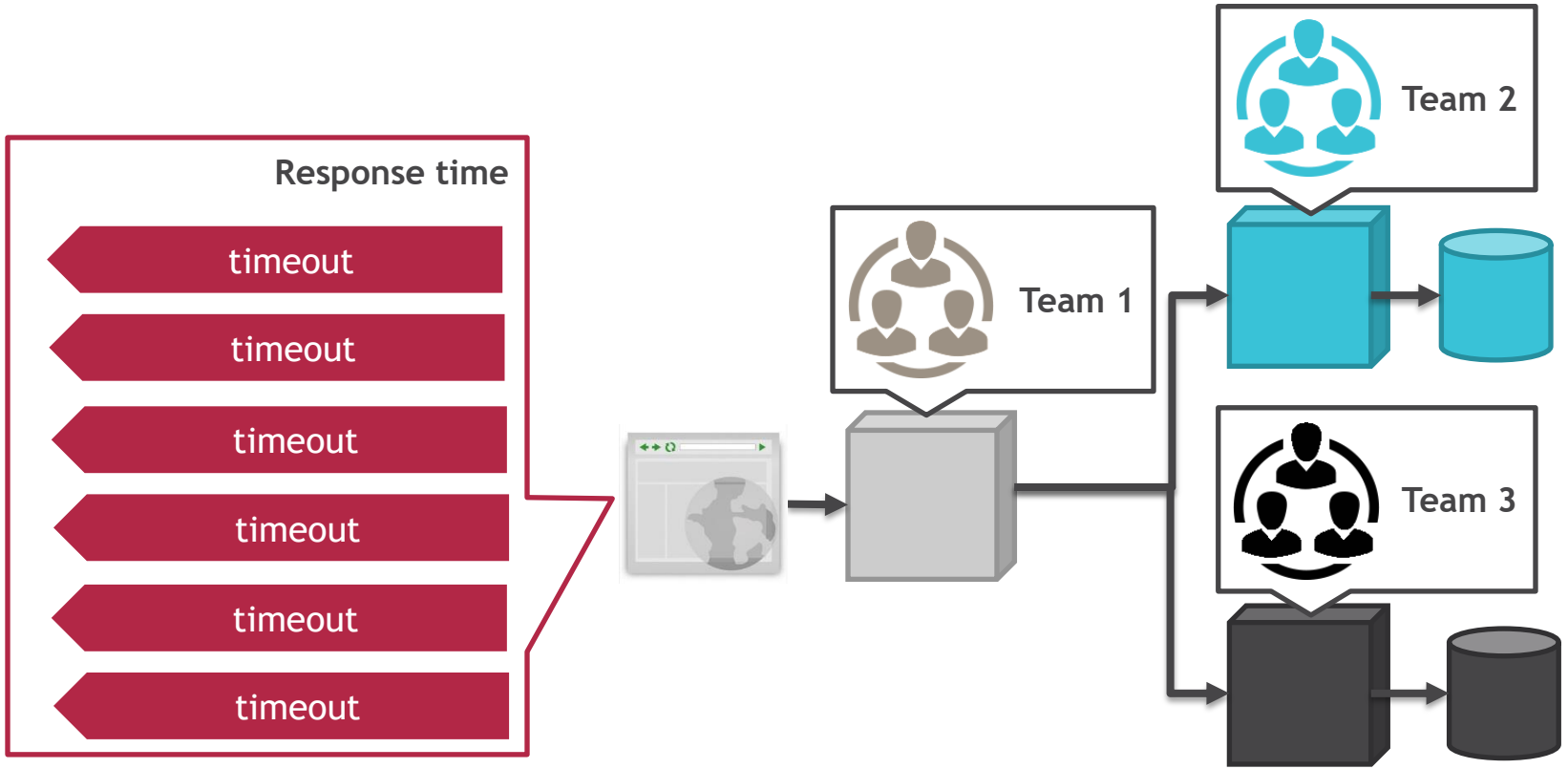
ONE SYSTEM

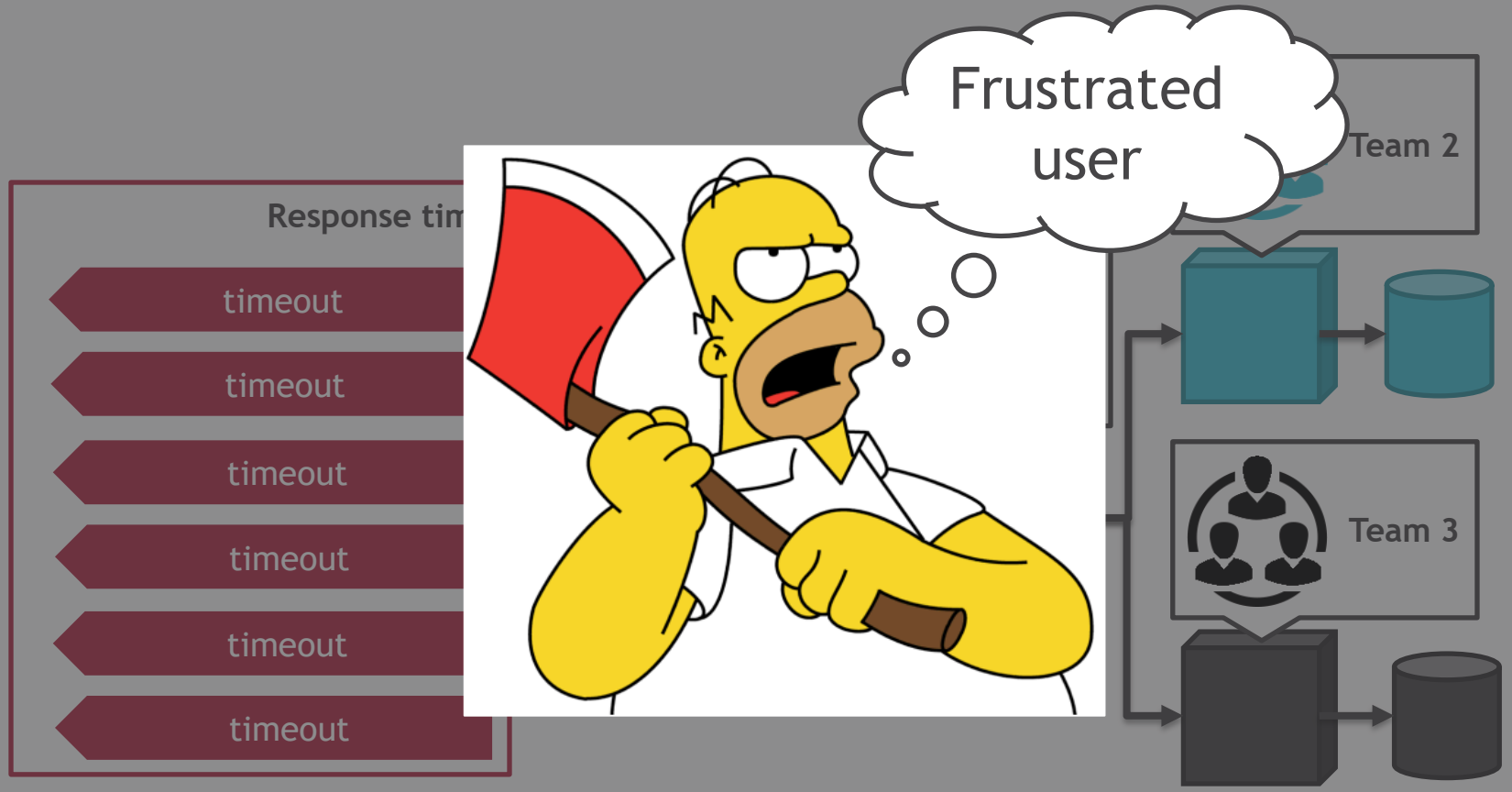



Team 2
Team 1
Team 3


Response time
150ms
150ms
150ms
150ms
150ms
150ms
Team 1
Team 2
Team 3


Response time
300ms
300ms
300ms
300ms
300ms
300ms
Team 1
Team 2
Team 3


Response time
timeout
timeout
timeout
timeout
timeout
timeout
Team 1
Team 2
Team 3


Frustrated user
Team 2
Team 3
Response tim
timeout
timeout
timeout
timeout
timeout
timeout

1. Whose fault?

1. ~~Whose fault?~~

