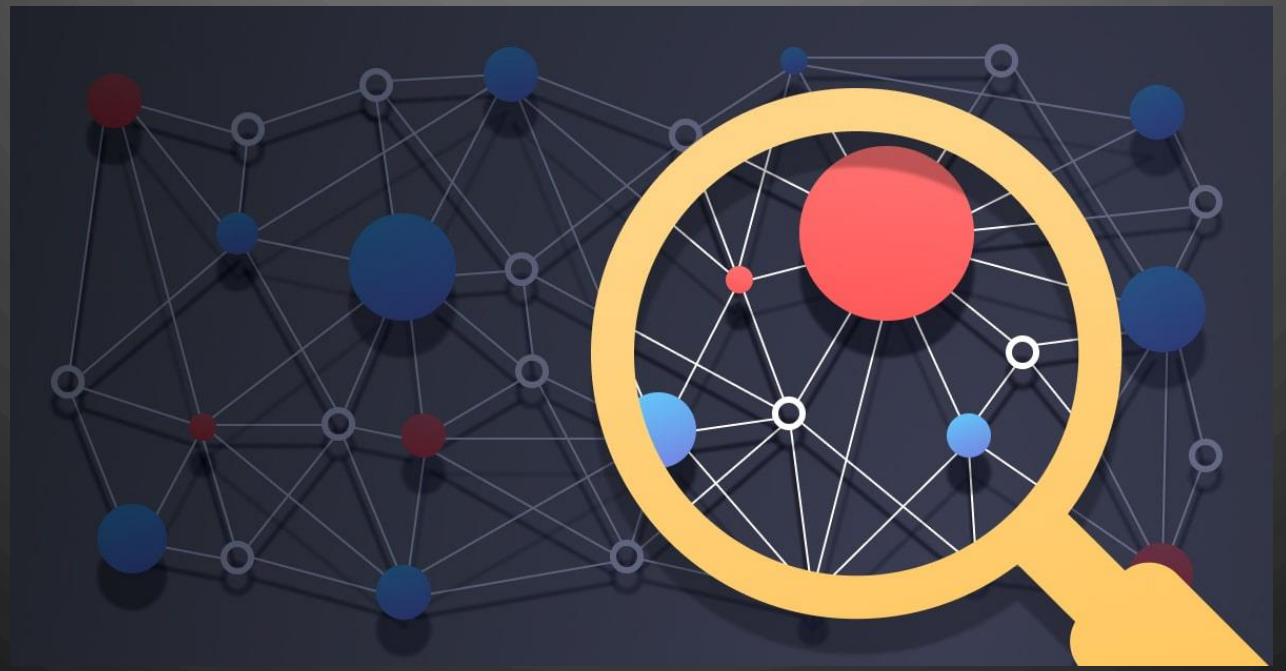
DISTRIBUTED TRACING



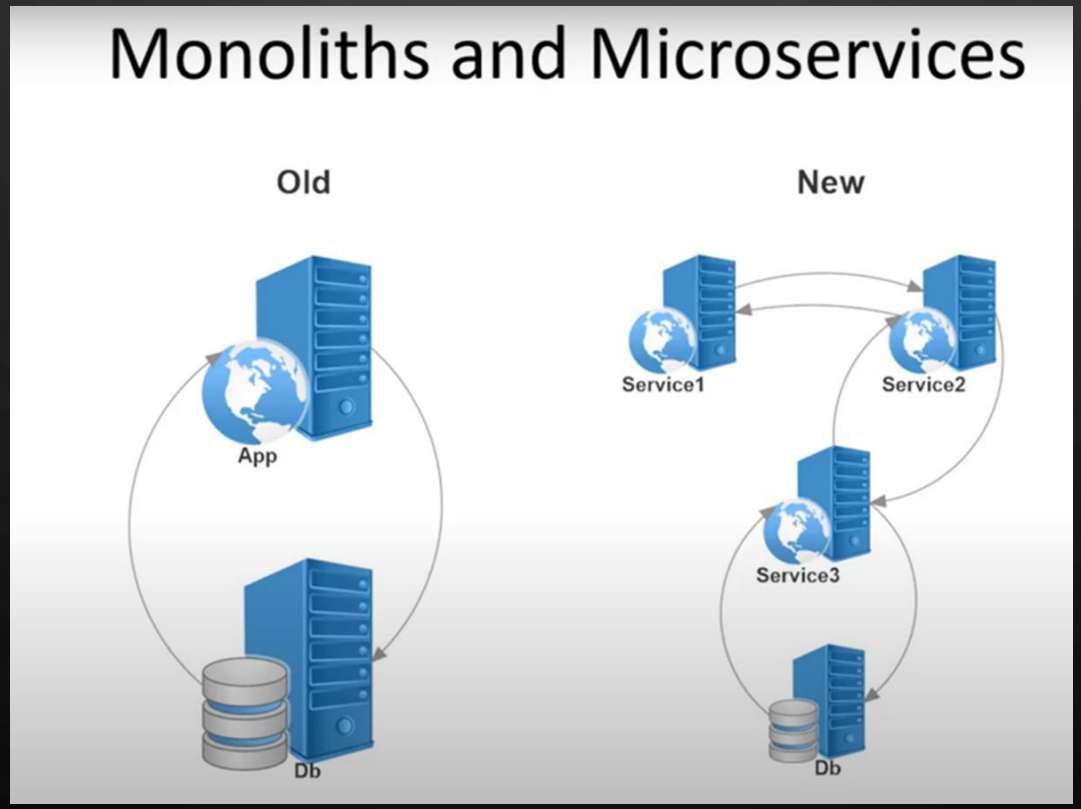
AGENDA

1. What is Distributed Tracing and how does it work?
2. Popular Distributed Tracing tools.
3. Zipkin and Sleuth – work logic.
4. OpenTelemetry.
5. Signoz.
6. Live demo.

What is Distributed Tracing?

