

**The return of...**

# **What we talk about when we talk about Observability Frameworks**

Timothy Mahoney

Open Source Observability Day 2024



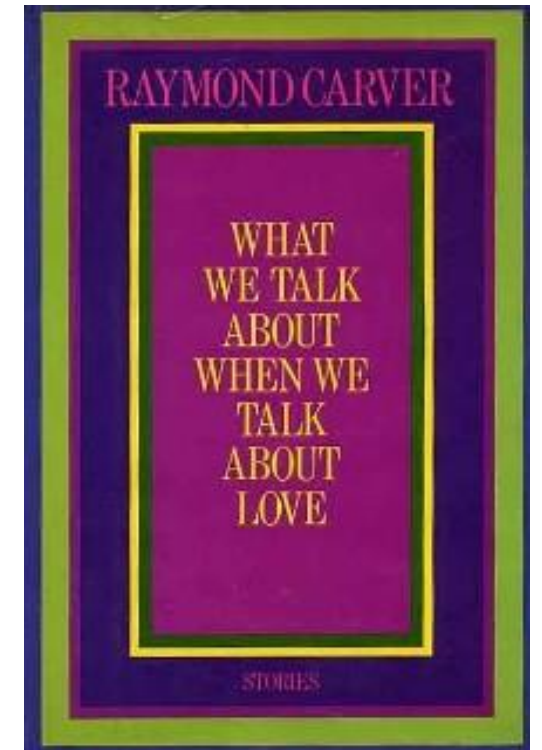
# Who am I?

Timothy Mahoney  
Senior Systems Engineer  
Observability Enablement Team  
IKEA Retail



# Trying to be clever with the title.

Haruki Murakami  
Raymond Carver



# Where we started

- Tooling Dependant
- Consistency Issues
- Lineage and Ownership Issues
- Taxonomy Issues
- Conflicting Standards and Requirements



# The Goal

An Observability Framework that:

- Tool Agnostic
- Embraces Open Source
- Lower Cognitive Load
- Encourages Collaboration