Where?

1. ~~Whose fault?~~

   Where?

2. Why?

1. ~~Whose fault?~~

   Where?

2. Why?

3. How to prevent?

PROFILERS, LOGS, METRICS?

logstash

kibana

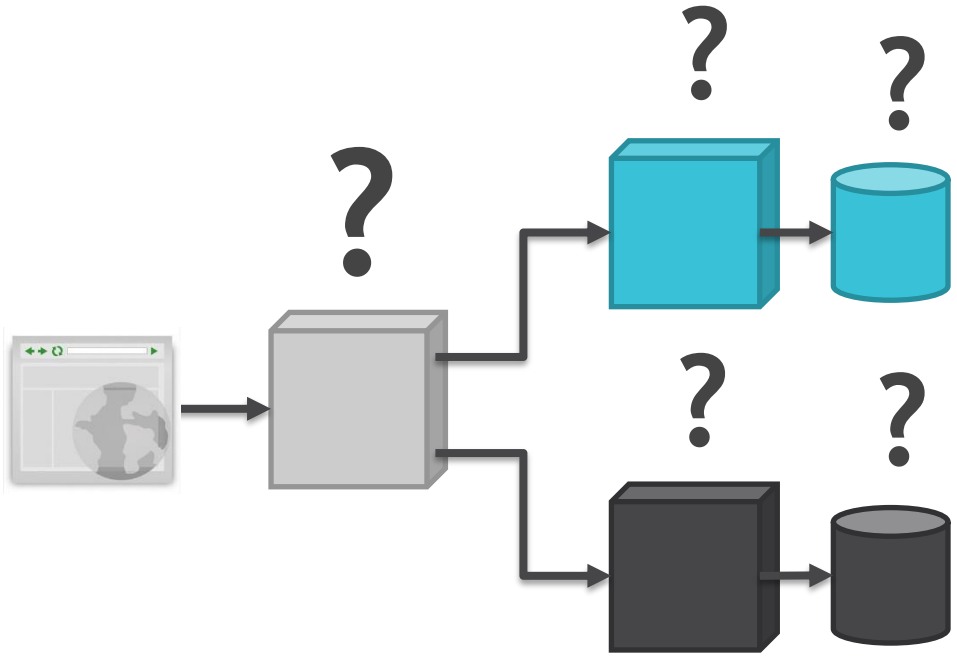
elasticsearch

Grafana

YourKit

JProfiler

VisualVM



*"When systems involve not just **dozens of subsystems** but **dozens of engineering teams**, even our best and most experienced engineers routinely guess wrong about the **root cause of poor end-to-end performance**"*

