Monitoring Series :

APPLICATION PERFORMANCE MONITORING

FUTURE

Infra Class : 12 Agustus 2020
Heronimus Tresy <heronimus.tresy@gdn-commerce.com>

In **previous** section ...

what we already had **Monitor** .. ?

- Server Monitoring
- Log Monitoring
- …...

ever heard **Application Performance Monitoring** (APM)..?

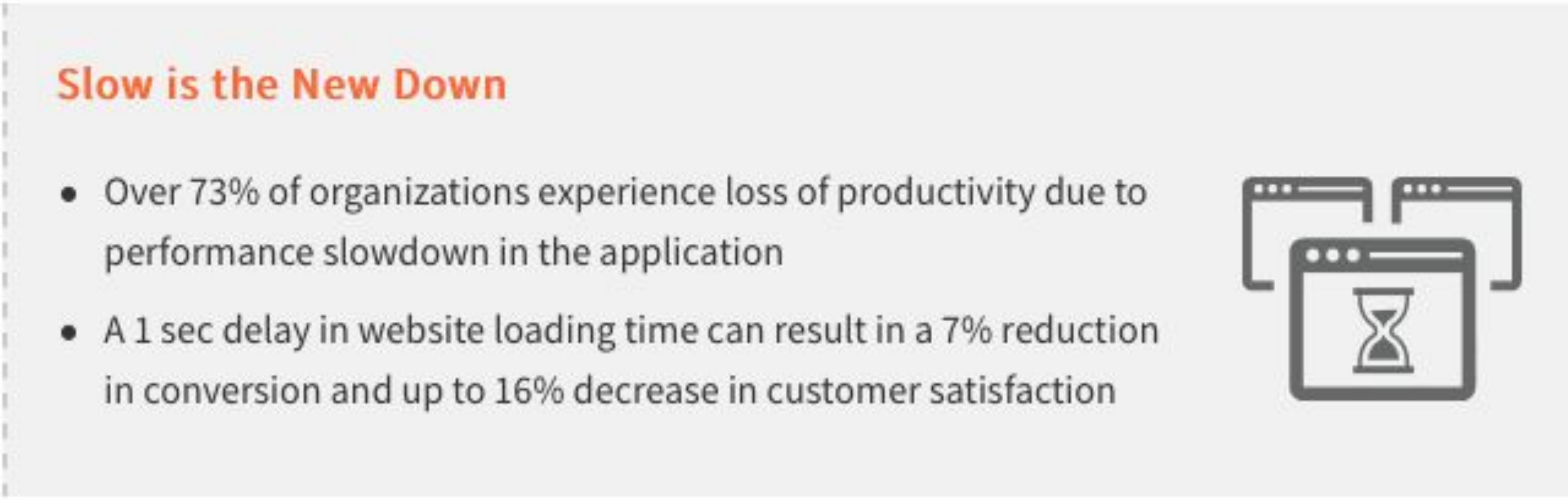
- Why we need APM
- Definition
- Performance metric
- Purposes
- APM Tools

Why we need APM?

IT Teams: We already build great infra architecture, all the apps already QA passed and run on production just fine, now can we relax?

Business: But why on earth our customer keep decreasing!

