

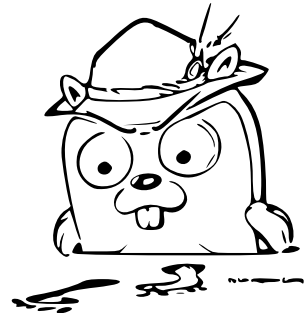
Tracing Microservices with OpenTracing



OPENTRACING



ZIPKIN



Jaeger

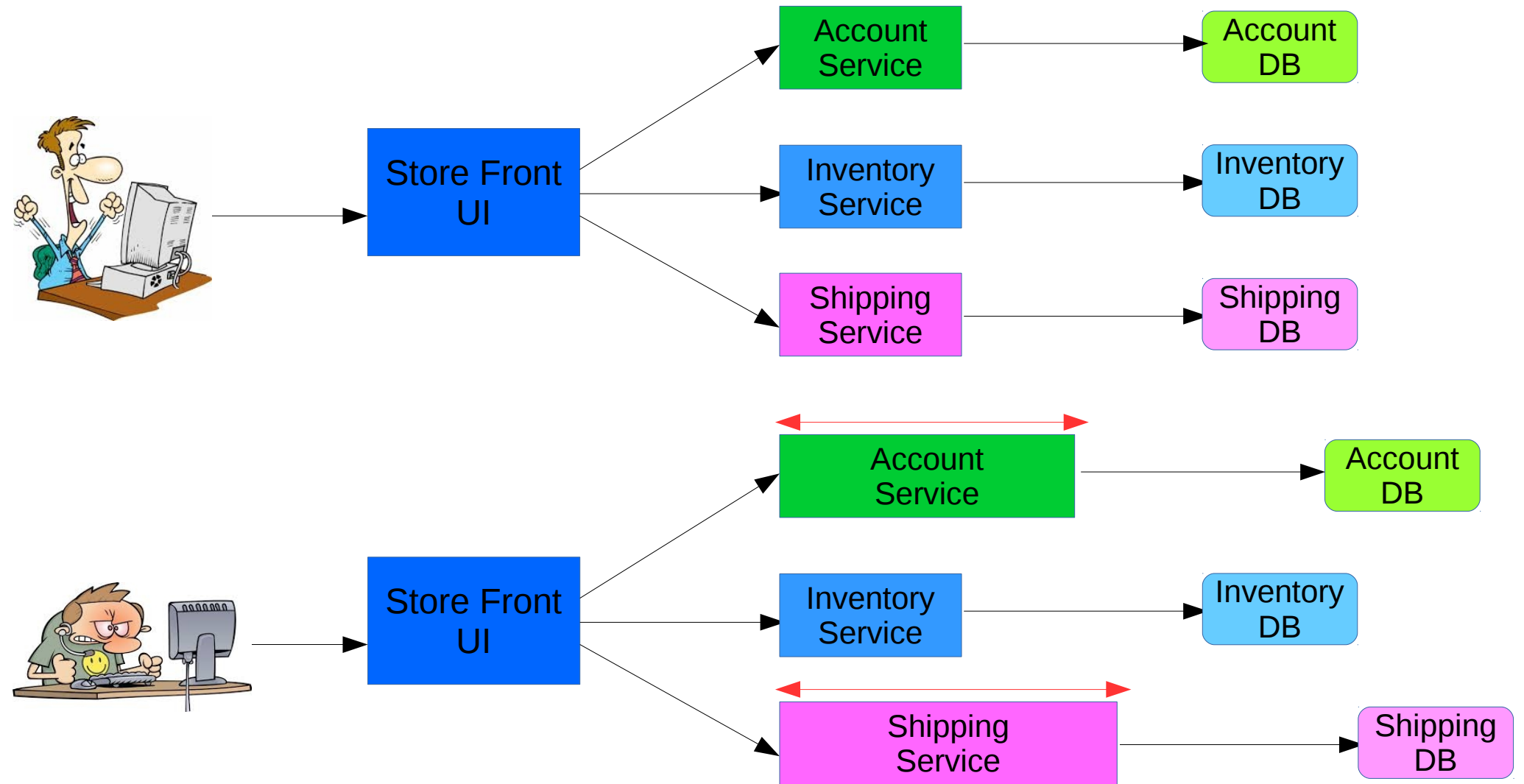




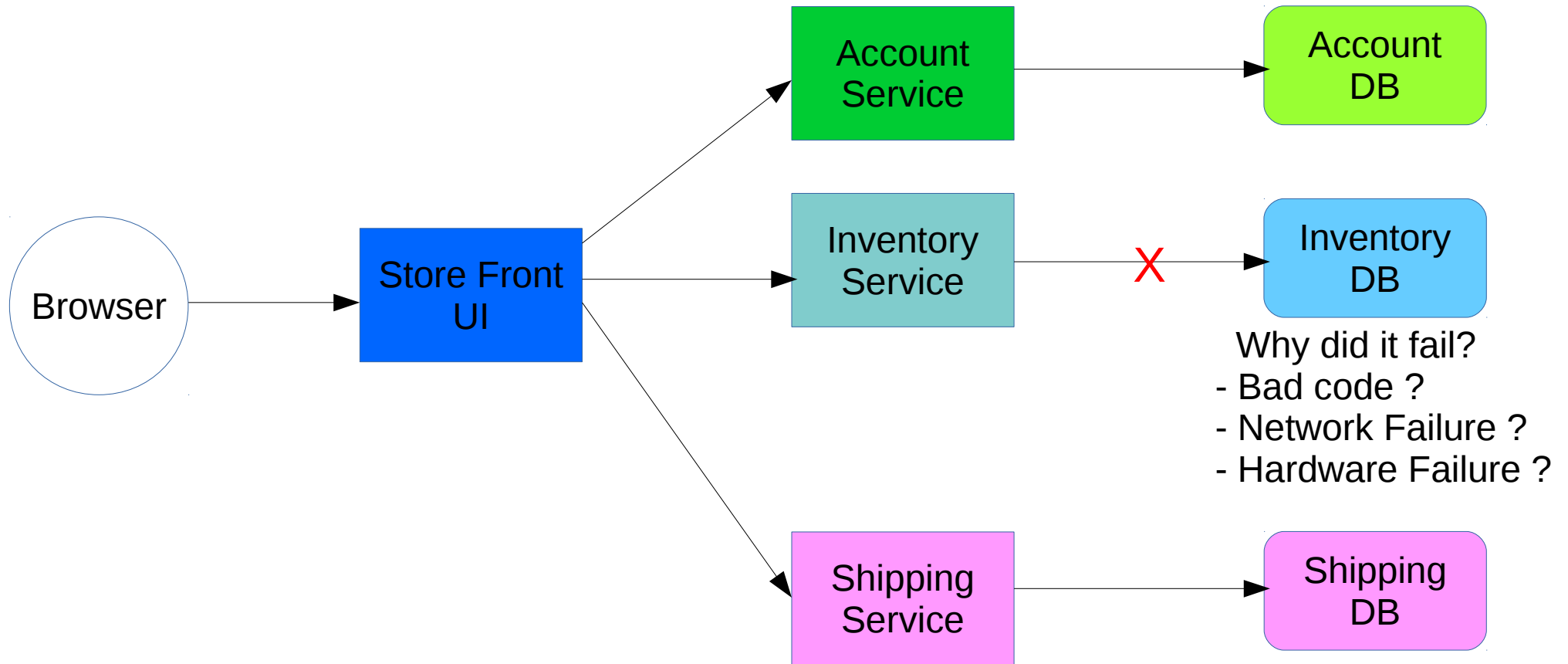
Why Tracing?



