Monitorama

Disintegrated Telemetry

The pain of monitoring asynchronous workflows

Johannes Tax

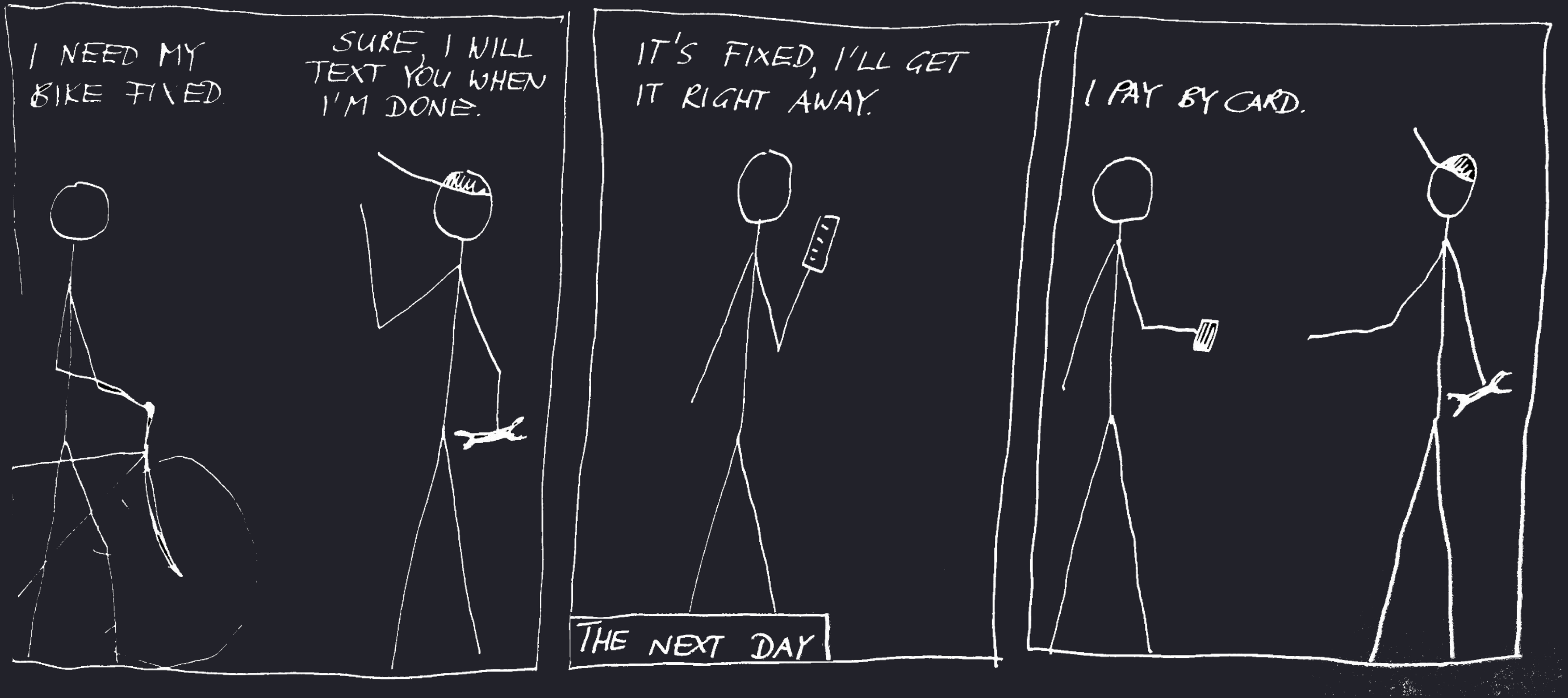
Principal Software Engineer at Grafana Labs

About this talk

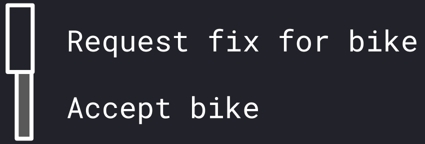
- Asynchronous workflows
  - Asynchronous communication between different services
  - Messaging and eventing
- Disintegration
  - the process of losing cohesion or strength
- Telemetry & Monitoring
  - Distributed Tracing
- Pain
  - an unpleasant sensory and emotional experience

Asynchronous workflows	

Example - Asynchronous workflow

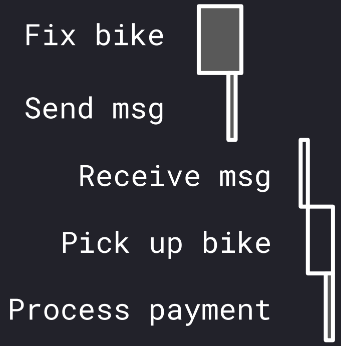

I NEED MY BIKE FIXED
SURE, I WILL TEXT YOU WHEN I'M DONE.
IT'S FIXED, I'LL GET IT RIGHT AWAY.
THE NEXT DAY
I PAY BY CARD.

