

November 9-11, 2021  
[www.dotnetconf.net](http://www.dotnetconf.net)

# .NET Conf

Discover the world of .NET





- 微软最有价值专家 郝冠军

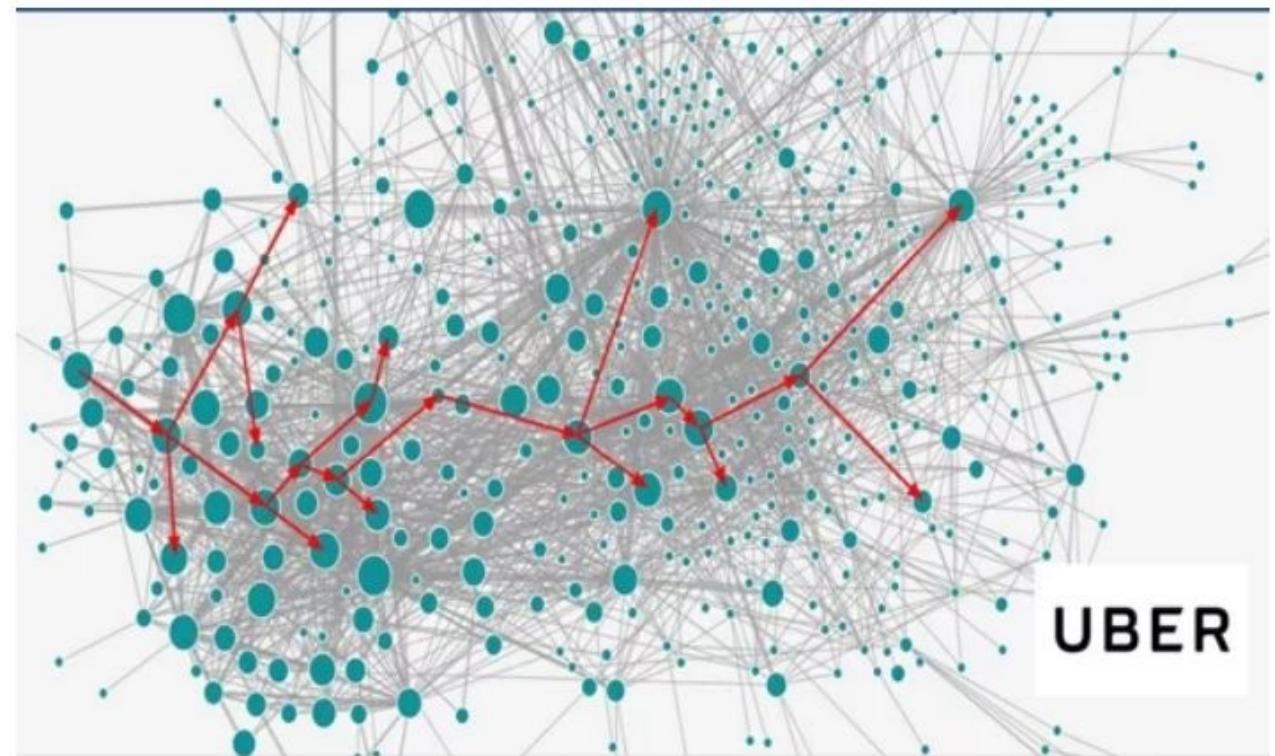


# 背景

- 2017.7 可观察性
- 云原生架构的特性
  - Modularity 模块化
  - Observability 可观察性
  - Deployability 可部署性
  - Testability 可测试性
  - Disposability 可处理性
  - Replaceability 可替换性