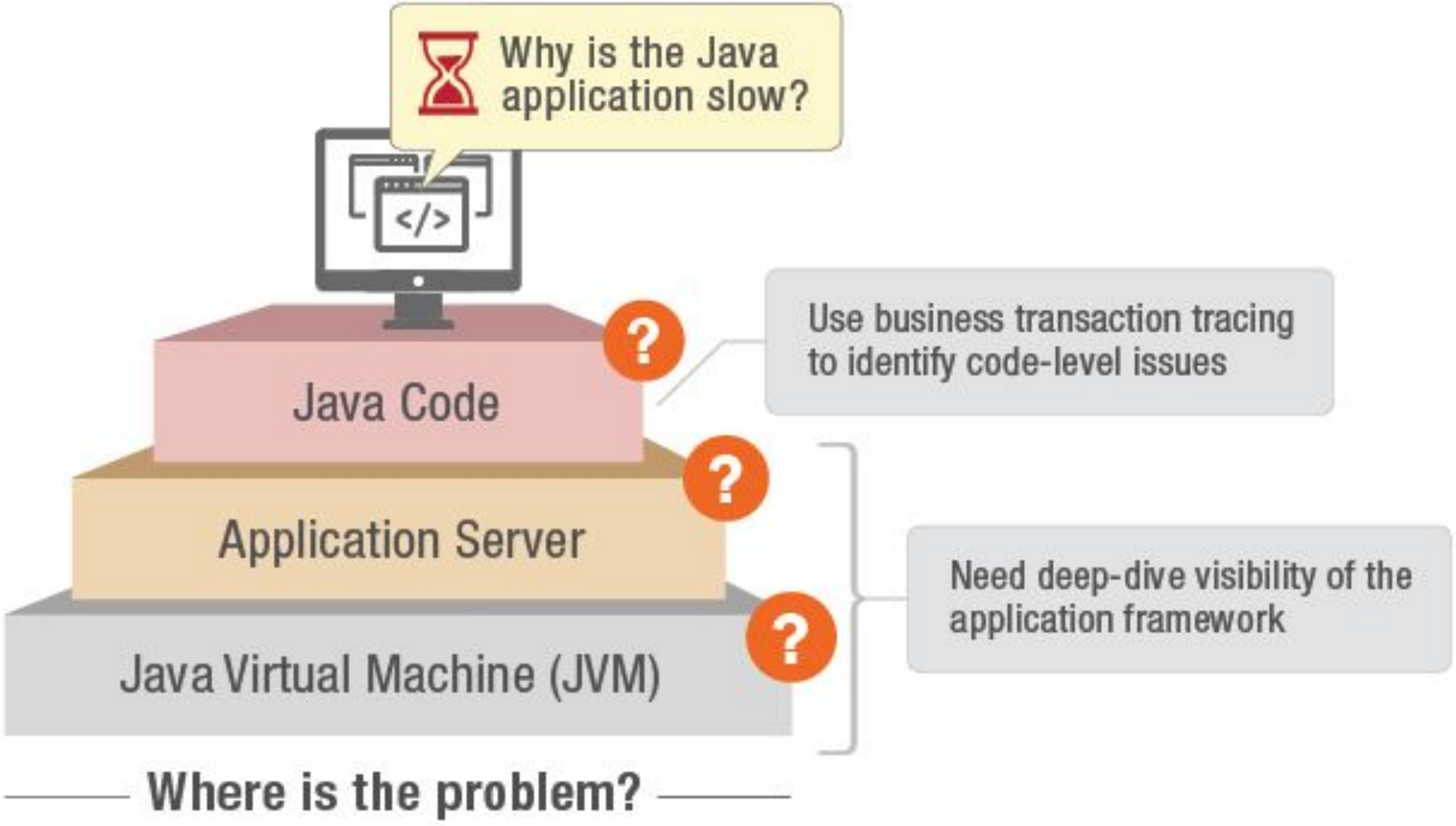
…. turns out that production apps **running slow**, can’t handle large request, and still have several unknown bugs left. Another restless day come for IT teams ...



Slow is the New Down

- Over 73% of organizations experience loss of productivity due to performance slowdown in the application
- A 1 sec delay in website loading time can result in a 7% reduction in conversion and up to 16% decrease in customer satisfaction

Slow is the New Down!



Why is the Java application slow?
Java Code
Application Server
Java Virtual Machine (JVM)
?
?
?
Use business transaction tracing to identify code-level issues
Need deep-dive visibility of the application framework
Where is the problem?