Example - Asynchronous workflow



Request fix for bike

Accept bike



Fix bike

Send msg

Receive msg

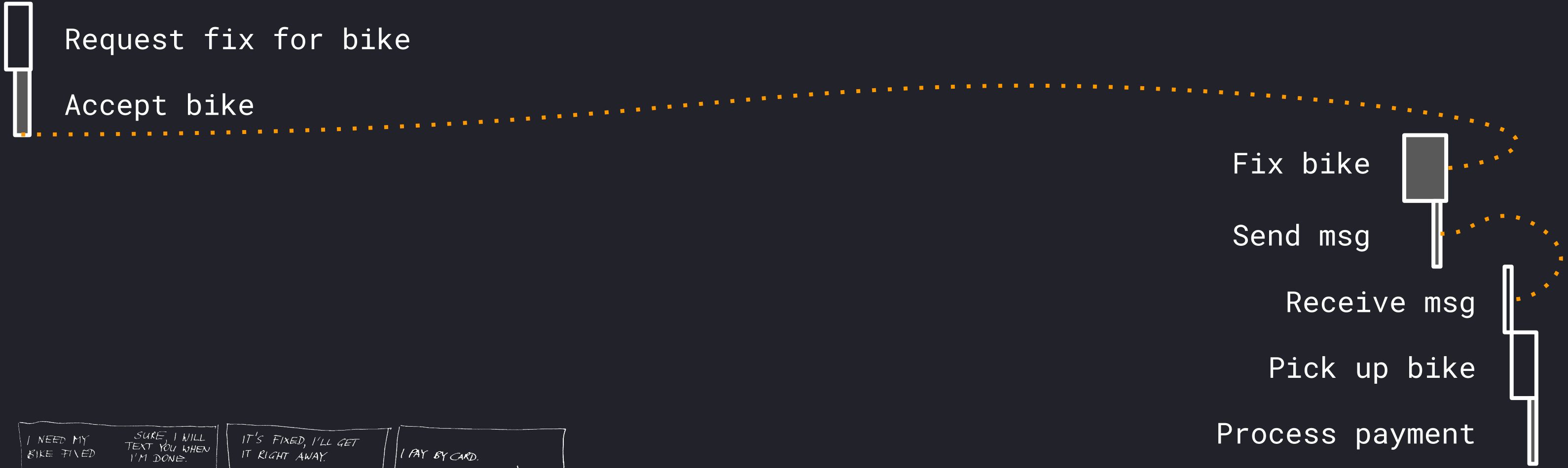
Pick up bike

Process payment

I NEED MY BIKE FIXED
SURE, I WILL TEXT YOU WHEN I'M DONE.
IT'S FIXED, I'LL GET IT RIGHT AWAY.
THE NEXT DAY
I PAY BY CARD.