https://research.google.com/pubs/pub36356.html

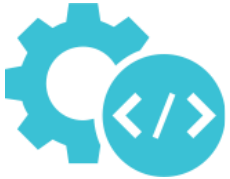
THE MOMENT WHEN YOU NEED

LATENCY TRACING

AGENDA

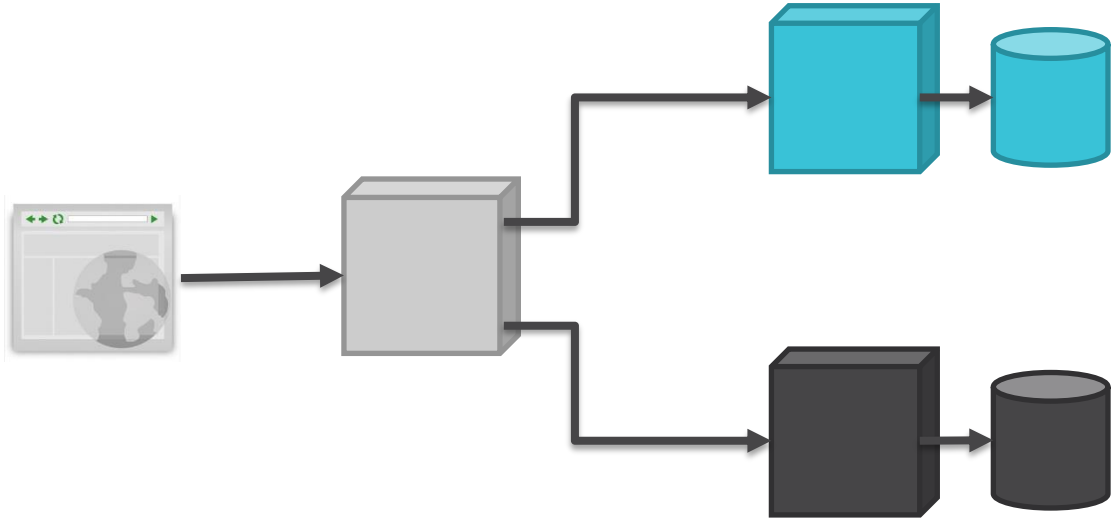
1. ~~What is latency tracing?~~

2. How it works?

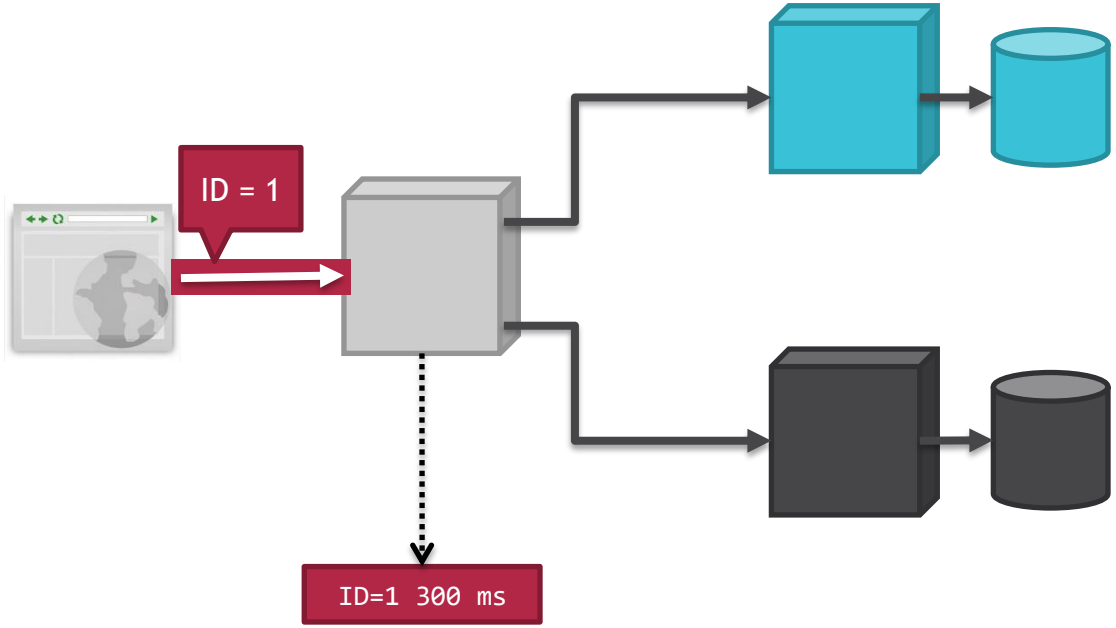


3. Live demo

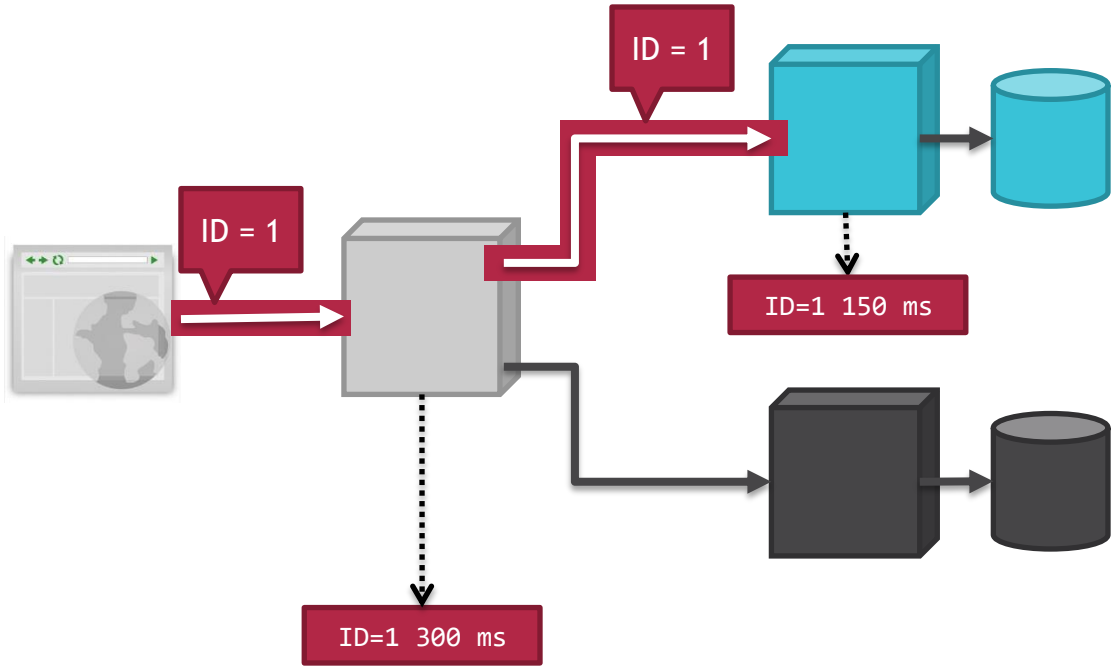
HOW IT WORKS



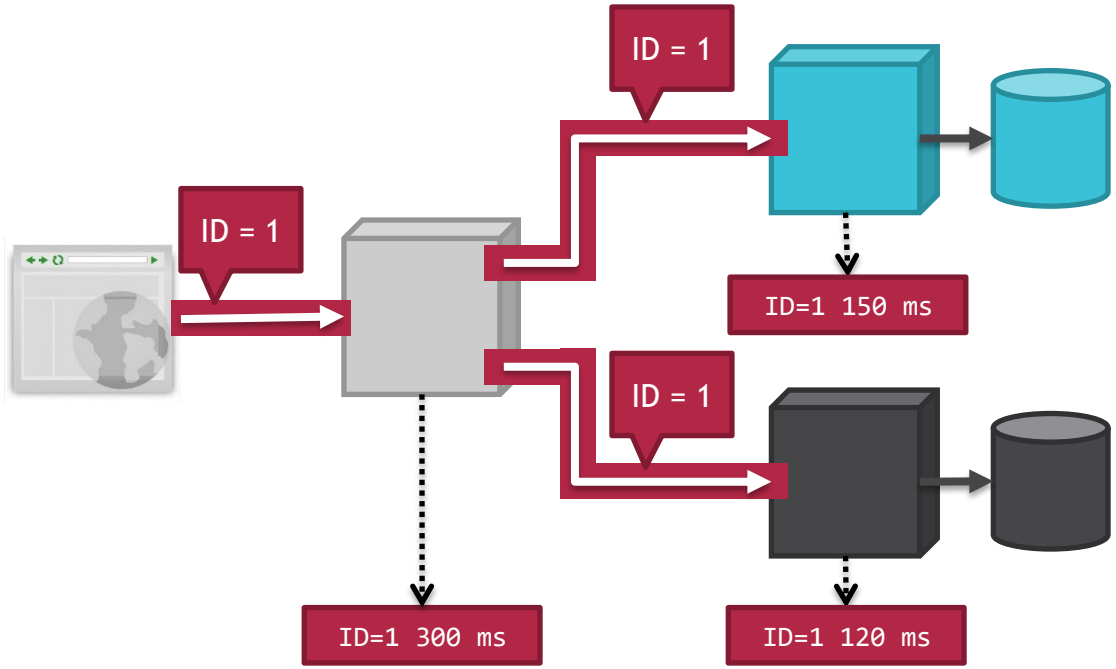
HOW IT WORKS



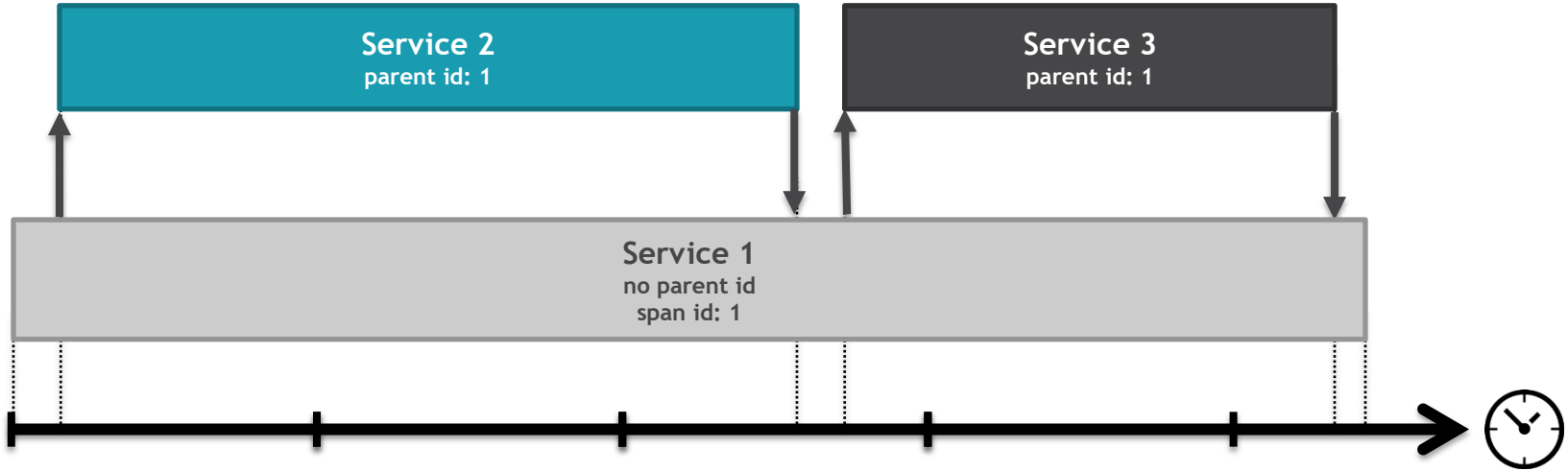
HOW IT WORKS



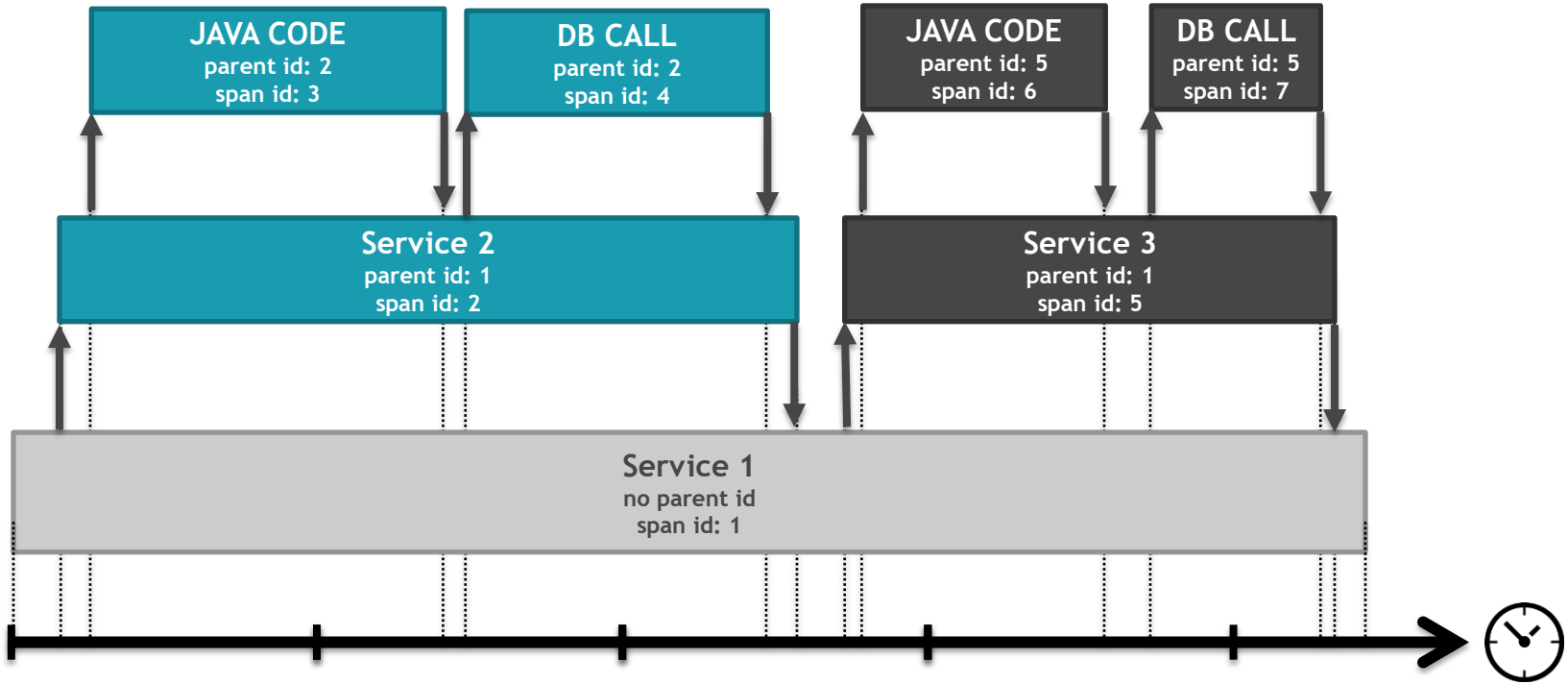
HOW IT WORKS



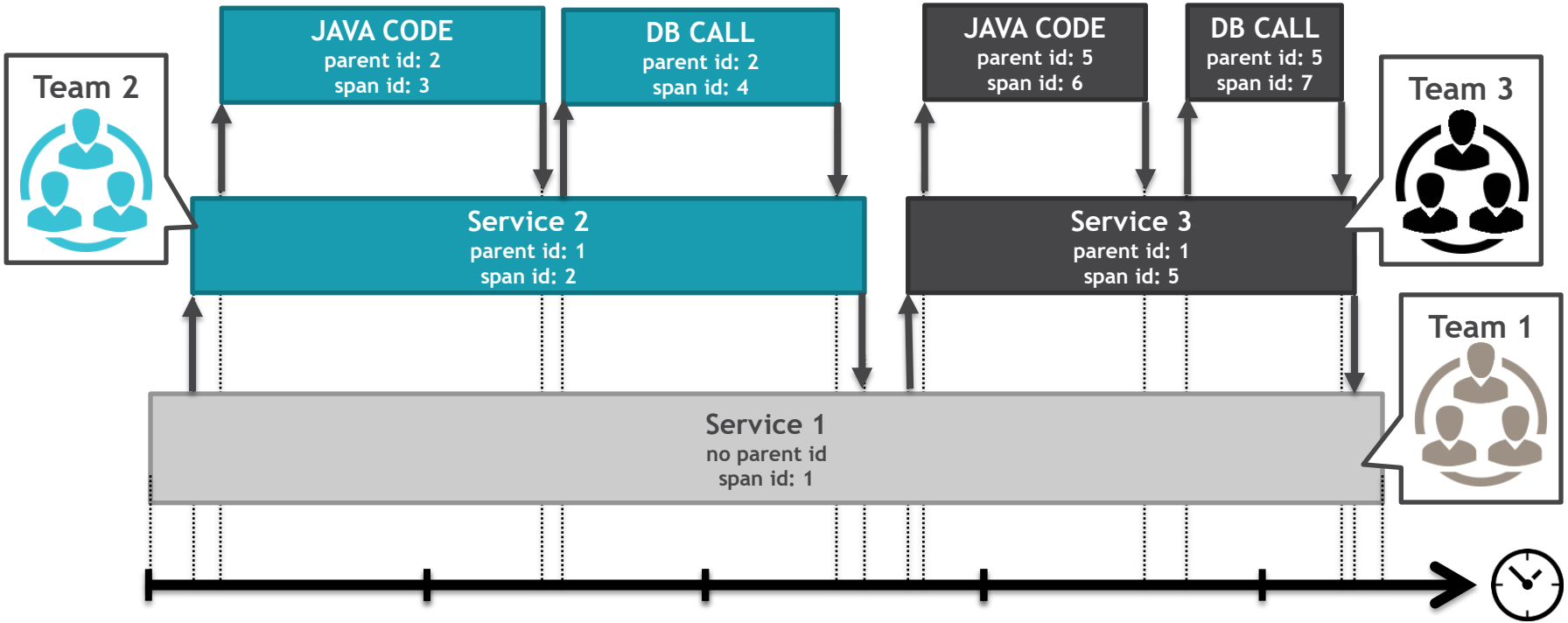
TRACES AND SPANS



