



OpenTelemetry from a developer perspective

Enrico Zimuel, Principal Software Engineer

May 16, 2023 - [CloudConf](#) Torino (Italy)

Observability and OpenTelemetry

- **Observability** lets us understand a system from the outside
- [OpenTelemetry](#) (OTel) is an open standard to enable observability
- With OTel an application can emit:
 - signals: **traces, metrics, and logs**

Manual instrumentation

- Changing the source code! 😡
- Adding OTel code for span, metrics and logs
 - **Span:** adding a start/end decorator
 - **Metrics:** invoke a function to set a value
 - **Logging:** usually handled by existing libraries

Automatic instrumentation

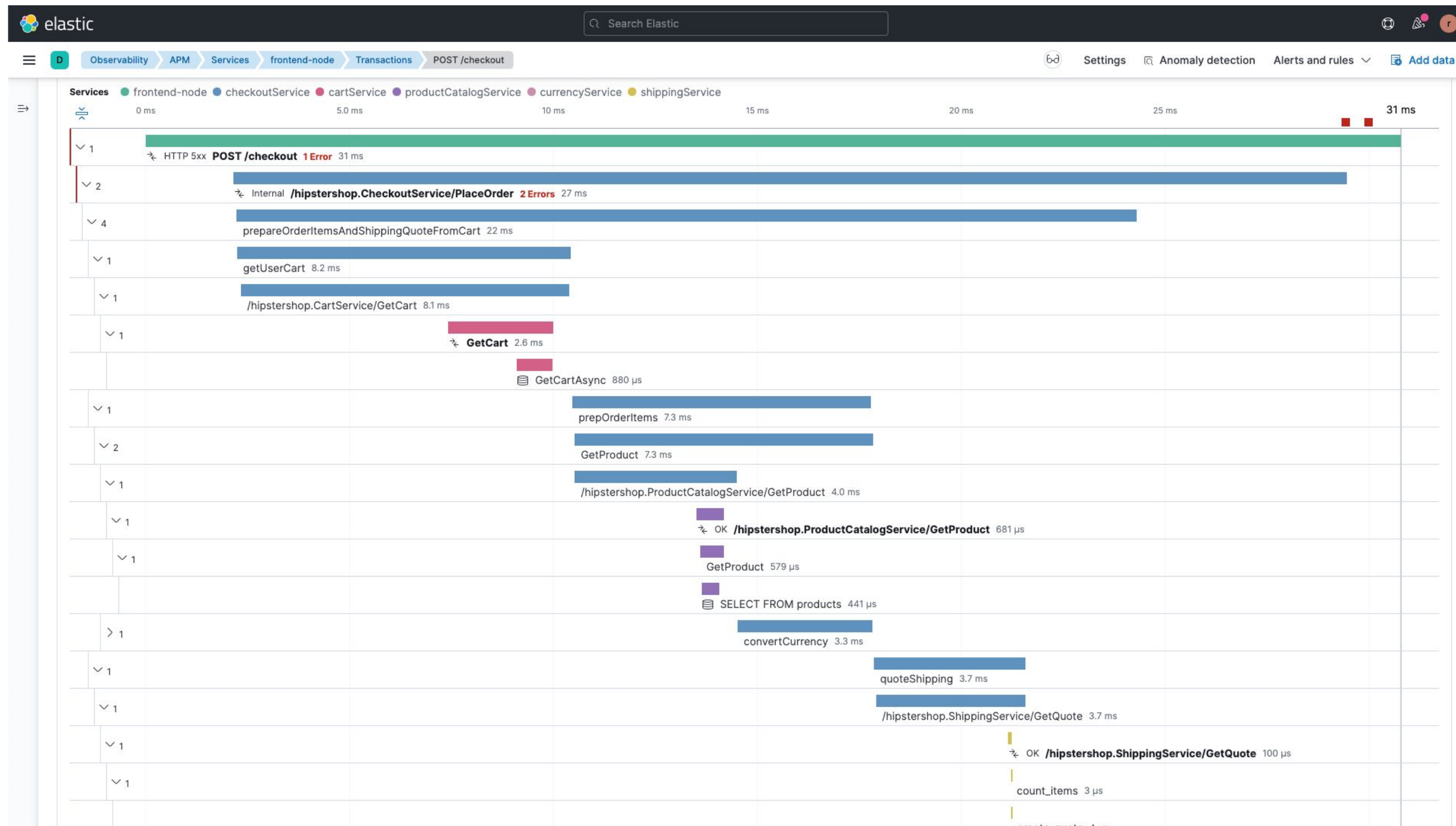
- No source code change 🙇
- Using OTel agent if available or commercial tools (eg. [Elastic APM agent](#))
- Typically configure the agent to intercept a function execution (eg. using annotation)