Asynchronous workflows	Disintegration

Temporal decoupling



Request fix for bike

Accept bike

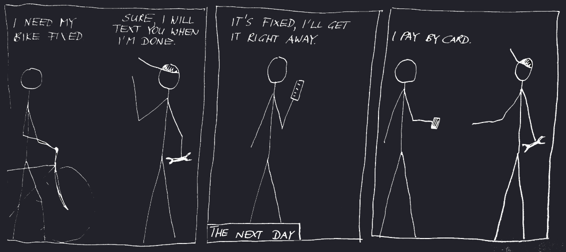
Fix bike

Send msg

Receive msg

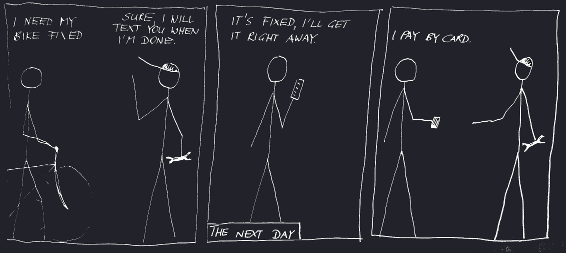
Pick up bike

Process payment

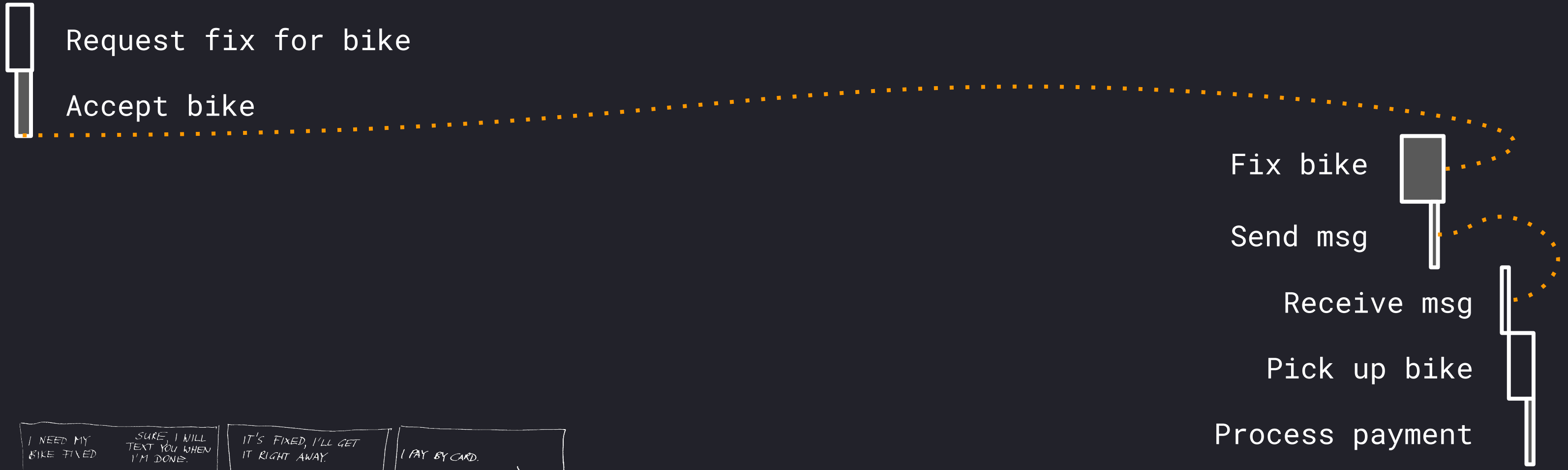

I NEED MY BIKE FIXED
SURE, I WILL TEXT YOU WHEN I'M DONE.
IT'S FIXED, I'LL GET IT RIGHT AWAY.
THE NEXT DAY
I PAY BY CARD.

Temporal decoupling

- Producers and consumers
  - Aren’t restricted by each other’s availability
  - Don’t have to run concurrently
- Increases reliability and resilience


I NEED MY BIKE FIXED
SURE, I WILL TEXT YOU WHEN I'M DONE.
IT'S FIXED, I'LL GET IT RIGHT AWAY.
THE NEXT DAY
I PAY BY CARD.

