



Introduction to distributed tracing  
within Kafka with Zipkin

# About Me



**DATA VIKING**

## Dominik Fries



CTO @THINKPORT

E: [dfries@thinkport.digital](mailto:dfries@thinkport.digital)

H: <https://thinkport.digital>

Code and Slides

<https://github.com/ThinkportRepo>

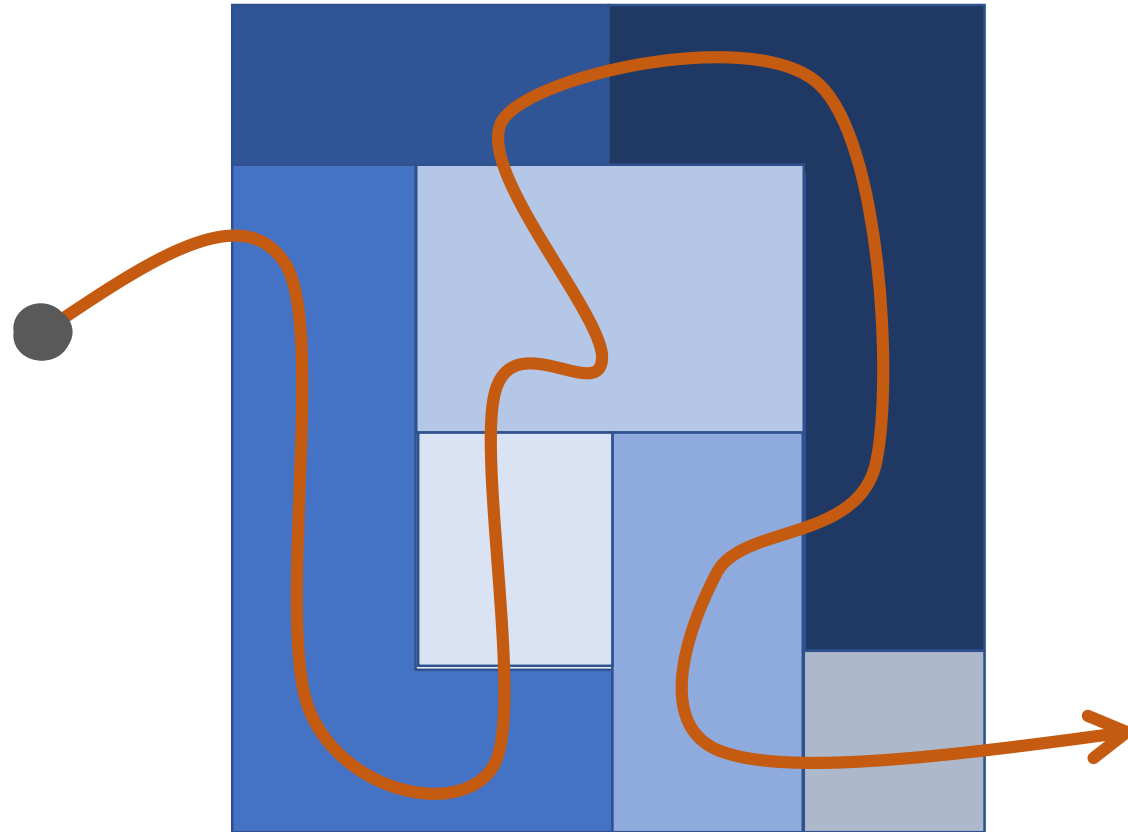
# In the beginning...