TRACES AND SPANS



TRACES AND SPANS



1. ~~Whose fault?~~

   Where?

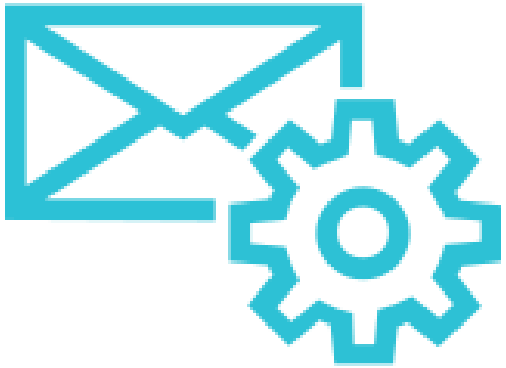
2. Why?

3. How to prevent?

HOW DO I ADD THIS TO MY PROJECT?

SO, THE PLAN IS

1. Pass request IDs between tiers
2. Measure and report processing time
3. Collect **traces** and **spans**



Communication protocols

- ✓ Pass trace/span IDs
- ✓ Use HTTP headers, JMS attrs
- ✓ Modify custom protocols

Entry points

- ✓ Intercept communication frameworks (HTTP, JMS, RPC, …)
- ✓ Start new traces

Method execution flow

- ✓ Measure execution time
- ✓ Report new spans
