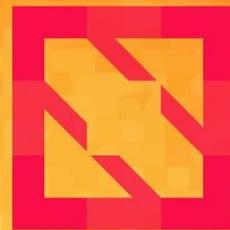






KubeCon



CloudNativeCon

North America 2019





KubeCon



CloudNativeCon

North America 2019

OpenTelemetry: The First Release, What's Next, and How to Get Involved



Presenters



KubeCon



CloudNativeCon

North America 2019



Sergey Kanzhelev
SWE at Microsoft
[@SergeyKanzhelev](https://twitter.com/SergeyKanzhelev)



Tristan Sloughter
SWE at Postmates
[@tsloughter](https://twitter.com/tsloughter)



Chris Kleinknecht
SWE at Google
[@c24t](https://twitter.com/c24t)



Morgan McLean
PM at Google
[@mtwo](https://twitter.com/mtwo)

OpenTelemetry



KubeCon



CloudNativeCon

North America 2019



OpenTelemetry makes robust, portable telemetry a built-in feature of cloud-native software.

OpenTelemetry



KubeCon



CloudNativeCon

North America 2019



API

Integrations

Libraries

Exporters

Collector

Contributors



KubeCon



CloudNativeCon

North America 2019



dynatrace



Postmates

DATADOG



SENTRY



Google Cloud



Uber

Contributors



KubeCon



CloudNativeCon

North America 2019



How Things Work: Membership



KubeCon



CloudNativeCon

North America 2019

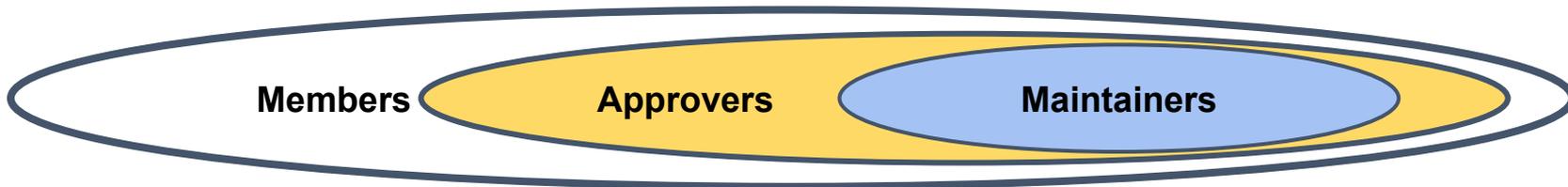
OpenTelemetry is free, vendor neutral set of libraries that anybody can use.

We value and welcome contributions!

Three levels of engagement thru contributions:

- Member
- Approver
- Maintainer

Project contains of multiple special interest groups.



How Things Work: Project Vision



KubeCon



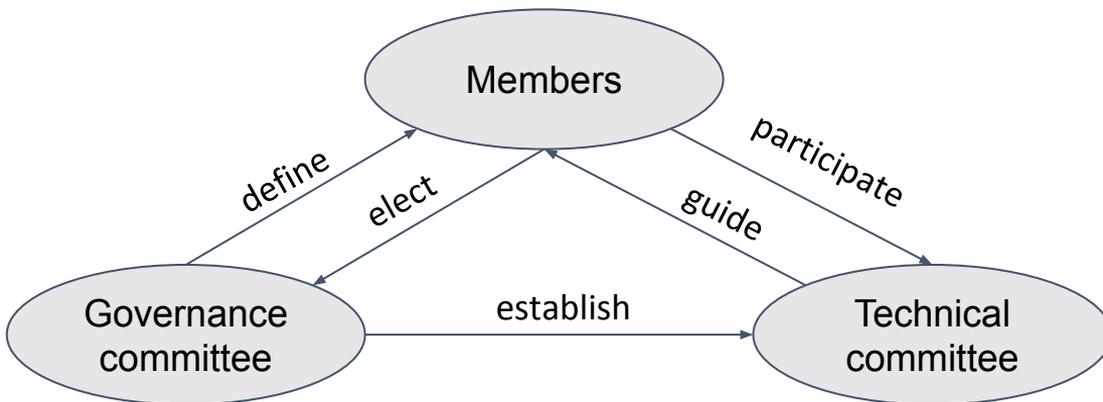
CloudNativeCon

North America 2019

“If you want to go fast, go alone, if you want to go far, go together”

There is a structure in place to ensure that we all building the same project:

- Project members are building the project
- Elected governance committee defines and uphold the project vision
- Technical committee shape the vision into the specifications



Telemetry in Action



KubeCon



CloudNativeCon

North America 2019



Telemetry in Action



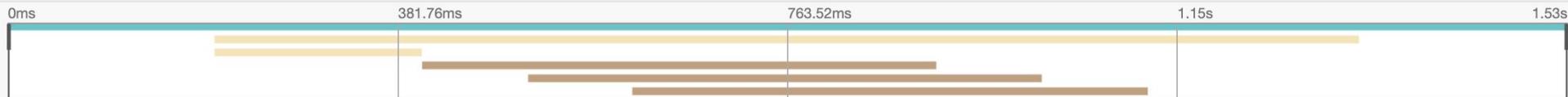
KubeCon



CloudNativeCon

North America 2019

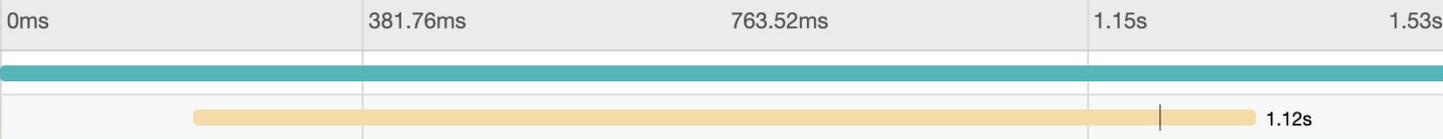
Trace Start **November 20 2019, 11:36:32.640** | Duration **1.53s** | Services **3** | Depth **3** | Total Spans **6**



Service & Operation

ParentSvc HTTP GET

KCNA19 KubeCon Talk



KubeCon Talk

Service: **KCNA19** | Duration: **1.12s** | Start Time: **202.49ms**

> **Tags:** internal.span.format = proto

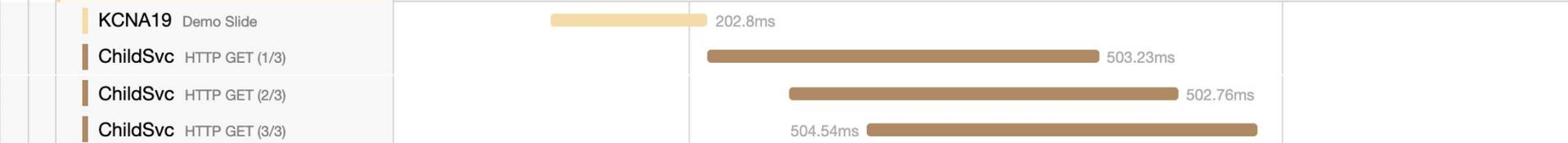
> **Process:**

> **Logs (1)**

> **1.22s:** message = applause | racuous = true

Log timestamps are relative to the start time of the full trace.