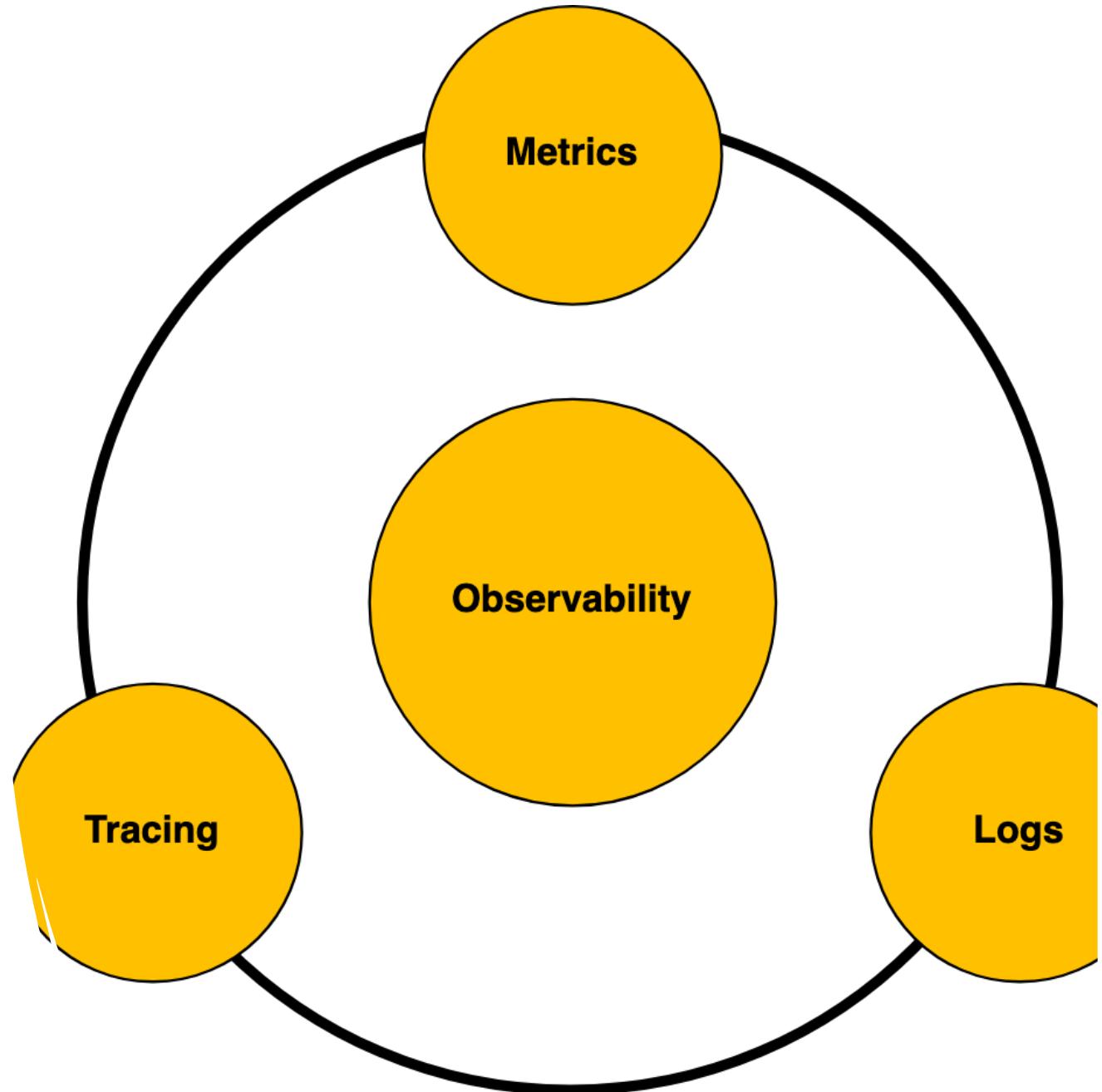
## 可观察性的三个维度

- 3 Dimensions
  - Tracing
  - Metrics
  - Logging



# 3 Pillars of Observability

---



- [The 3 Pillars of System Observability: Logs, Metrics, and Tracing - IOD \(iamondemand.com\)](#)

# Observability vs Monitoring

- Observability
  - Focus on Application
  - Developer
- Monitoring
  - Focus infrastructure
  - IT
- [Observability vs Monitoring: What's The Difference - THCBin Tech Blog](#)

# Metrics, logs, and traces

- Metrics
  - Aggregated summary statistics.
- Logs
  - Detailed debugging information emitted by processes.
- Distributed Tracing
  - Provides insights into the full lifecycles, aka traces of requests to a system, allowing you to pinpoint failures and performance issues.

Structured data can be transmuted into any of these!

# Tracing concepts in a nutshell

- Span
  - Represents a single unit of work in a system.
  - Typically encapsulates: operation name, a start and finish timestamp, the parent span identifier, the span identifier, and context items.
- Trace
  - Defined implicitly by its spans. A trace can be thought of as a directed acyclic graph of spans where the edges between spans are defined as parent/child relationships.
- DistributedContext
  - Contains the tracing identifiers, tags, and options that are propagated from parent to child spans

# Tracing concepts

trace

/messages

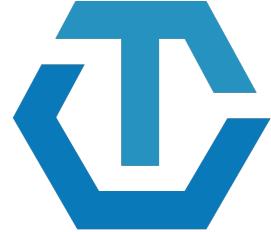
span

auth

cache.Get

mysql.Query

cache.Put



## OPENTRACING

- 370 Contributors
- 5360 stars on GitHub
- 100s of supported Integrations with OSS libraries and frameworks