Temporal decoupling causes disintegration



Request fix for bike

Accept bike

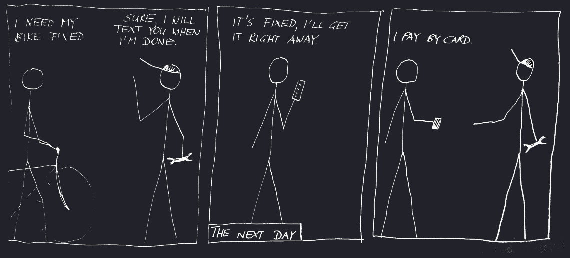
Fix bike

Send msg

Receive msg

Pick up bike

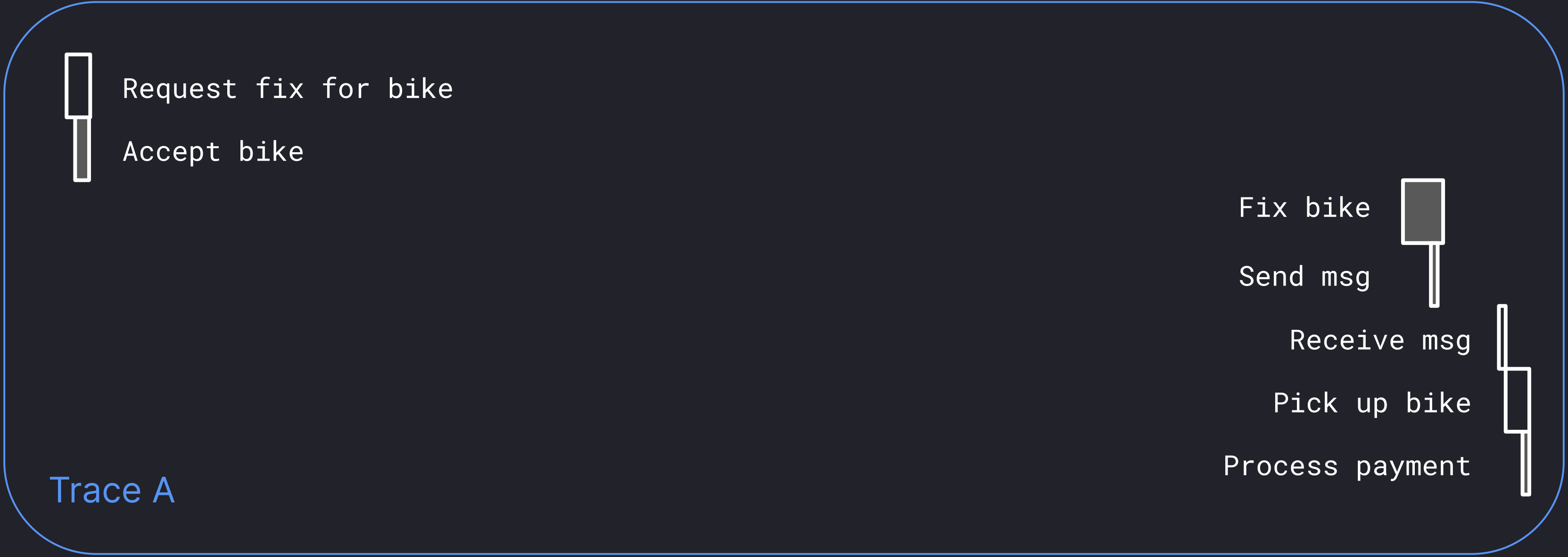
Process payment


I NEED MY BIKE FIXED.
SURE, I WILL TEXT YOU WHEN I'M DONE.
IT'S FIXED, I'LL GET IT RIGHT AWAY.
THE NEXT DAY
I PAY BY CARD.

Monitoring	
Asynchronous workflows	Disintegration

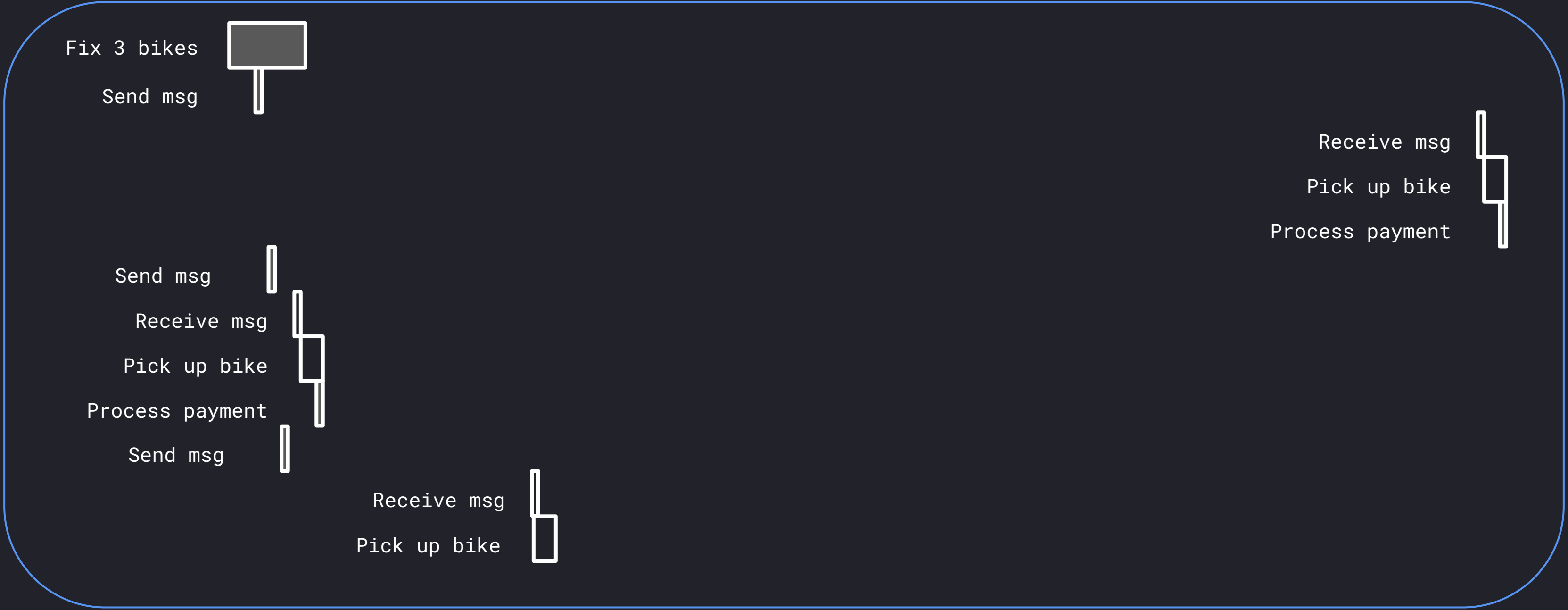
How to model an asynchronous workflow?

As a single trace:



How to model an asynchronous workflow?

As a single trace:

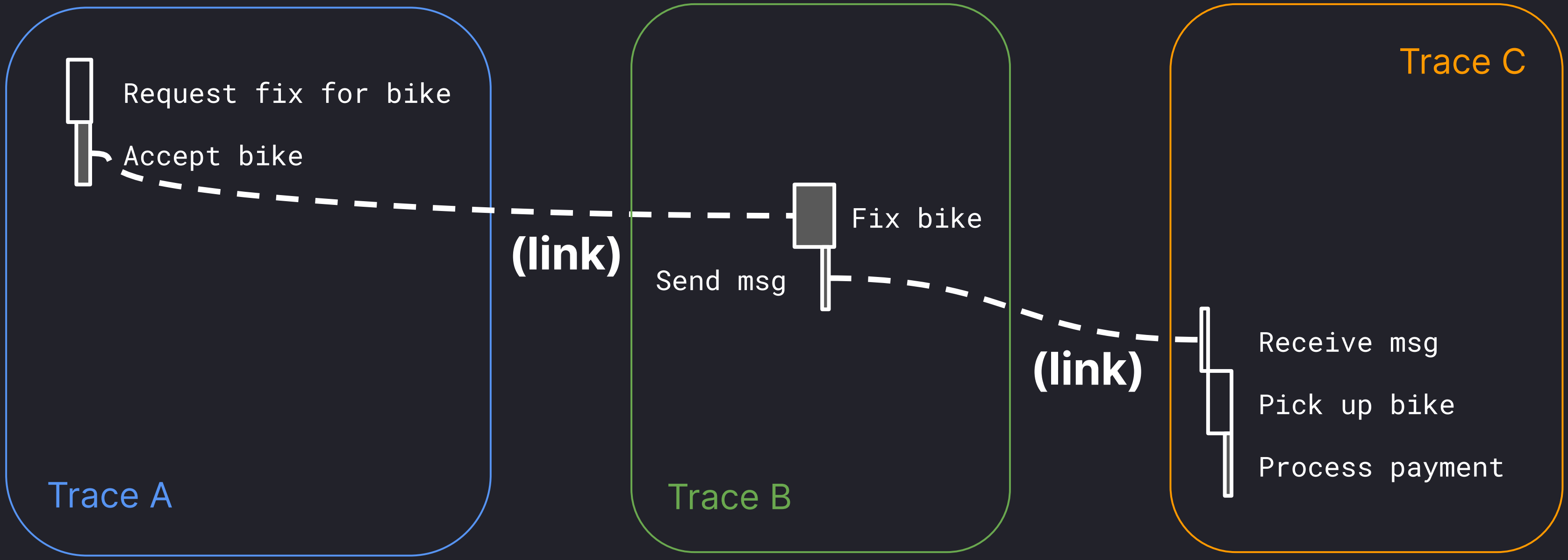


Span relationships

- Parent-child relationship
  - 1:n relationship
  - One span can have only one parent
  - Constitutes a trace
- Span links
  - n:n relationship
  - Any span can have any number of links
  - Relates spans from different traces

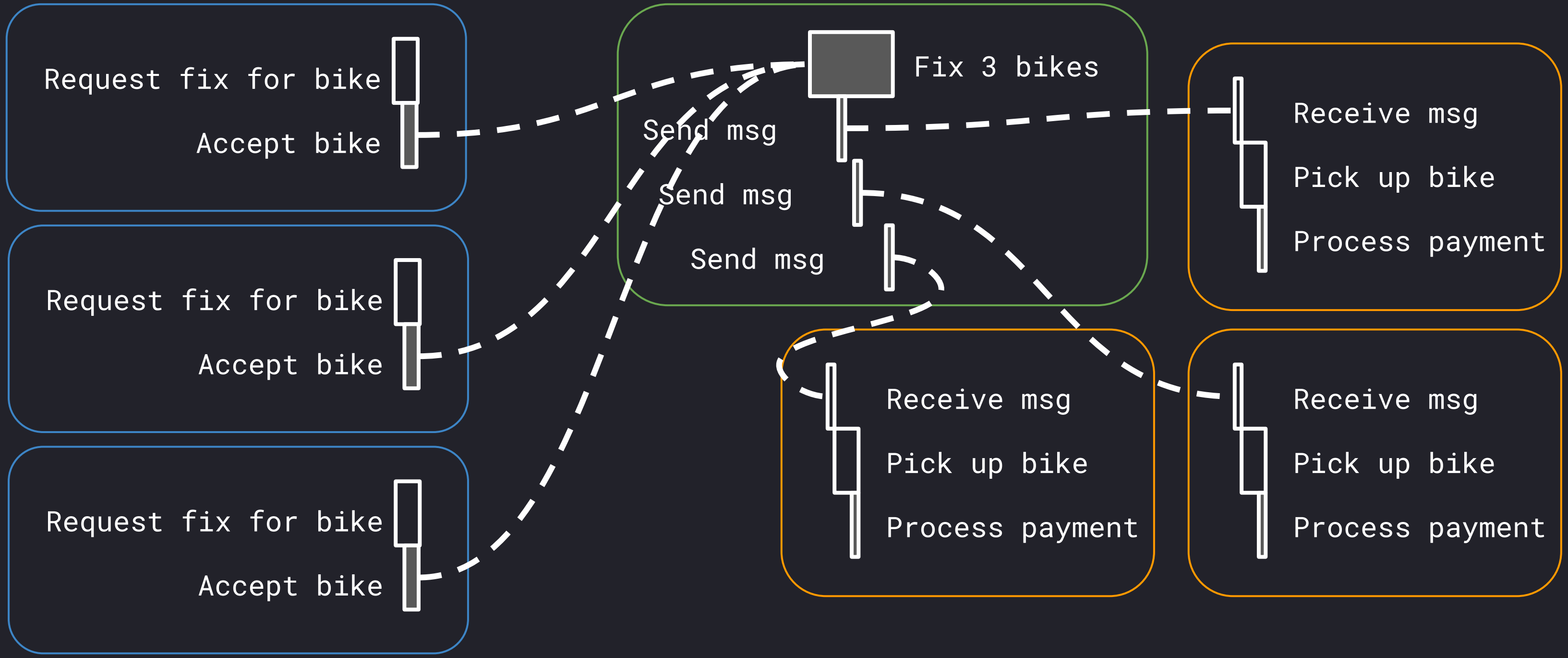
How to model an asynchronous workflow?