SpanID: b51f82572fc552c [🔗](#)



Telemetry in Action



KubeCon



CloudNativeCon

North America 2019

```
1 Span(  
2     name="kubekon_talk",  
3     context=SpanContext(trace_id=0xdead, span_id=0xbeef)),  
4     events=[{  
5         'name': 'applause',  
6         'attributes': {'racuous': True},  
7         'timestamp': '2019-11-21T12:00:00Z'  
8     }],  
9     parent=SpanContext(trace_id=0xdead, span_id=0x0b0e)),  
10    start_time='2019-11-21T10:55:00Z',  
11    end_time='2019-11-21T12:25:00Z'  
12 )
```

Telemetry in Action

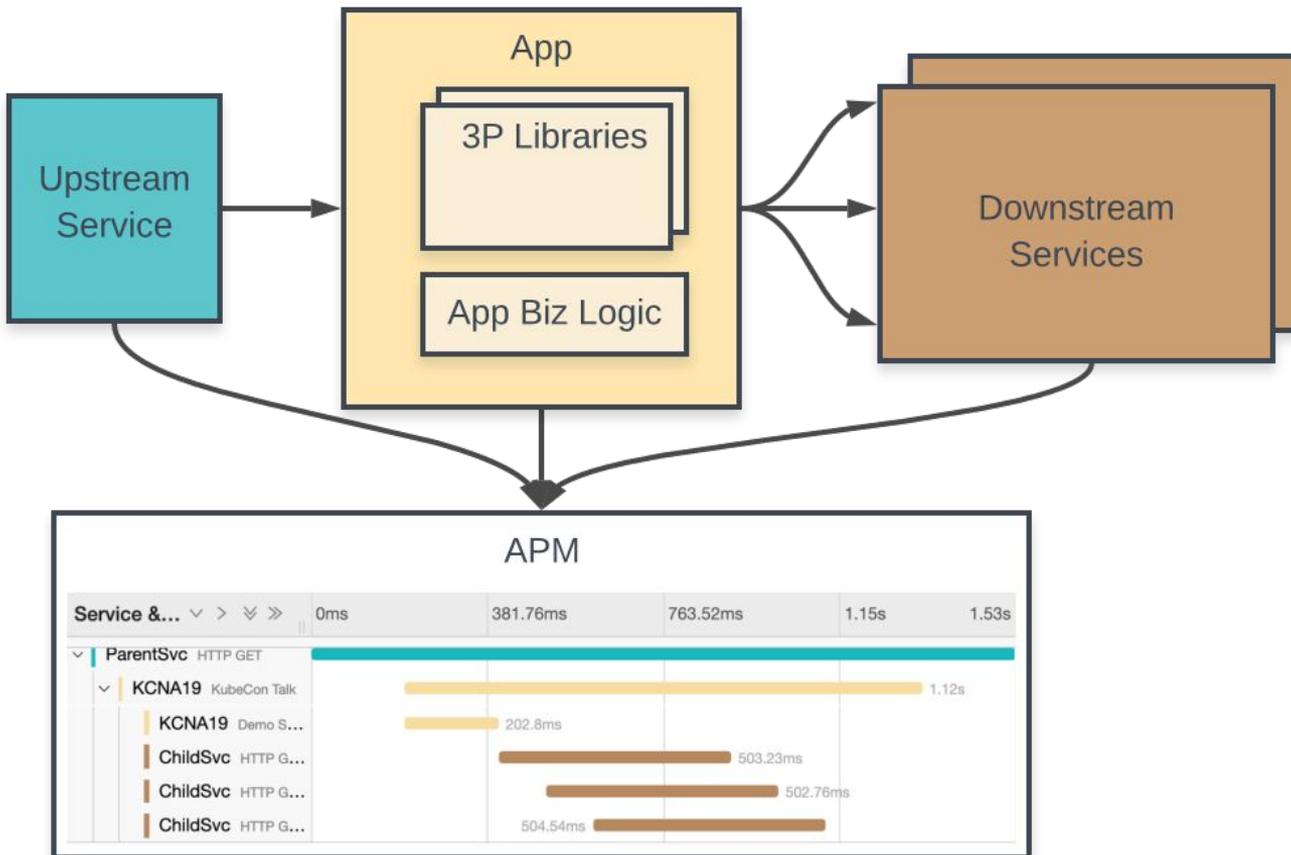


KubeCon



CloudNativeCon

North America 2019



Telemetry in Action



KubeCon



CloudNativeCon

North America 2019

OpenTelemetry enables you to:

- Instrument application and library code
- Propagate context between services
- Export telemetry data to APM backends



Telemetry in Action



KubeCon

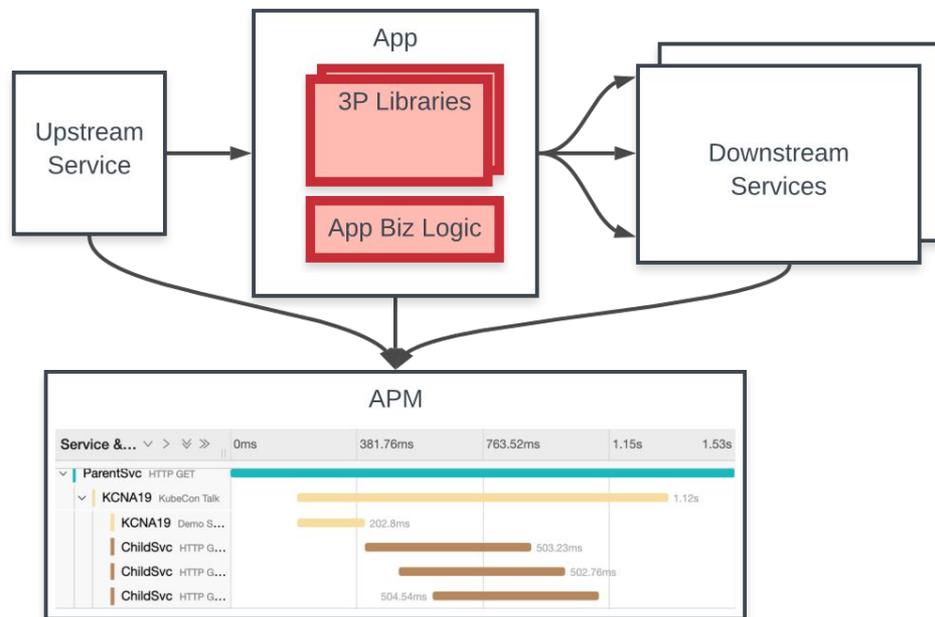


CloudNativeCon

North America 2019

Instrument application and library code

- Generate spans and metrics
- Only depend on API package, low overhead by default
- May get "for free" via libraries or auto-instrumentation



Telemetry in Action



KubeCon

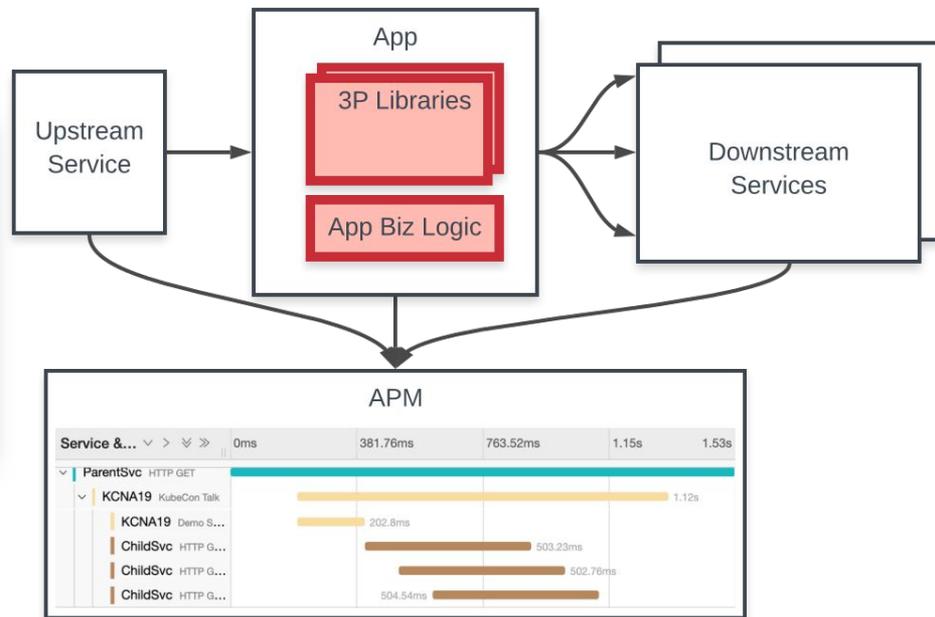


CloudNativeCon

North America 2019

Instrument application and library code

```
1 from opentelemetry import trace
2
3 tracer = trace.tracer()
4
5 # Create a span!
6 with tracer.start_as_current_span("kubeccon_talk") as span:
7     give_kubeccon_talk()
8     span.add_event("applause", {"racuous": True})
```