Example: Java annotation

```
import io.opentelemetry.instrumentation.annotations.WithSpan;

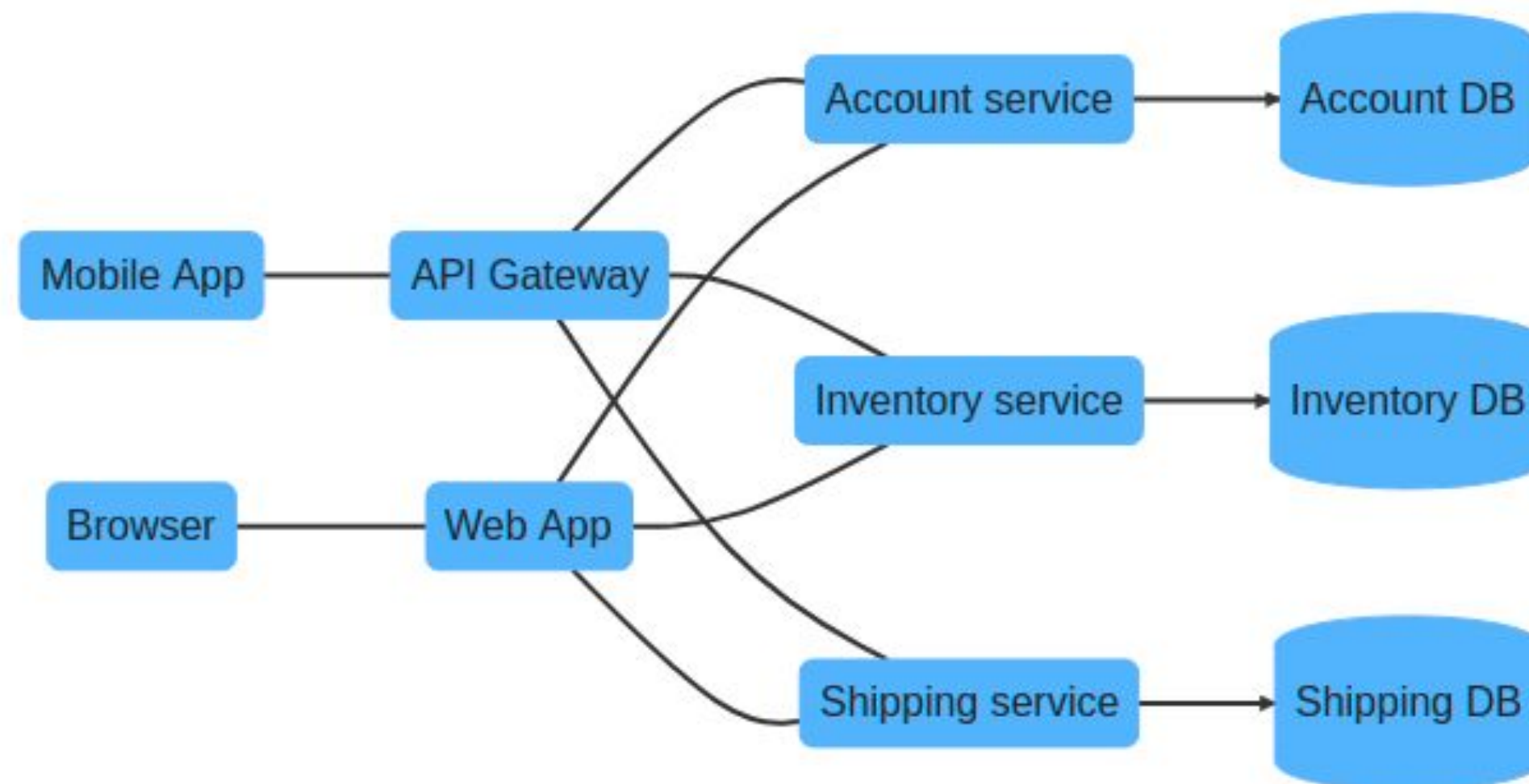
public class MyClass {
    @WithSpan
    public void myMethod() {
        <...>
    }
}
```

