



The Human Side of Airbnb's Microservice Architecture

Jessica Tai | QCon Plus | May 20, 2021



@jessicamtai

A photograph of a young woman with long dark hair, smiling at the camera. She is wearing a black sweater over a striped shirt and large gold hoop earrings. She is holding a fluffy, orange and white Corgi dog. They are positioned on a grassy cliff edge, with a vast ocean and distant hills visible in the background.

Hi, I'm Jessica.
I've been at Airbnb for 7 years.

@jessicamtai

Architecture evolution

Migration hits and misses

What's next

Architecture evolution

Migration hits and misses

What's next

@jessicamtai

Architecture evolution

Migration hits and misses

What's next

@jessicamtai

Architecture evolution

@jessicamtai



@jessicamtai

Monolith

2008 - 2017

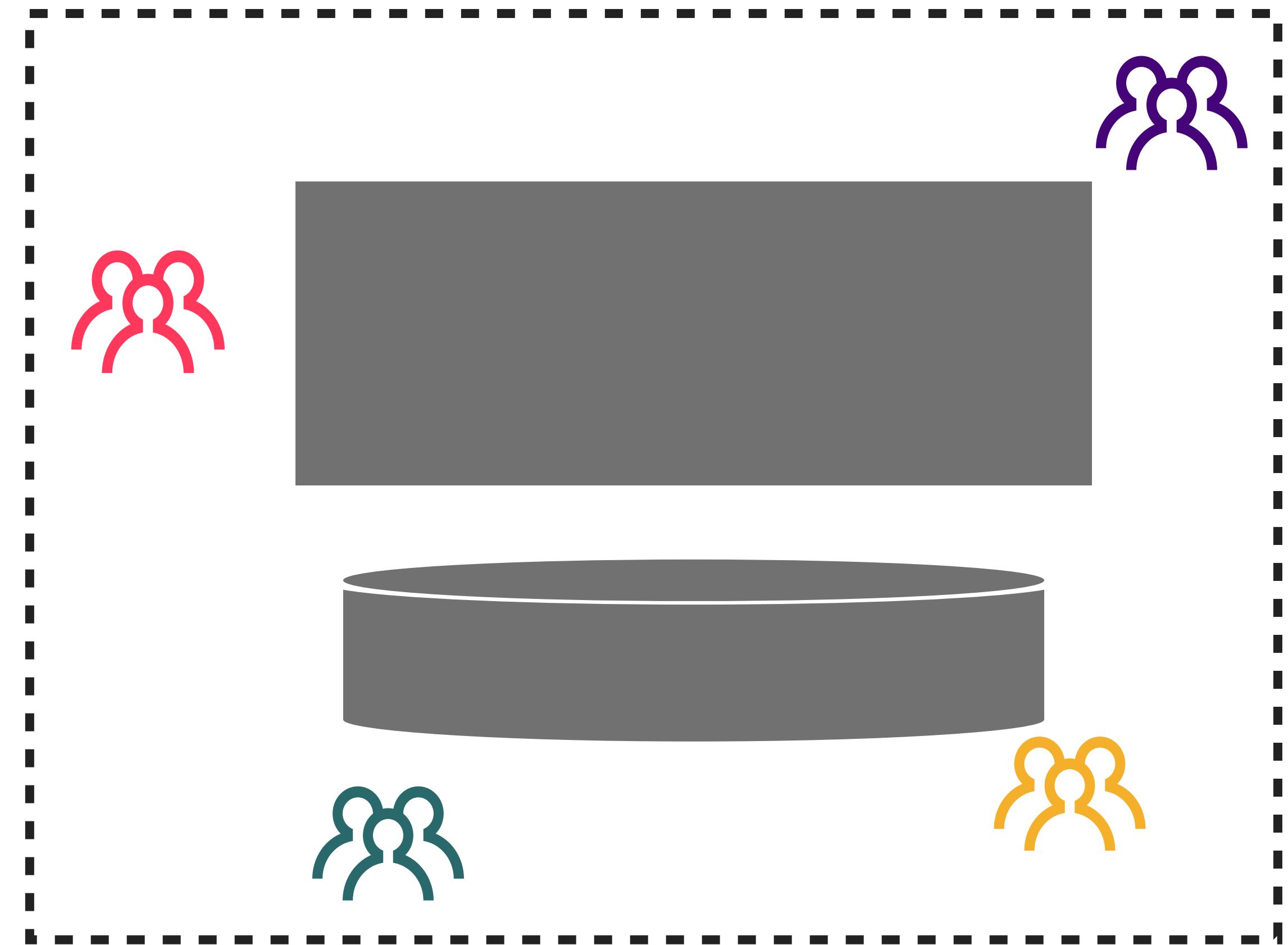


@jessicamtai

Monolith

2008 - 2017

More full-stack engineers

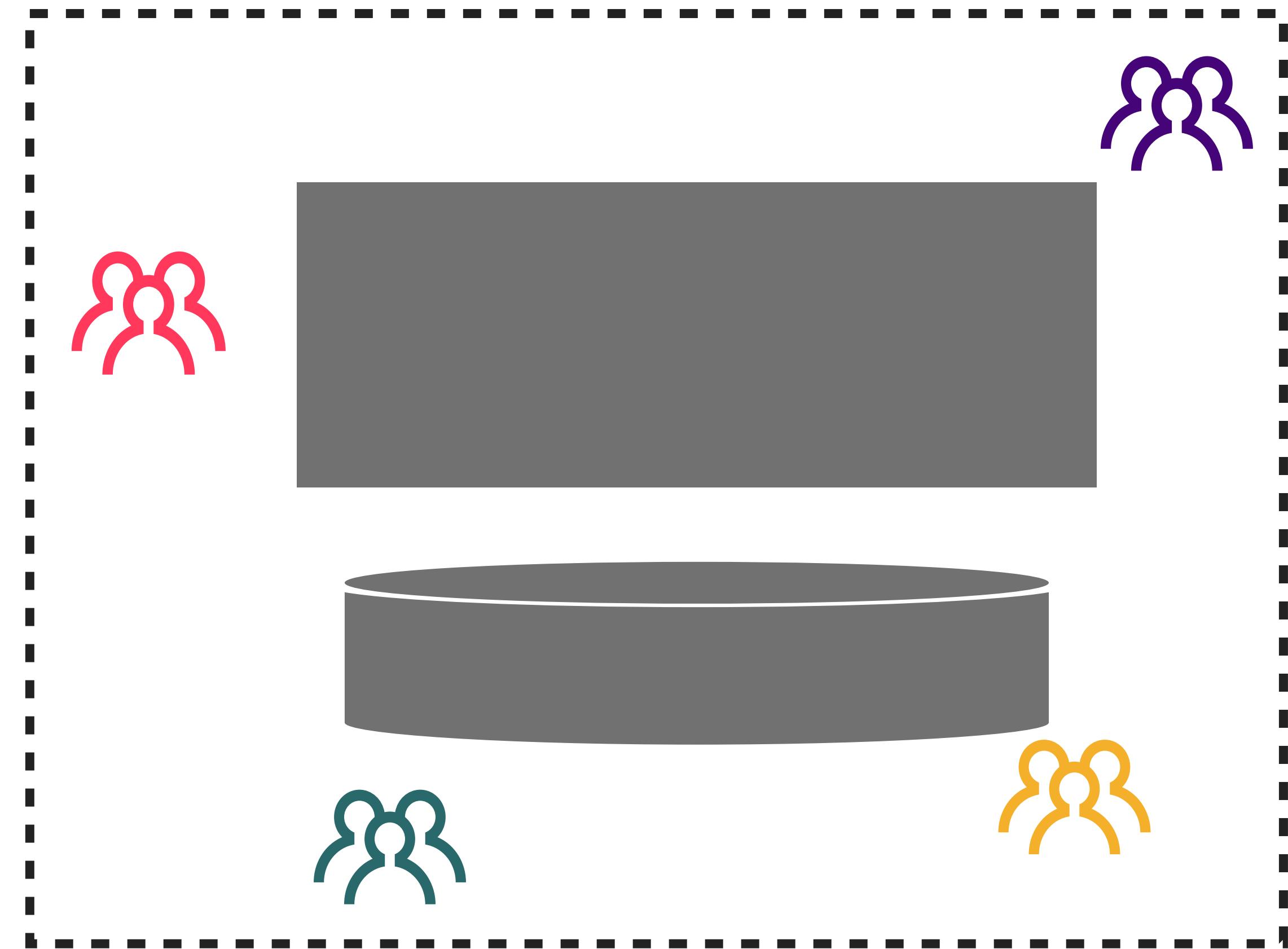


@jessicamtai

Monolith

2008 - 2017

Features completed
within teams, less dependencies

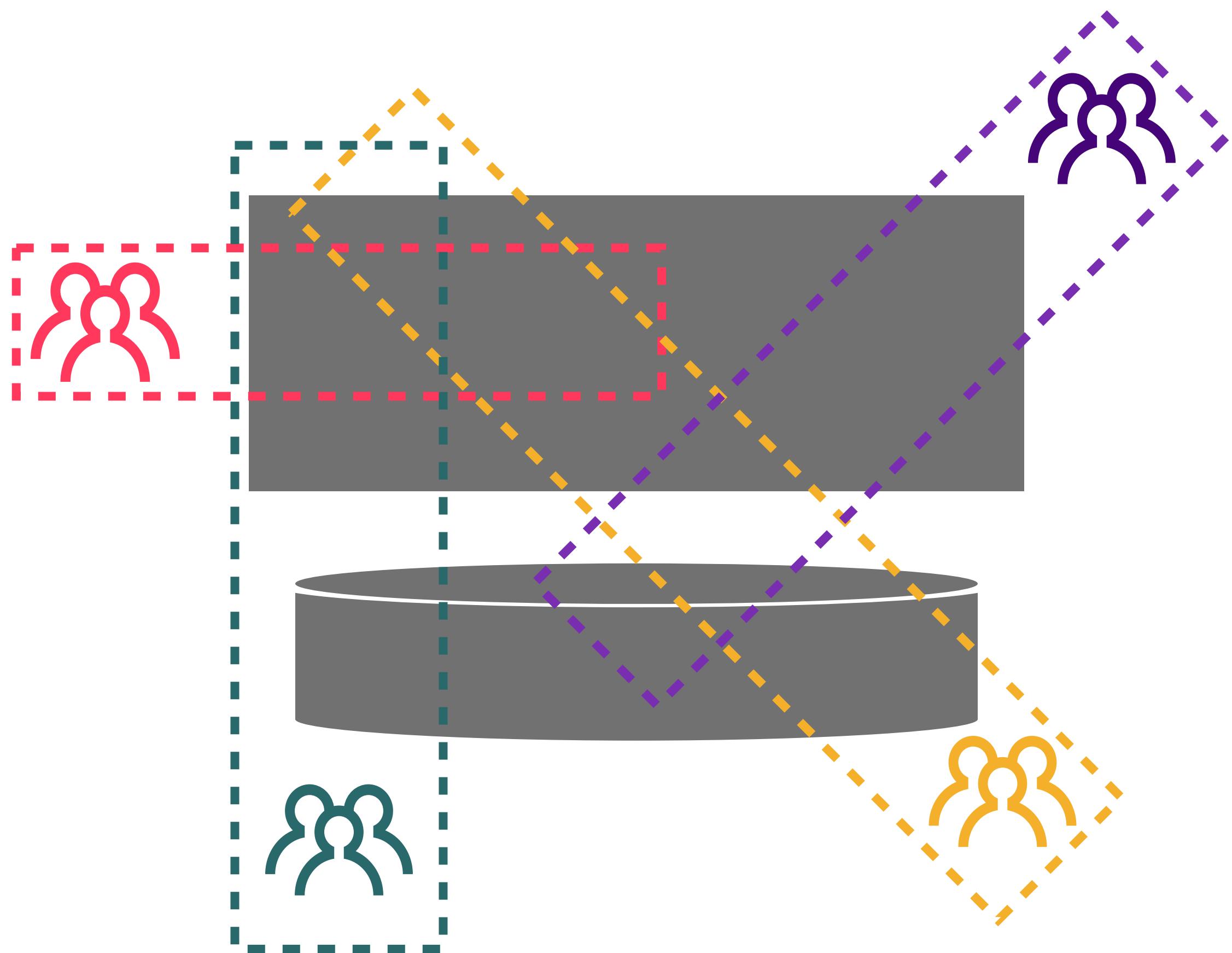