there was the monolith

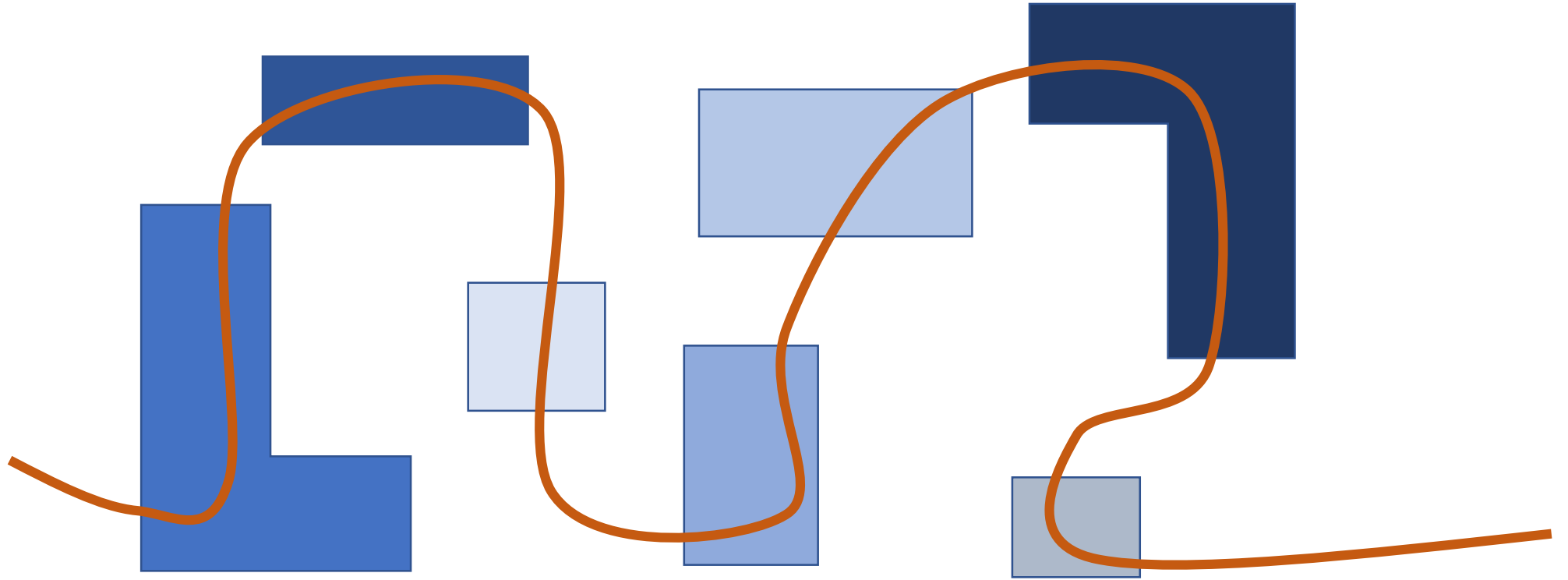
**No** system boundaries

**No** distribution

**One** application

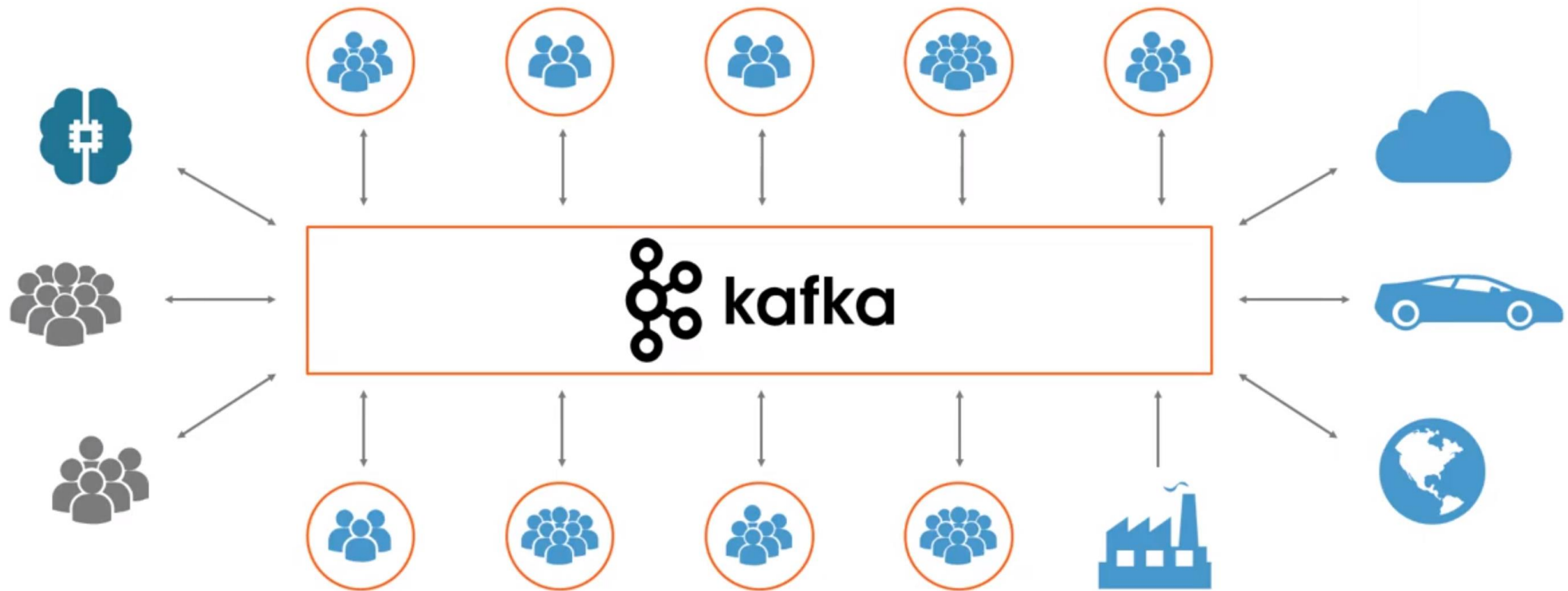


But everthing changed ...