Waterfall diagrams



Distributed traces

- Track a call on multiple services (using [W3C Trace Context](#))

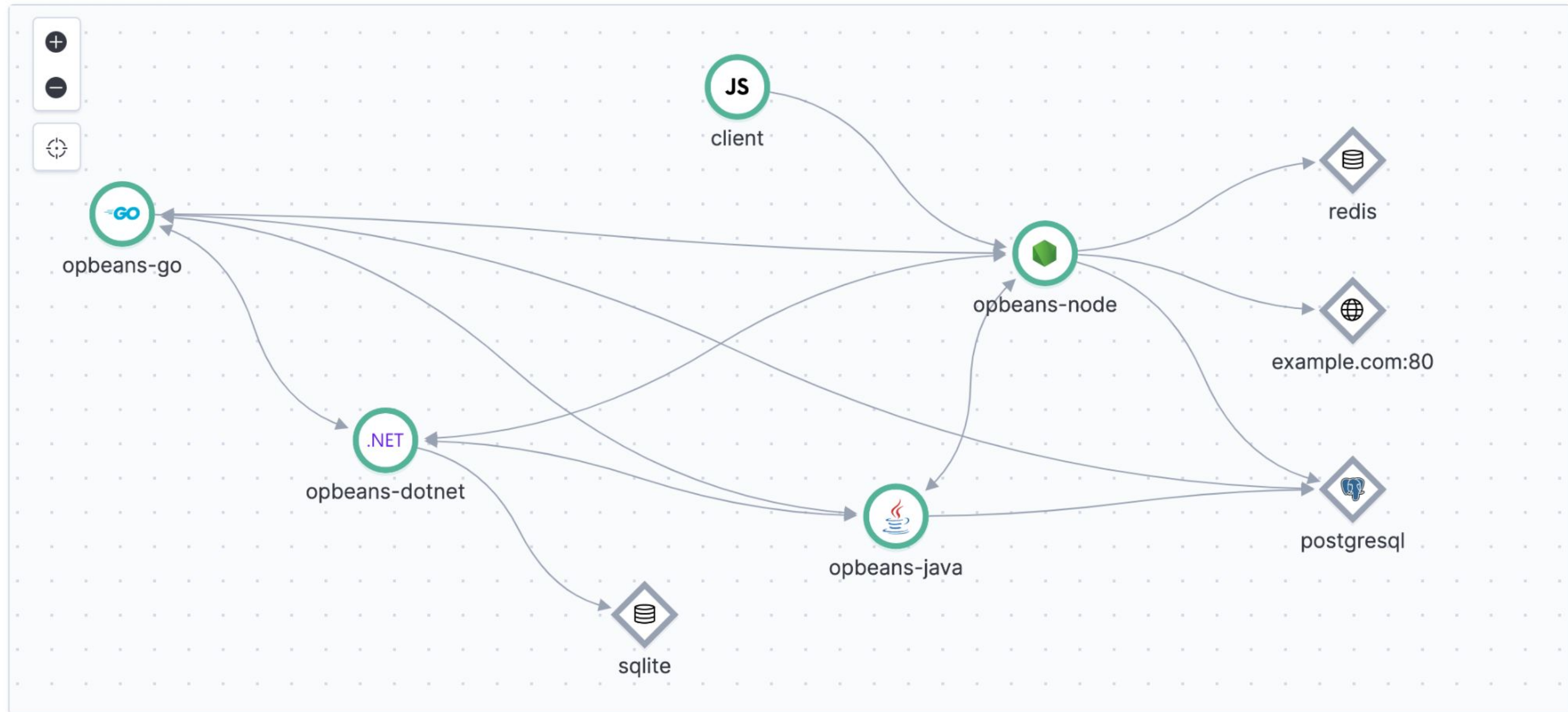


OTel supported languages

Languages	Traces	Metrics	Logs	Automatic
C++	Stable	Stable	Experimental	
.NET	Stable	Stable	Mixed	YES
Go	Stable	Beta		
Java	Stable	Stable	Experimental	YES
Javascript/TypeScript	Stable	Stable	Development	
PHP	Beta	Beta	Alpha	YES
Python	Stable	Stable	Experimental	YES
Ruby	Stable			YES
Rust	Beta	Alpha		
Swift	Stable	Experimental		