- ✓ Capture method arguments
- ✓ Thread locals for trace/span IDs

Asynchronous invocation

- ✓ Intercept new thread starting
- ✓ Pass trace/span IDs to the new threads

WHAT NEEDS TO BE CHANGED

Communication protocols

- ✓ Pass trace/span IDs
- ✓ Use HTTP headers, JMS attrs
- ✓ Modify custom protocols

Entry points

- ✓ Intercept communication frameworks (HTTP, JMS, RPC…)
- ✓ Start new traces

Method execution flow

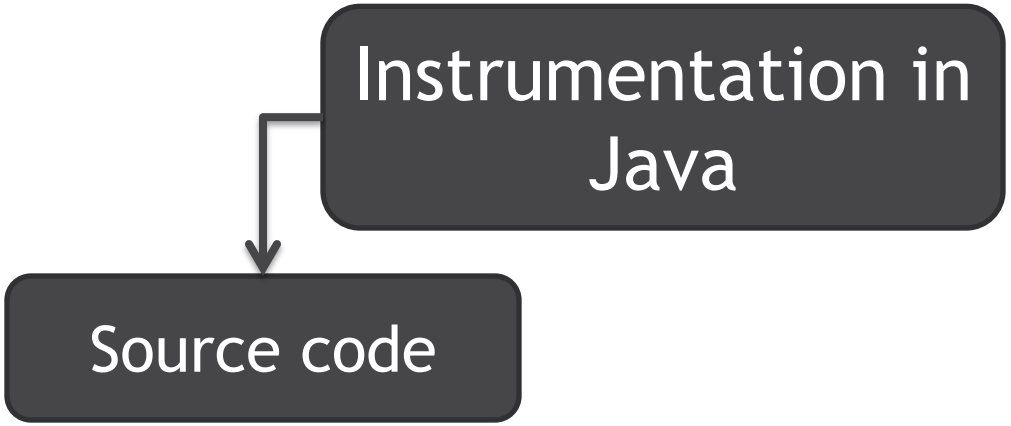
- ✓ Measure execution time
- ✓ Report new spans
- ✓ Capture method arguments
- ✓ Thread locals for trace/span IDs