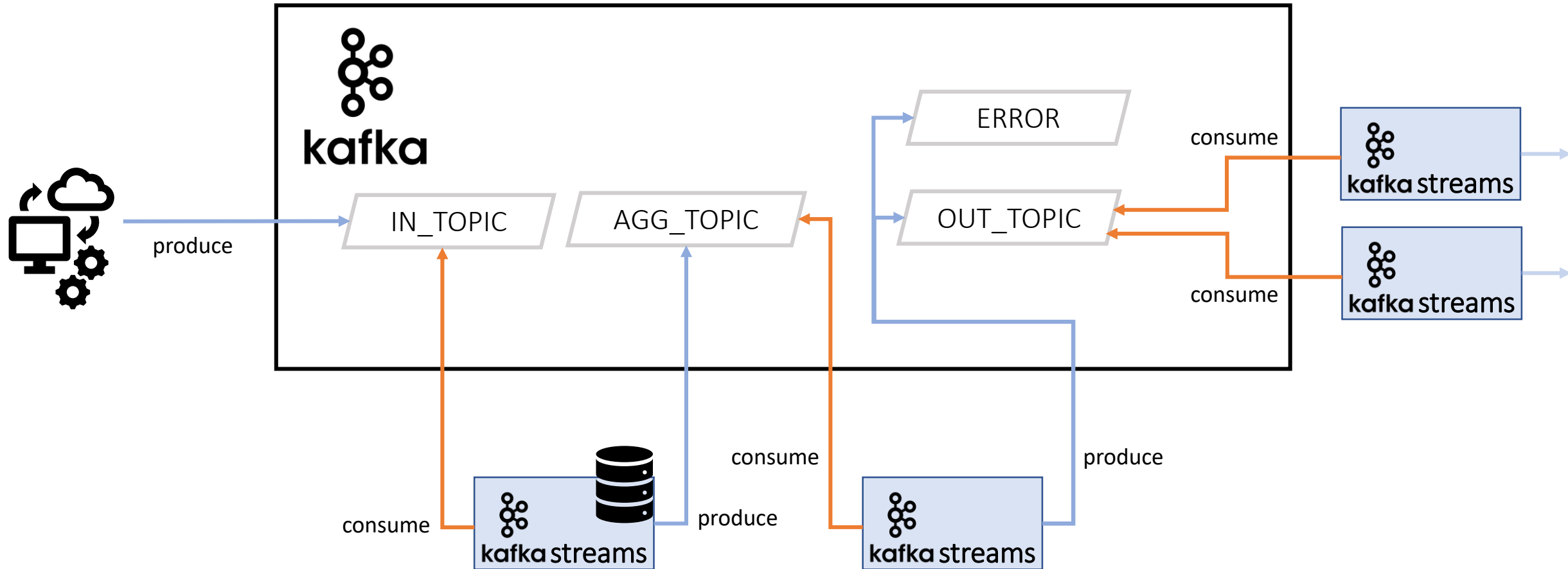


... when the MicroServices appeared

# Decouple the MicroServices with Kafka



# Event Driven Architecture



HOW CAN WE SUPPORT



THIS IN PRODUCTION



**Honest Status Page**

@honest\_update

 Follow



We replaced our monolith with micro services so that every outage could be more like a murder mystery.