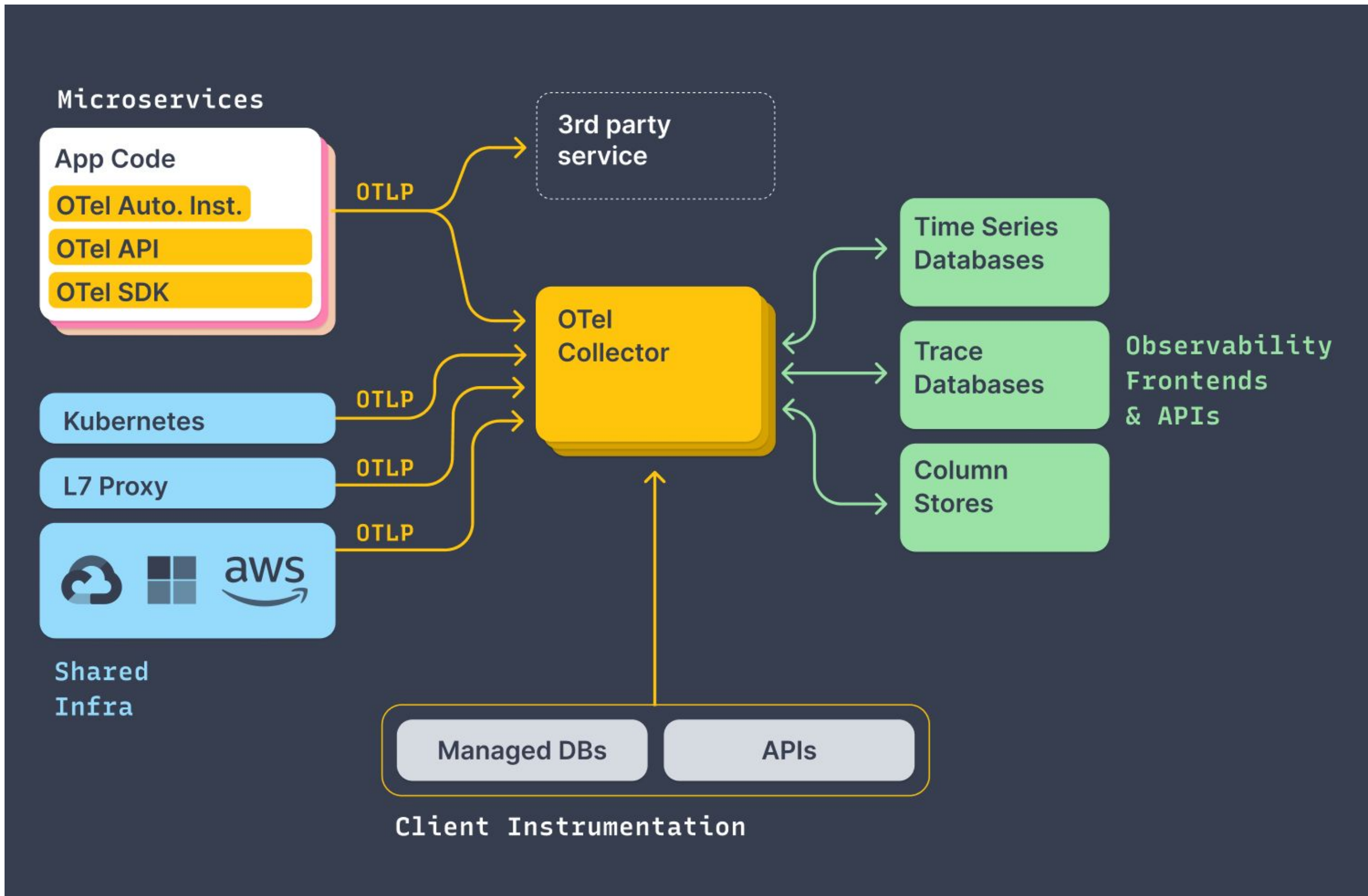
More information: <https://opentelemetry.io/docs/instrumentation/>

Elastic Service Map



Observability backend

- The OTel SDK or OTel agent can emit:
 - directly to the Observability backend
 - use [OTel Collector](#) to aggregate and scale



Source: <https://opentelemetry.io/docs/>

Thanks!

More information about [OpenTelemetry](#)
and the [Elastic initiative](#) about OTel

Contacts: enrico.zimuel@elastic.co