## OpenCensus

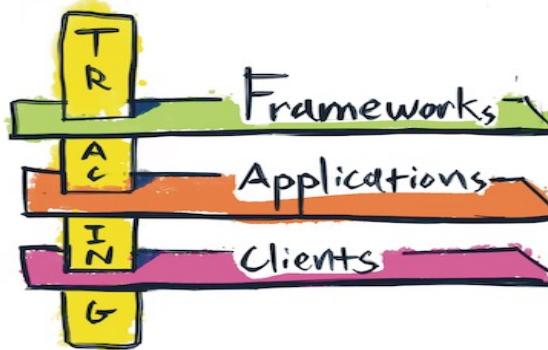
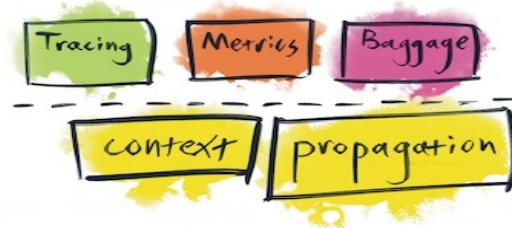
- 390 Contributors
- 3392 stars on GitHub
- Backed by Google, Microsoft, Omnitron, Postmates, Dynatrace

**OpenTracing and OpenCensus have merged form OpenTelemetry!**



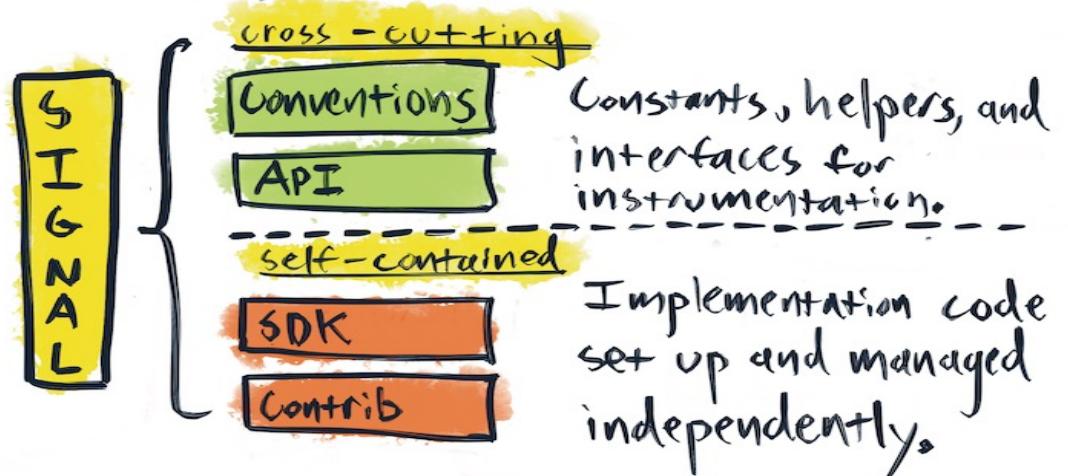
## OpenTelemetry Architecture

OpenTelemetry is designed as a set of independent observability tools, called Signals, which are built on top of a shared mechanism for context propagation.

