OTel?

Oh, tell!

Phillip

Vorathep Sumetphong

Tech Specialist bitkub

Open Source Contributor

My background

agoda

Silicon Valley Insight

bitkub

Software Development, Today

- Software Development has changed (again!)
- Some Code vs No Code
- No need to provision servers or wire up network

What was the shortest amount of time taken to build a software?

What was the longest software you have ever operated?

Somethings don’t change

Deploy apps != Operate

- More complex applications
- Harder to pinpoint issues

What we still need

- Understand application performance from user’s point of view
- Understand what/how much resources are being consumed

We still need ‘observability’

My Goal, Today

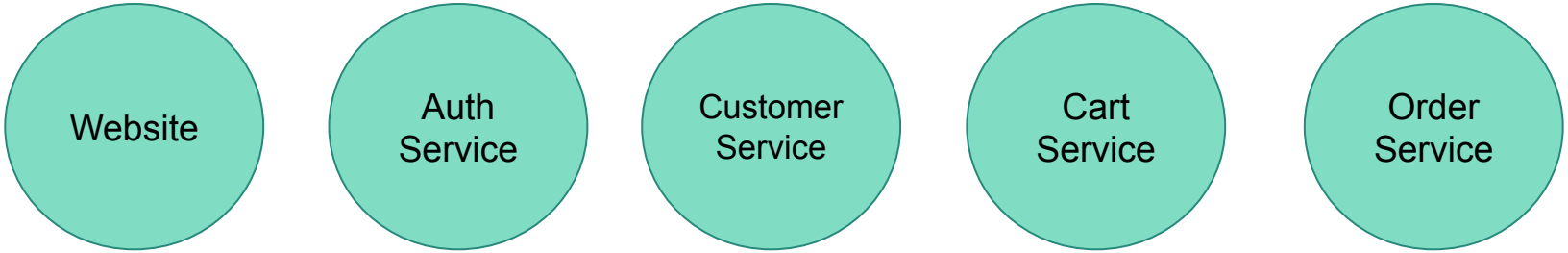
- Understanding Observability
- Three Pillars vs Single Braid Paradigms
- OpenTelemetry

What are we observing?

Transactions

A transaction represents all the actions a distributed systems needs to execute in order for the service to do something useful.

Checkout Transaction

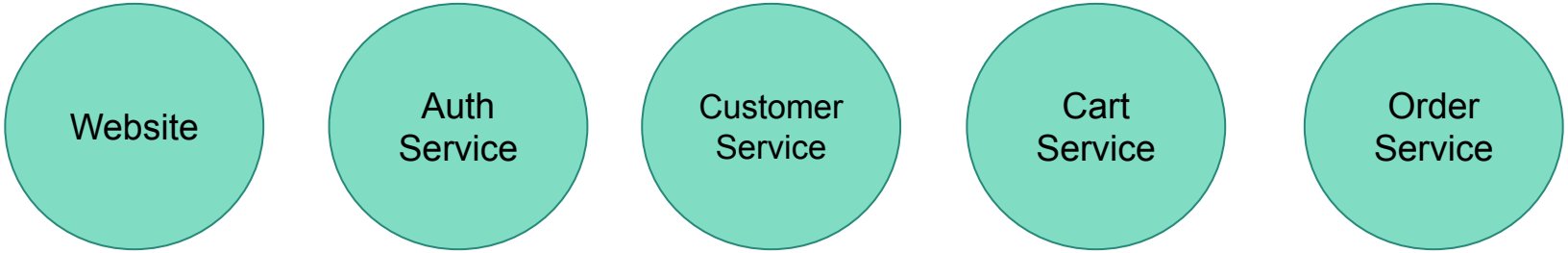


Resources

Transactions use up Resources.