Distributed Tracing is a convienient way to observe activity of an application based on microservices architecture. It tracks all requests going through services in the application.

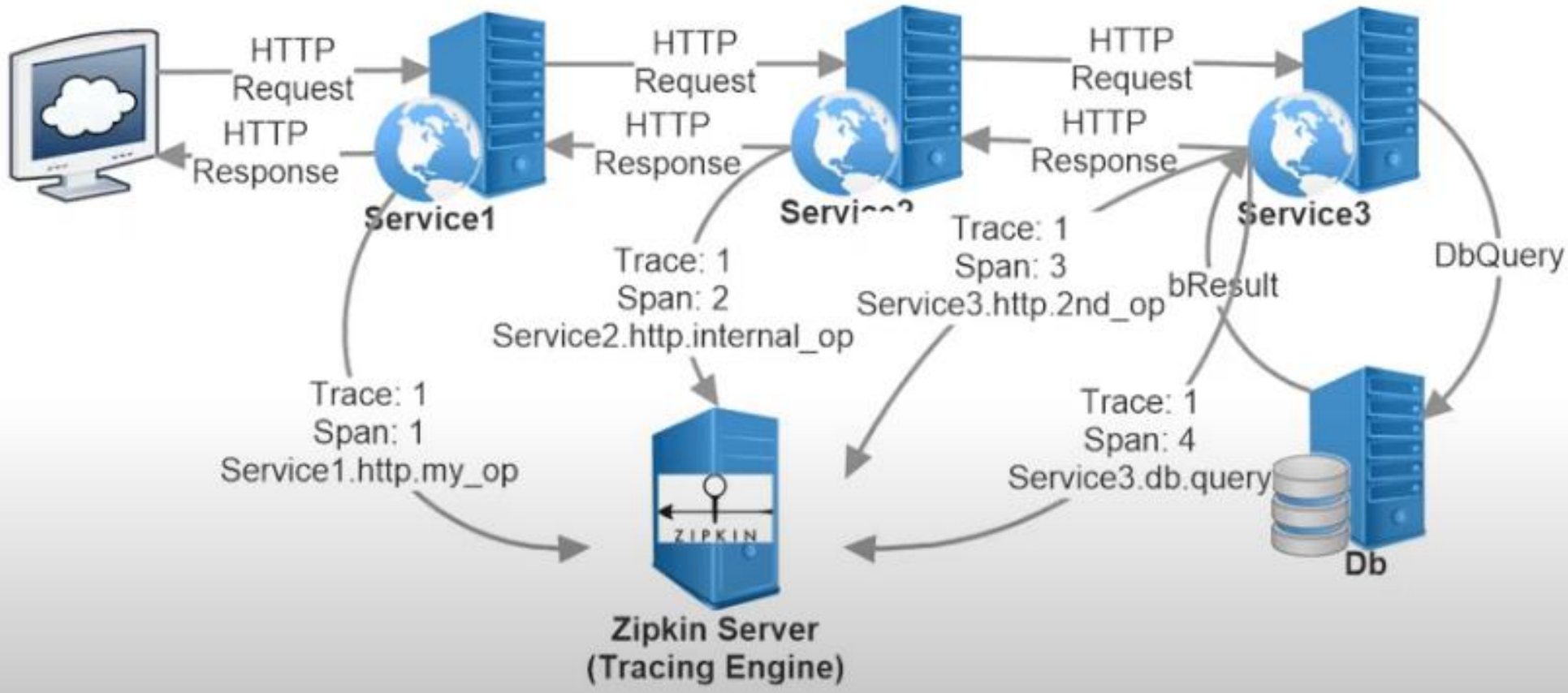
Distributed Tracing improves debug and diagnostic. It helps to better understand dependencies beetween services.