https://www.eginnovations.com/blog/what-is-application-performance-monitoring/

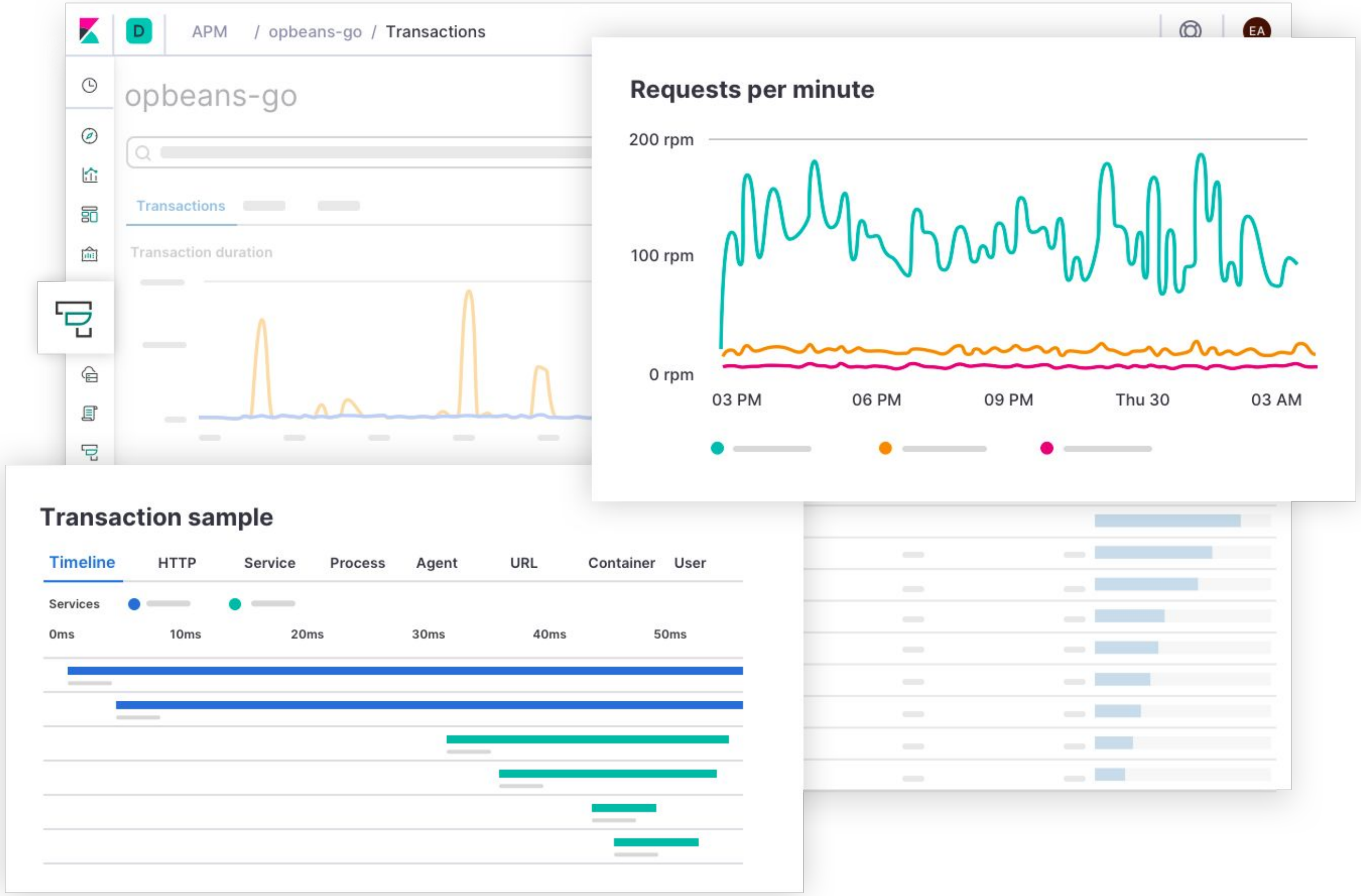
Definition: APM Monitoring & Management

- Monitoring and management of performance and availability of **software applications**
- Detect and diagnose complex application **performance problems** to maintain an **expected level of service**.
- APM is “the translation of **IT metrics into business meaning**”

https://www.apmdigest.com/the-anatomy-of-apm-4-foundational-elements-to-a-successful-strategy

Performance Metric

- Transaction record (trans. per second, pages per second, request per minute)
- Response times
- Error rate
- …..