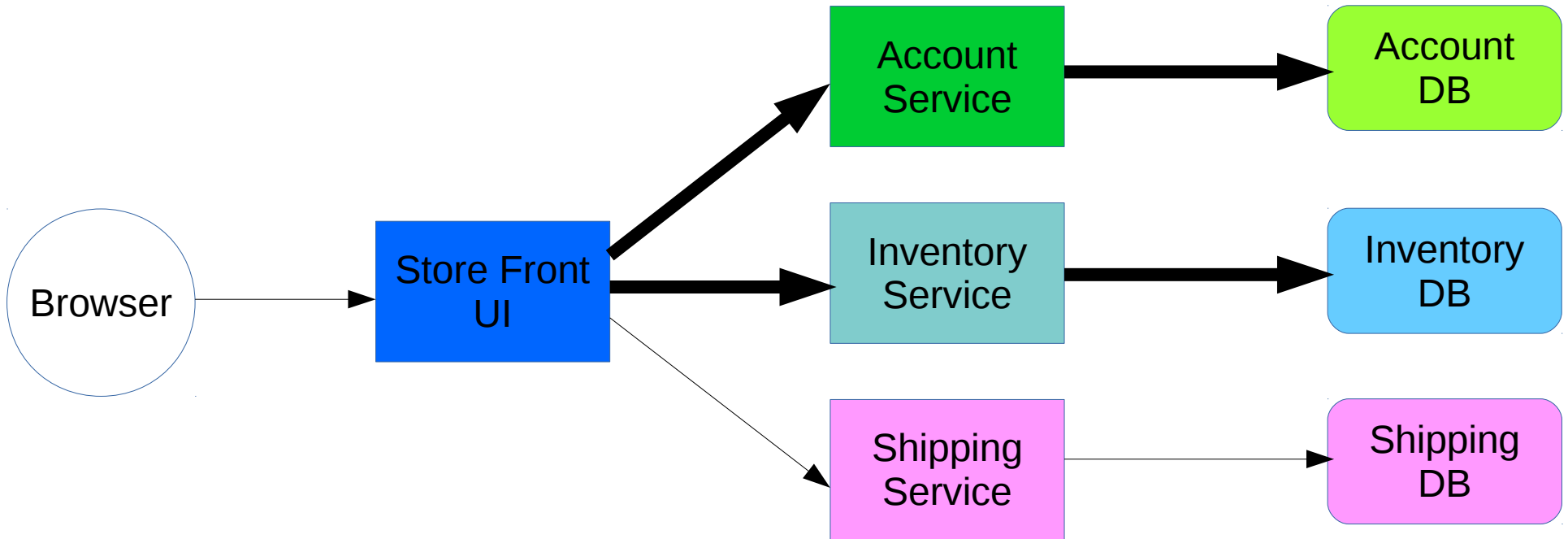
Track Latency



Root Cause Analysis

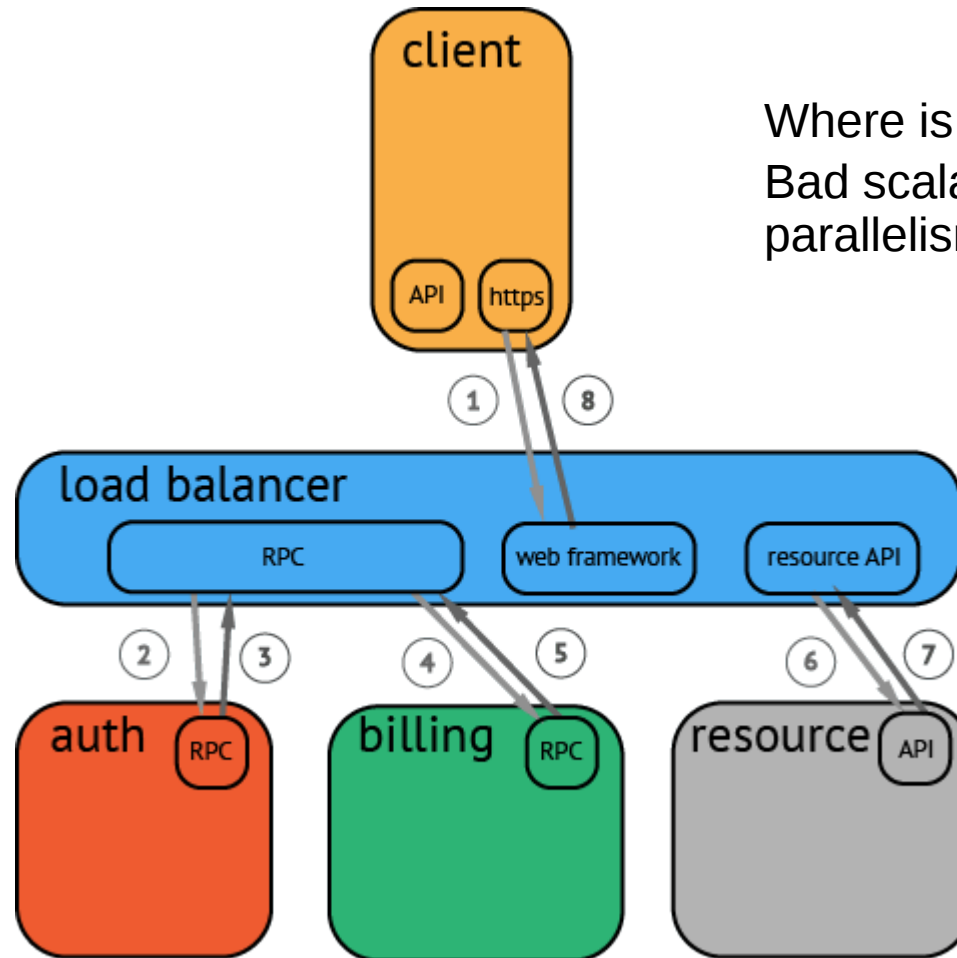


Critical Paths