Telemetry in Action



KubeCon



CloudNativeCon

North America 2019

Propagate context between services

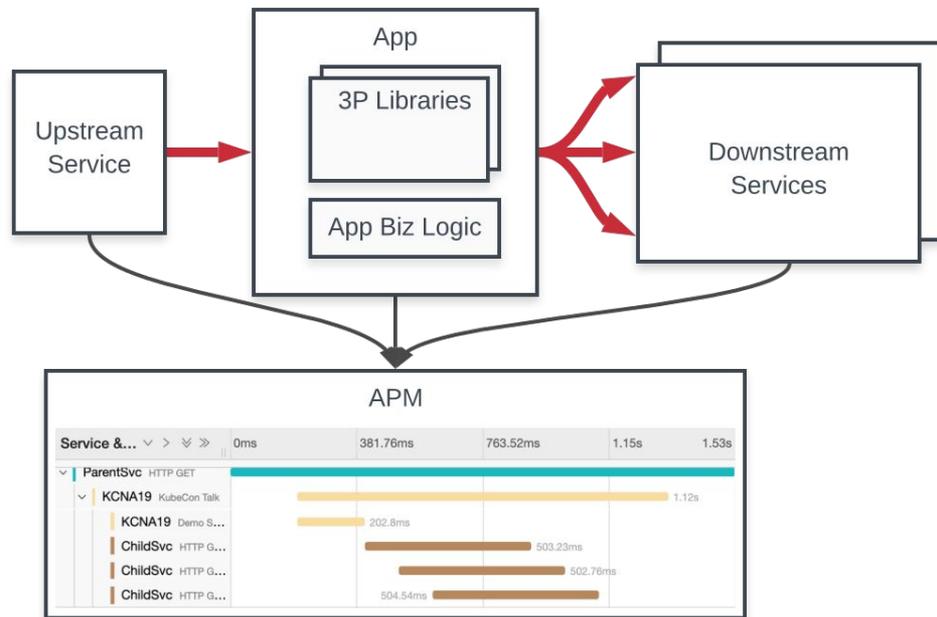
Incoming

- Deserialize request metadata
- Inject into application context

Outgoing

- Serialize some application context
- Attach as request metadata

E.g. [w3c/trace-context](https://w3c.github.io/trace-context/)



Telemetry in Action



KubeCon

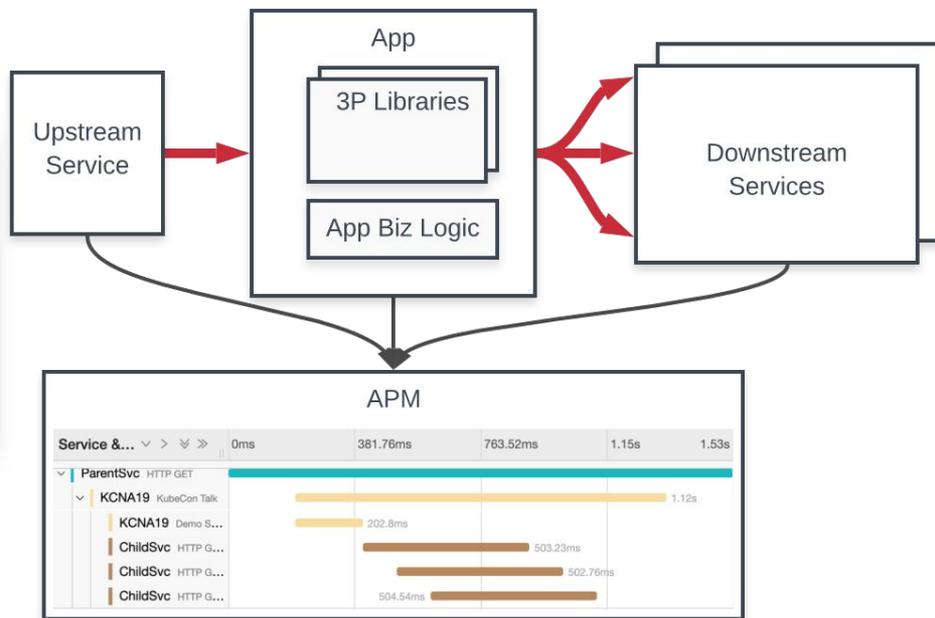


CloudNativeCon

North America 2019

Propagate context between services

```
1 # Accept B3/zipkin trace headers
2 propagators.set_global_httptextformat(B3Format())
3
4 # Handle incoming requests via Flask
5 flask_app.wsgi_app = OpenTelemetryMiddleware(flask_app.wsgi_app)
6
7 # Add metadata to outgoing requests via Requests
8 opentelemetry.ext.http_requests.enable(tracer())
```



Telemetry in Action



KubeCon

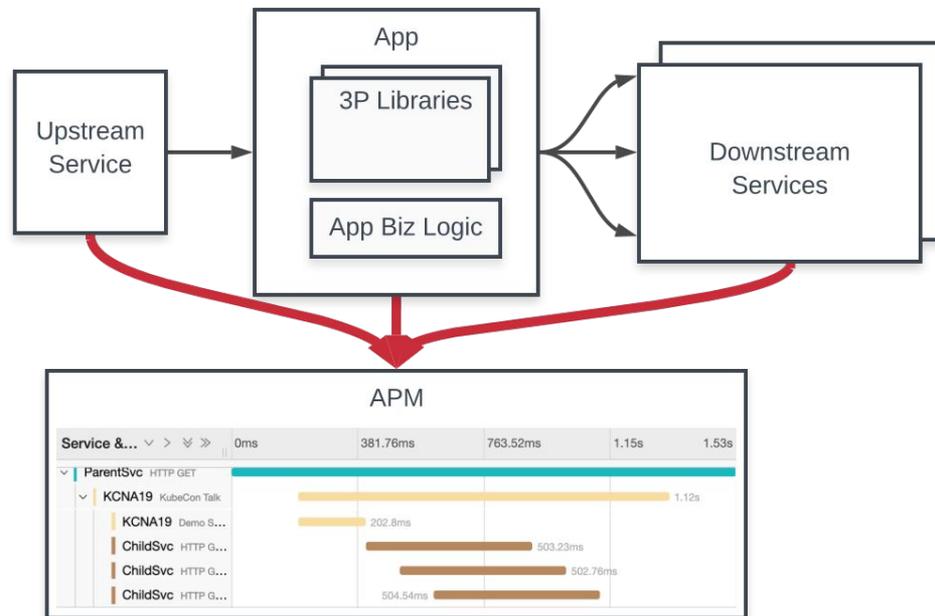


CloudNativeCon

North America 2019

Export telemetry data to APM backends

- APM vendors maintain exporters
- May export to multiple backends at once
- Export to agent/collector to swap other exporters out at run time