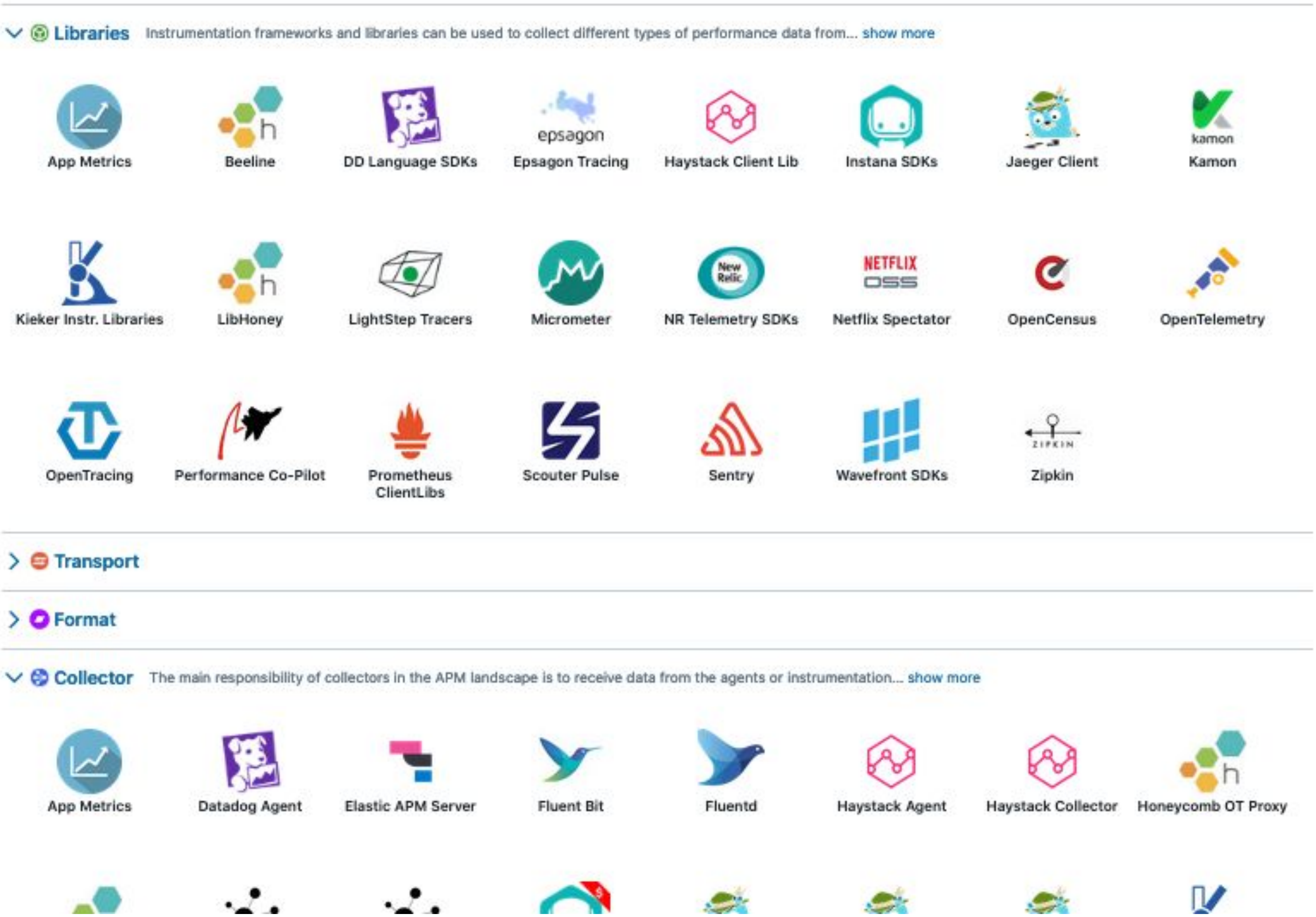


Purposes

- Problem Detection
- **End User Experience** Monitoring
- Reporting & Application **Data Analytics**

Application Performance Monitoring Tools

Many APM stack tools available ...



Libraries Instrumentation frameworks and libraries can be used to collect different types of performance data from... show more

Transport

Format

Collector The main responsibility of collectors in the APM landscape is to receive data from the agents or instrumentation... show more