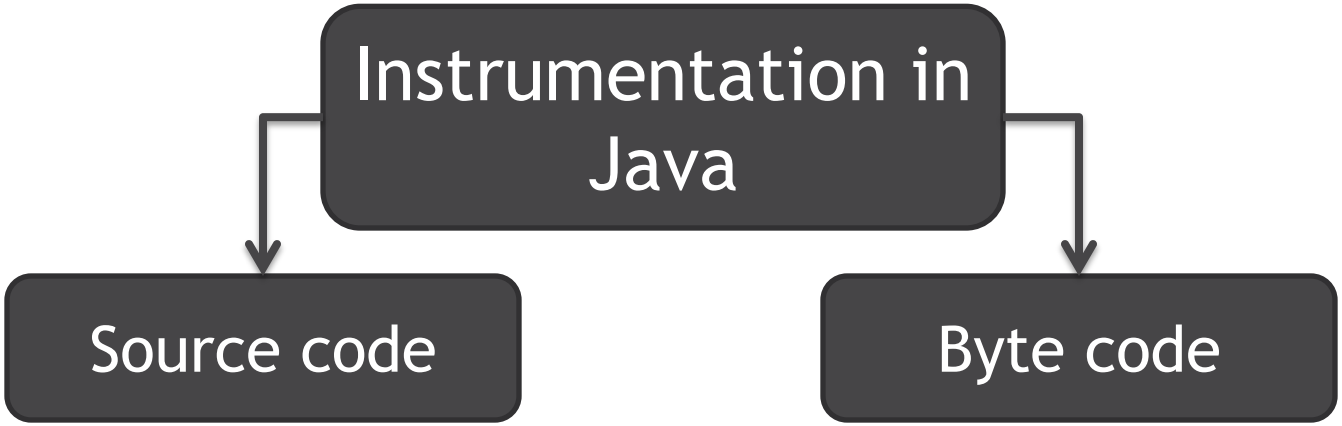
Asynchronous invocation

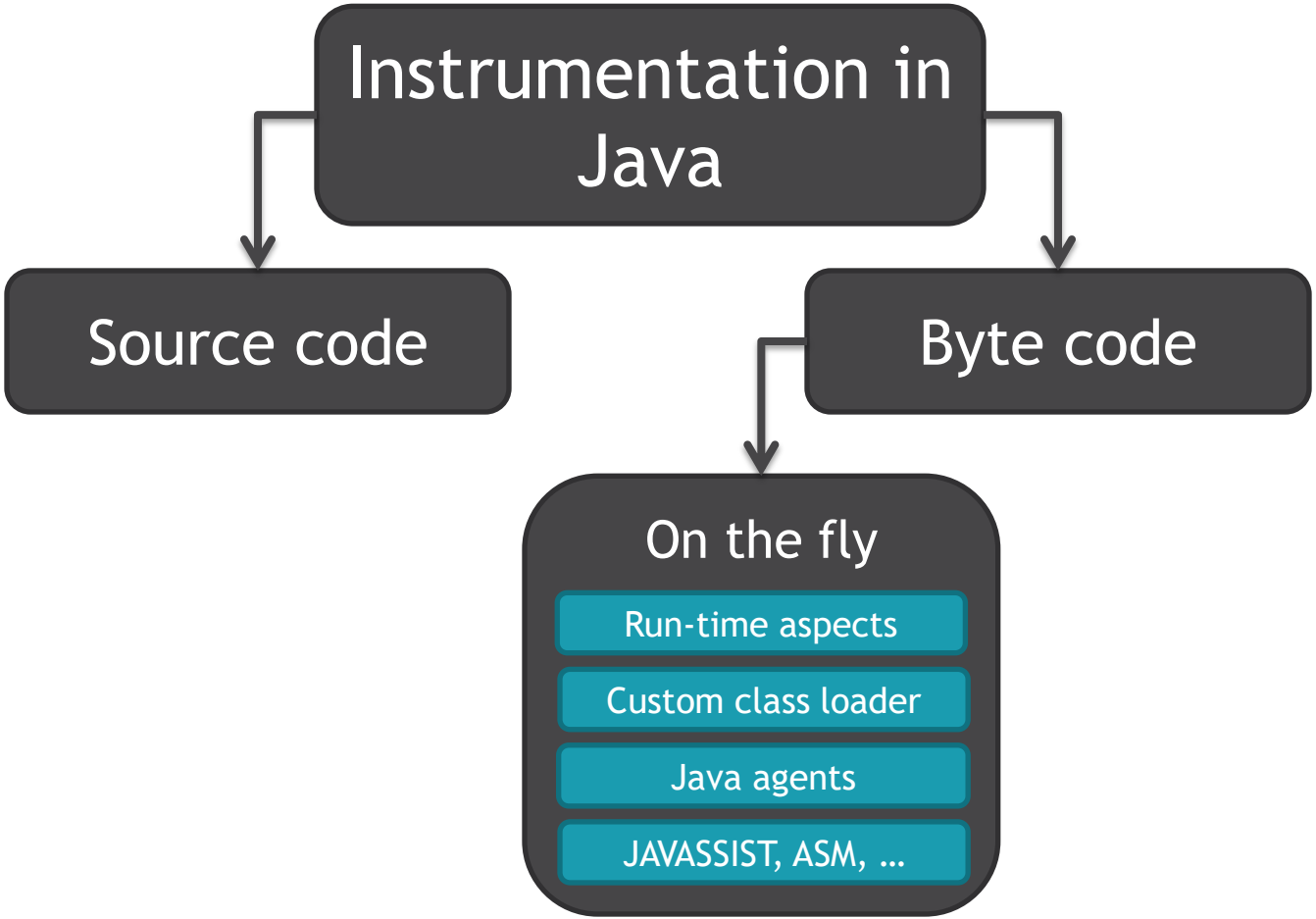
- ✓ Intercept new thread starting
- ✓ Pass trace/span IDs to the new threads

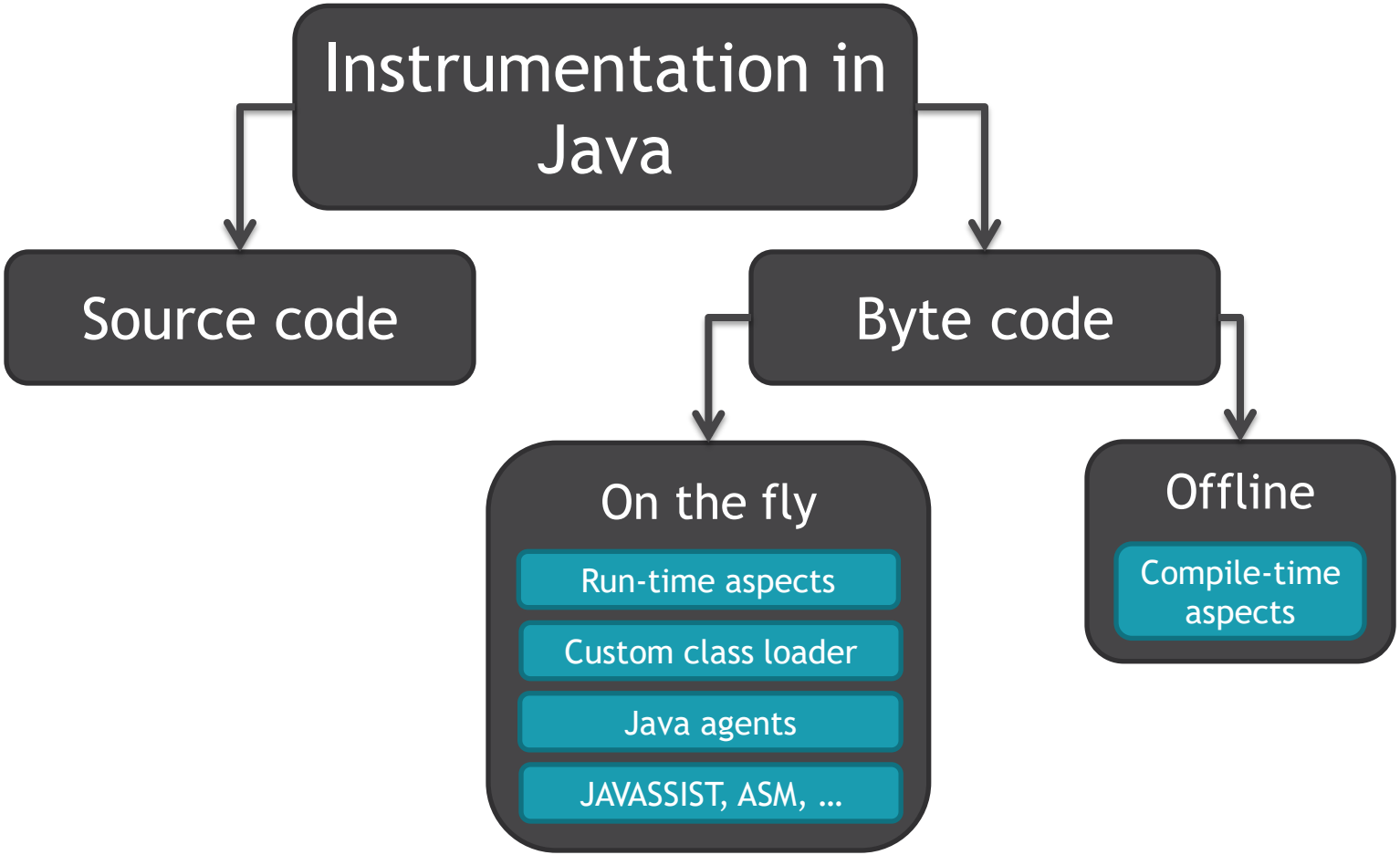
HOW DO I MODIFY MY JAVA APP?