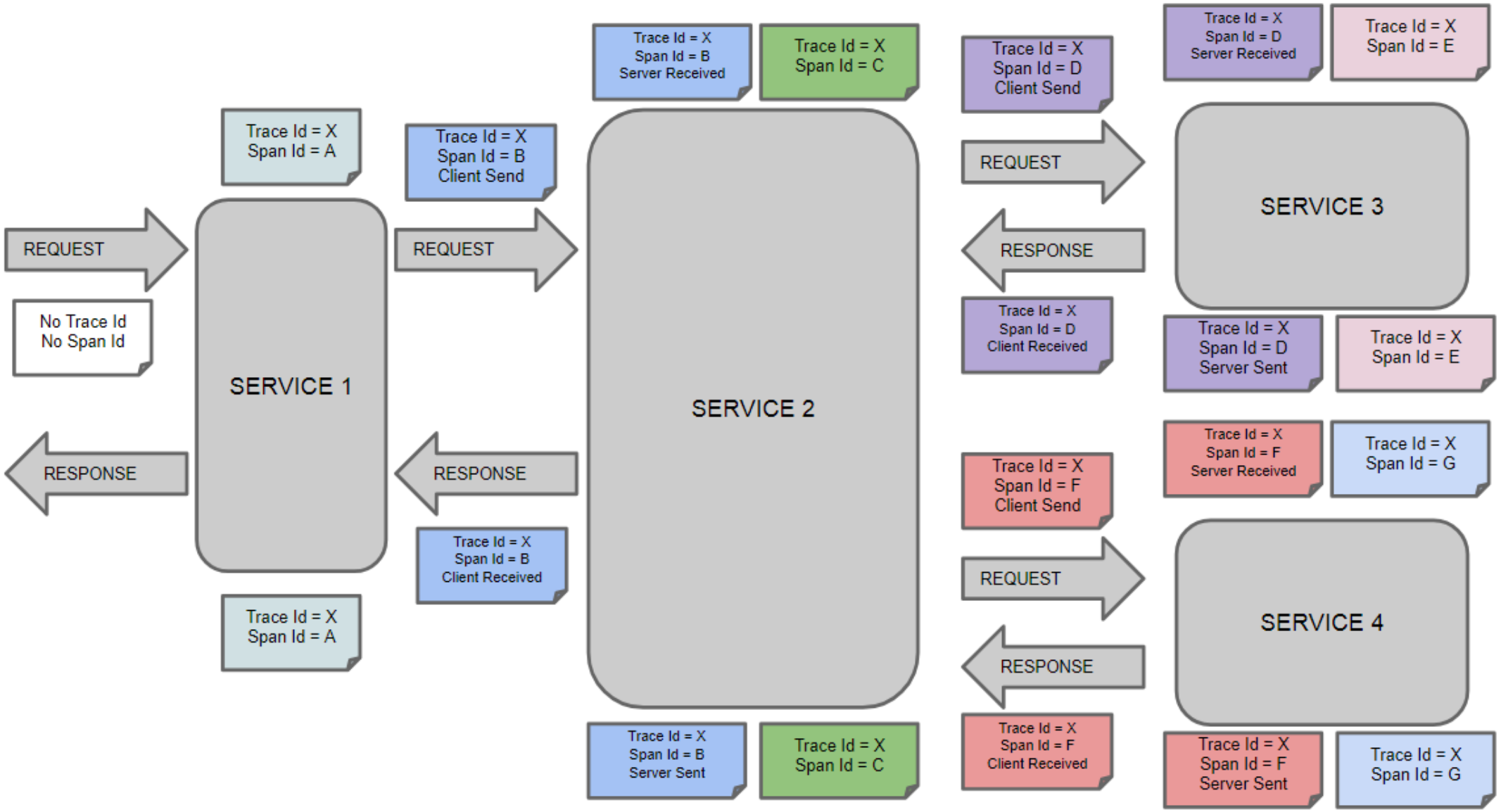


How does Distributed Tracing work?

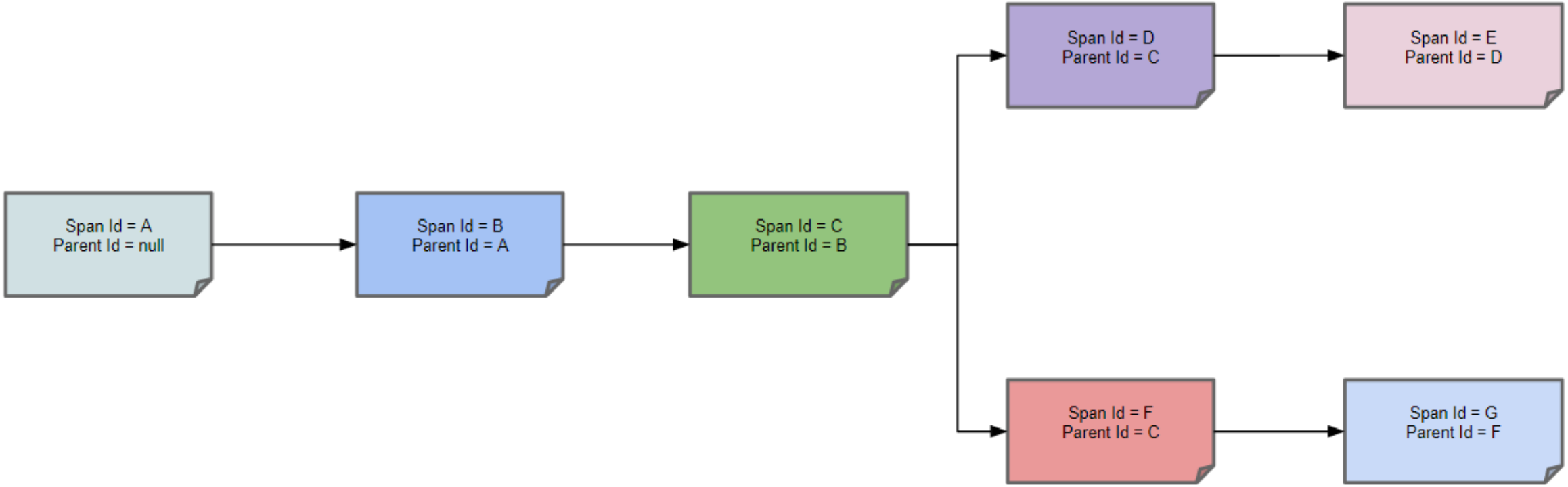
Distributed Tracing starts when a user make an interraction within the application, for example sends an HTTP request. The HTTP requests gets a unique *Trace ID*. When it goes through services, each operation (span), executed as a result of this request, gets a unique ID (*span ID*) and parent ID, so the ID of an operation which caused current operation.