RETWEETS

2,882

LIKES

2,256



4:10 PM - 7 Oct 2015



18



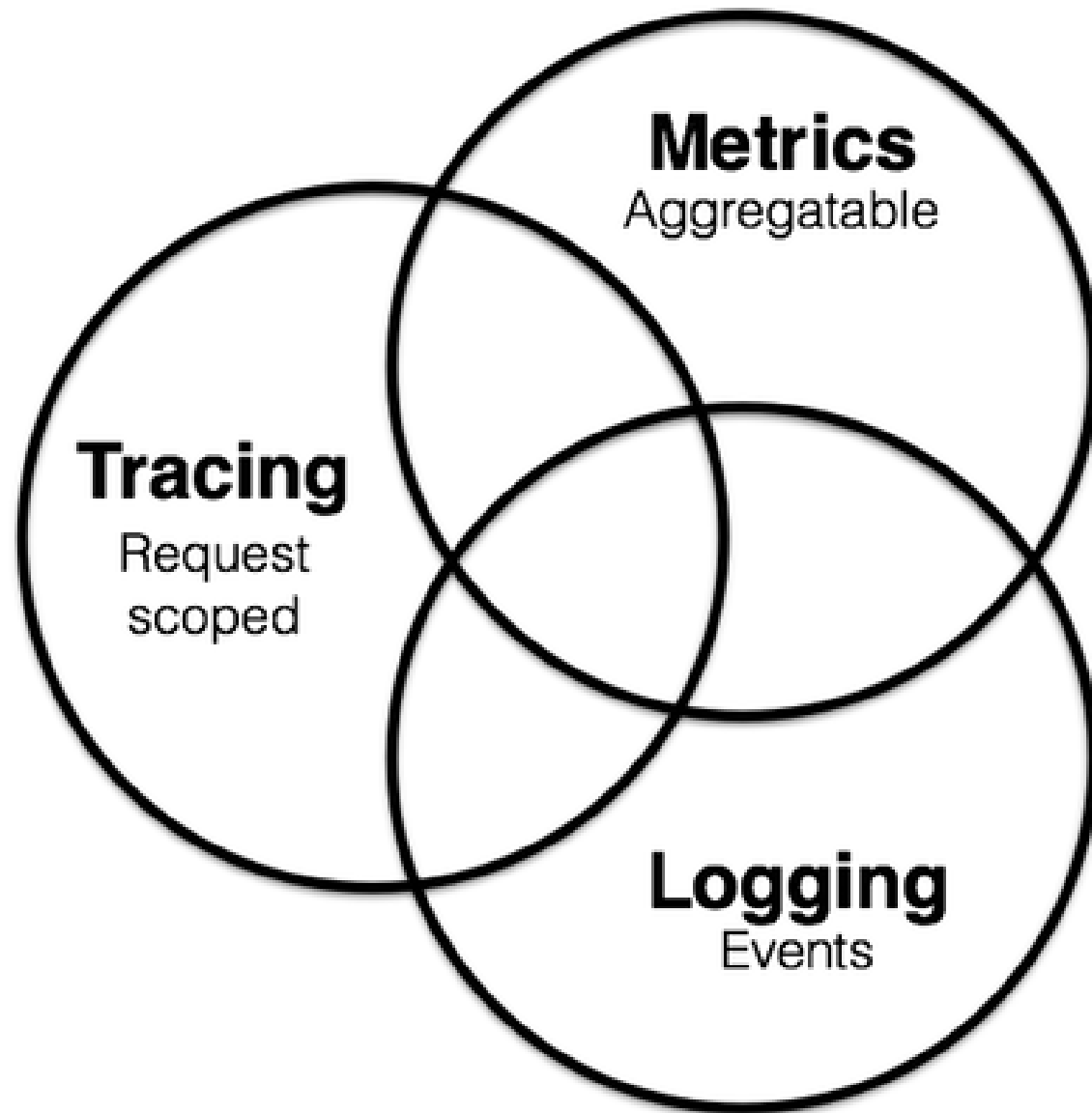
2.9K



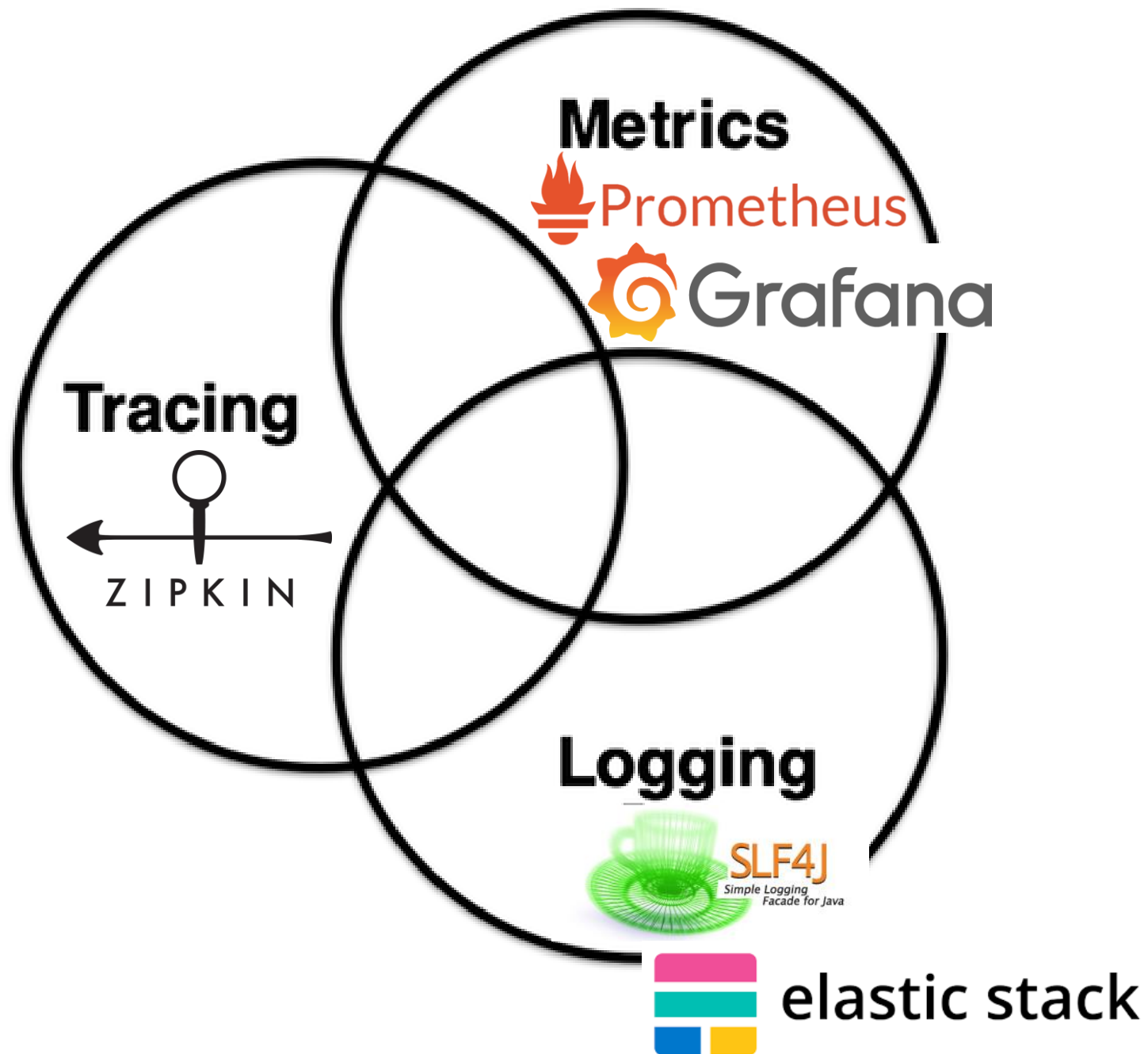
2.3K



The Three  
Pillars of  
Observability