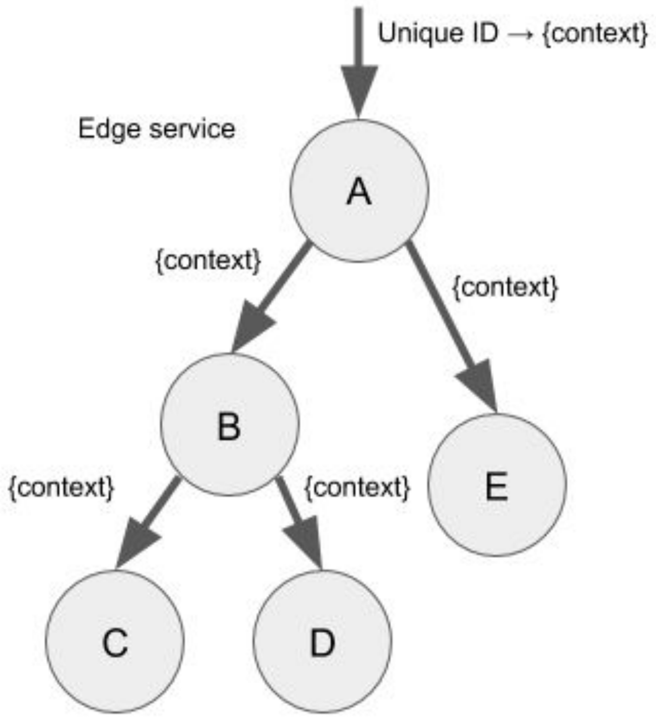
openapm.io/landscape

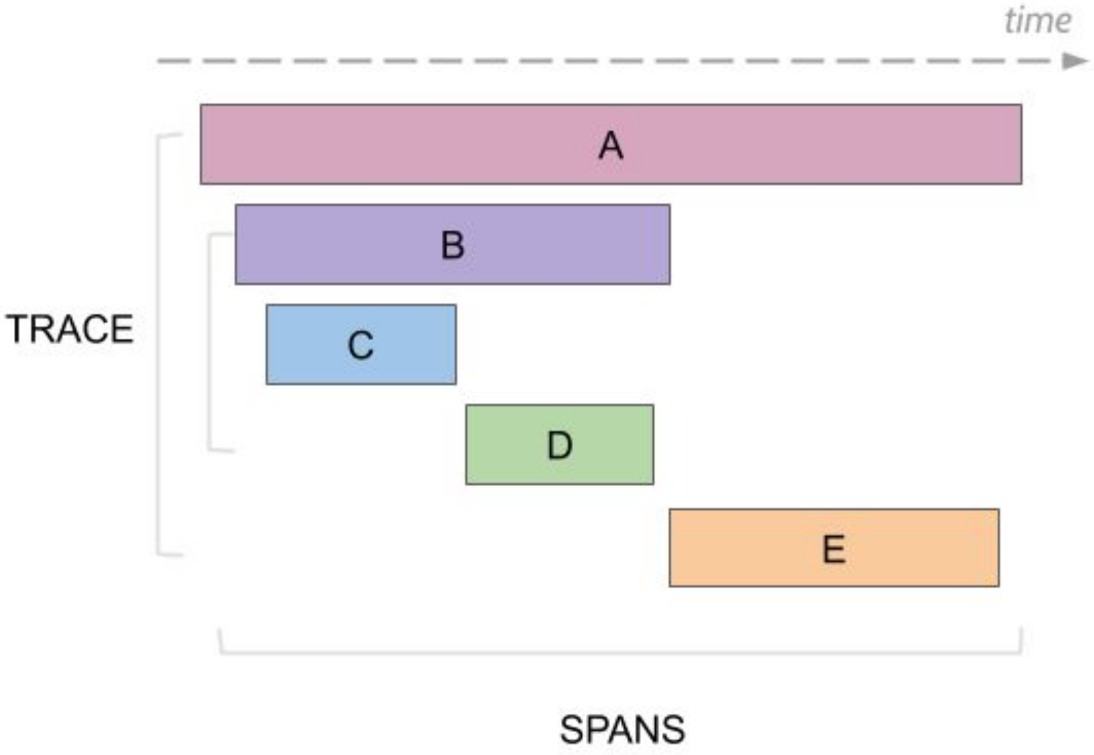
Terminology

OpenTracing Standard

Vendor-neutral APIs and instrumentation for distributed tracing application performance monitoring.

https://github.com/opentracing/specification/blob/master/specification.md

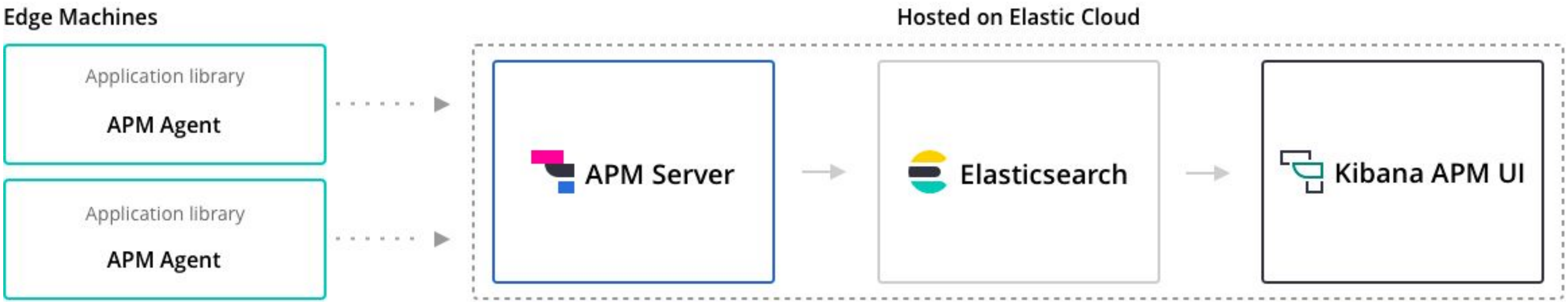




Elastic APM stack

for App Perf. Monitoring System.

Elastic APM Architecture



Edge Machines

Application library

APM Agent

Application library

APM Agent

Hosted on Elastic Cloud

APM Server

Elasticsearch

Kibana APM UI

APM Agent/Library need to be setup within the Apps.