Telemetry in Action



KubeCon

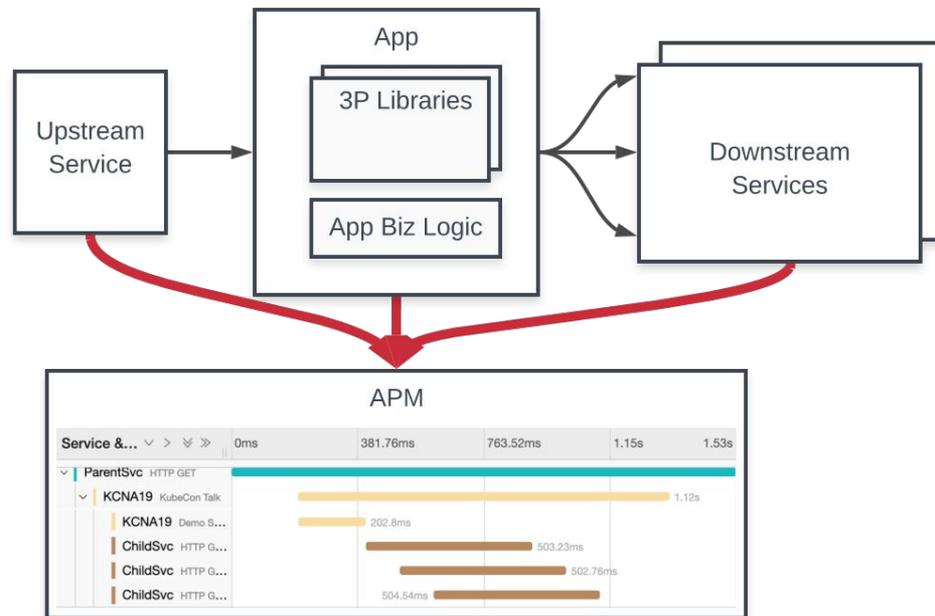


CloudNativeCon

North America 2019

Export telemetry data to APM backends

```
1 tracer = trace.tracer()
2
3 tracer.add_span_processor(
4     BatchExportSpanProcessor(
5         JaegerSpanExporter(
6             service_name="KubeCon Demo",
7             **jaeger_config
8         ),
9         **batching_config
10     )
11 )
```



Architecture

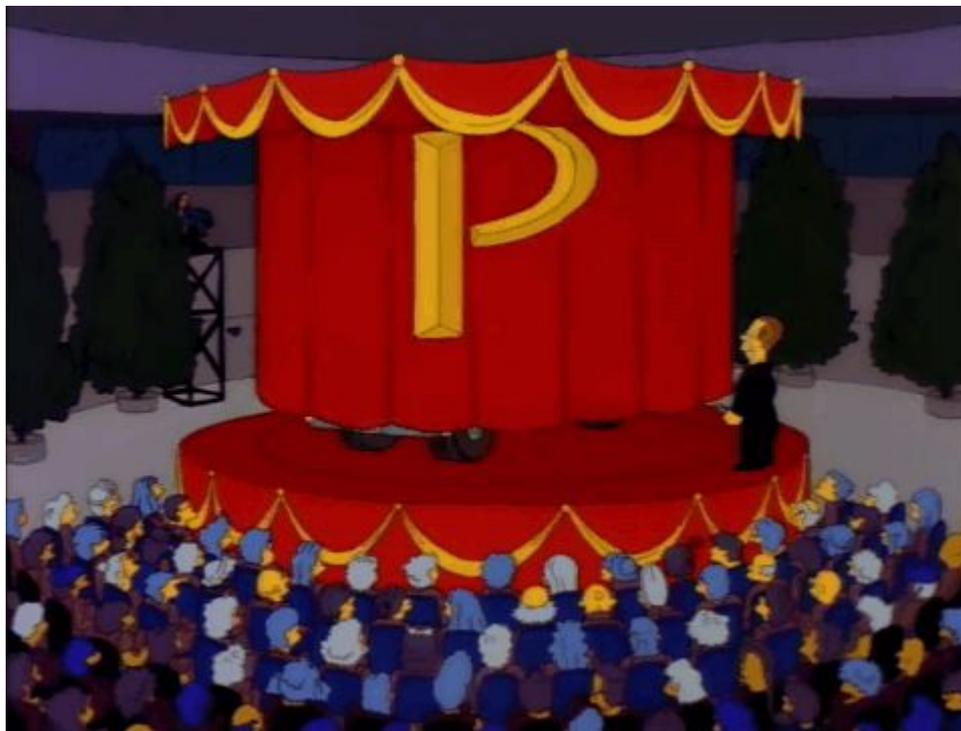


KubeCon



CloudNativeCon

North America 2019



Architecture

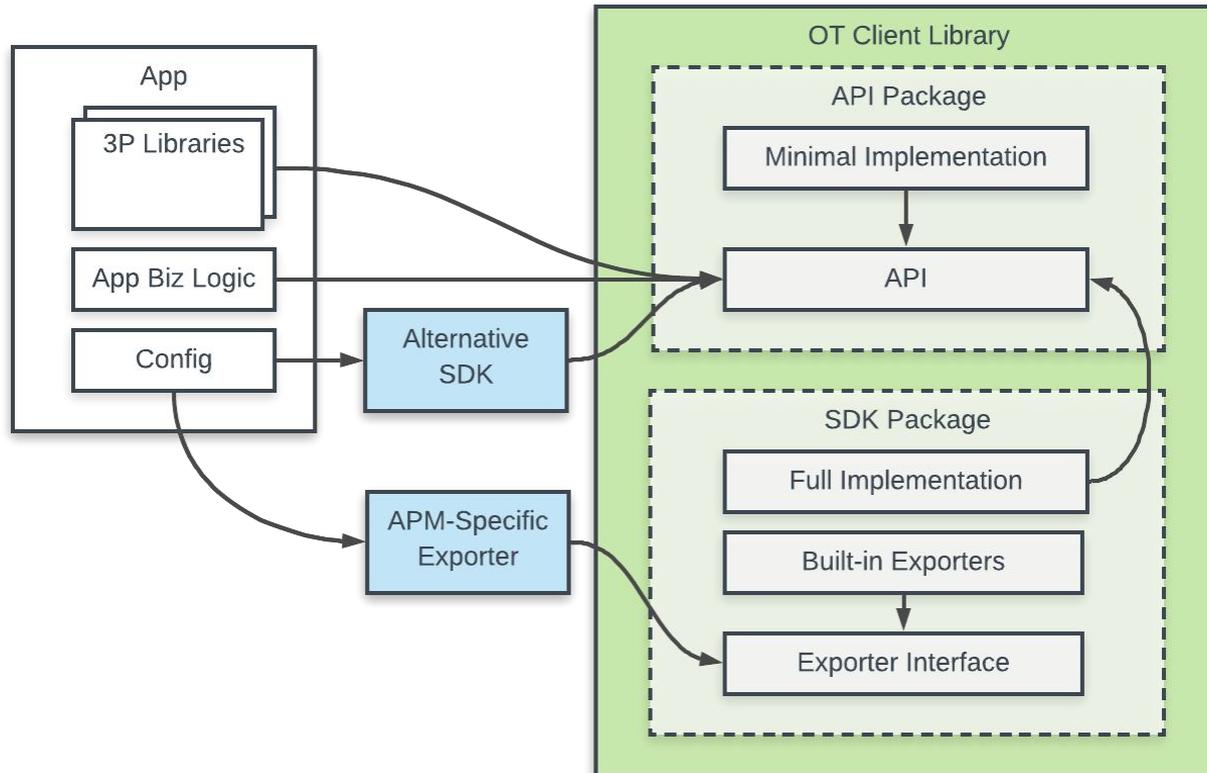


KubeCon



CloudNativeCon

North America 2019



OpenTelemetry Personas



KubeCon



CloudNativeCon

North America 2019

OpenTelemetry is designed for multiple users, each with different use cases and goals

