KubeCon

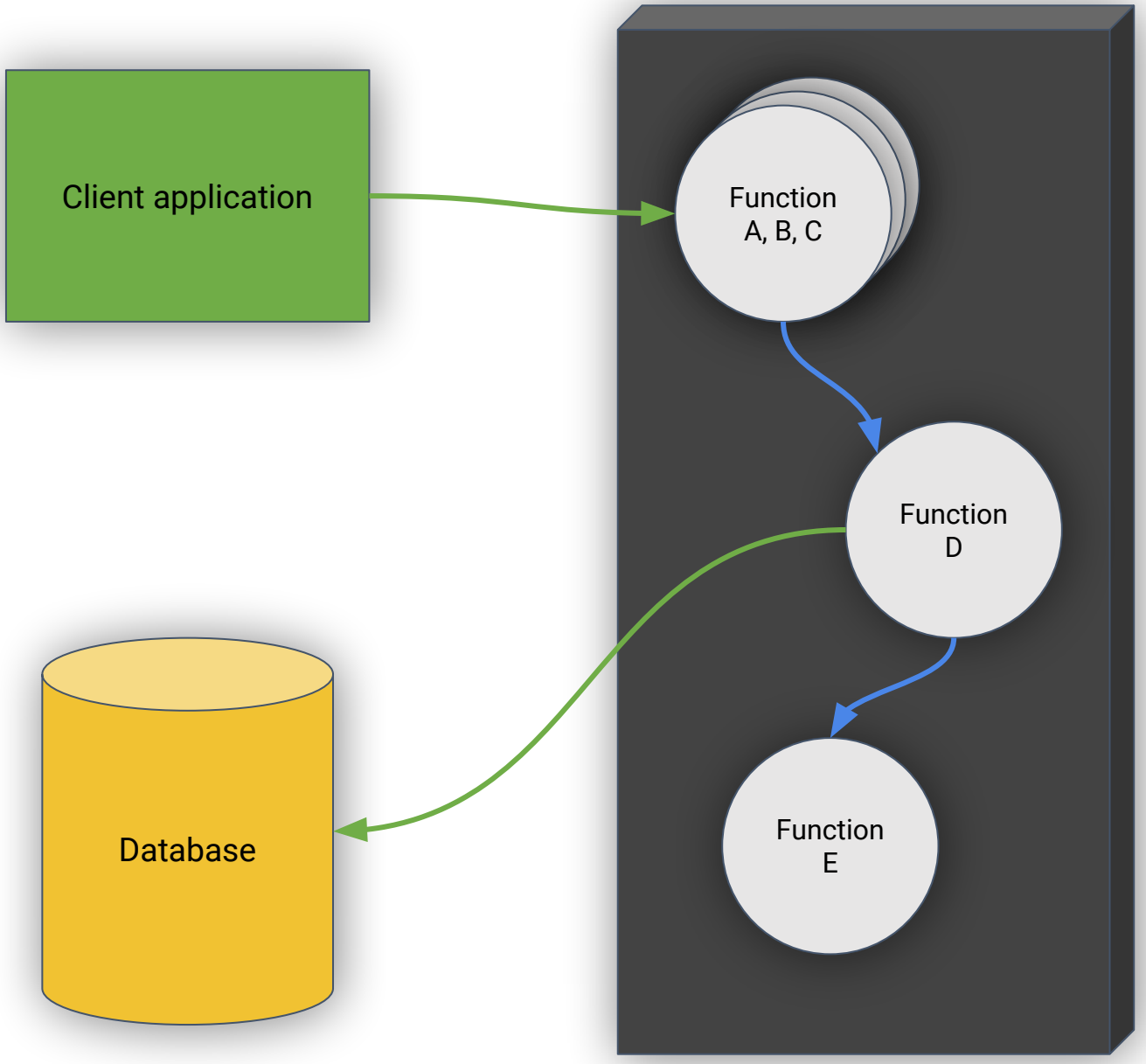
CloudNativeCon

Europe 2018

OpenCensus and Istio

Morgan McLean - Product Manager, Google
Varun Talwar - Founder, Startup

This was an image of the 2001: Space Odyssey monolith, but it's probably copyrighted so I removed it



Critical instrumentation:

Understanding how the system works

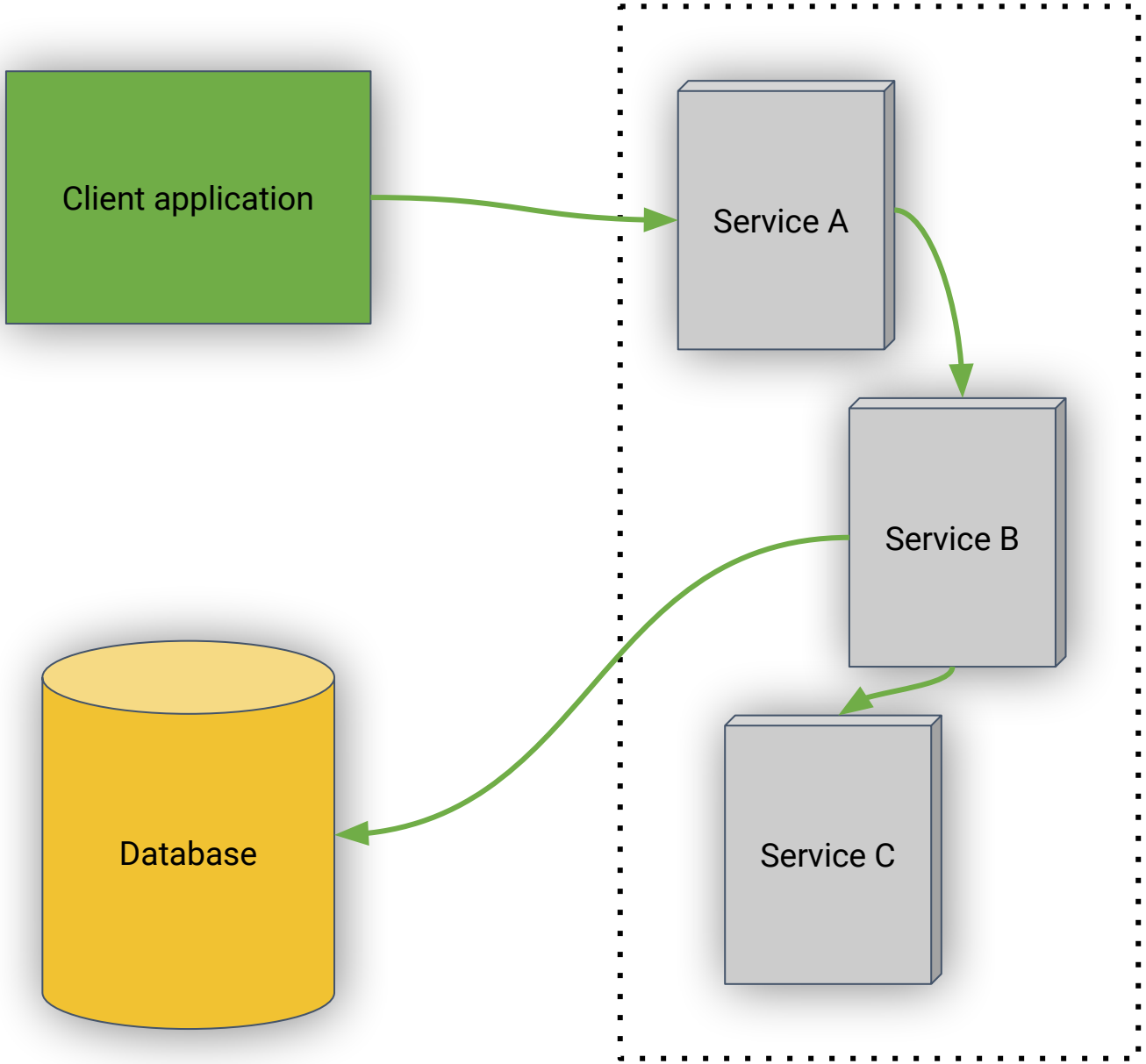
- Static code analysis
- Local debugging

Basic monitoring

- System-level metrics:
  - CPU consumption
  - Memory consumption
- Application-level metrics:
  - Endpoint latency
  - Custom metrics
- Logs

Performance tuning

- Offline / local testing
- Offline / local debugging
- Profiling
- Non-distributed “transaction” traces



Critical instrumentation:

Understanding how the system works

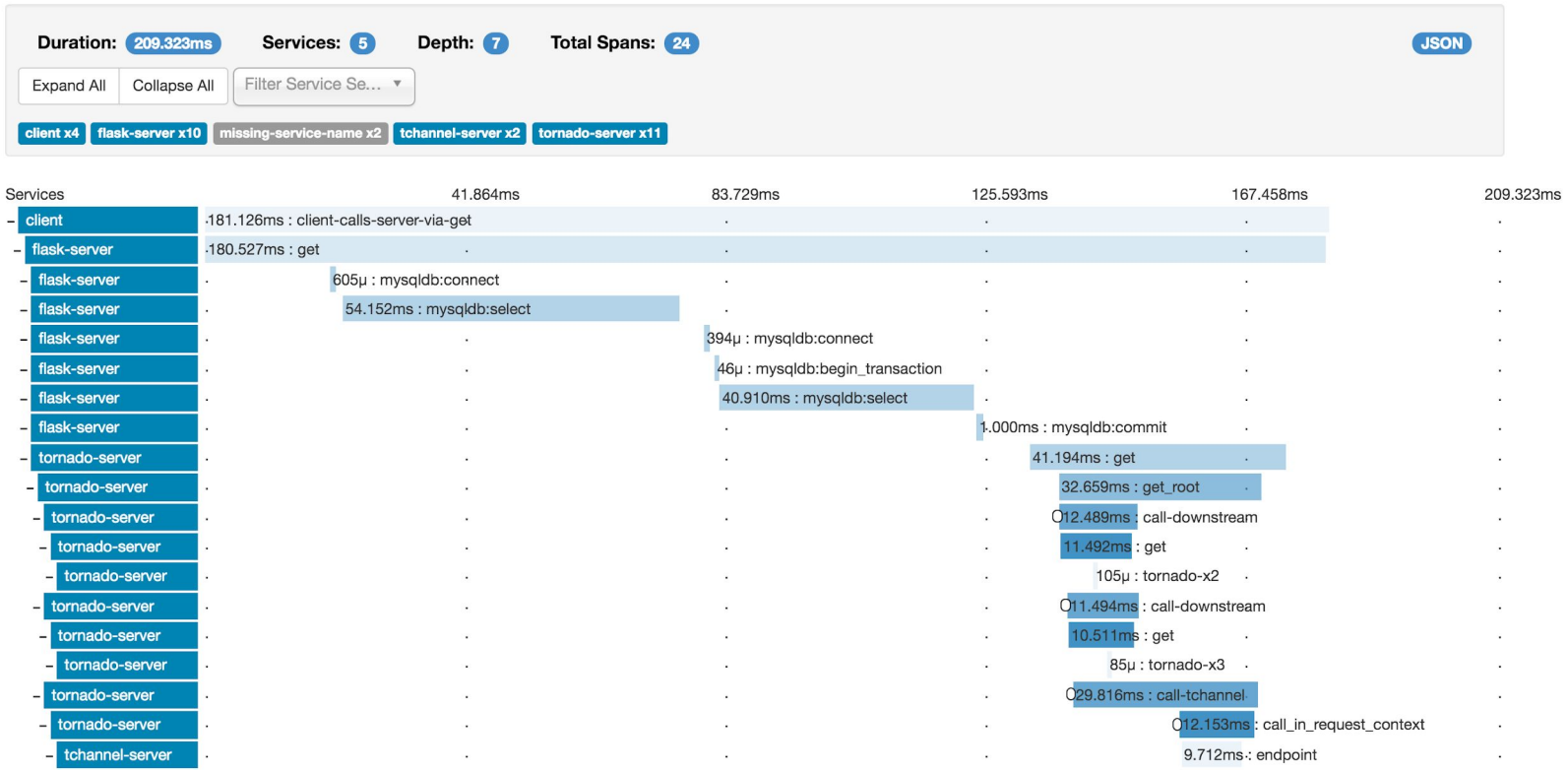
- Distributed tracing

Basic monitoring

- System-level metrics:
  - CPU consumption
  - Memory consumption
- Application-level metrics:
  - Endpoint latency of each node
  - RPC stats
  - Custom metrics
- Logs

Performance tuning

- Distributed tracing
- Production debugging
- Continuous profiling



Distributed tracing gives us:

- Application topology
- A view into *how* certain requests are handled
- An understanding of where the system is performing poorly
- An understanding of where errors are occuring
- Correlation / context for other signals like logs, errors, metrics, profiles, etc.