- Database can be locked
- Services an only handle many concurrent requests

Checkout Transaction

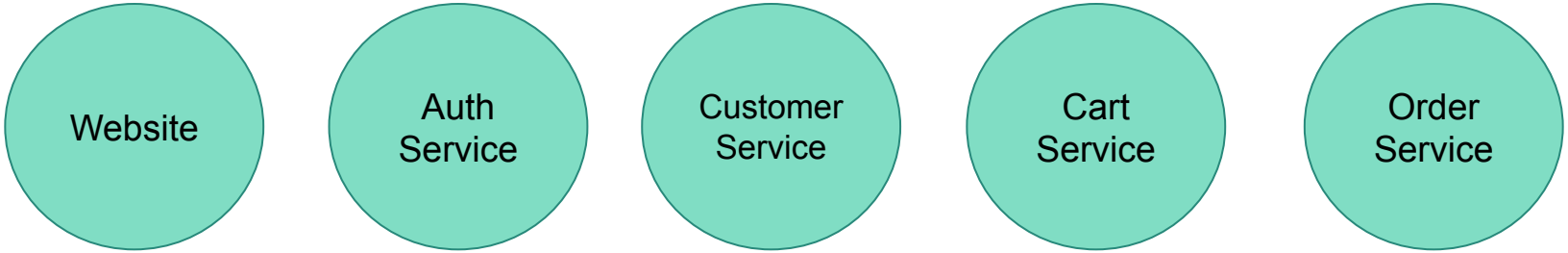


Real World Scenario

Noc Team: Users are not able to checkout!

Urgent fix is required! What to do?

Checkout Transaction



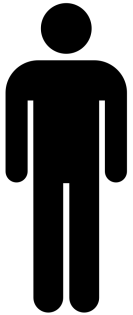
Three Pillars of Observability

Logging
Recording the individual events that make up a transaction.

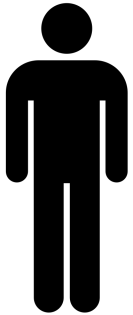
Metrics
Recording the individual events that make up a transaction.

Tracing
Measuring the latency of operations and identifying performance bottlenecks in a transaction.

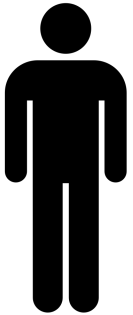
Three Browser Tabs



Person A
I want to understand the transactions that their program is executing.
I will add logs



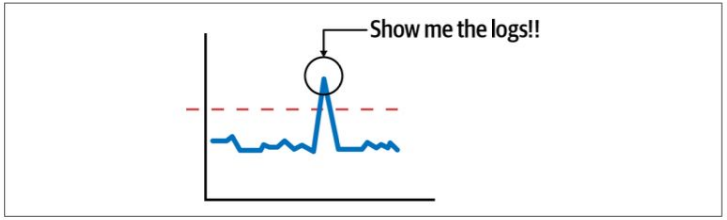
Person B
I want to monitor the resources and also capture metrics, but I don't want logs of Person A

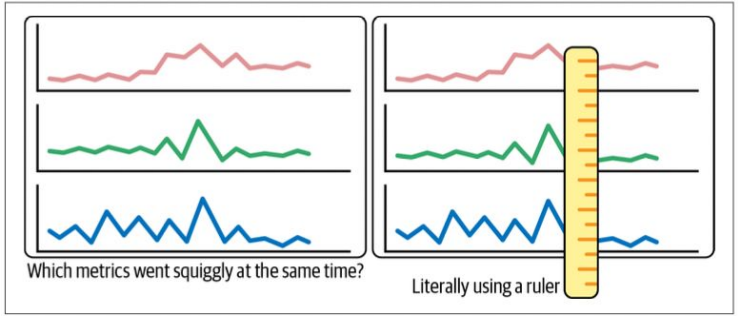


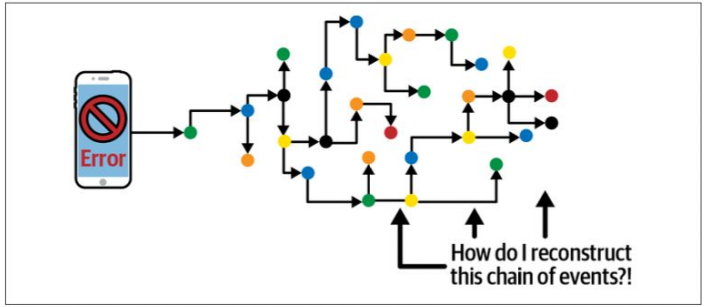
Person C
I want to find bottlenecks of my code, those logs of Person A don't work, I will use some third party lib!