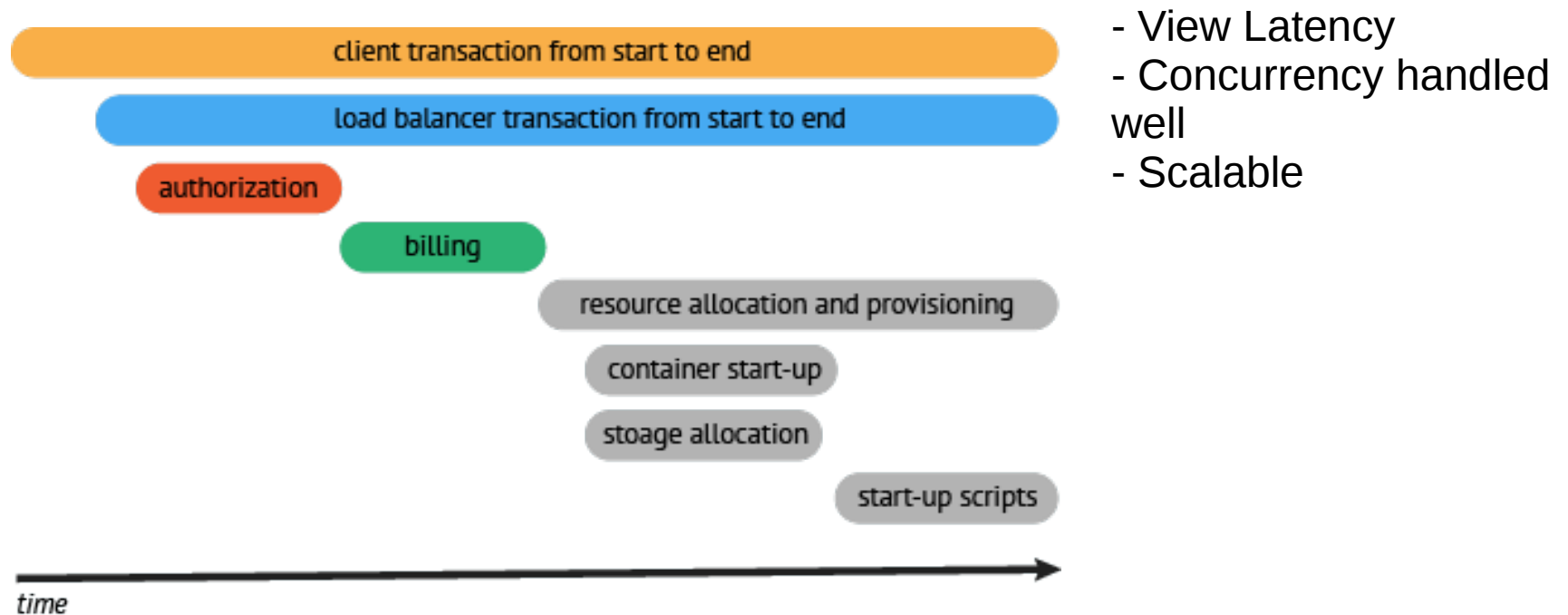
- Identify Critical paths
- With limited resources, allocate larger resource share to the critical paths
- Optimize critical paths

