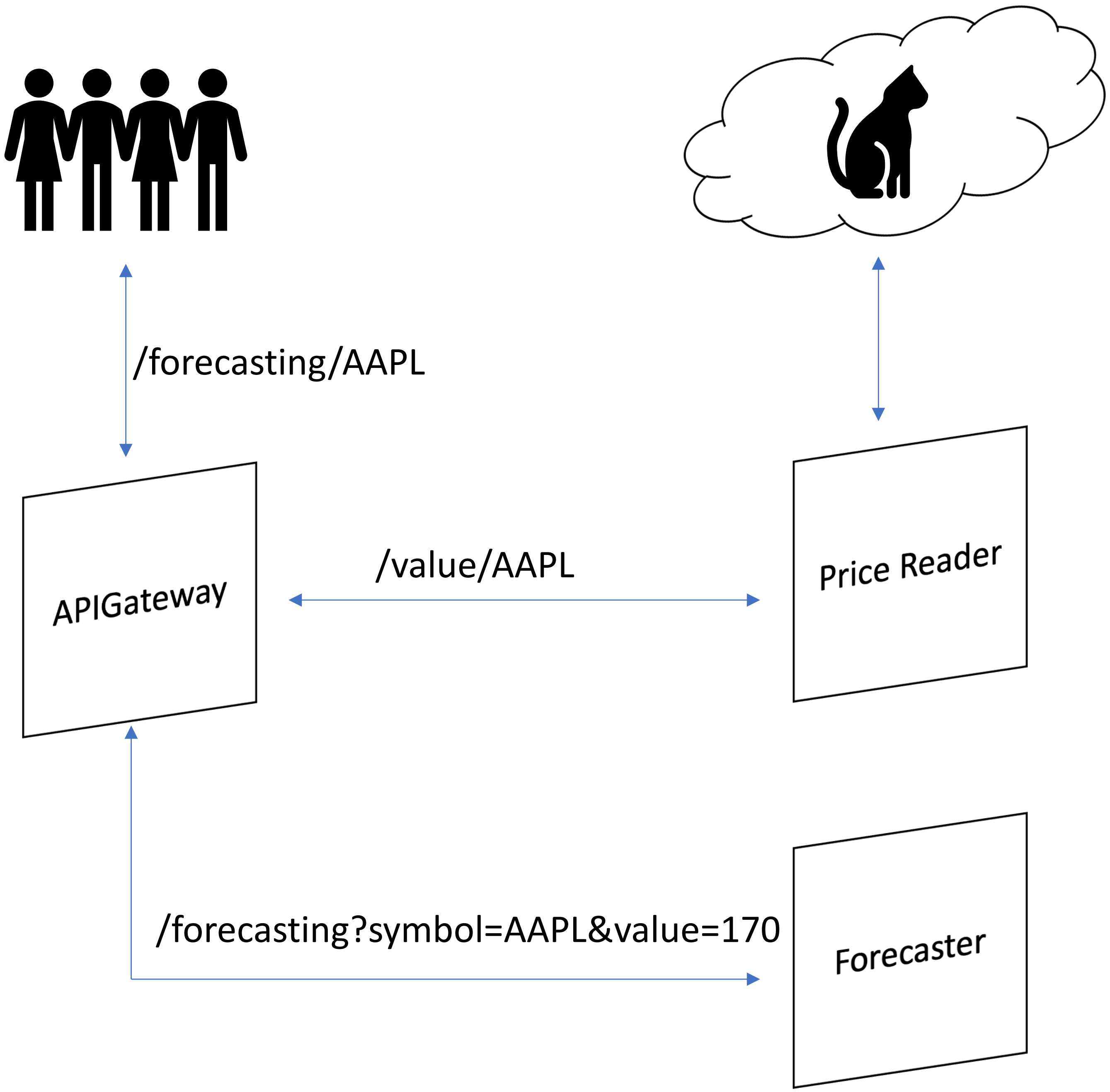
Logging is not (distributed) Tracing

Questions (to the audience) free!



/forecasting/AAPL

APIGateway

/value/AAPL

Price Reader

/forecasting?symbol=AAPL&value=170

Forecaster

Logging is

Storing execution information for later analysis

Logging…

```
printf( “Error\n” );
std::cout << “Error\n”;
```

What?

Define your goal and audience

Decide what to log

Silence → Noise

What?

Context

Severity levels and Categories

Not only for troubleshooting

Not only for humans

Where

Log to a file (and rotate logs) and, maybe, centralized (Graylog, ELK Stack…)

How?

Use a logging library

spdlog::error(“Error");