- Application developers
- Library owners
- APM vendors



Persona: Application Owner



KubeCon

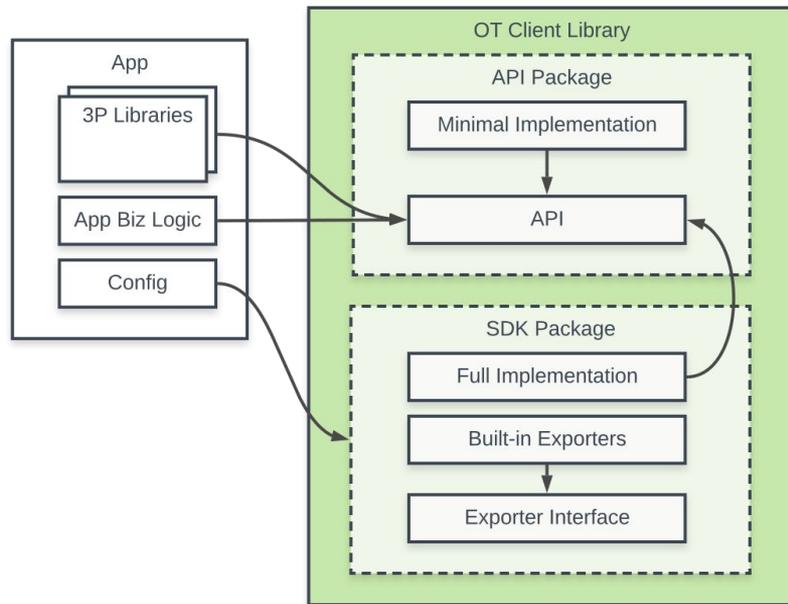


CloudNativeCon

North America 2019

End-users/developers/operators of applications that need instrumentation
(Postmates, Walmart, etc.)

- Application code is instrumented with OT API package
- Load a specific version of the SDK, which may vary by deployment



Persona: Application Owner



KubeCon

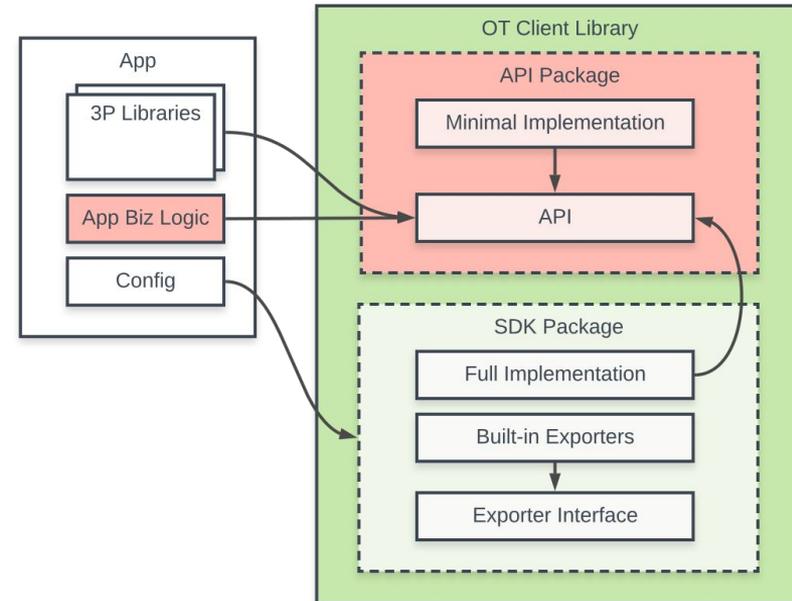


CloudNativeCon

North America 2019

End-users/developers/operators of applications that need instrumentation
(Postmates, Walmart, etc.)

- Application code is instrumented with OT API package
- Load a specific version of the SDK, which may vary by deployment



Persona: Application Owner



KubeCon

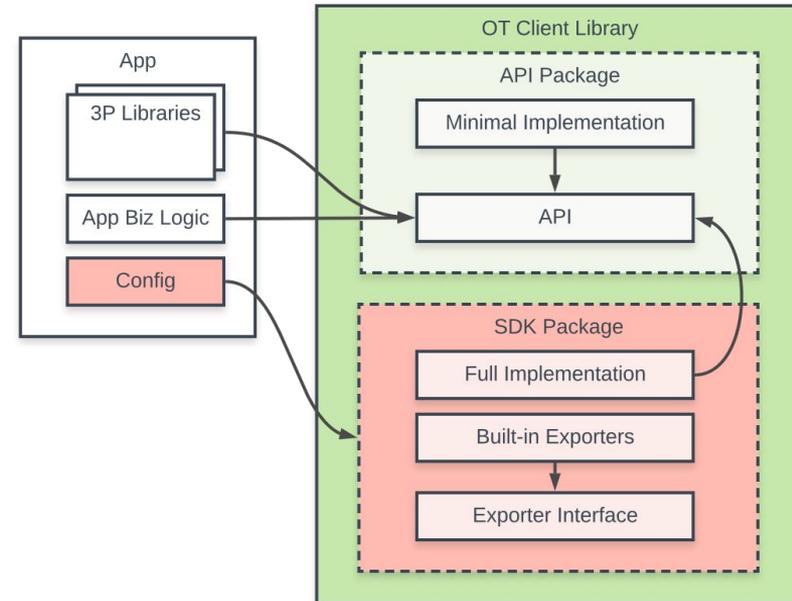


CloudNativeCon

North America 2019

End-users/developers/operators of applications that need instrumentation
(Postmates, Walmart, etc.)

- Application code is instrumented with OT API package
- Load a specific version of the SDK, which may vary by deployment



Persona: Application Owner



KubeCon

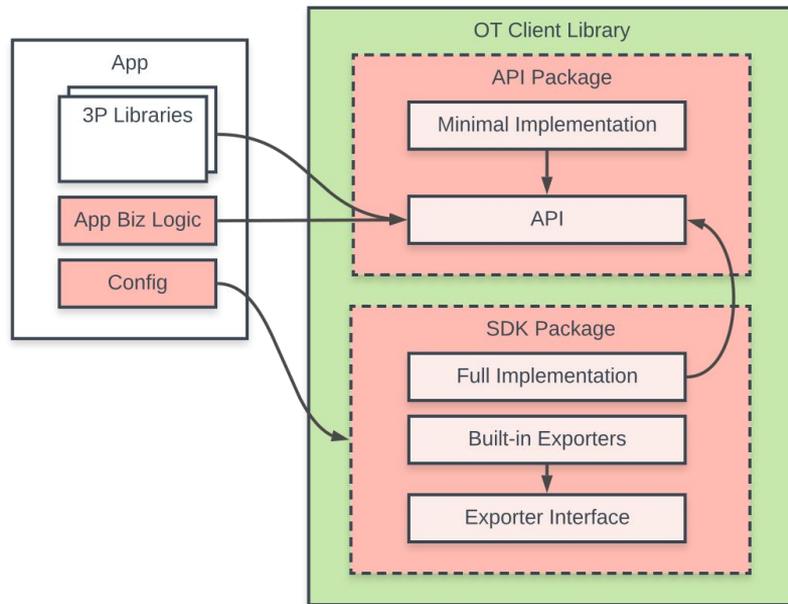


CloudNativeCon

North America 2019

End-users/developers/operators of applications that need instrumentation
(Postmates, Walmart, etc.)