Collecting Distributed Traces




REQUEST
No Trace Id
No Span Id
SERVICE 1
RESPONSE
Trace Id = X
Span Id = A
Trace Id = X
Span Id = B
Client Send
REQUEST
RESPONSE
Trace Id = X
Span Id = B
Client Received
Trace Id = X
Span Id = A
Trace Id = X
Span Id = B
Server Received
Trace Id = X
Span Id = C
SERVICE 2
Trace Id = X
Span Id = B
Server Sent
Trace Id = X
Span Id = C
Trace Id = X
Span Id = D
Client Send
REQUEST
RESPONSE
Trace Id = X
Span Id = D
Client Received
Trace Id = X
Span Id = D
Server Received
Trace Id = X
Span Id = E
SERVICE 3
Trace Id = X
Span Id = D
Server Sent
Trace Id = X
Span Id = E
Trace Id = X
Span Id = F
Client Send
REQUEST
RESPONSE
Trace Id = X
Span Id = F
Client Received
Trace Id = X
Span Id = F
Server Received
Trace Id = X
Span Id = G
SERVICE 4
Trace Id = X
Span Id = F
Server Sent
Trace Id = X
Span Id = G

TRACE


Span Id = A
Parent Id = null
Span Id = B
Parent Id = A
Span Id = C
Parent Id = B
Span Id = D
Parent Id = C
Span Id = E
Parent Id = D
Span Id = F
Parent Id = C
Span Id = G
Parent Id = F

POPULAR DISTRIBUTED TRACING TOOLS

Implemented in Java.
Created by Twitter.
Open-source.

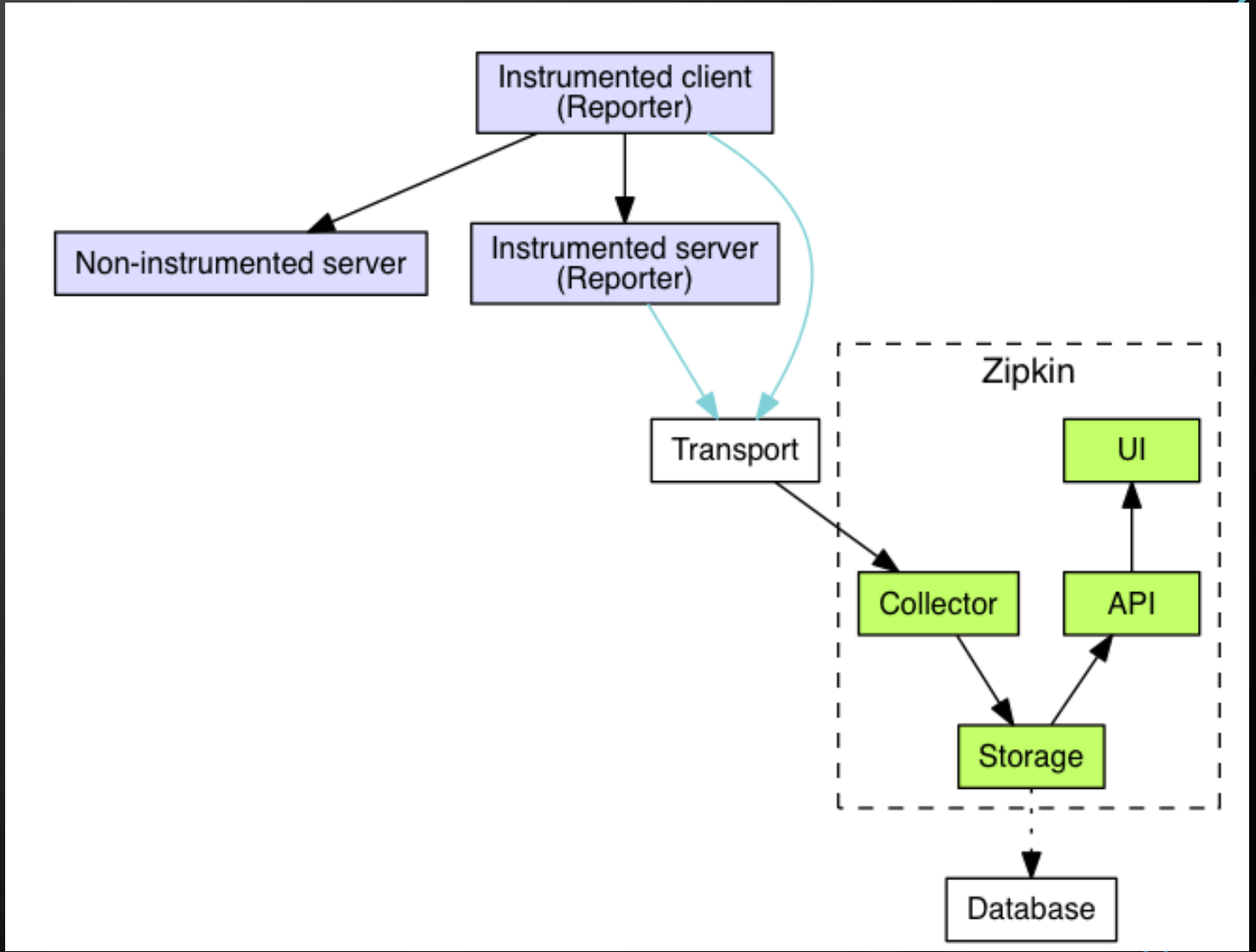
Implemented in Go.
Created by Uber.
Open-source.