# Groundwork

Cognitive overload  
Documentational debt  
Technical debt



# Contributors

Team Discussion  
Observability Slack  
Workshops  
Forums  
Conferences

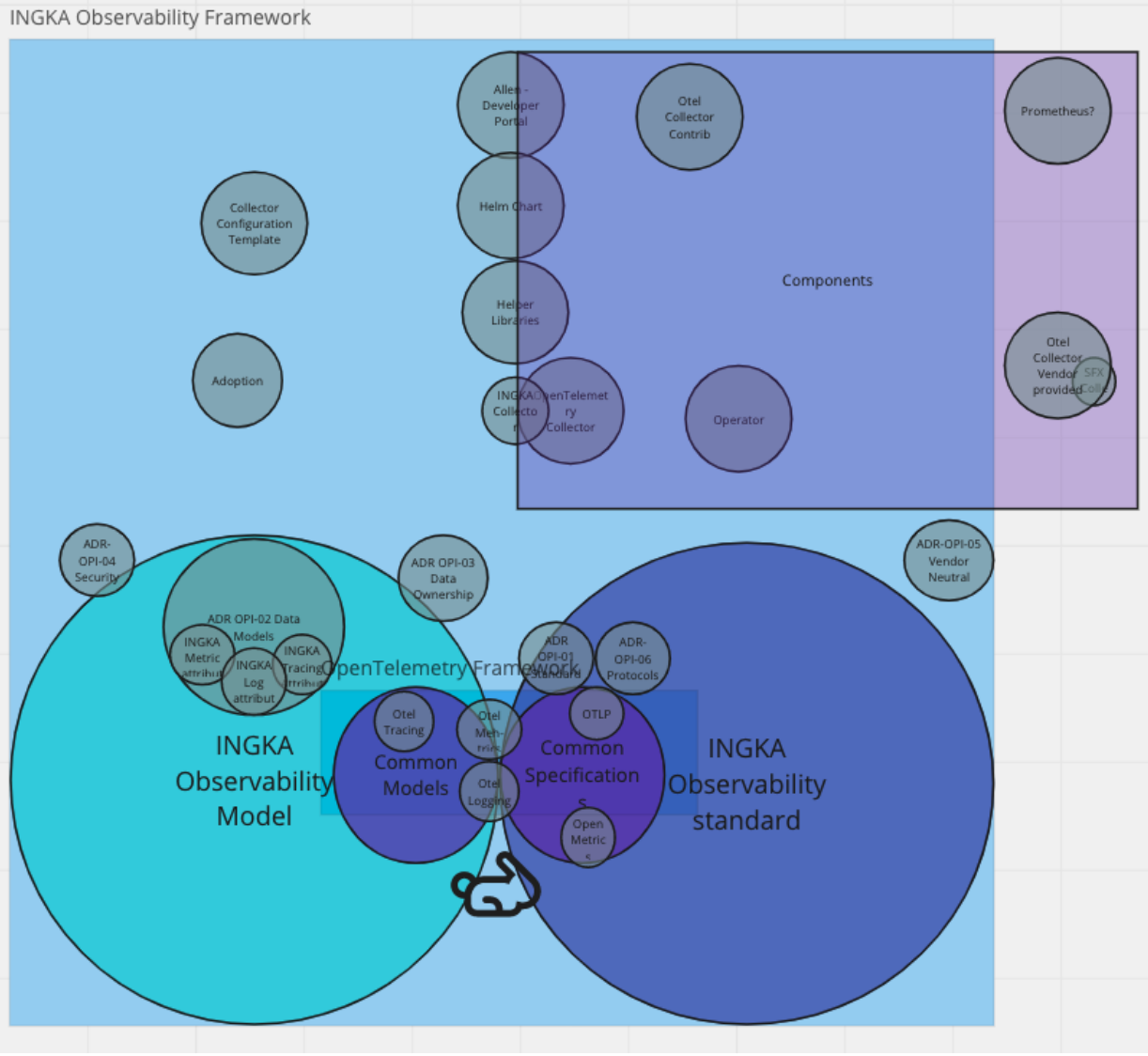


# The Detractors



The cult of the dumpster fire emoji.





# Do we even know what we are talking about?

- Don't assume a common understanding of concepts.
- Don't be afraid to ask trivial questions.
- What do users actually want?

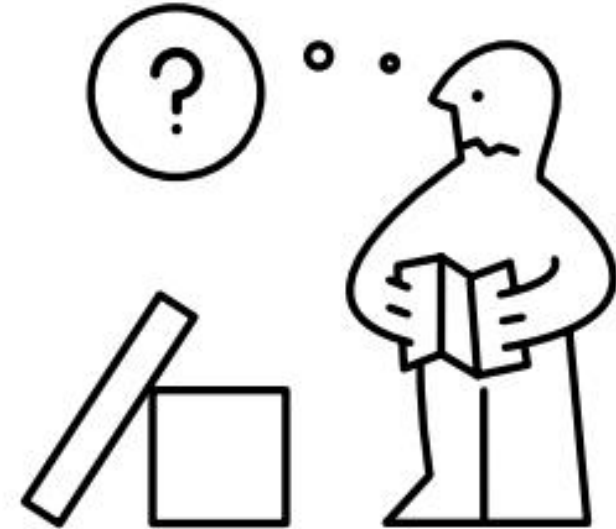
# Are we understood?

Language barriers

Modal Verbs RFC2119

Vocabulary != Authority

Not good use of ChatGPT



# How and where do we collaborate?

- Silence is a terrible form of consensus.
- Creating Community is Critical



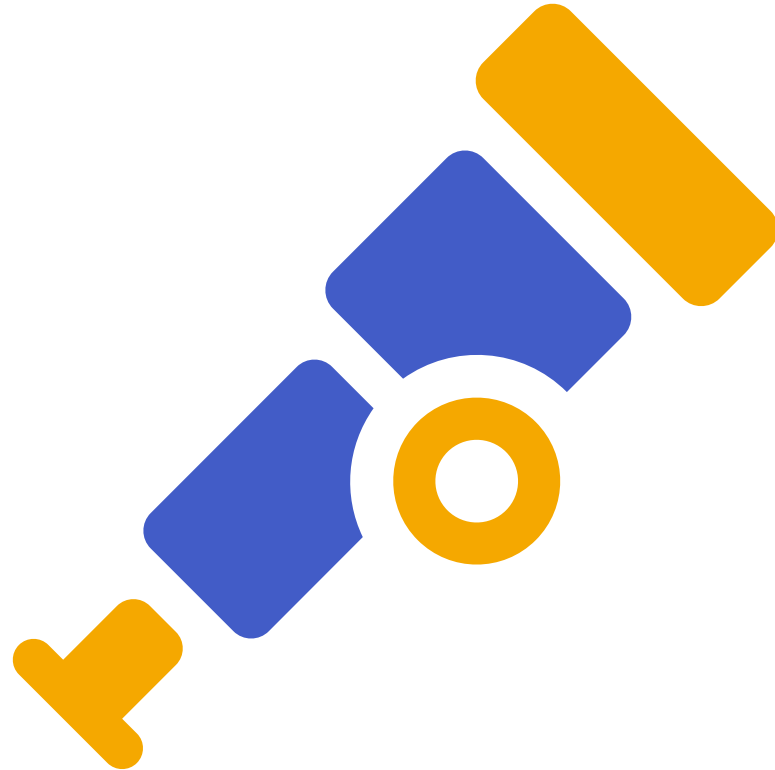
# Do we really need a Framework?