- Application code is instrumented with OT API package
- Load a specific version of the SDK, which may vary by deployment



Persona: Library Owner



KubeCon

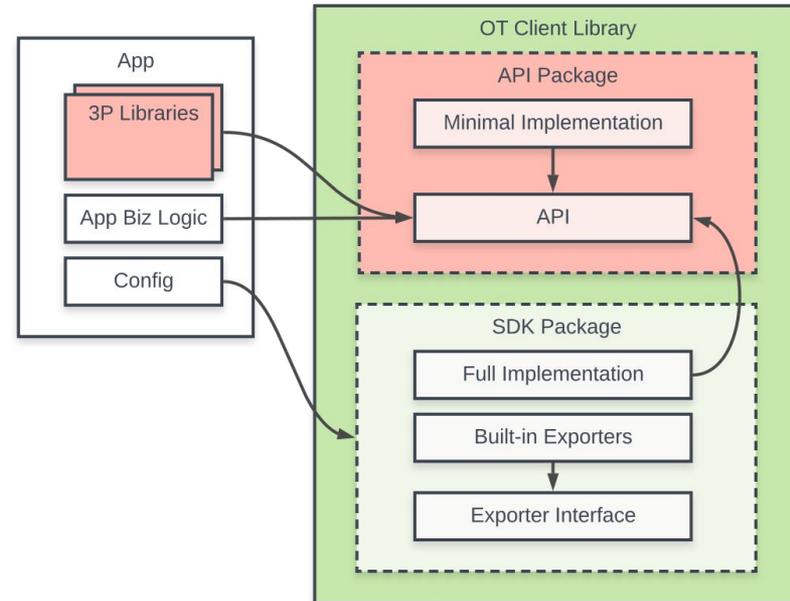


CloudNativeCon

North America 2019

Library owners/maintainers who want to maintain integrations (MongoDB, JDBC, etc.)

- Value performance, ease of maintenance
- Should depend on API package only
- Libraries instrumented with OpenTracing or OpenCensus will continue to work with OpenTelemetry



Persona: APM Vendor



KubeCon

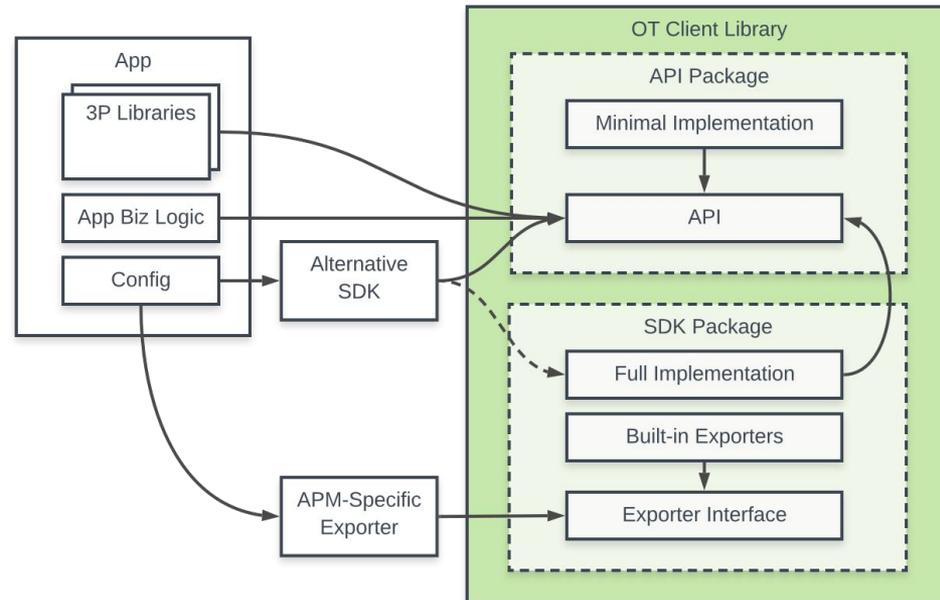


CloudNativeCon

North America 2019

Telemetry vendors who need to build exporters and understand where the market is going (Dynatrace, New Relic, Datadog, etc.)

- May need advanced features specific to their APM service
- Write custom exporters to send telemetry data to their agent or service
- May extend the SDK, e.g to act on span start and end events
- May ship their own SDKs to replace those that ship with client libraries



Persona: APM Vendor



KubeCon

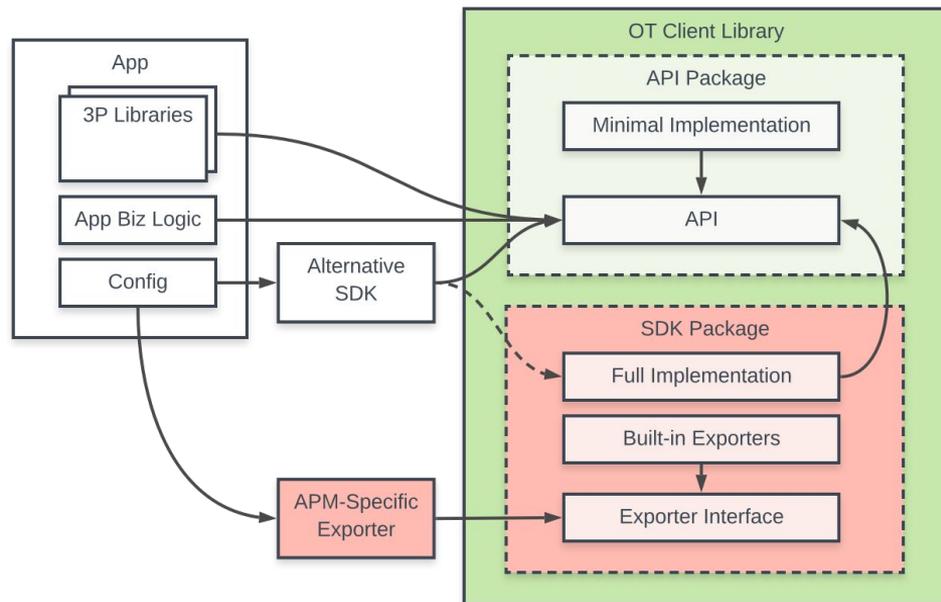


CloudNativeCon

North America 2019

Telemetry vendors who need to build exporters and understand where the market is going (Dynatrace, New Relic, Datadog, etc.)

- May need advanced features specific to their APM service
- Write custom exporters to send telemetry data to their agent or service
- May extend the SDK, e.g to act on span start and end events
- May ship their own SDKs to replace those that ship with client libraries



Persona: APM Vendor



KubeCon

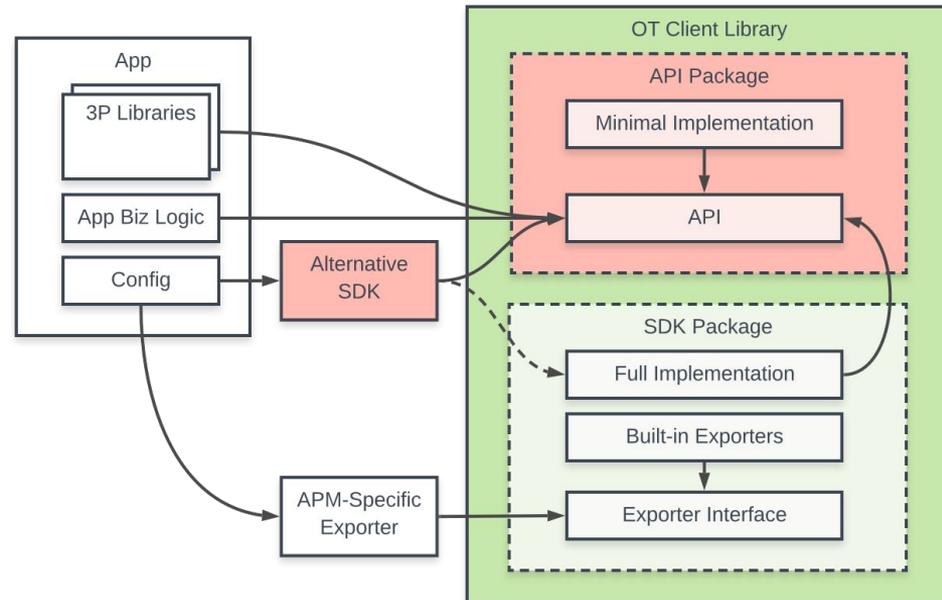


CloudNativeCon

North America 2019

Telemetry vendors who need to build exporters and understand where the market is going (Dynatrace, New Relic, Datadog, etc.)