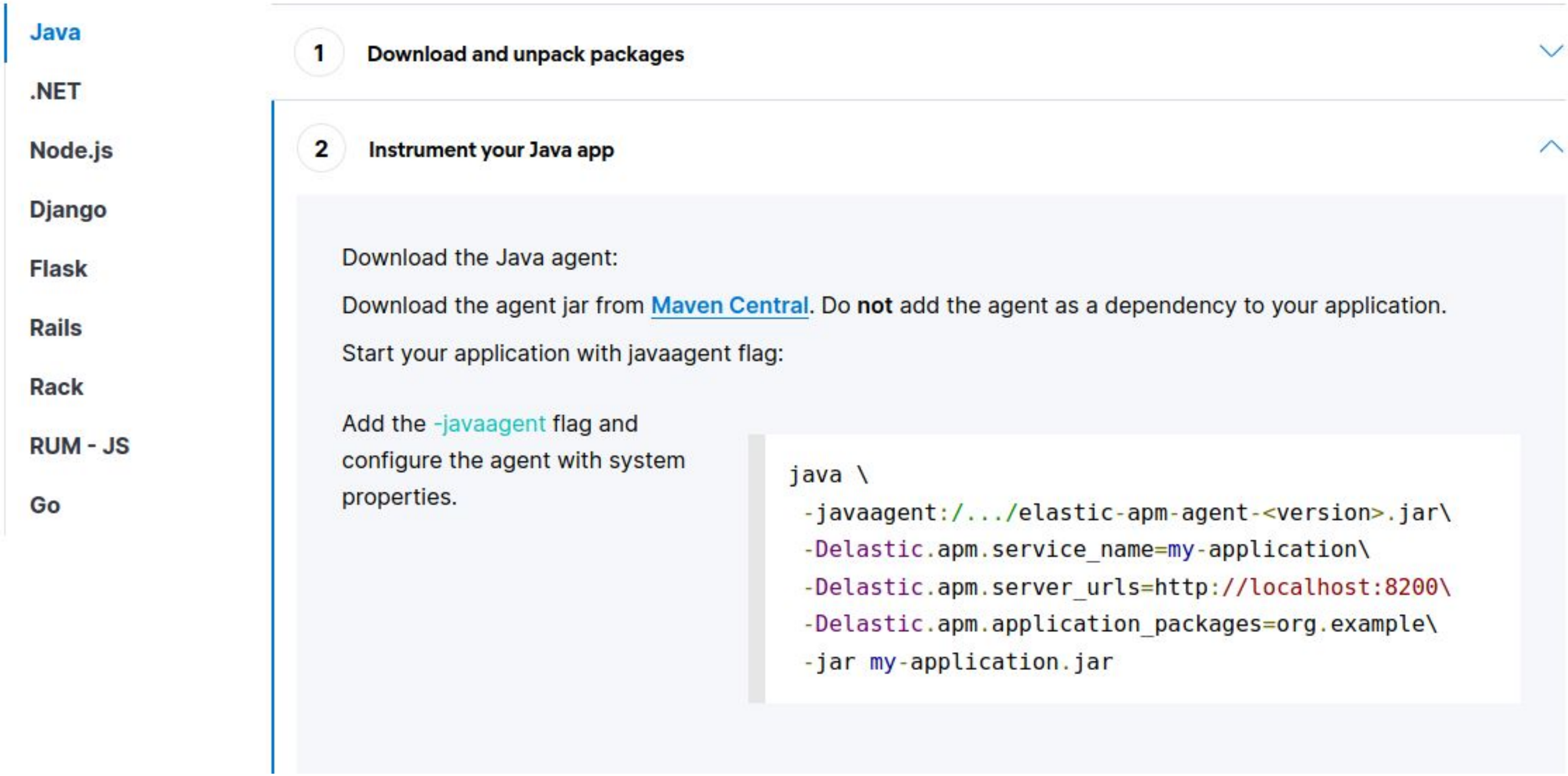
Collect Metrics

Store Metrics

Visualize

Instrumenting Apps using Elastic APM

https://www.elastic.co/apm
https://www.elastic.co/guide/en/apm/get-started/current/agents.html



1 Download and unpack packages

2 Instrument your Java app

Download the Java agent:

Download the agent jar from Maven Central. Do **not** add the agent as a dependency to your application.

Start your application with javaagent flag:

Add the -javaagent flag and configure the agent with system properties.

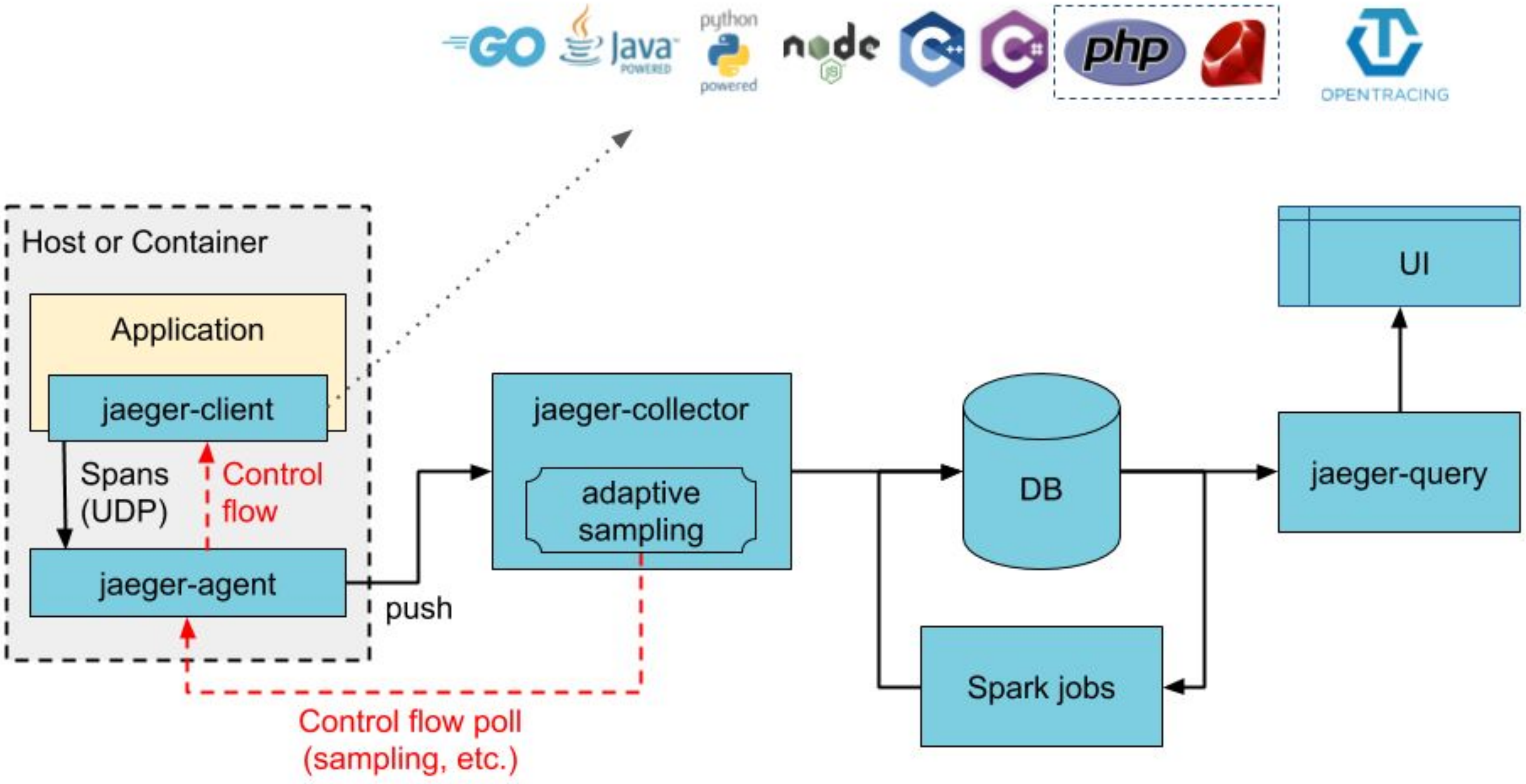
```
java \
 -javaagent:/.../elastic-apm-agent-<version>.jar\
 -Delastic.apm.service_name=my-application\
 -Delastic.apm.server_urls=http://localhost:8200\
 -Delastic.apm.application_packages=org.example\
 -jar my-application.jar
```