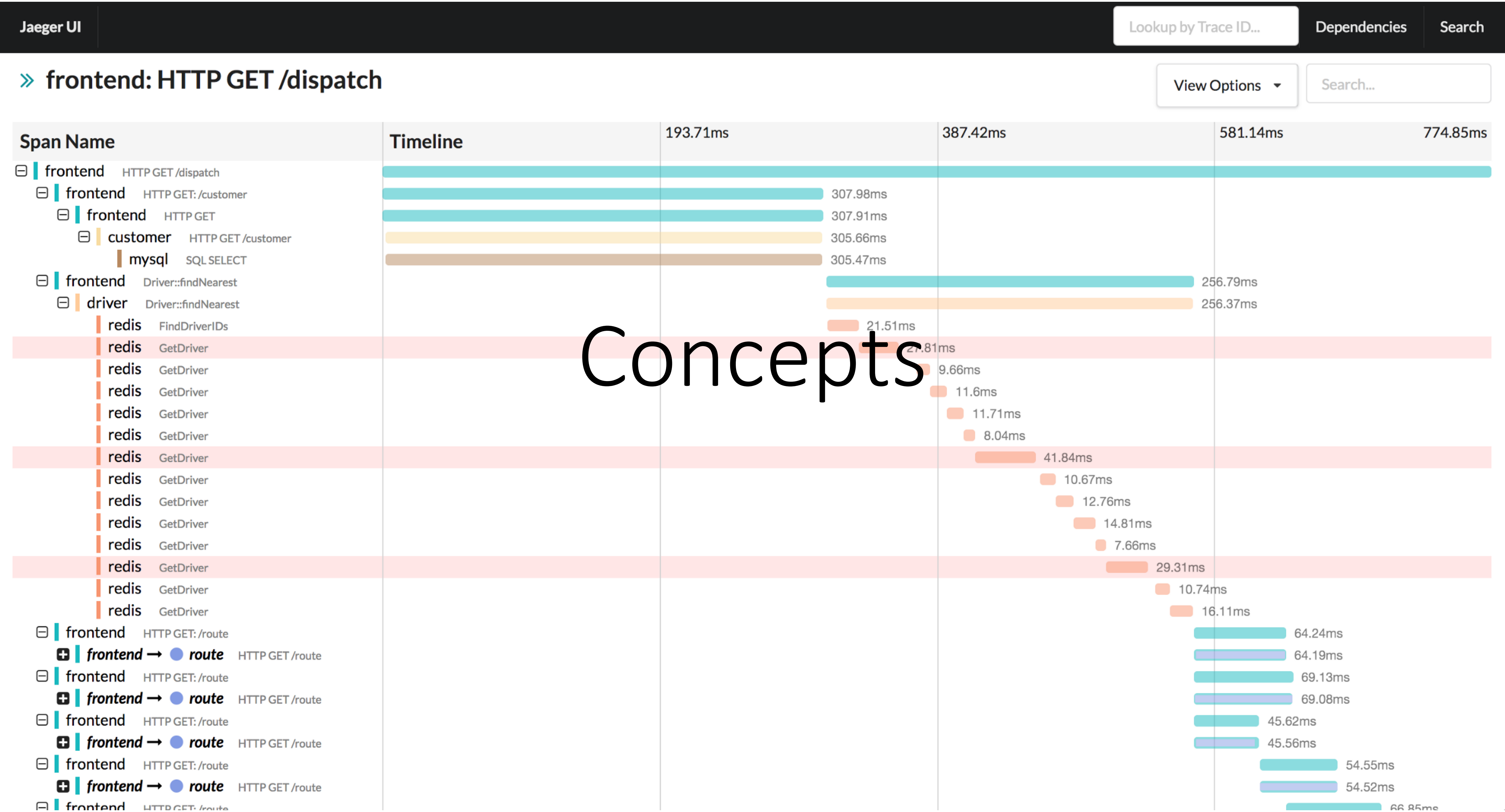
spdlog

https://github.com/gabime/spdlog/wiki/1.-QuickStart

Distributed Tracing is

a method used to profile and monitor applications, especially those built using a microservices architecture

Concepts



Jaeger UI | Lookup by Trace ID... | Dependencies | Search

» frontend: HTTP GET /dispatch

View Options | Search...

Span Name		Timeline
frontend	HTTP GET /dispatch	
frontend	HTTP GET: /customer	307.98ms
frontend	HTTP GET	307.91ms
customer	HTTP GET /customer	305.66ms
mysql	SQL SELECT	305.47ms
frontend	Driver::findNearest	256.79ms
driver	Driver::findNearest	256.37ms
redis	FindDriverIDs	21.51ms
redis	GetDriver	27.81ms
redis	GetDriver	9.66ms
redis	GetDriver	11.6ms
redis	GetDriver	11.71ms
redis	GetDriver	8.04ms
redis	GetDriver	41.84ms
redis	GetDriver	10.67ms
redis	GetDriver	12.76ms
redis	GetDriver	14.81ms
redis	GetDriver	7.66ms
redis	GetDriver	29.31ms
redis	GetDriver	10.74ms
redis	GetDriver	16.11ms
frontend	HTTP GET: /route	64.24ms
frontend → route	HTTP GET /route	64.19ms
frontend	HTTP GET: /route	69.13ms
frontend → route	HTTP GET /route	69.08ms
frontend	HTTP GET: /route	45.62ms
frontend → route	HTTP GET /route	45.56ms
frontend	HTTP GET: /route	54.55ms
frontend → route	HTTP GET /route	54.52ms
frontend	HTTP GET: /route	66.85ms

Tracing

Tracing tracks the progression of a single user request as it is handled by other services that make up an application.

Trace

A directed acyclic graph (DAG) of Spans

Span

A span represents a single operation in a trace

a Span has

Name, start and finish timestamps

Tags: [{ key, value }]

Logs: [{ key, value, timestamp }]

and a SpanContext

Information needed to refer to a distinct Span across a process boundary and zero or more Baggage Items as [{ key, value }] that cross process boundaries

and References

ChildOf: a Span is the child of a parent Span

FollowsFrom: the new Span is independent of the parent Span

Tigres, Leones

OpenTracing (CNCF) vs OpenCensus (Google)

Jaeger

An OpenTracing Tracer (and an OpenCensus Exporter)

Why?

C++ client

Demo

https://github.com/david-antiteum/logging

The End