Application metrics give us:

- Latency, RPC stats at every level of the stack
- Custom metrics

Why is this hard?

- Ideally, traces and stats should be automatic for each RPC
  - This requires integrations with every language, RPC framework, storage client, API client, etc.
    - Projects like OpenCensus and OpenTracing are targeting this
- Libraries vs. agents?
  - Agents can be very slick
  - Libraries can be managed through source control, have less ‘magic’, and provide an API

What does OpenCensus provide?

- An implementation for tracing, application stats, and tags targeting 8 languages
  - **C++**, **Java**, **Go**, Python, PHP, Ruby, *.Net, node.js*
  - Integrations with web / RPC frameworks, storage clients, etc.
  - APIs for interacting with spans, stats, tags
  - Exporters
- Realtime unsampled analysis with z-pages

/z pages

TraceZ Summary

Span Name	Running	Latency Samples									Error Samples
		[>0us]	[>10us]	[>100us]	[>1ms]	[>10ms]	[>100ms]	[>1s]	[>10s]	[>100s]	
HttpServer/traceconfigz	0	0	0	0	0	0	0	0	0	0	0
HttpServer/tracez	1	0	0	0	0	1	0	0	0	0	0
Recv.helloworld.Greeter.SayHello	0	0	10	10	10	7	1	0	0	0	0

Span Name: Recv.helloworld.Greeter.SayHello

Finished Requests 10

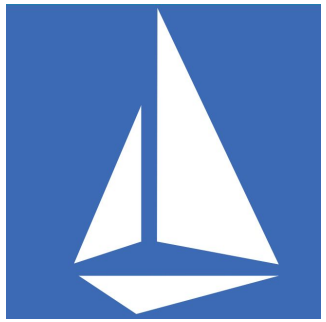
```
When                            Elapsed(s)
-----------------------------------------
2017/12/02-21:37:57.472000      0.002787     TraceId: 27845009b7298988c90207e89802c8ce SpanId: 274398e41b4a06d5 ParentSpanId: 1b8cd50723d30705
          21:37:57.472110        .   110 ... Received message_id=0 message_size=0
          21:37:57.474761        .  2651 ... Sent message_id=0 message_size=60414
                                             Status{canonicalCode=OK, description=null}
                                             Attributes:{}
2017/12/02-21:37:32.335000      0.001002     TraceId: 0e0f2910418068502dafef9f7610483c SpanId: fca7b1ed3ff53a0f ParentSpanId: 86d5e57bd86c4611
          21:37:32.335268        .   268 ... Received message_id=0 message_size=0
          21:37:32.335957        .   689 ... Sent message_id=0 message_size=49331
                                             Status{canonicalCode=OK, description=null}
                                             Attributes:{}
2017/12/02-21:37:21.259000      0.005406     TraceId: 460eb57a99e97c46809fbe3e0780d5d6 SpanId: eb54811c23eac071 ParentSpanId: 07294967d63f43b0
          21:37:21.259083        .    83 ... Received message_id=0 message_size=0
          21:37:21.264380        .  5296 ... Sent message_id=0 message_size=61007
                                             Status{canonicalCode=OK, description=null}
                                             Attributes:{}
2017/12/02-21:27:45.180000      0.006234     TraceId: 67d266124964df9dce976573ab1bcafc SpanId: 520aa99cbbc3286c ParentSpanId: 18e9b2ad811f8761
          21:27:45.180117        .   117 ... Received message_id=0 message_size=0
          21:27:45.186216        .  6099 ... Sent message_id=0 message_size=47647
                                             Status{canonicalCode=OK, description=null}
                                             Attributes:{}
```

Tags

- Tags are key value pairs that are used to store information about metrics
- Define your dimensions and view recorded data by dimensions
- Very powerful concept
  - *ip=10.32.103.12*
  - *user-agent=curl/1.0*
  - *coupon=discount-1f1acdbe3*

How does this work at Google?