Jaeger **(+Opentracing)**
APM stack

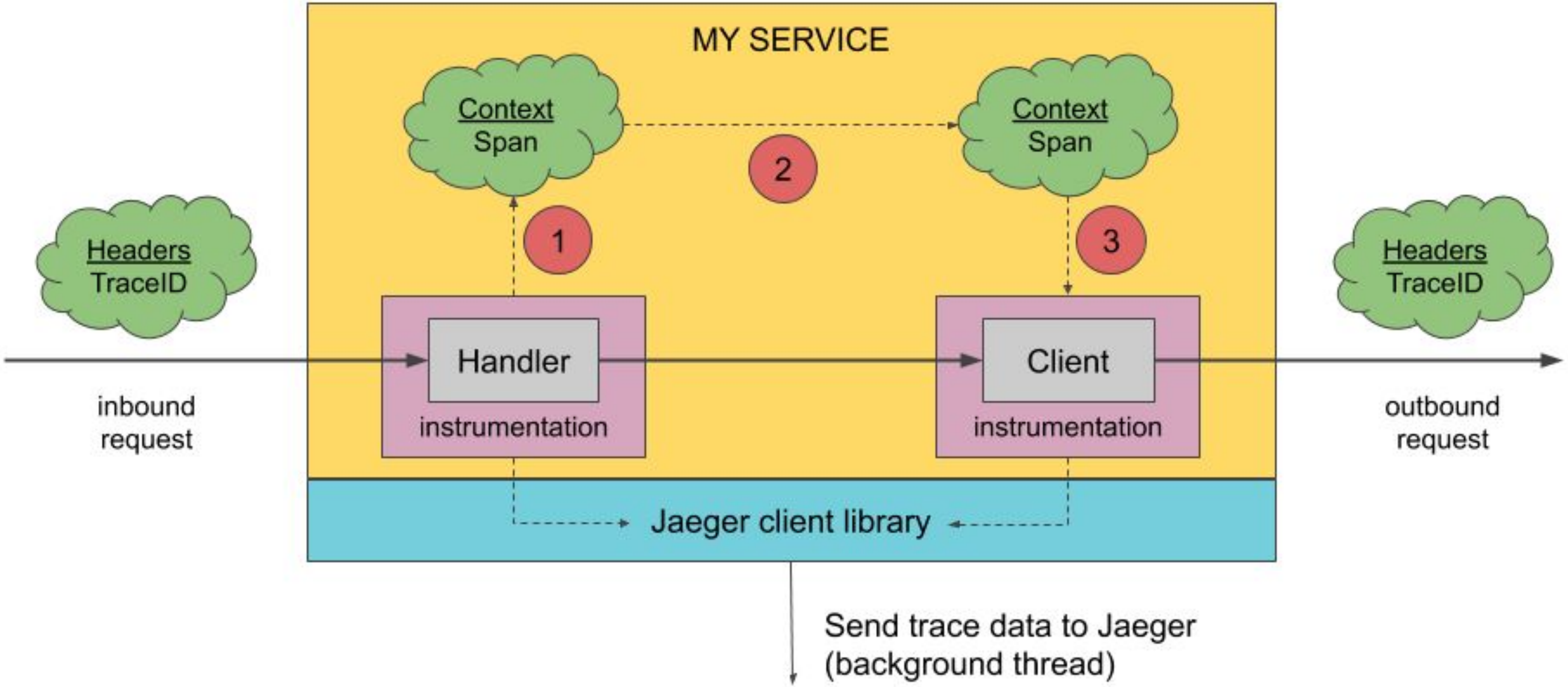
for App Perf. Monitoring System.