- Engineers are going to solve the problems that are here and now
- Tool sprawl
- Knowledge gaps



# We Love OpenTelemetry

Data Model  
Specifications  
Components  
OTEPs



# Components

OpenTelemetry Components  
IKEA Specific Components



# Standards

CNCF is not a standards institution

Defined in ADRs

Extension of Common Industry Standards



# Data Model

- Ensuring Enterprise-wide consistency in observability signals.
- Unambiguous Mapping

OTLP Field	GCP <small>(googlecloudexporter)</small>	Azure <small>(azuremonitorexporter)</small>	AliCloud <small>(alibabacloudlogserviceexporter)</small>	Splunk <small>(splunkhecexporter)</small>	Loki <small>(lokiexporter)</small>
Timestamp	timestamp ISO 8601 2022-10-27T13:59:38.312892Z	timestamp [UTC] ISO 8601 2022-10-27T13:59:38.312892Z	timeUnixNano Unix Time (ns) 1666879178312952000	time Unix Time (s with decimal) 1666879178.313	tsNs Unix Time (ns) 1666879178312952000
TraceId	trace <small>TraceId mixed with project info projects/{projectname}/traces/{traceid}</small>	operation_id	traceID	trace_id	traceid
SpanId	spanId	operation_ParentId	spanID	span_id	spanid
SeverityText	severity	n/a	severityText	otel.log.severity.text	severity
SeverityNumber	n/a	n/a	severityNumber	otel.log.severity.number	n/a
Body	textPayload	message	content	event	body
Resource	Controlled by exporter*	n/a	resource	fields**	resource





# What is the minimum viable set of attributes for an atomic unit of observability?

## Specific Attributes:

- Deployment Environment
- Service Version
- Domain
- Legal Company  
Sub Domain
- System Name
- Team Name
- Framework Version



# Insuring Adoption

- Users and Teams want to be shown solutions rather than possibilities.
- Clear benefit to teams and company
- Attributes mandatory on ingestion



```

;
g, DiagConsoleLogger, DiagLogLevel } = require('@opentelemetry/api');

lemetry = require('@opentelemetry/sdk-node');

  postgres and http instrumentation
eTracerProvider } = require('@opentelemetry/sdk-trace-node');
isterInstrumentations } = require('@opentelemetry/instrumentation');
pInstrumentation } = require('@opentelemetry/instrumentation-http');
ressInstrumentation } = require('@opentelemetry/instrumentation-express');

r trace exporter
ource } = require('@opentelemetry/resources');
anticResourceAttributes } = require('@opentelemetry/semantic-conventions');
pleSpanProcessor } = require('@opentelemetry/sdk-trace-base');
PTraceExporter } = require('@opentelemetry/exporter-trace-otlp-http');
gerExporter } = require('@opentelemetry/exporter-jaeger');

ger(new DiagConsoleLogger(), DiagLogLevel.DEBUG);

rovider
der = new NodeTracerProvider({
  new Resource({ [SemanticResourceAttributes.SERVICE_NAME]: 'warehouse_app',
ResourceAttributes.DEPLOYMENT_ENVIRONMENT]: 'development',
ResourceAttributes.INGKA_LEGAL_COMPANY]: 'ingka',
ResourceAttributes.INGKA_SYSTEM_NAME]: 'warehouse',
ResourceAttributes.INGKA_VERSION]: 'v2.0.0',

trumentations({
tations: [
ently to be able to have auto-instrumentation for express
eed the auto-instrumentation for HTTP.
ttpInstrumentation(),
ressInstrumentation(),
pInstrumentation({

```

# Examples

Give teams a place to collaborate and share examples.

Include Framework attributes in the examples.



# Demos and Labs

Demonstrate to teams how using Observability and the Observability Framework will solve problems.

# Where are we heading?



# Engineering Baseline

Observability and Monitoring as a Practice and not a Checkbox



# Challenges

- Bought Services and SaaS
- Adopting a Consistent Event Structure
- Frontend Observability and Privacy
- Legacy Systems



# Observability as a Data Product

- Ingka attributes correlate to other business data.
- System information can be shared as a data product.





# Data Observability

- Freshness
- Quality
- Value
- Schema
- Lineage



# Conclusions

- Make the framework something teams want to adopt.
- Ensure your consumers are invested in collaboration.
- Don't present your case in a way that can be easily dismissed in an emoji
- Face your imposter syndrome