A simple trace visualization



Where is the component latency?
Bad scalability, bad response to
parallelism

Standard trace for a distributed app





Distributed Tracing Terminology



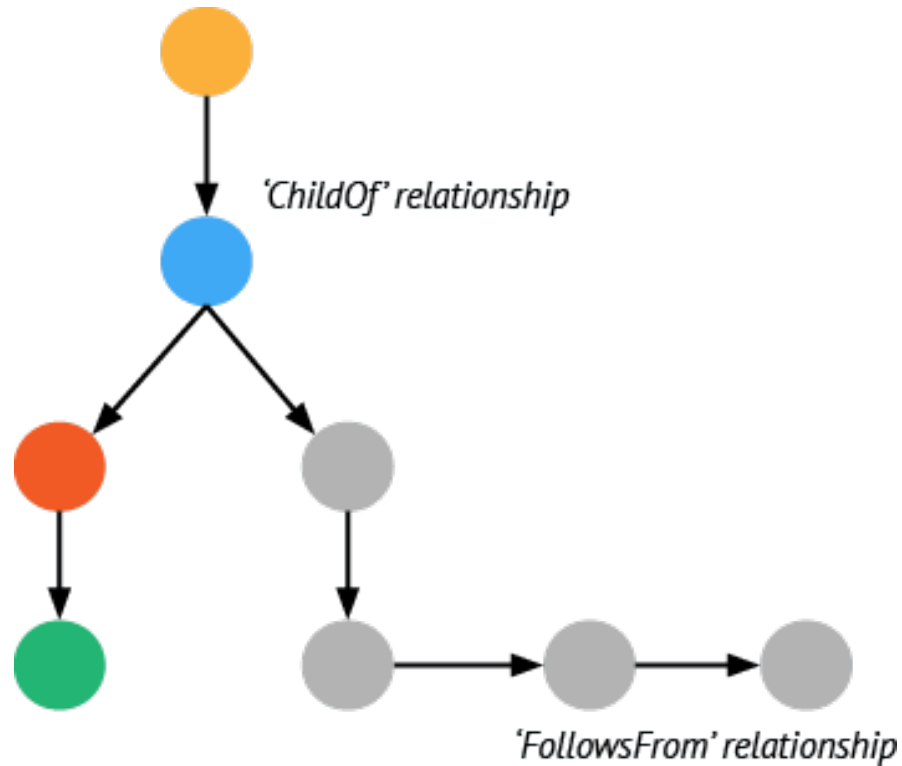
Span

- **Operation Name**
- **Start/Finish Timestamp**
- **Span Tags and Logs [key:value]**
- **Span Context**
- **References**