As multiple traces:



Tracing at the next level

As multiple traces:



Monitoring	Pain
Asynchronous workflows	Disintegration

Pain points

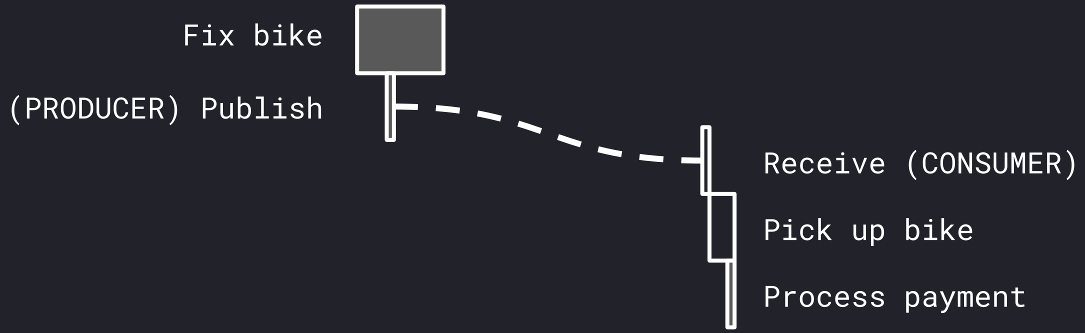
- No established best-practice
- As a single trace
    - Large and confusing traces for complex scenarios
    - Impossible to model certain workflows
- As multiple traces
    - Overkill for simple workflows
    - Varying levels of support from different tools and vendors
    - Breaks established sampling solutions
- Scenario-specific
    - Lack of consistency
    - Scalability challenges

OpenTelemetry Semantic Conventions

- Define span names, attributes names, and span relationships
- OpenTelemetry messaging workgroup
  - Stable semantic conventions for messaging traces and metrics
  - On OpenTelemetry’s roadmap for 2024