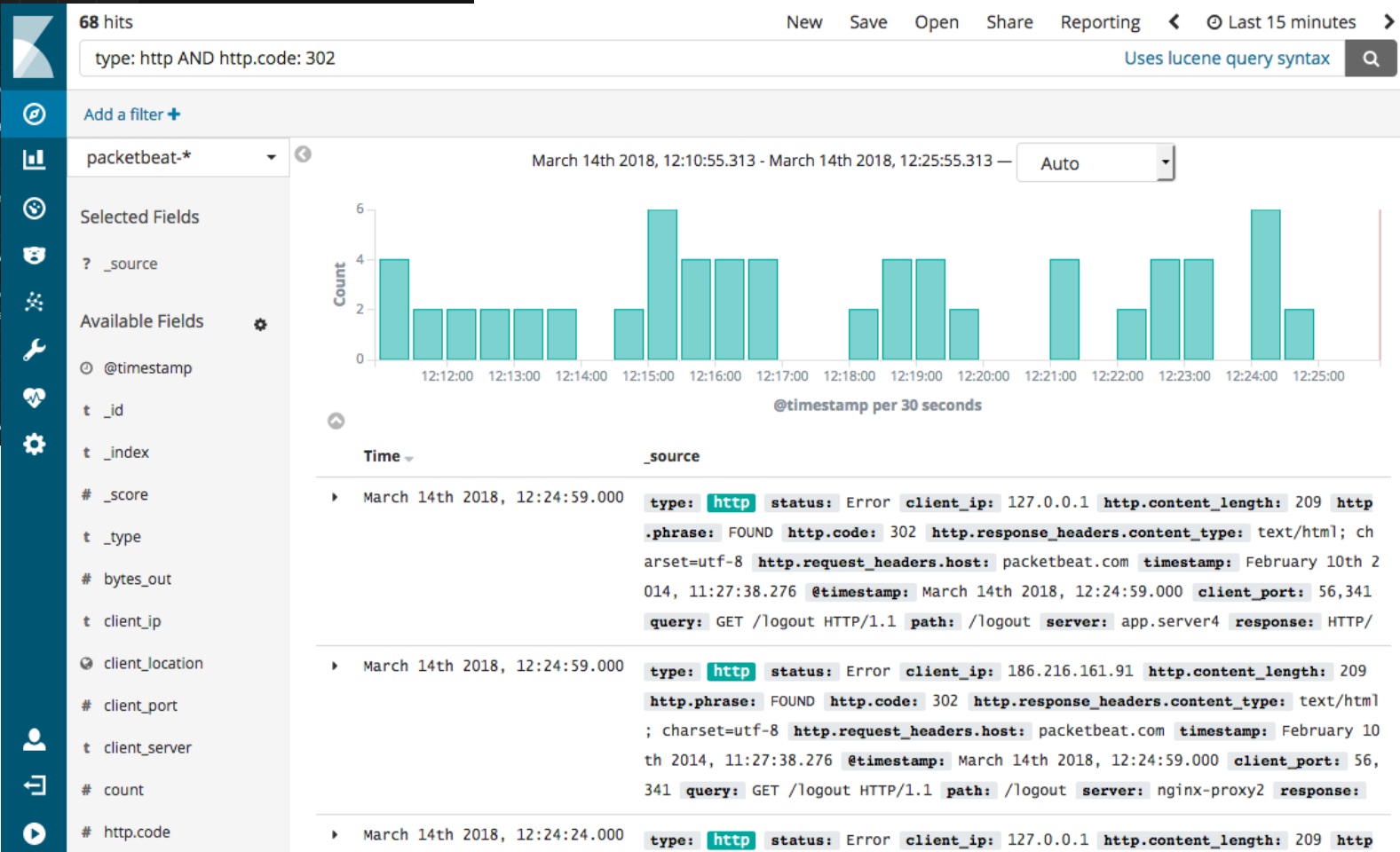
# The Three Pillars of Observability



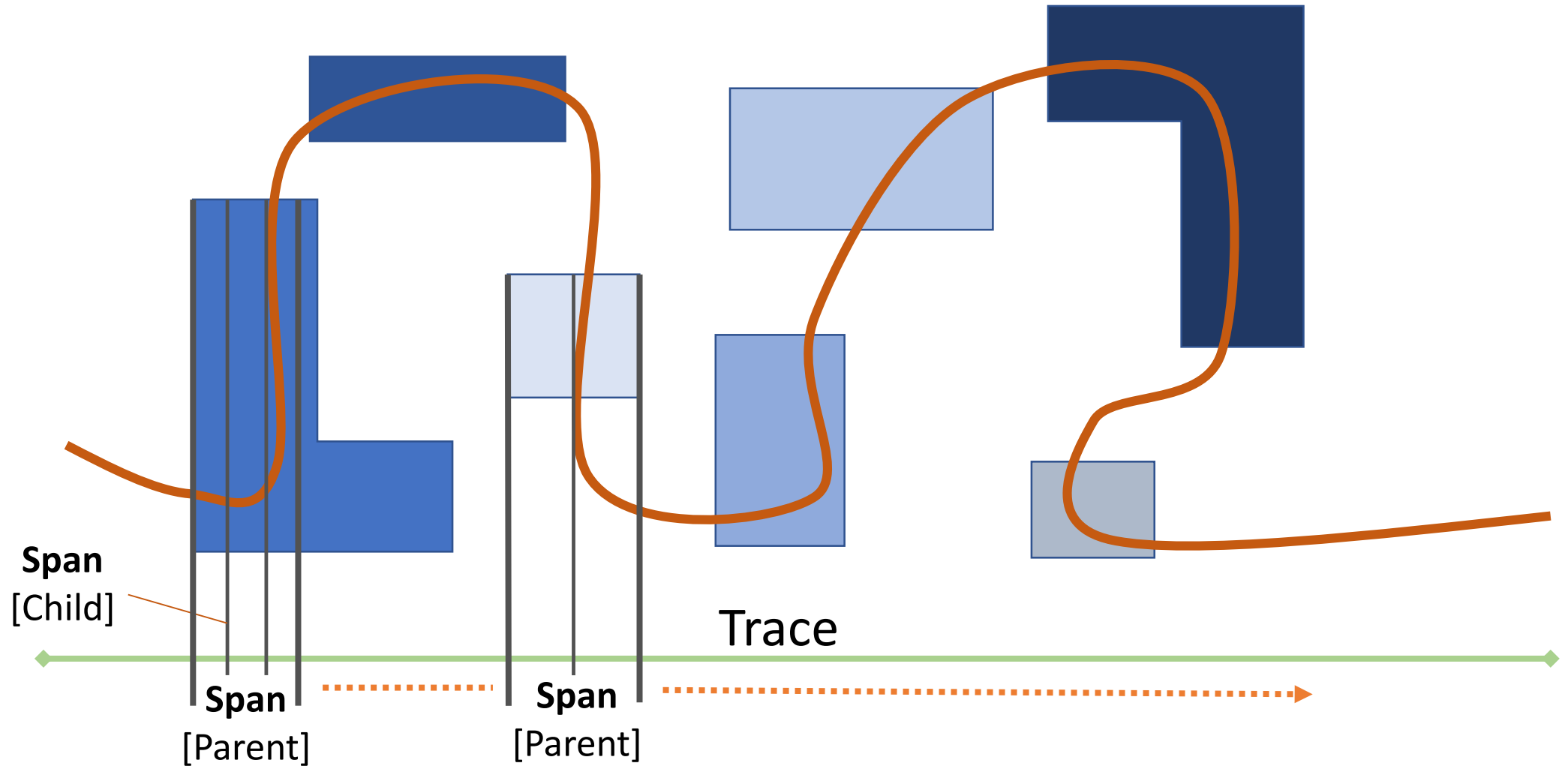
# Metric dashboards



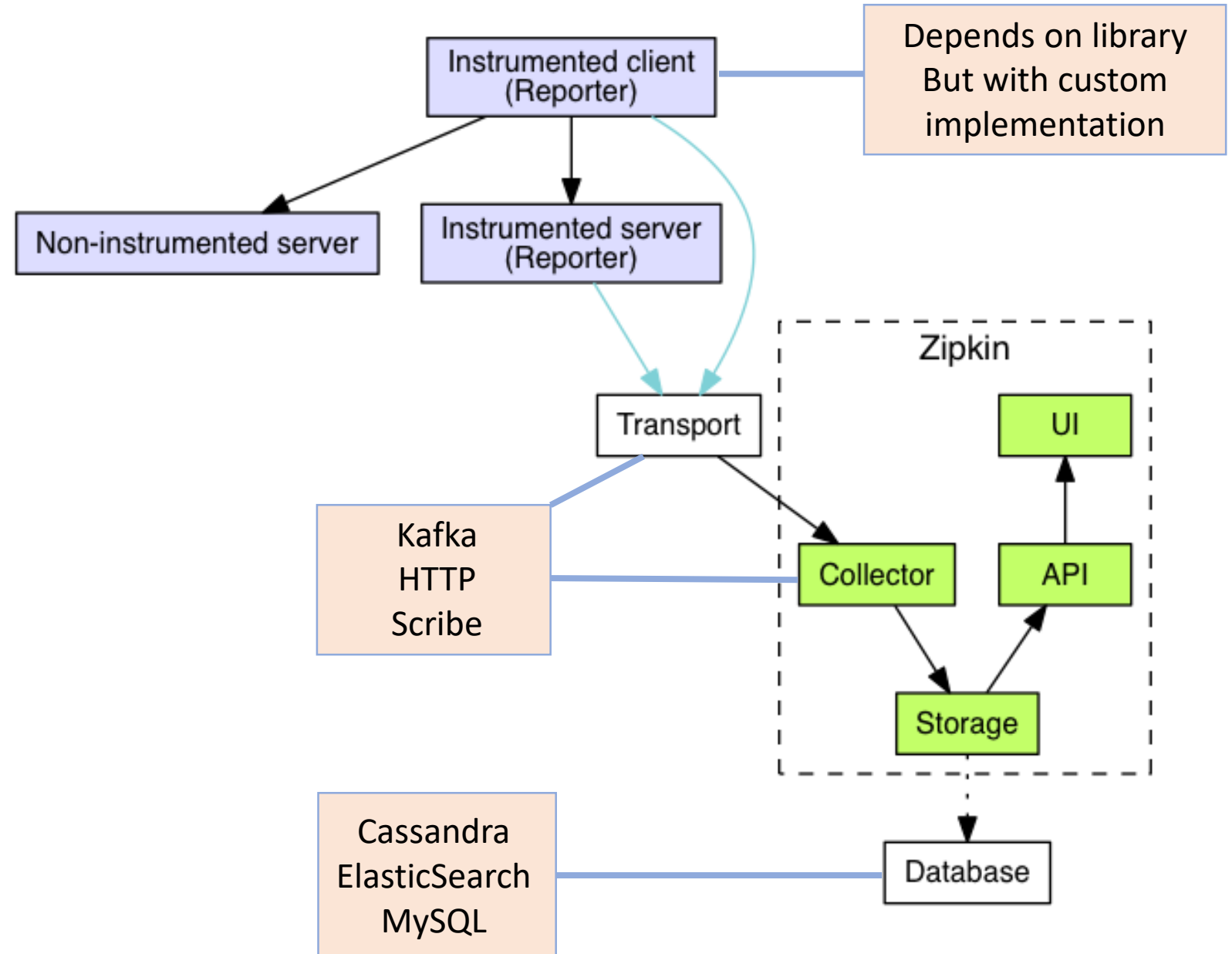