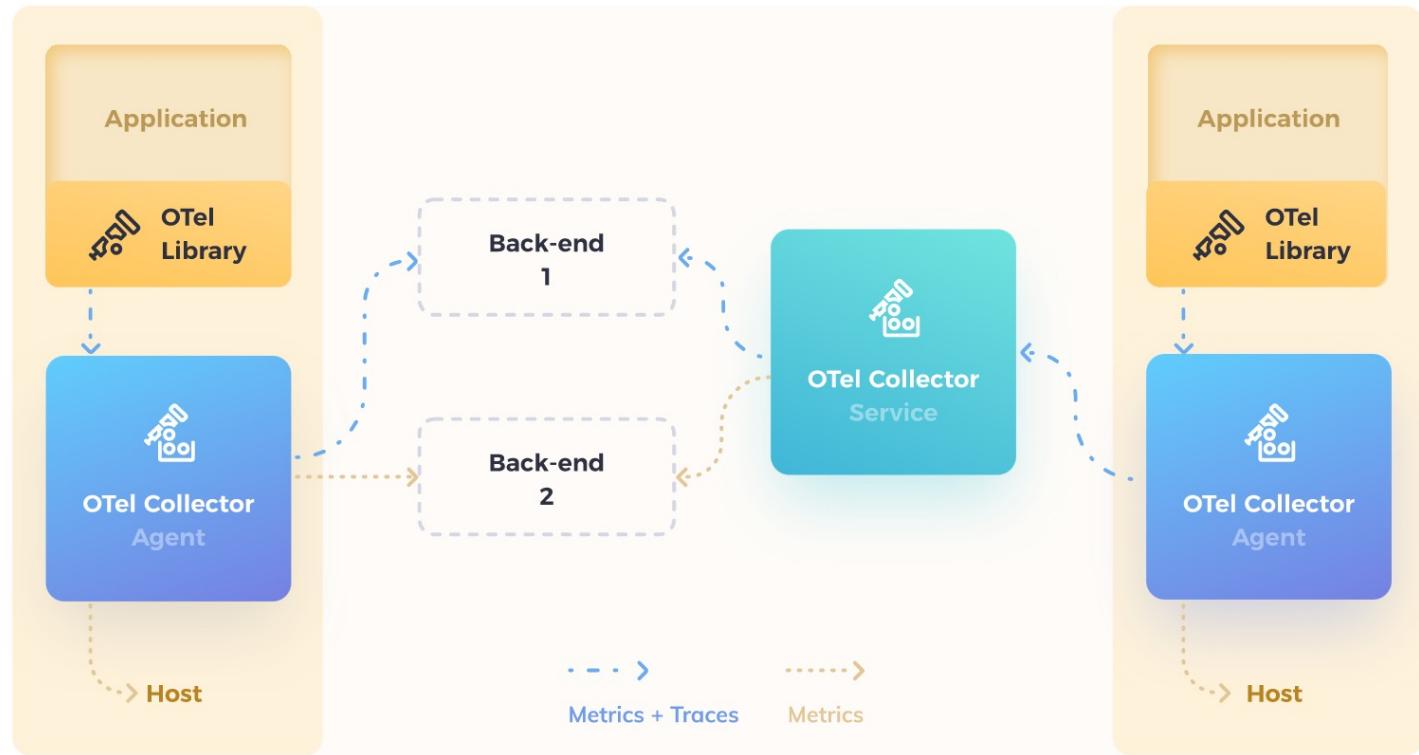


Signals work as cross-cutting concerns, mixed into many libraries.

The portion of each signal which is imported as a cross-cutting concern is kept separate from the portion which can be managed independently by the application owner.



# OpenTelemetry



## REFERENCE ARCHITECTURE

# OpenTelemetry .NET Status and Releases

- [OpenTelemetry .NET API](#)
  - The API only surfaces necessary abstractions to instrument an application/library
- [OpenTelemetry .NET SDK](#)
  - OpenTelemetry SDK is a reference implementation of the OpenTelemetry API.
- Status
  - Tracing : 1.1
  - Metrics : Alpha
  - Logging : Beta

See: <https://opentelemetry.io/docs/net/>

# Tracing

- Attribute
  - Available in current span.
- Baggage
  - Baggage can be propagated out of proc using [Propagators](#).
  - OpenTelemetry SDK ships a BaggagePropagator and enables it by default.
  - <https://github.com/open-telemetry/opentelemetry-dotnet/blob/main/src/OpenTelemetry.Api/README.md#baggage-api>
- Event
  - As a log in span

# OpenTelemetry.API vs .NET Activity API

- Span => Activity
- Attribute => Tag
  - Tags in Activity represents the OpenTelemetry Span Attributes.

# Instrumentations

- OpenTelemetry.Instrumentation.AspNet
  - OpenTelemetry.Instrumentation.AspNet.TelemetryHttpModule
- OpenTelemetry.Instrumentation.AspNetCore
- OpenTelemetry.Instrumentation.Http
- OpenTelemetry.Instrumentation.GrpcNetClient
- OpenTelemetry.Instrumentation.SqlClient
- OpenTelemetry.Instrumentation.StackExchangeRedis

# Exporters

- OpenTelemetry.Exporter.Console
- OpenTelemetry.Exporter.InMemory
- OpenTelemetry.Exporter.Jaeger
- OpenTelemetry.Exporter.OpenTelemetryProtocol.Logs
- OpenTelemetry.Exporter.OpenTelemetryProtocol
- OpenTelemetry.Exporter.Prometheus
- OpenTelemetry.Exporter.Zpages
- OpenTelemetry.Exporter.Zipkin

# Demo

End