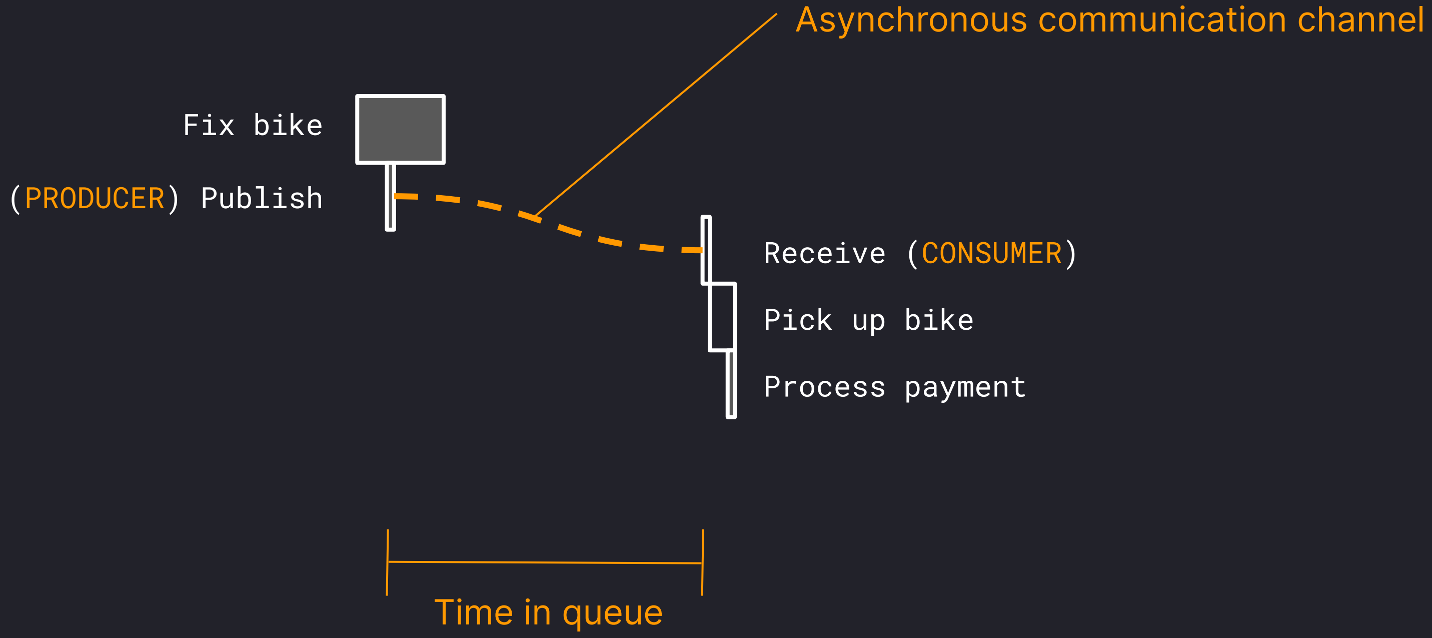
OpenTelemetry Semantic Conventions

Require a link between message producer and consumer spans



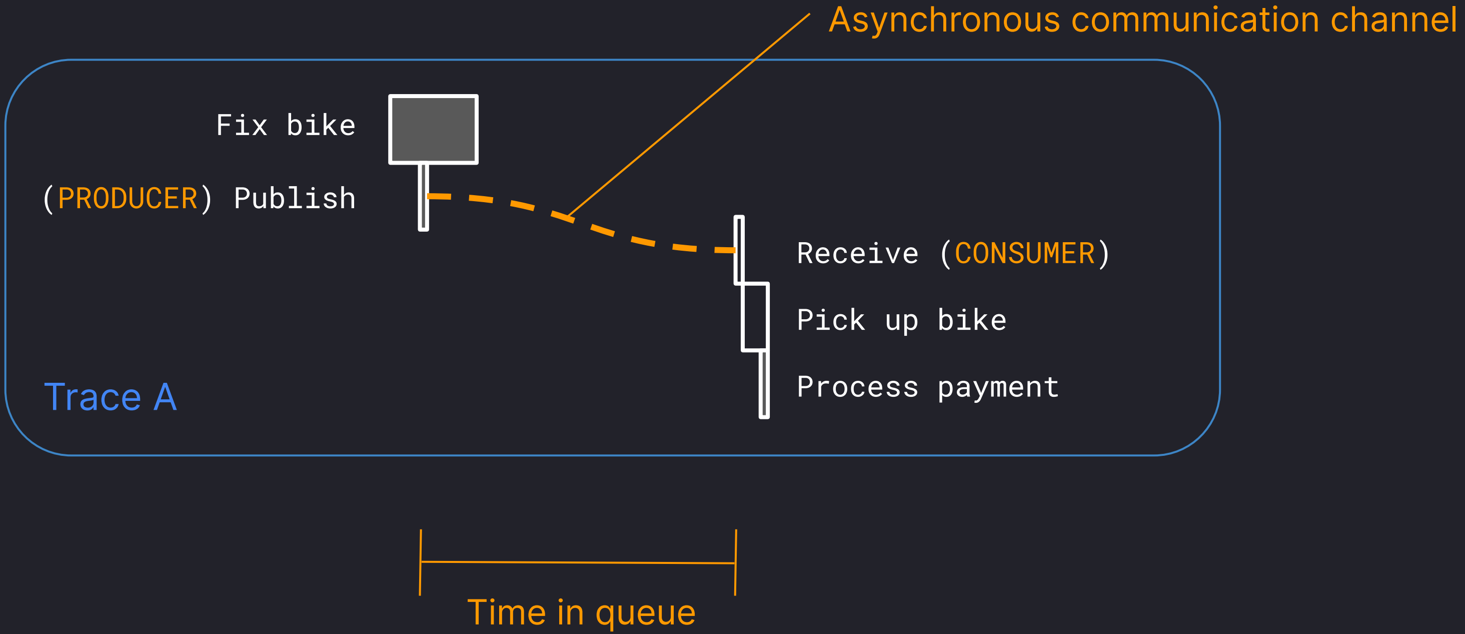
OpenTelemetry Semantic Conventions

Require a link between message producer and consumer spans



OpenTelemetry Semantic Conventions

Require a link between message producer and consumer spans



Asynchronous communication channel

Fix bike

(PRODUCER) Publish

Receive (CONSUMER)