HOW DOES ZIPKIN WORK?

Zipkin consists of four components:

- Collector,
- Storage,
- Search,
- Web UI.

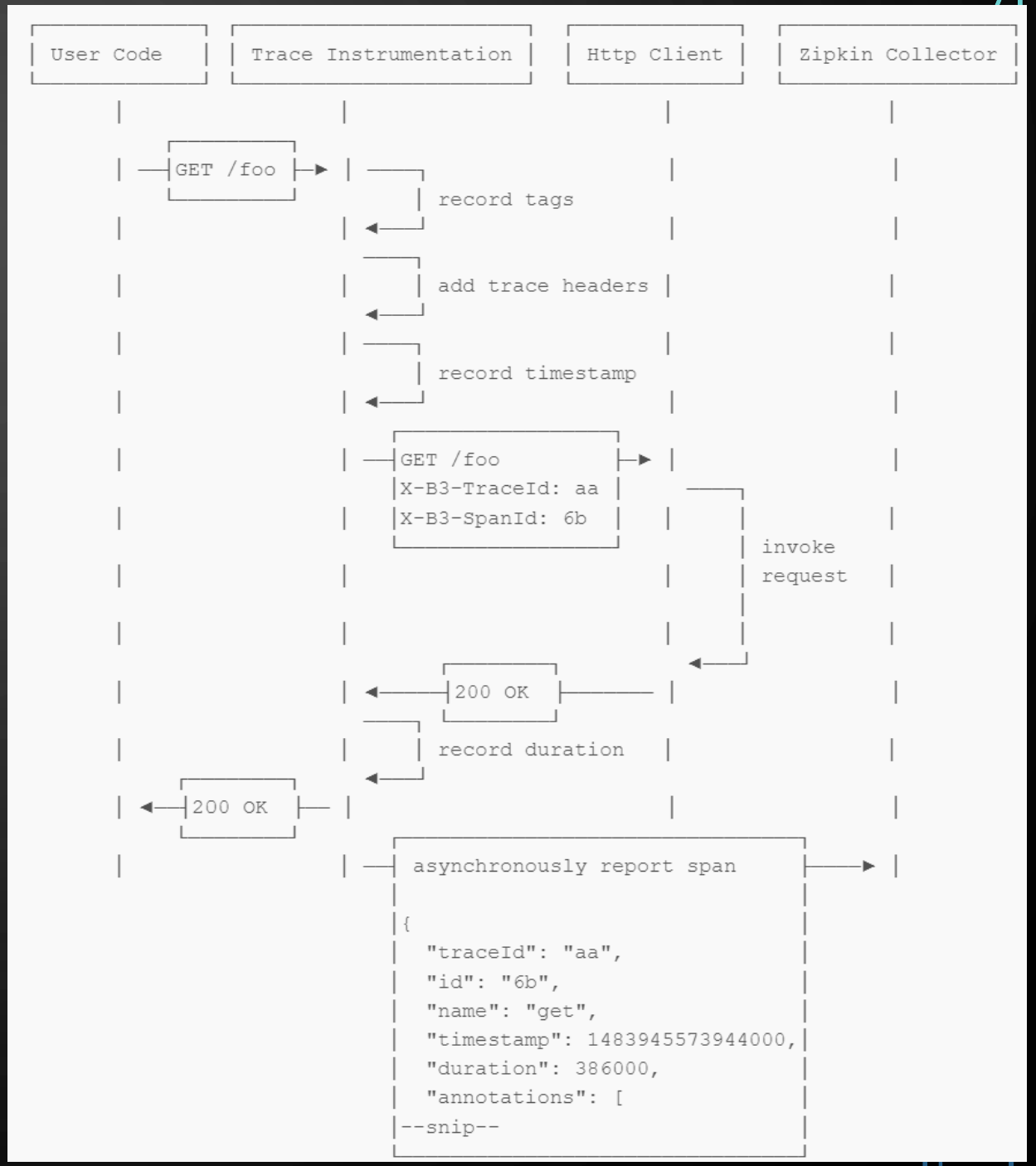
Collector validates input data, then sends it to Storage. The user can access the data in any moment through Search component and Web UI.