Three Browser Tabs

Noc Team: There's a bug!







That's, not fun.

What if we could identify *how*,
not *what* is changing?

~~Three Pillars~~ Single Braid

- Logs linked with transactions
- Metrics linked with logs
- Each data-point linked with resources

End Result: A single traversable graph containing all the data.

Value of Structured Data

To build better tools, we need better data

Telemetry must have two qualities to support high-quality automated analysis:

- All data points must be connected in a graph with proper indexing.
- All data points that represent common operations must have well-defined keys and values.

Structured Data: Attributes

Key value for all data structures.

Set of conventions given.

```
{
  "http.method": "GET",
  "http.target": "path/123",
  "http.host": "www.example.com",
  "http.scheme": "https",
  "http.status_code": 200
}
```

Structured Data: Events

- An Event is a timestamp with a set of attributes.

Static Context

Decided when program starts
Eg. example-service

Dynamic Context

Decided through the transaction
Eg. http.method

```
{
  "http.method": "GET",
  "http.target": "path/123",
  "http.host": "www.example.com",
  "http.scheme": "https",
  "http.status_code": 200,
  "service.name": "example-service",
  "service.id": "1231",
  "service.version": "1.1",
  "timestamp": "12/12/1995 14:15:11 UTC",
  "message": "an event"
}
```

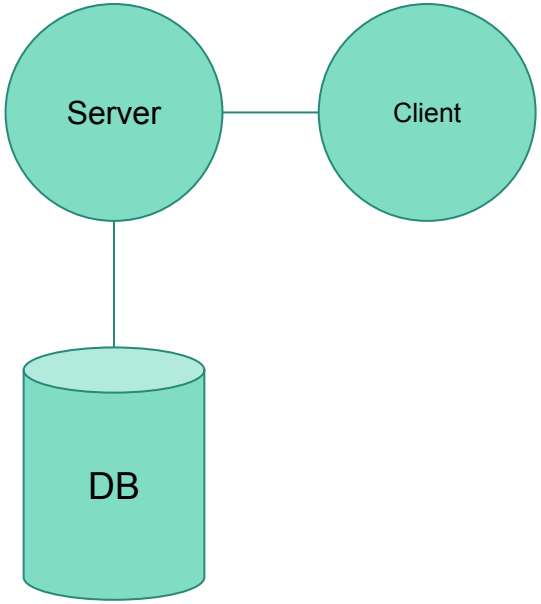
Structured Data: Resources

Observing Services and Machines

Static Context

Resources describe the physical and virtual infrastructure that a program is consuming.

What are resources?
Services, containers, deployments,regions & ++

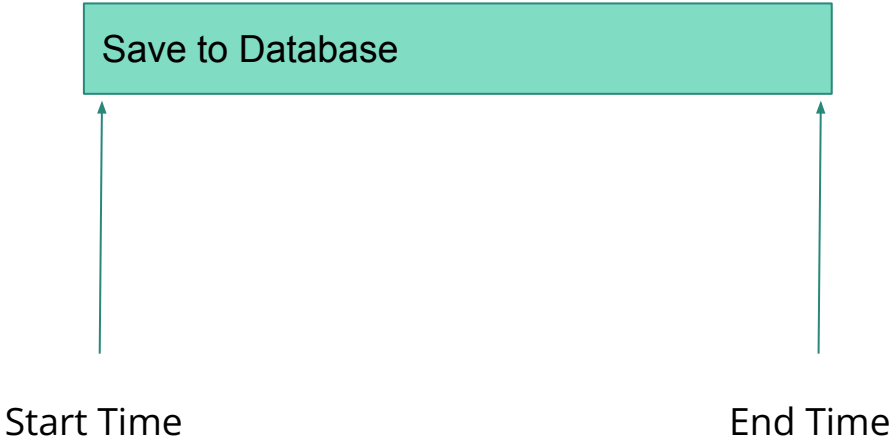


Structured Data: Spans

Observing Transactions

Dynamic Context

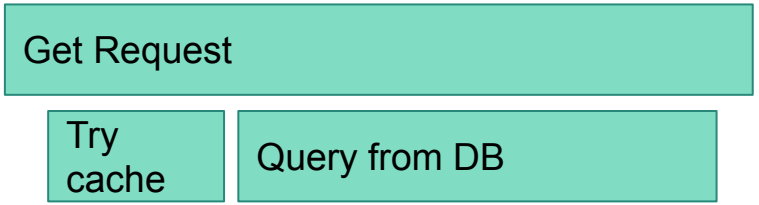
A span has an operation name, start time, a duration, and set of attributes



Structured Data: Tracing

Like logging but only better

A trace is a graph with organized spans associated with resources.