Pick up bike

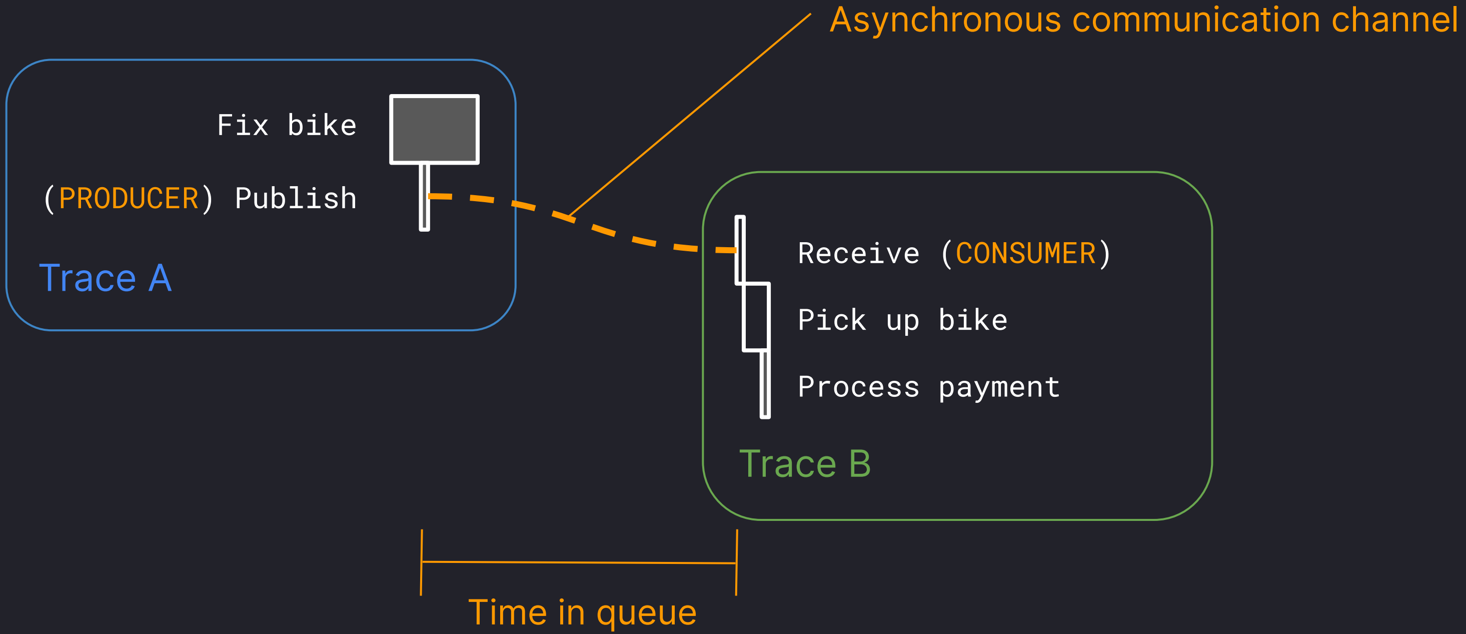
Process payment

Trace A

Time in queue

OpenTelemetry Semantic Conventions

Require a link between message producer and consumer spans



Pain points

- No established best-practice
- As a single trace
  - Large and confusing traces for complex scenarios
  - Impossible to model certain workflows
- As multiple traces
  - Overkill for simple workflows
  - Varying levels of support from different tools and vendors
  - Breaks established sampling solutions
- Scenario-specific
  - Lack of consistency
  - Scalability challenges

Pain points

- No established best-practice
- Large and confusing traces for complex scenarios
- Impossible to model certain workflows
- Varying levels of support from different tools and vendors
- Overkill for simple workflows
- Breaks established sampling solutions
- Lack of consistency
- Scalability challenges

❌ Impossible
🟠 Not yet solved
✅ Solved

Pain points

- No established best-practice
    - 🟠 **Draft solution as part of an open standard**
- Large and confusing traces for complex scenarios
    - ✅ **Traces can be broken up**
- Impossible to model certain workflows
    - ✅ **All workflows can be modelled**
- Varying levels of support from different tools and vendors
    - 🟠 **Stable semantic conventions and instrumentations as a catalyst**
- Overkill for simple workflows
    - ✅ **Single-trace modelling is possible**
- Breaks established sampling solutions
    - 🟠 **Currently no link-based sampling solutions**
- Lack of consistency
    - ✅ **Reliable invariants across different modelling options**
- Scalability challenges
    - ✅ **Possibility to switch modelling options without losing invariants**

❌ **Impossible**

🟠 **Not yet solved**

✅ **Solved**

Monitoring	Pain
Asynchronous workflows	Disintegration

“How should I instrument?”

- There’s a solution, it brings tracing to the next level, BUT:
    - the instrumentation standard isn’t stable,
    - there exist very few instrumentation libraries,
    - context propagation standards aren’t stable,
    - it breaks existing sampling approaches,
    - and it’s not well supported by most vendors.

Work to be done

- Instrumentation standard
  - Traces
  - Metrics
- Instrumentation libraries
- Context propagation
  - W3C Trace Context AMQP First Public Working Draft
  - W3C Trace Context MQTT First Public Working Draft
  - CloudEvents Specification Extension for Distributed Tracing
- Sampling
- Vendor support for span links

Get in touch

- Monitorama Slack
  - #talk-johannes-tax
- CNCF Slack
  - #otel-messaging
  - @Johannes Tax
- Weekly meeting Thursday 8am PST
- GitHub issue