Jaeger-Opentracing Architecture

https://www.jaegertracing.io/docs/1.18/architecture/



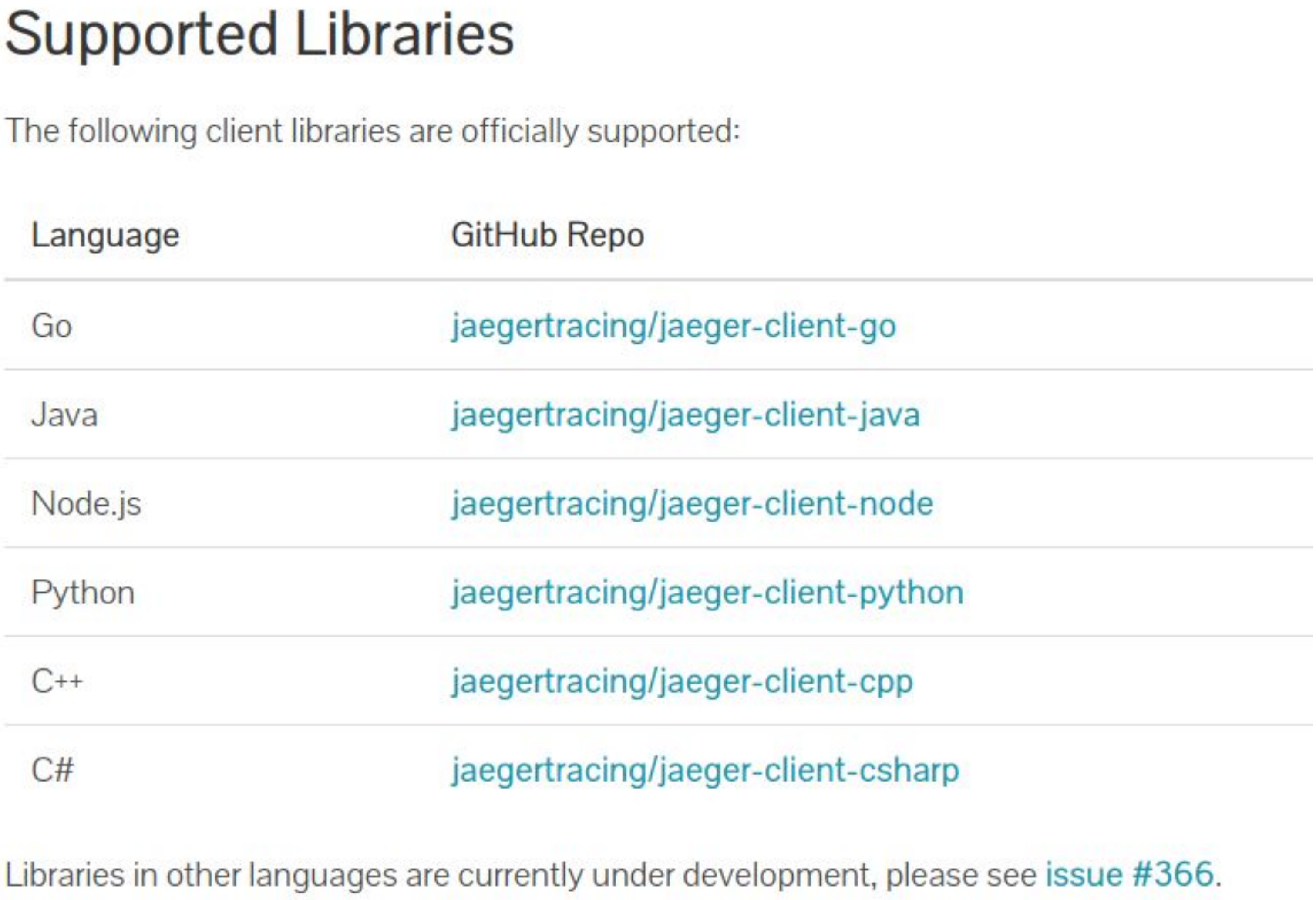
Instrumenting Apps using Jaeger-Opentracing



https://www.jaegertracing.io/docs/1.18/apis/

Instrumenting Apps using Jaeger-Opentracing

https://www.jaegertracing.io/docs/1.18/client-libraries/



Supported Libraries

The following client libraries are officially supported:

Language	GitHub Repo
Go	jaegertracing/jaeger-client-go
Java	jaegertracing/jaeger-client-java
Node.js	jaegertracing/jaeger-client-node
Python	jaegertracing/jaeger-client-python
C++	jaegertracing/jaeger-client-cpp
C#	jaegertracing/jaeger-client-csharp

Libraries in other languages are currently under development, please see issue #366.

Demo !

https://github.com/heronimus/guide-apm-monitoring

- **Other Useful Link**

https://www.elastic.co/apm

https://openapm.io/

https://www.jaegertracing.io/docs/1.18/getting-started/

https://zipkin.io/pages/architecture.html

https://github.com/yurishkuro/opentracing-tutorial

https://medium.com/opentracing/tracing-http-request-latency-in-go-with-opentracing-7cc1282a100a