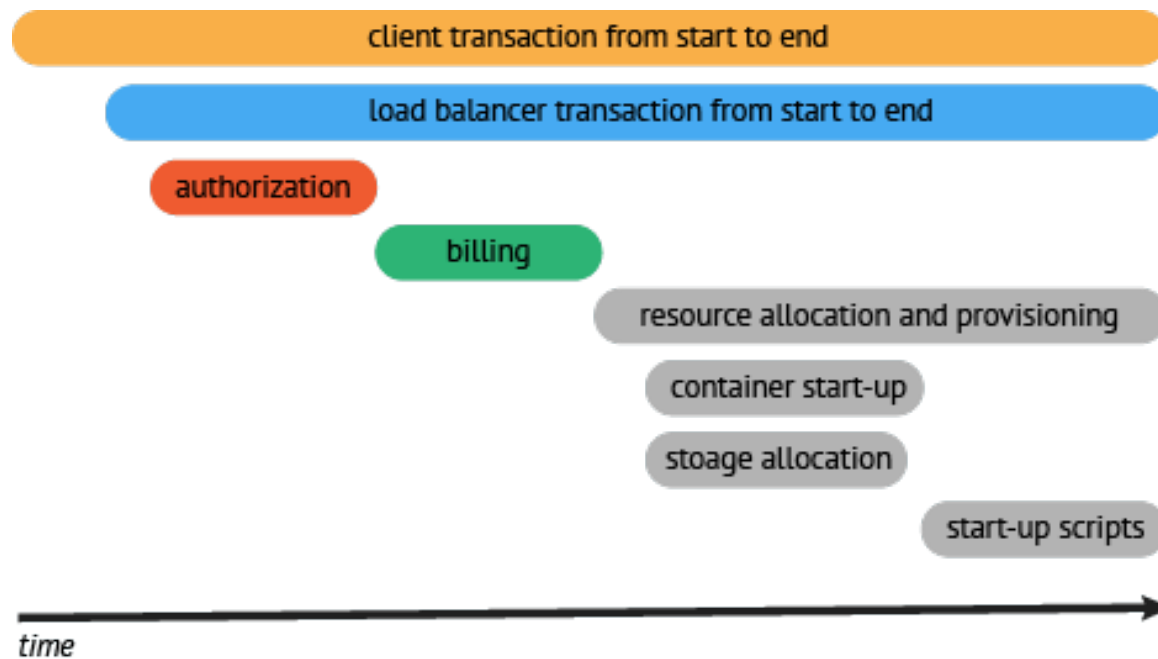
Span Relationships

- **ChildOf**
- **FollowsFrom**



Trace

- Set of spans sharing a single root span (trace id).
- Shows the path a request takes through the system.