Instrumentation in Java


Instrumentation in Java
Source code


Instrumentation in Java
Source code
Byte code


Instrumentation in Java
Source code
Byte code
On the fly
Run-time aspects
Custom class loader
Java agents
JAVASSIST, ASM, …


Instrumentation in Java
Source code
Byte code
On the fly
Run-time aspects
Custom class loader
Java agents
JAVASSIST, ASM, …
Offline
Compile-time aspects

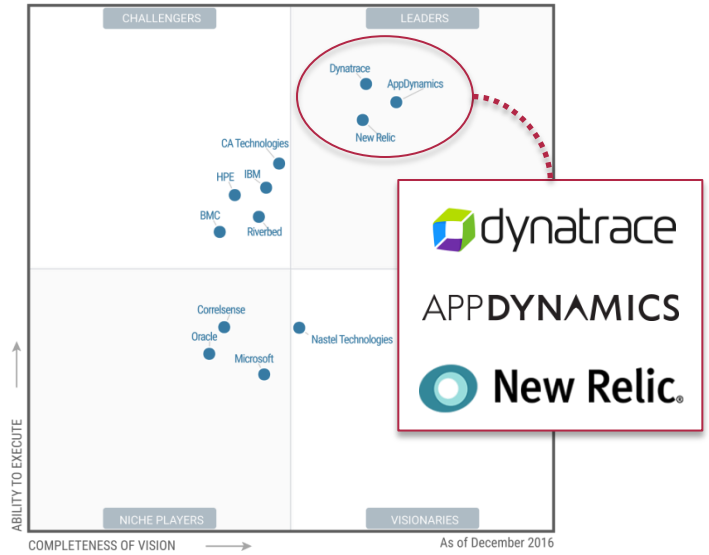
ANY EXISTING TOOLS?

COMMERCIAL

OPEN-SOURCE

Gartner

Magic Quadrant for Application Performance Monitoring Suites (21 December 2016)



https://www.gartner.com/doc/reprints?id=1-3OGTPY9&ct=161221

DZone / Java Zone

Java Performance Monitoring: 5 Open Source Tools You Should Know (19 January 2017)

stagemonitor

www.stagemonitor.org

PINPOINT

github.com/naver/pinpoint

Glowroot
Open source Java APM

glowroot.org

kamon

kamon.io

MoSKito
Essential

www.moskito.org

ZIPKIN

zipkin.io

https://dzone.com/articles/java-performance-monitoring-5-open-source-tools-you-should-know

OPENTRACING

A vendor-neutral open standard

for distributed tracing

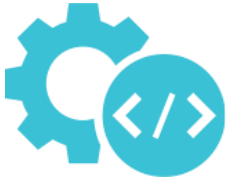
http://opentracing.io

```
Tracer tracer = ...;
Span parentSpan = ...;
Span span = tracer
    .buildSpan("someWork")
    .asChildOf(parentSpan.context())
    .withTag("foo", "bar")
    .start();
try {
    // Do things
} finally {
    span.finish();
}
```