Tracing at Google is automatically built into every service, made possible by Google's internal platform



OpenCensus + Istio

Envoy

- Configuration for Envoy stats
- Application and Envoy level Tracing + exporters

Mixer

- Aggregation
- Provide Mixer adapter for OpenCensus that generates trace spans from Mixer report calls. (Planned for summer)

- OC Lib
  - Tags, Pass headers

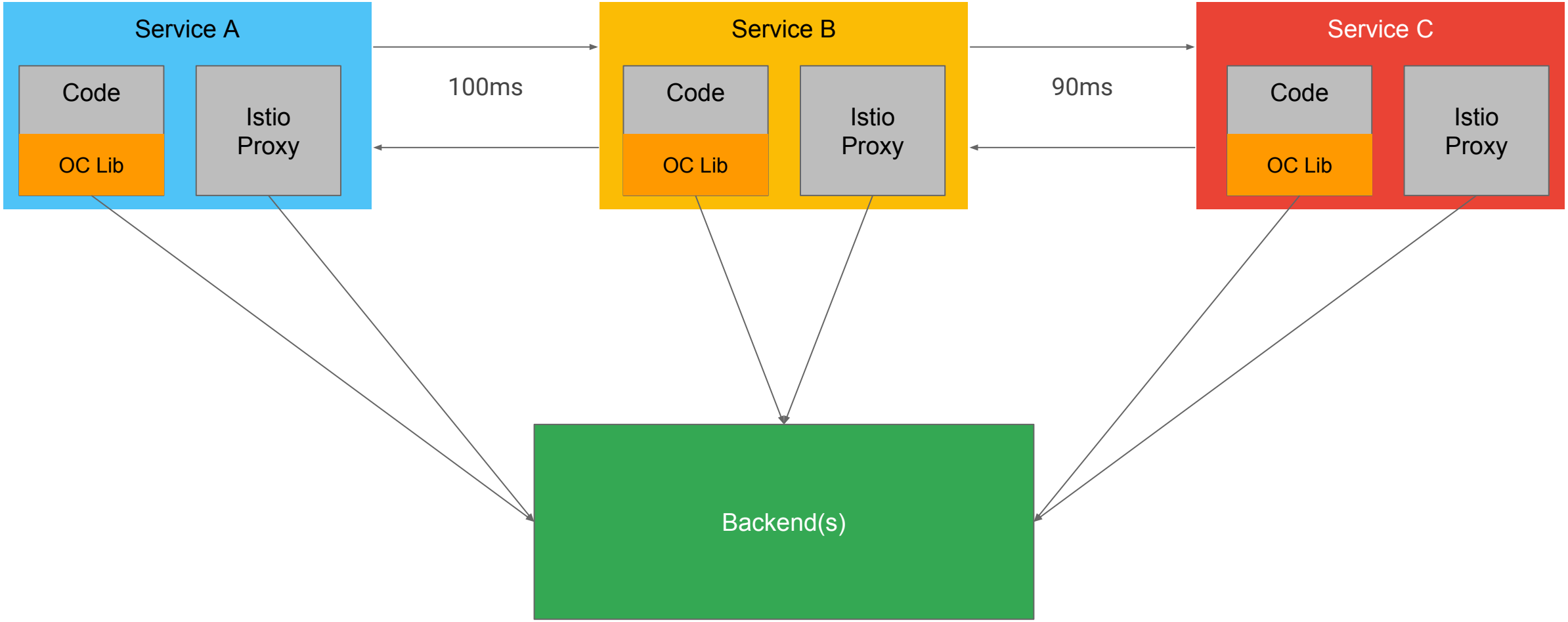
Doesn’t Envoy Provide This?

Yes, but:

- You have to perform correlations yourself
- No automatic spans from client libraries
- You can’t create custom spans, stats, tags, etc.
- No z-pages

Use OpenCensus for app instrumentation, builtin Envoy tracer for additional spans

OpenCensus + Istio



KubeCon

CloudNativeCon

Europe 2018

Hello World

gRPC and OpenCensus example

Code here: https://github.com/rakyll/opencensus-grpc-demo

What's coming next?

Istio + OpenCensus

- Completing Envoy support
- Mixer adapter for OpenCensus

OpenCensus

- More integrations: MongoDB, Spring, etc.
- Completing libraries across all 8 languages
- Client-side instrumentation for mobile and web apps
- More intelligent sampling
- Logs? Profiles? Errors?


KubeCon
CloudNativeCon
Europe 2018
Please visit opencensus.io
for more details