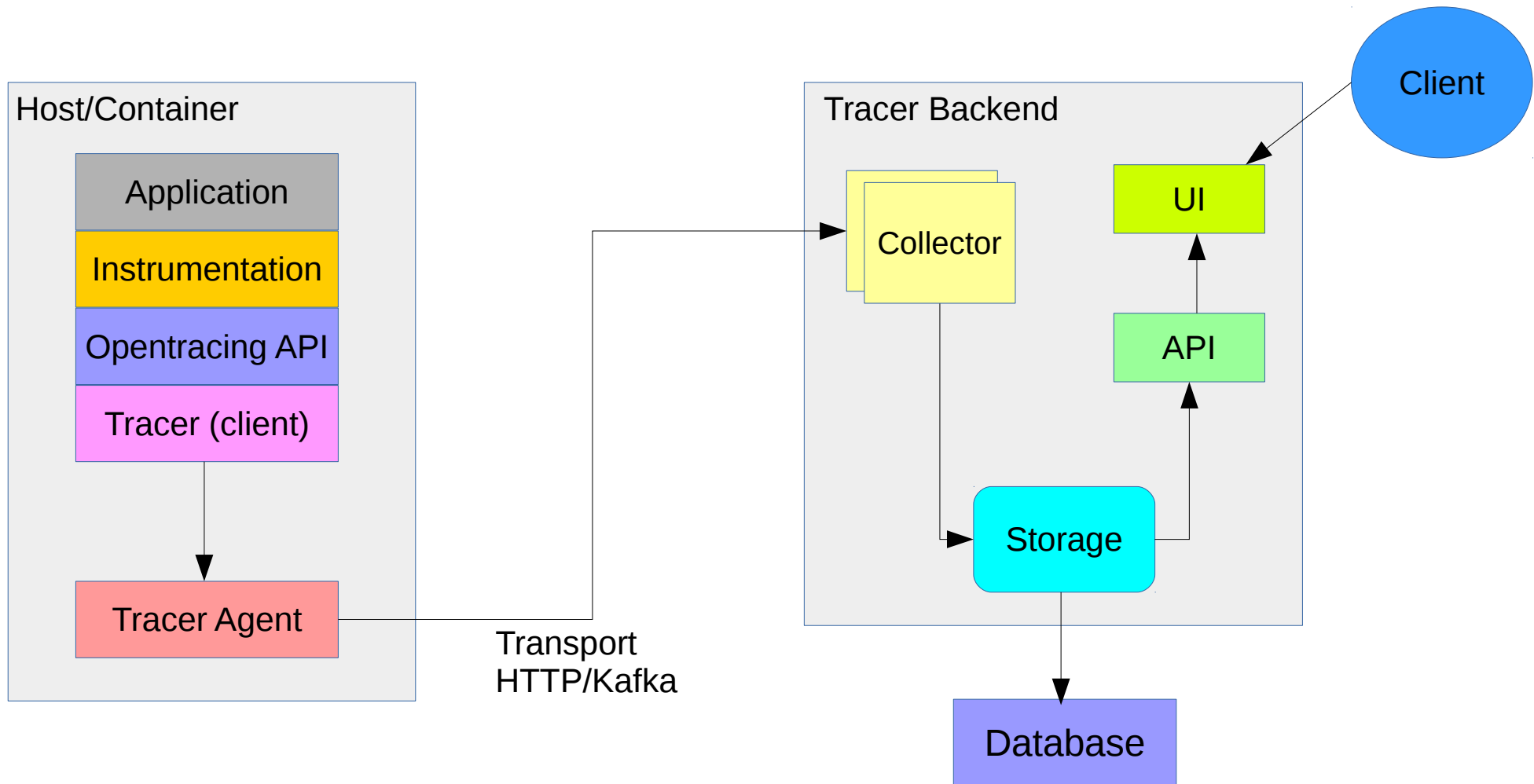




Distributed Tracers



Distributed tracer architecture



Distributed Tracers

- **Dapper (Google)**
- **Zipkin (Twitter)**
- **Jaeger (Uber)**
- **Appdash**

Add the tracer in the application code.



Distributed Tracers

- Dapper
- Zipkin
- Jaeger
- Appdash

**Add the tracer in the application code.
Which one to use?**



Distributed Tracers

- Dapper
- Zipkin
- Jaeger
- Appdash

Add the tracer in the application code.

Which one to use?

Need to change my tracer.



Distributed Tracers

- Dapper
- Zipkin
- Jaeger
- Appdash

Add the tracer in the application code.

Which one to use?

Need to change my tracer.

No, I don't want to change my entire source code.