Structured Data: Metrics

- Aggregated Events

Visualized as

- Gauge
- Histogram
- Count
- Threshold

Auto Analysis

Automated correlation analysis

Some examples

- Extreme latency is highly correlated with kafka.node = 6
- Increase error rate is highly correlated with version = 1.3
- Traffic spike is highly correlated with username = phillip

Auto Analysis during war room

Intuition is a problem

- Misguided
- Missing understanding

Automated Correlation detection

- Machines detect potential problems
- Test hypothesis that the machine detects

How to implement Single Braid?

Oh tell, OpenTelemetry (Otel)

OpenTelemetry is a set of APIs, SDKs, tooling and integrations that are designed for the creation and management of telemetry data such as traces, metrics and logs.

Benefits

- Vendor Agnostic instrumentation
- Collector Binary
- E2E implementation to generate, emit, process and export telemetry data
- Full control of data with ability to send to multiple destinations parrelly through configuration
- Open Standard configurations

Otel is not observability backend like Jaegar or Prometheus.

The Single Braid

Events

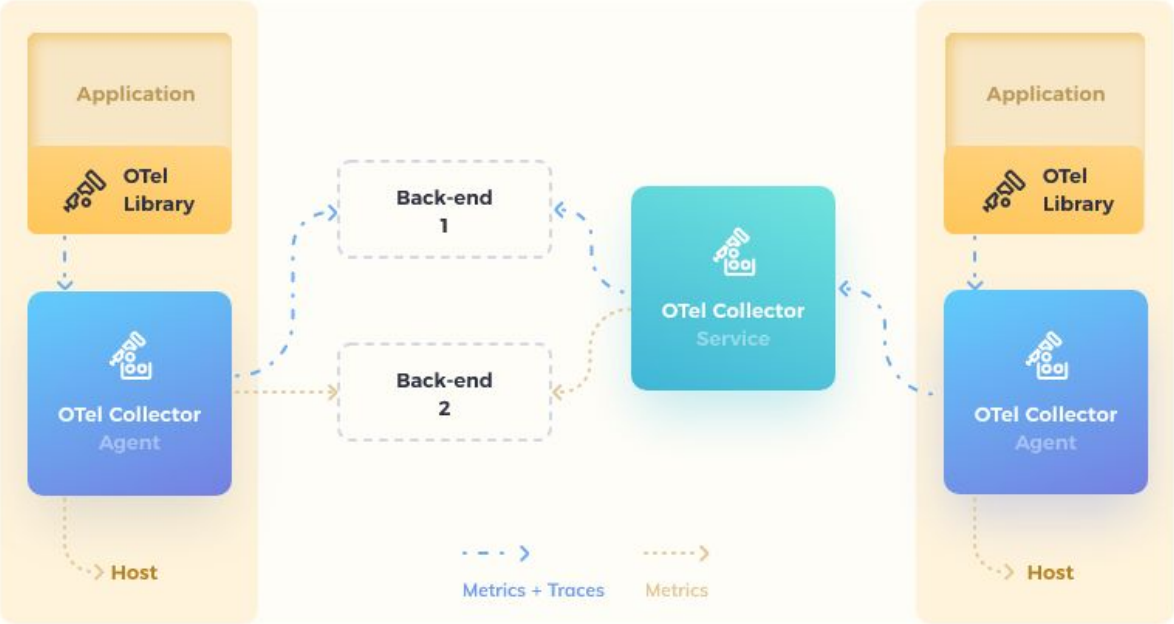
Span

Resources

Metrics

Traces

Otel: Architectural Overview



Understanding Collector

Some Demo

Q&A

Can be about anything

Do you have any feedback or questions?

vorathep055@gmail.com