AGENDA

1. ~~What is latency tracing?~~

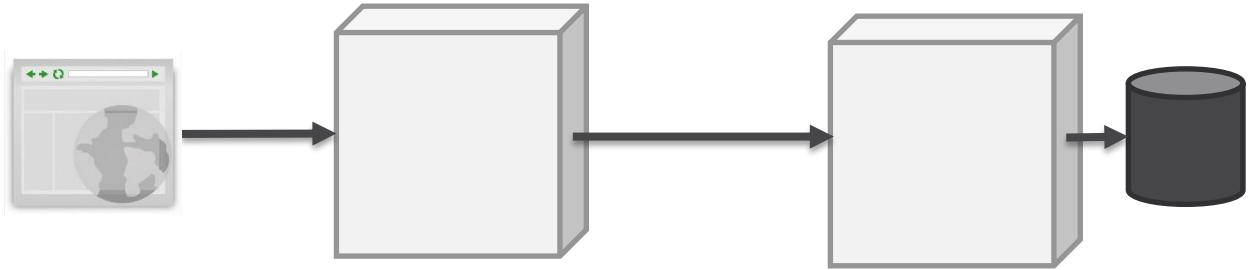
2. ~~How it works?~~

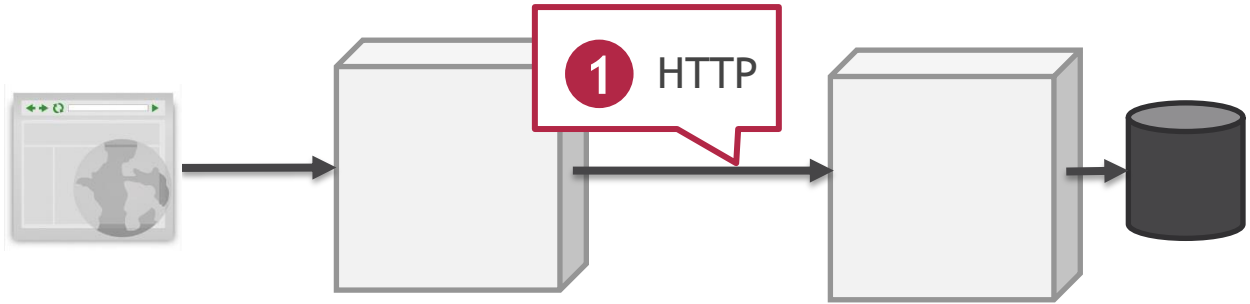


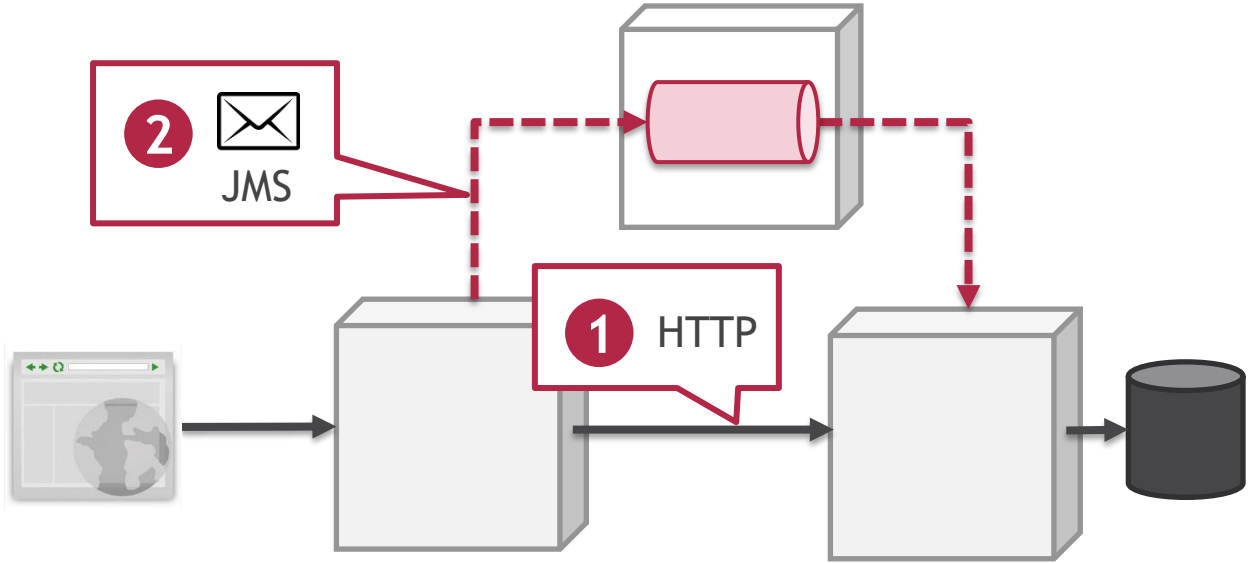
3. Live demo

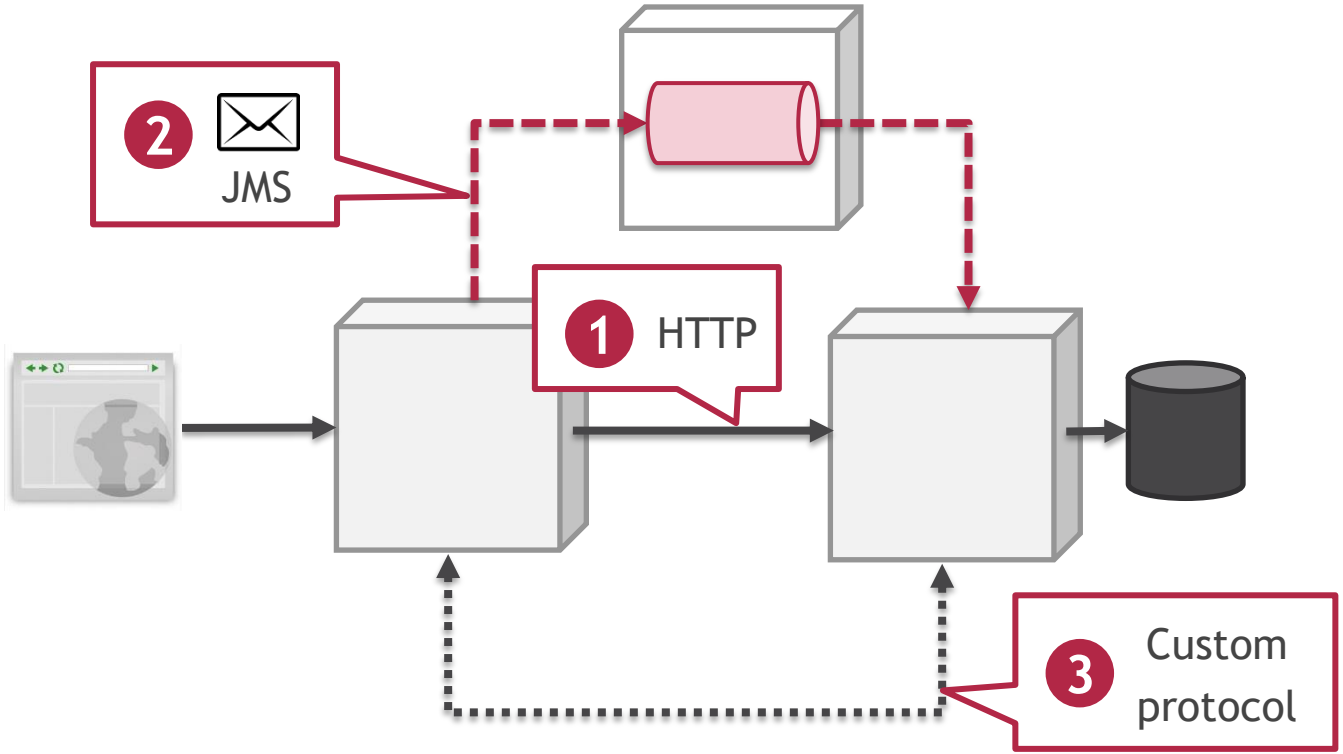
I'M GOING TO SHOW

http://github.com/kslisenko/java-performance




1
HTTP


2
JMS
1
HTTP


2
JMS
1
HTTP
3
Custom
protocol


LET’S GO!
github.com/kslisenko/java-performance

TAKE AWAYS

LATENCY TRACING ISSUES AND LIMITATIONS

1. Computation and I/O overhead
2. Custom protocols
3. Reactive streams, batch processing
4. Security and privacy

Latency tracing

- ✓ Must have — for microservices
- ✓ Better — in production
- ✓ At least — at performance testing


ЖЫЗНЬ БОРОДОЙ
latency tracing
ЖЫЗНЬ БЕЗ БОРОДЫ
latency tracing


CHICAGO
JAM WELCOMES BELLE AND SEBASTIAN
CIAL GUEST JULIEN BAKER
CHASE
JAM WELCOMES
BELLE AND SEBASTIAN
AUGUST 16
THANK YOU!
QUESTIONS?