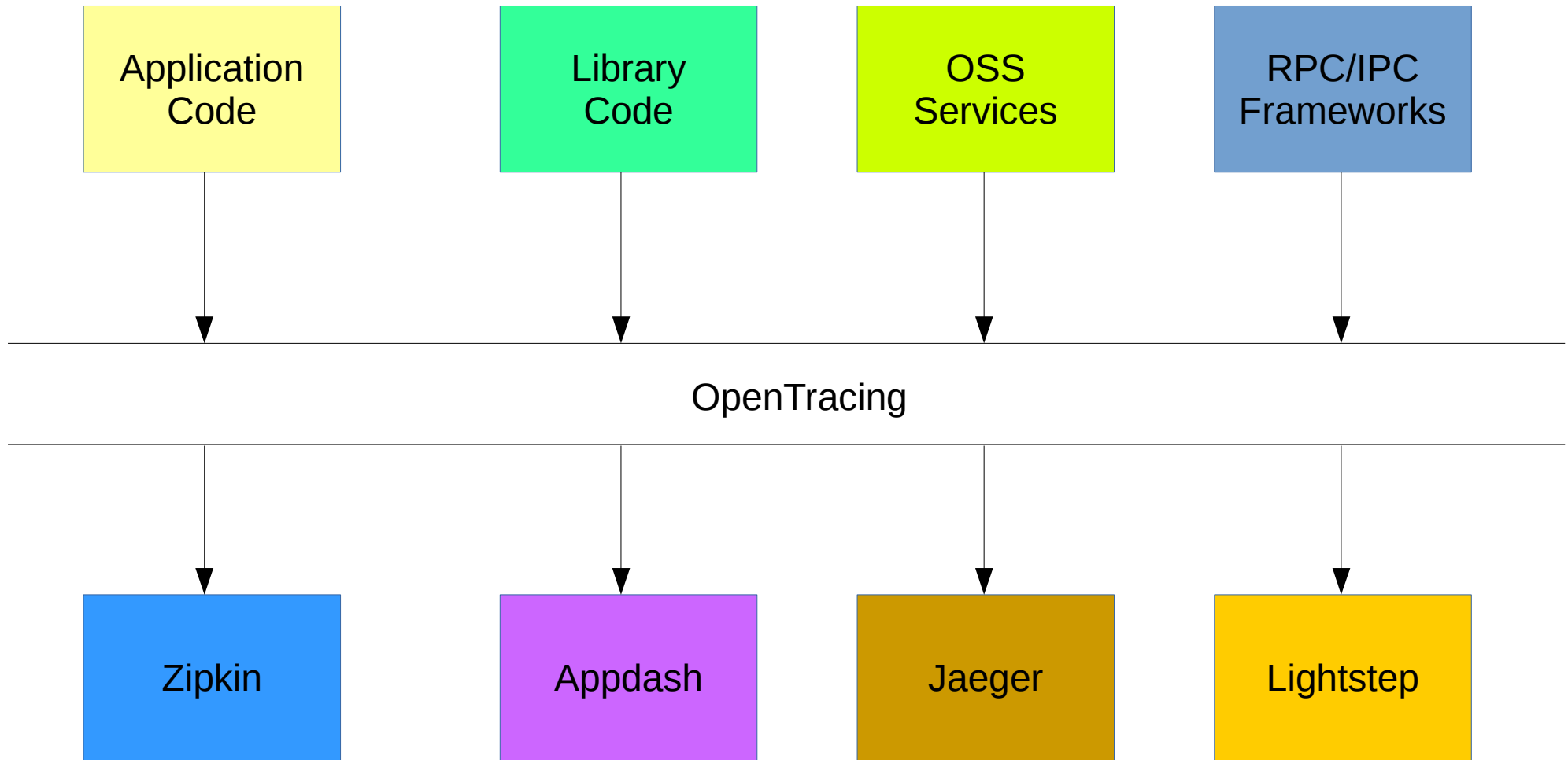
Enter OpenTracing...



Opentracing

- **Abstraction from different tracers.**
- **Consistent**
- **Expressive**
- **Vendor-Neutral APIs (Important)**
- **Switching between tracers ($O(1)$)**
- **Standardizes the behavior both within and between processes**

How does it all fit in?



Demo



perfevents and OpenTracing

- **Counter Abstraction from platform h/w**
- **Resource bottlenecks**
- **Workload characterization**
- **Quality of Service**

<https://github.com/opentracing-contrib/perfevents/>



Try it Yourself...

For OpenTracing and Zipkin :

- <https://github.com/opentracing>**
- <https://github.com/openzipkin/zipkin-go-opentracing>**

For platform analysis with OT + zipkin :

<https://github.com/platform-tracing/>