- May need advanced features specific to their APM service
- Write custom exporters to send telemetry data to their agent or service
- May extend the SDK, e.g to act on span start and end events
- May ship their own SDKs to replace those that ship with client libraries



Persona: APM Vendor



KubeCon

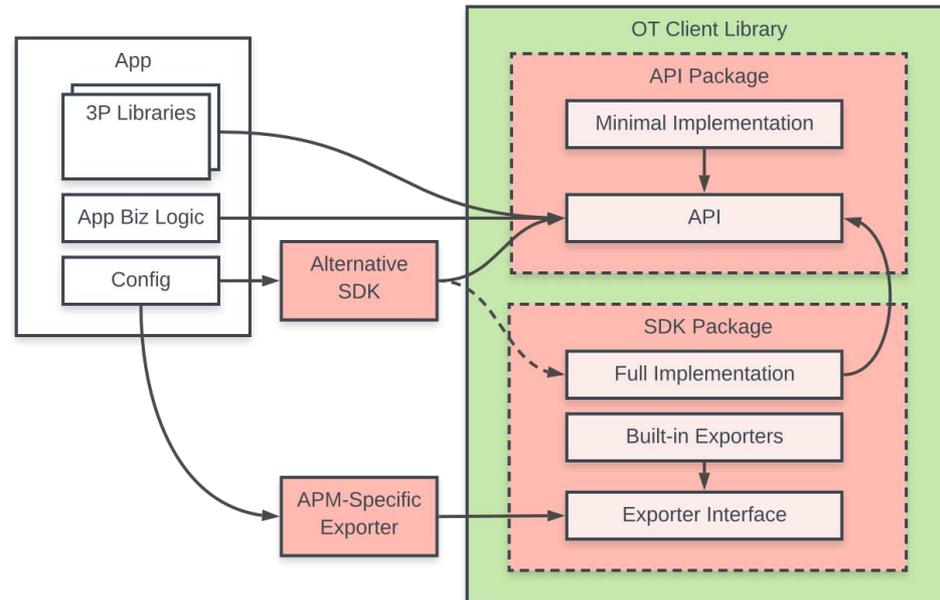


CloudNativeCon

North America 2019

Telemetry vendors who need to build exporters and understand where the market is going (Dynatrace, New Relic, Datadog, etc.)

- May need advanced features specific to their APM service
- Write custom exporters to send telemetry data to their agent or service
- May extend the SDK, e.g to act on span start and end events
- May ship their own SDKs to replace those that ship with client libraries



OpenTelemetry Components

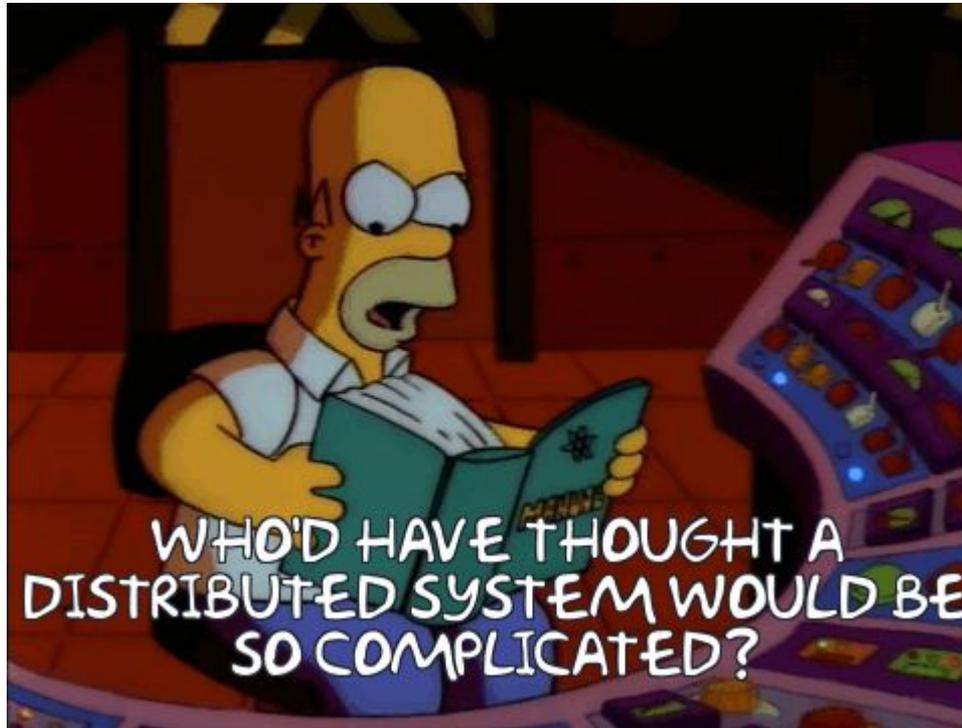


KubeCon



CloudNativeCon

North America 2019



OpenTelemetry Components



KubeCon



CloudNativeCon

North America 2019

OpenTelemetry includes:

- Client libraries in multiple languages
- Specification, data format, and semantic conventions
- Integrations for other libraries/frameworks/etc.
- Exporters for APMs
- A standalone collector

Component: Client Libraries



KubeCon



CloudNativeCon

North America 2019

Clients in multiple languages that generate telemetry data

E.g. [opentelemetry-python](#)

- Tracing
- Metrics
- Logs (someday)
- Separate API and SDK packages
- Other utilities
 - Context propagation, both in- and intra-process
 - Resource detection

Component: Specification



KubeCon



CloudNativeCon

North America 2019

The specification describes how client libraries should behave, which components belong to API and SDK packages in each language

[opentelemetry-specification](#)

- Includes a data format for exporters and agent/collector protocol
- Includes semantic conventions for field names

Component: Exporters



KubeCon



CloudNativeCon

North America 2019

Exporters format and ship telemetry data to specific APM backends

E.g. [opentelemetry-ext-jaeger](#)

- Exporter interface included in the SDK package
- SDK also includes utilities for exporters: batching, retrying, etc.
- Configured application-wide: switch APMs, switch exporters without changing application code

Component: Integrations



KubeCon



CloudNativeCon

North America 2019

Multiple kinds of integrations and extensions:

- Framework (web/service)
 - Deserialize request metadata, inject into application context
 - Generally creates a span, metrics for each handled request
- Protocol/transport
 - E.g. W3C-HTTP, gRPC
 - Text and binary format options
 - Framework integrations may use protocol integrations
- Other libraries, especially databases
 - Integrations only depend on API package
 - E.g. [opentelemetry-ext-pymongo](#)

Component: Collector



KubeCon



CloudNativeCon

North America 2019

Standalone agent or service that provides smart trace sampling, metrics aggregation, and other advanced features

[opentelemetry-collector](#)

- App exports to OT agent, which exports to other APM backends
- Single binary with two deployment options:
 - "Agent" sidecar that runs on same host/pod as app
 - "Collector" standalone application that runs independently

What should you do next?



KubeCon



CloudNativeCon

North America 2019



If you are already involved



KubeCon



CloudNativeCon

North America 2019

- Keep it up!
- Help others get involved
- Tell your friends and coworkers!
- Join a [SIG](#), if you haven't already



To start using OpenTelemetry



KubeCon



CloudNativeCon

North America 2019

- Head over the docs <https://opentelemetry.io/docs/>
- Choose your language and checkout the Quick Start
- If no Quick Start exists yet, see the individual [repos on Github](#)

Tracing

- [Quick Start](#)
- [API Reference](#)

This page contains documentation for OpenTelemetry Ruby.

Quick Start

Please note that this library is currently in *alpha*, and shouldn't be used in production environments.

The API and SDK packages are available on rubygems.org, and can installed via `bundler` :

```
bundle install opentelemetry-api
bundle install opentelemetry-sdk
```

From there, you should be able to use opentelemetry as per the following:

Tracing

- [Quick Start](#)
- [API Reference](#)

This page contains documentation for OpenTelemetry Go.

