OpenTelemetry or OpenTelemessy:

Solving Observability Problems While Creating New Ones

Narendra Reddy Sanikommu

Senior Software Engineer, Nvidia

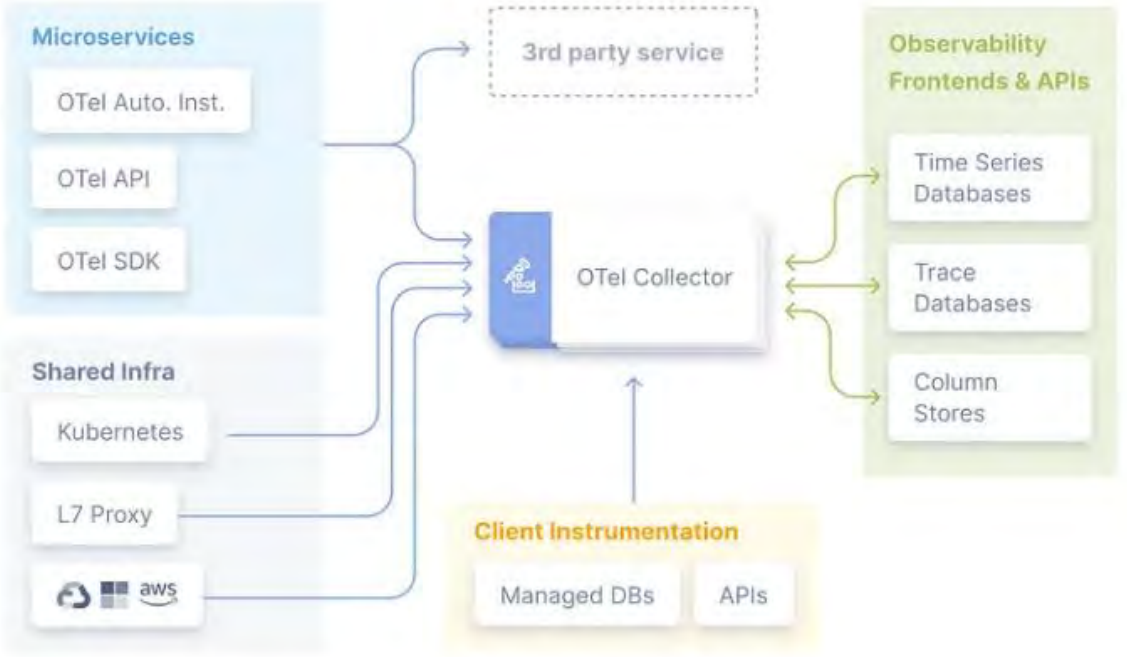
Agenda

- *Open Telemetry Intro*
- *The Promise - Core Problems Open Telemetry Addresses*
- *The Reality - A Complex Beast*
- *When Does OpenTelemetry Make Sense?*
- *Strategies for Taming the Telemessy*
- *Future Outlook - 2025 and Beyond*

Introduction

Introduction

- OpenTelemetry is a collection of APIs, SDKs, and tools to instrument, generate, collect, and export telemetry data (metrics, logs, and traces)



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

The Promise - Core Problems Open Telemetry Addresses

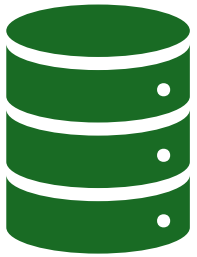
The Promise - Core Problems OTel Addresses

Vendor Lock-in Elimination

Pre-OTel: Proprietary agents, SDKs, and APIs

OTel: Vendor-neutral approach

Instrument once, export anywhere

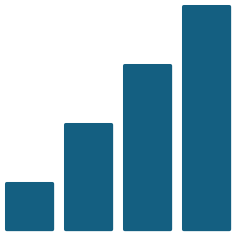


Unified Standards

Merged OpenTracing and OpenCensus

Standardized telemetry collection across languages

The Promise – comprehensive approach



Complete Signal Coverage

Unified support for metrics, traces, and logs

Correlation between signals for deeper insights

Separation of Concerns

Collection decoupled from analysis

OTel Collector: powerful aggregator/forwarder

Choose any compatible backend

The Promise – Developer Experience

Auto-Instrumentation

- Reduced manual instrumentation burden
- Automatic capture of HTTP requests, DB queries
- Support for multiple languages

Future-Proof Foundation

- Modular design for emerging needs
- Broad community adoption (2nd largest CNCF project)
- Active development roadmap

The Reality - A Complex Beast

The Reality - A Complex Beast

Terminology Maze

Overwhelmingly technical lexicon

Multiple layers of abstraction

Too many layers of indirection

Architecture Complexity

Context propagation

Provider components

Exporters, processors, SDKs, collectors

Each with unique configurations

The Reality –Implementation Challenges

Language Implementation Inconsistency

Varying quality and feature support across languages

Challenges in polyglot environments

Documentation: comprehensive but overwhelming

Maturity Spectrum

Different components at varying maturity levels

Metrics and logging still developing

"Getting started" cited as top barrier to adoption

The Reality – Troubleshooting Quagmire

Multi-Layer Debug Nightmares

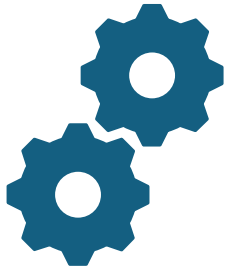
- Problems across instrumentation, collection, export
- Issues spanning multiple components

Collector Configuration Hell

- Complex YAML configurations
- Mismatched component names
- Authentication and connectivity issues
- "Try to file an issue and you'll be bounced between at least three repos"

When Does Open Telemetry Make Sense?

when does Open Telemetry Make Sense?



Large, Diverse Technology Stacks

Standardized telemetry across languages/frameworks

Value exceeds implementation complexity

Strategic Vendor Independence

Freedom from proprietary monitoring solutions

Flexibility in choosing observability providers

Risk reduction for long-term investments

when does Open Telemetry Make Sense?

Cloud-Native Environments

Ideal for distributed systems

Monitoring containerized applications

Kubernetes integration

Data Flexibility Requirements

Control over telemetry data collection and routing

Filtering capabilities to reduce noise/costs

Custom tags for organization and searching

Strategies for Taming the Telemessy

Strategies for Taming the Telemessy

Start with Auto-Instrumentation

Begin with language-specific auto-instrumentation

Use OpenTelemetry EKS Add-on for Kubernetes

Focus first on tracing, then expand to metrics and logs

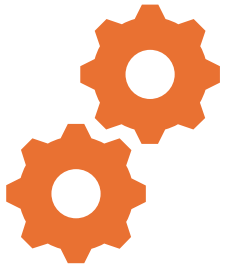
Follow Latest Best Practices

Leverage semantic conventions

Optimize attribute usage

Use local exporters for troubleshooting

Strategies for Taming the Telemessy



Iterative Implementation

Start small with single services

Focus on critical user flows

Expand gradually as expertise grows

Community Resources

Look beyond official documentation

Vendor distributions can simplify adoption

Active community support and extensions

Future Outlook - 2025 and Beyond

Future Outlook – 2025 and Beyond

Promising Developments

- Semantic conventions reaching stability
- OpenTelemetry Collector approaching v1.0
- Profiling signal advancing
- GenAI observability integration

Ongoing Challenges

- Documentation remains problematic
- Complexity management
- Standardizing edge cases

Thank You