ZIPKIN LOGIC PART 2 - SLEUTH

As we mentioned before Zipkin Collector receive data, so it has to get that data from something else and that data has to be in specific format.

This functionality is provided by Sleuth. It adds neccessary headers like Trace ID, duration of the operation and etc. After all, it sends that data to Zipkin Collector.



~~OPEN TRACING~~ -------> OPEN TELEMETRY

OpenTelemetry is a collection of tools, APIs, and SDKs. Use it to instrument, generate, collect, and export telemetry data (metrics, logs, and traces) to help you analyze your software's performance and behavior.

OpenTelemetry was created by combining OpenCensus and OpenTracing.

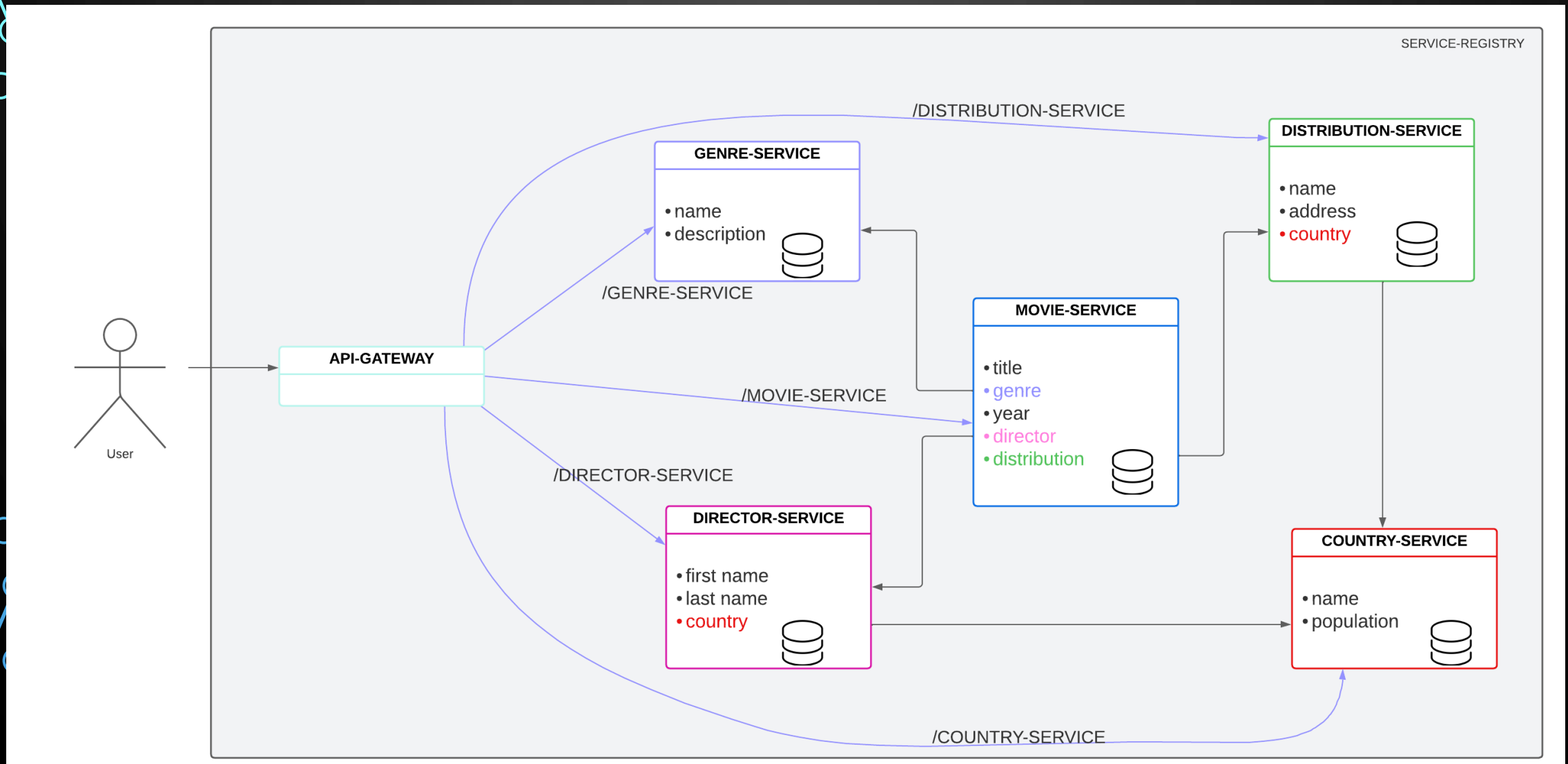
OpenCensus was a set of metric and distributed tracing tools.

OpenTracing - Vendor-neutral APIs and instrumentation for distributed tracing.

SIGNOZ

Signoz is an open-source tool providing nice looking WEB UI for visualizing data collected by OpenTelemetry.

DISTRIBUTED TRACING IN ACTION WITHIN A SIMPLE MOVIE APP



USEFUL LINKS AND SOURCES

- https://www.splunk.com/en_us/data-insider/what-is-distributed-tracing.html
- https://www.dynatrace.com/news/blog/what-is-distributed-tracing/
- https://www.techtarget.com/searchapparchitecture/tip/3-distributed-tracing-tools-perfect-for-microservices
- https://zipkin.io/pages/architecture.html
- https://newrelic.com/resources/ebooks/quick-introduction-distributed-tracing
- https://howtodoinjava.com/spring-cloud/spring-cloud-zipkin-sleuth-tutorial/
- https://opentracing.io/docs/overview
- https://eng.uber.com/distributed-tracing
- https://newrelic.com/resources/ebooks/quick-introduction-distributed-tracing
- https://docs.spring.io/spring-cloud-sleuth/docs/current/reference/html/getting-started.html#getting-started
- https://cloud.spring.io/spring-cloud-sleuth/reference/html
- https://www.baeldung.com/spring-cloud-sleuth-get-trace-id
- https://ryanharrison.co.uk/2021/08/06/distributed-tracing-spring-boot-jaeger.html
- Other not mentioned sources come from each tool's official website.

THANK YOU