Quick Start

Please note that this library is currently in *alpha*, and shouldn't be used in production environments.

First let's install the API and SDK packages

```
go get -u go.opentelemetry.io/otel
```

From there, you should be able to use opentelemetry as per the following:

Already using OpenTracing/Census?



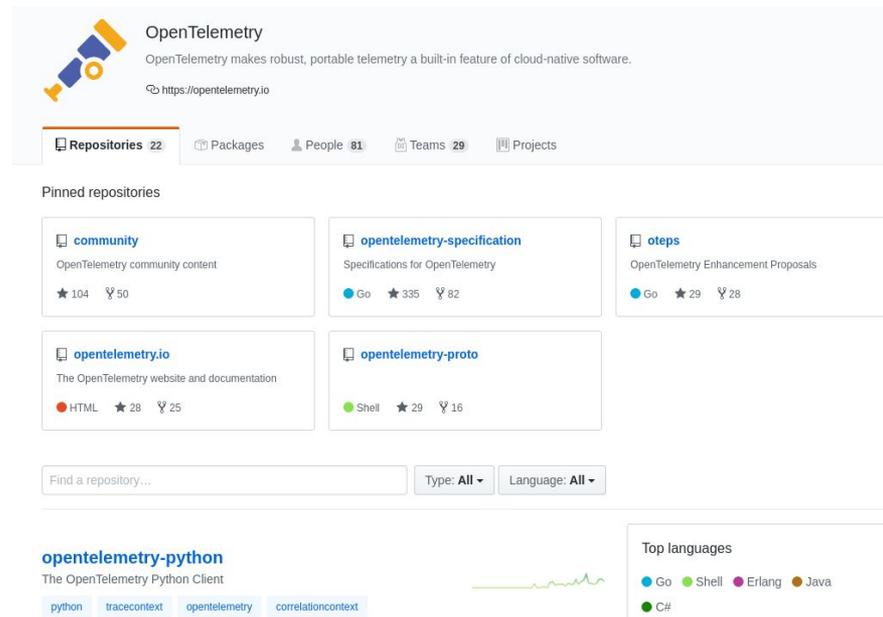
KubeCon



CloudNativeCon

North America 2019

- Take a look at the OpenTelemetry implementations for the languages you are using
- Try them out and send feedback
- Try the [OpenTelemetry collector](#) for collecting OpenTracing data
- Look at the available OT/OC shims



The screenshot shows the GitHub repository page for OpenTelemetry. At the top, there is a header for "OpenTelemetry" with a description: "OpenTelemetry makes robust, portable telemetry a built-in feature of cloud-native software." and the URL "https://opentelemetry.io". Below this, there are statistics for "Repositories 22", "Packages", "People 81", "Teams 29", and "Projects".

The "Pinned repositories" section displays four items:

- community**: OpenTelemetry community content, 104 stars, 50 forks.
- opentelemetry-specification**: Specifications for OpenTelemetry, 335 stars, 82 forks.
- opentelemetry.io**: The OpenTelemetry website and documentation, 28 stars, 25 forks.
- opentelemetry-proto**: 29 stars, 16 forks.

Below the pinned repositories is a search bar "Find a repository..." and filters for "Type: All" and "Language: All".

The "opentelemetry-python" repository is highlighted, showing it is "The OpenTelemetry Python Client" with tags for "python", "tracecontext", "opentelemetry", and "correlationcontext". To the right, a "Top languages" section shows a bar chart and a list of languages: Go, Shell, Erlang, Java, and C#.

Want to start contributing?



KubeCon



CloudNativeCon

North America 2019

- <https://opentelemetry.io/get-involved/>
- Find the OpenTelemetry repo you'd like to contribute to
- Fork it!
- Look for “good first issues” in the Issue tracker and update the issue to note that you are working on it
- Or just find something interesting to you or missing that you need

Filters ▾ Labels 18 Milestones 5 New issue

Clear current search query, filters, and sorts

<input type="checkbox"/>	🚨 1 Open ✓ 1 Closed	Author ▾	Labels ▾	Projects ▾	Milestones ▾	Assignee ▾	Sort ▾
<input type="checkbox"/>	Error logging with EventSource area: SDK good first issue up-for-grabs 16						
	#85 opened on Jun 3 by SergeyKanzhelev SDK v0.2 compl...						

Your org wants to give support



KubeCon



CloudNativeCon

North America 2019

- Meet us after the presentation
- Introduce yourself on gitter [open-telemetry/community](https://gitter.im/open-telemetry/community)



W3C Trace Context



KubeCon



CloudNativeCon

North America 2019

“Standards form the fundamental building blocks for product development by establishing consistent protocols that can be universally understood and adopted.”

OpenTelemetry makes telemetry a built-in feature of cloud-native software:

- best practices
- great libraries
- standards

Join W3C Distributed Tracing WG: <https://github.com/w3c/distributed-tracing-wg>

Trace Context

W3C Proposed Recommendation 21 November 2019



Non-code Participation



KubeCon



CloudNativeCon

North America 2019

There are many ways to participate including many non-code contributions:

- Share your story via blogs and events
- Help to reach out to more people
- Plan and triage work
- Do reviews, file issues and share feedback

Join our community!

M

 Sergey Kanzhelev
Oct 29 · 1 min read

Welcome new OpenTelemetry governance committee members

Last week we held our first OpenTelemetry governance meeting. Out of twelve amazing candidates representing the companies, cloud vendors, and APM providers, four were elected to represent...

Read more...

2

 Austin Parker
Oct 24 · 4 min read

Repositories 18 Packages 2 People 28 Projects

Pinned repositories

 OpenTelemetry OpenTelemetry community website	 opentelemetry-specification Specifications for OpenTelemetry
 opentelemetry.io The OpenTelemetry website and documentation	 opentelemetry-go

How To Start Contributing to Op

This post originally appeared on the LightStep blog.

Read more...

1

 Morgan McLean
Oct 10 · 3 min read

OpenTelemetry Monthly Update

Welcome to the fourth monthly OpenTelemetry community update. If you are new to the OpenTelemetry community, you should check out our [community mailing list](#) and subscribe to the shared calendar (web, gCal, iCal). We also have a very popular...

Read more...

15



The image shows a screenshot of the OpenTelemetry GitHub profile and a tweet. The GitHub profile includes the OpenTelemetry logo, the name "OpenTelemetry" with the handle "@opentelemetry", and a bio stating "OpenTelemetry makes robust, portable telemetry a built-in feature of cloud-native software." It also shows the repository "opentelemetry.io" and the date "Joined April 2019". The tweet, from the same account, is pinned and dated May 21. The tweet text reads "And a roadmap to the future:" and features a graphic with a target icon, a plus sign, and the OpenTelemetry logo. Below the graphic, the tweet text says "Merging OpenTracing and OpenCensus: A Roadmap to Convergence" and includes a link to a Medium article.

Tell us how you use OpenTelemetry



KubeCon



CloudNativeCon

North America 2019

OpenTelemetry has no way of knowing how you use the project.

Tell us about your scenarios:

- What environments you use it
- How do you use it, what do you like the most
- What's missing

This and other feedback you can share via various communication options:

Gitter: <https://gitter.im/open-telemetry/community>

GitHub: <https://github.com/open-telemetry/community>

E-mails: cncf-opentelemetry-community@lists.cncf.io

SIG and community meetings: [calendar](#)



Q&A



KubeCon



CloudNativeCon

North America 2019

Governance committee joins us for panel Q&A

- Ben Sigelman, LightStep
- Bogdan Drutu, Google
- Constance Caramanolis, Splunk
- Liz Fong-Jones, Honeycomb
- Morgan James McLean, Google
- Sarah Novotny, Microsoft
- Sergey Kanzhelev, Microsoft
- Ted Young, LightStep
- Yuri Shkuro, Uber