# Logging

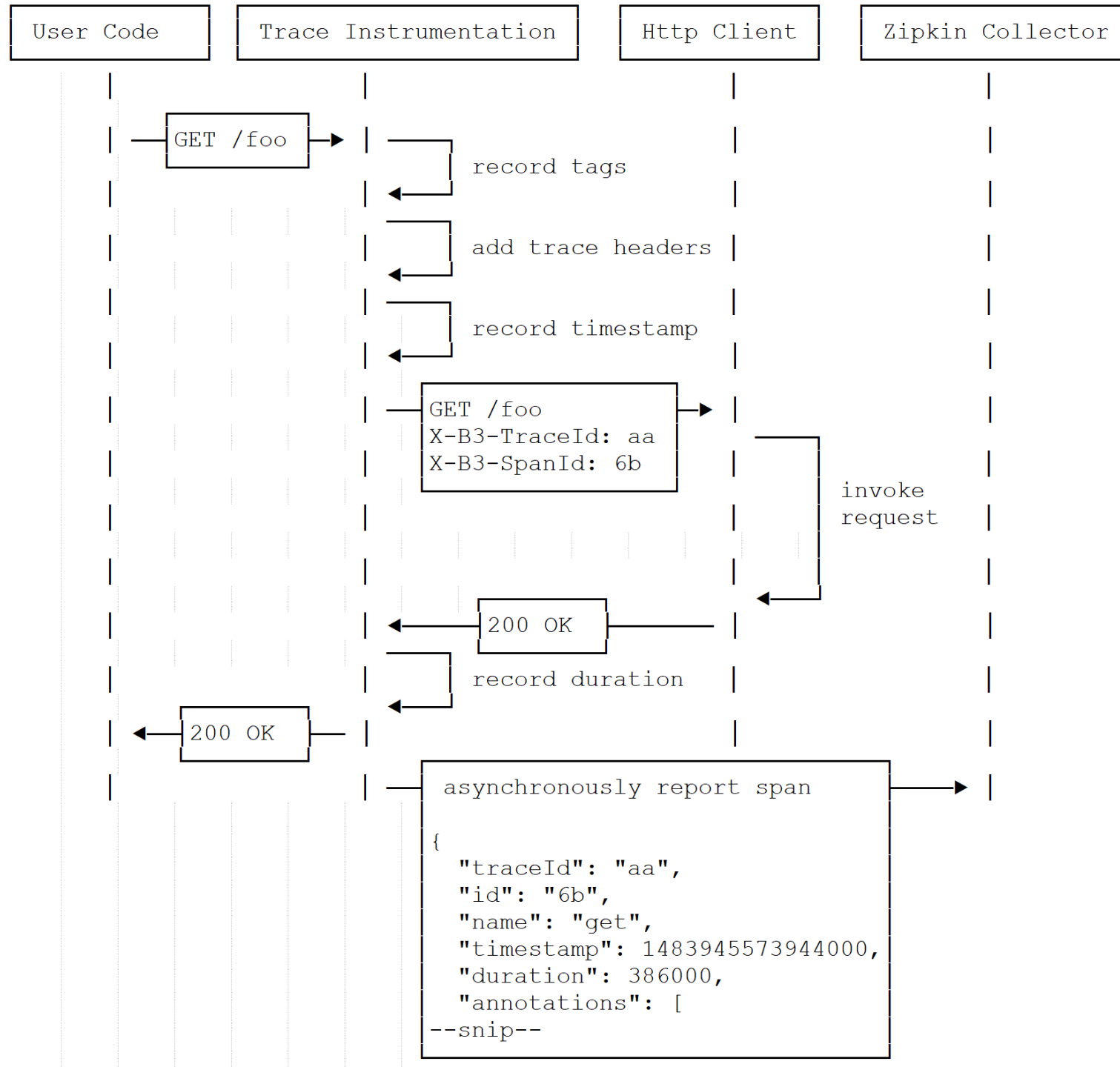


# What is a trace



# Zipkin Architecture

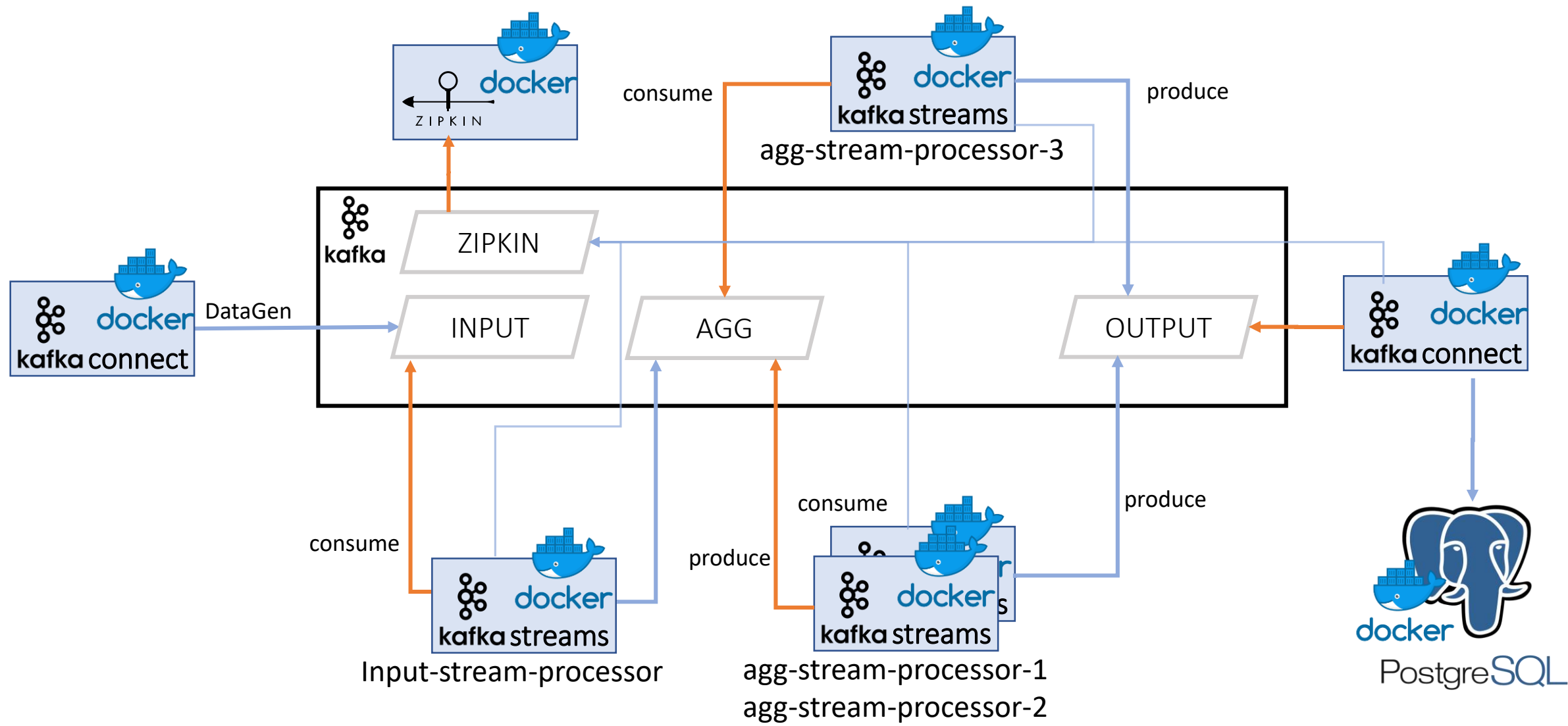




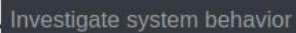
# DEMO



# Demo Architecture







Find a trace

[View Saved Trace](#)

## Dependencies

Go to trace

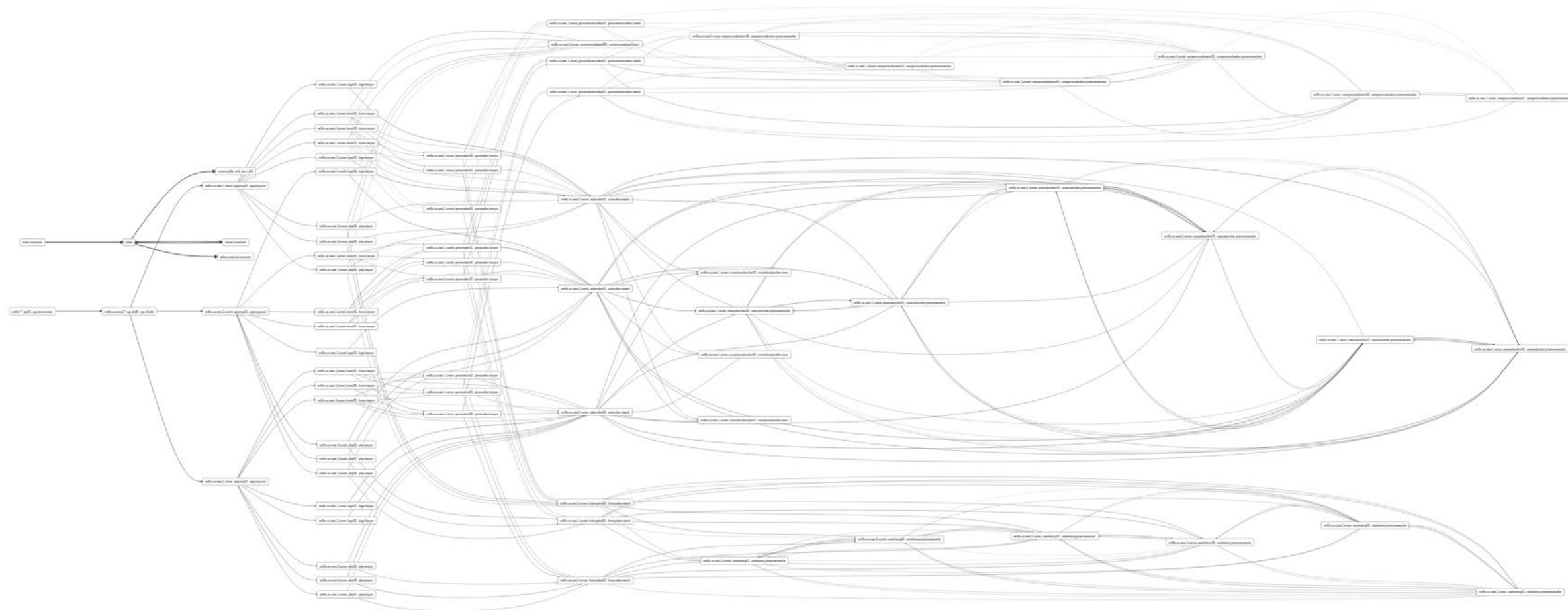
Start time	2018-09-11
------------	------------

22:15

End time	2018-09-12
----------	------------

22:15

## Analyze Dependencies



Code and slides @

<https://github.com/ThinkportRepo>

Questions and discussions

E: [dfries@thinkport.digital](mailto:dfries@thinkport.digital)

M: +176 23524634