@jessicamtai

Monolith

2008 - 2017

Confusing team ownership +
unowned code



@jessicamtai

**Slow deploys resulted in
slower developer velocity.**

Microservices

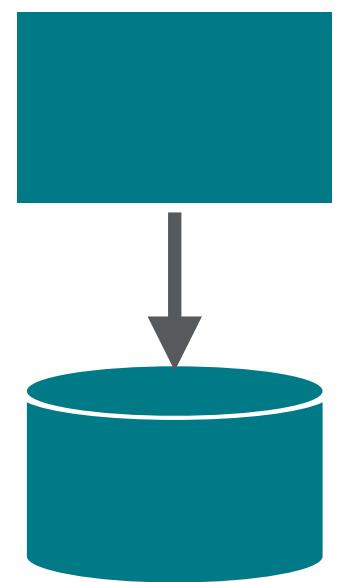
2017 - 2020

@jessicamtai

Microservices

2017 - 2020

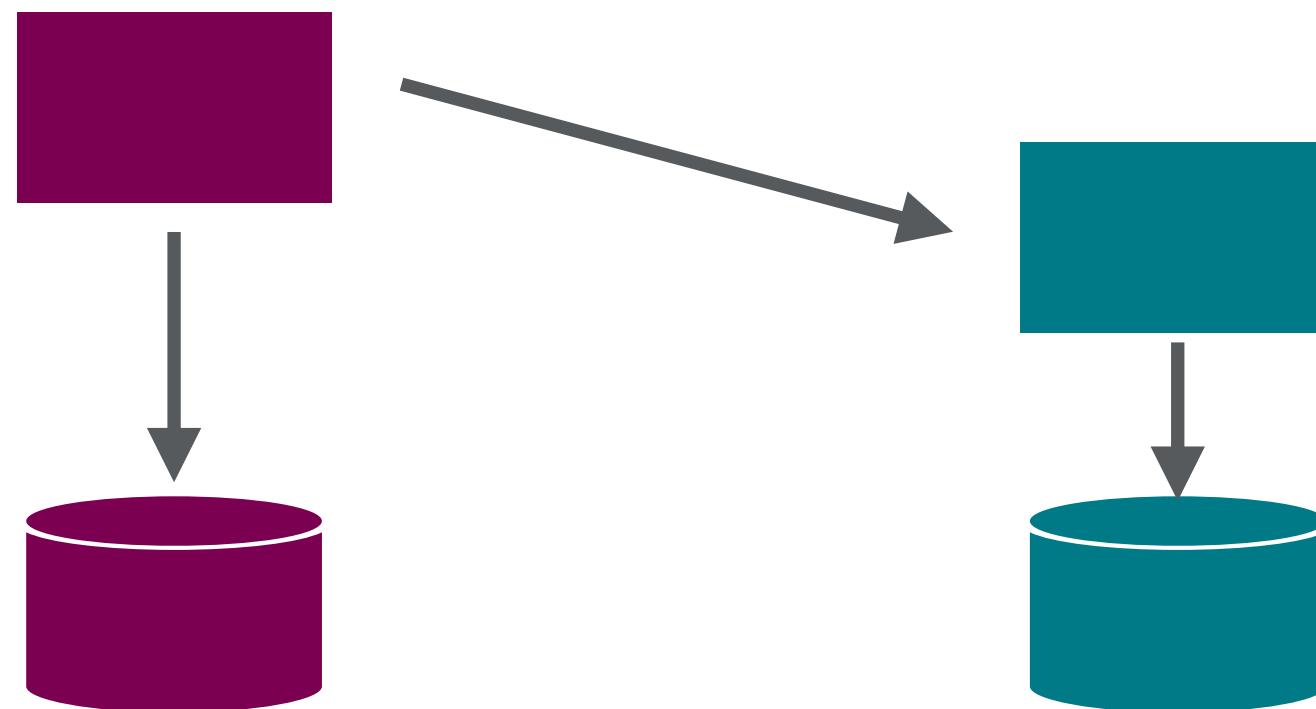
Data fetching service



Microservices

2017 - 2020

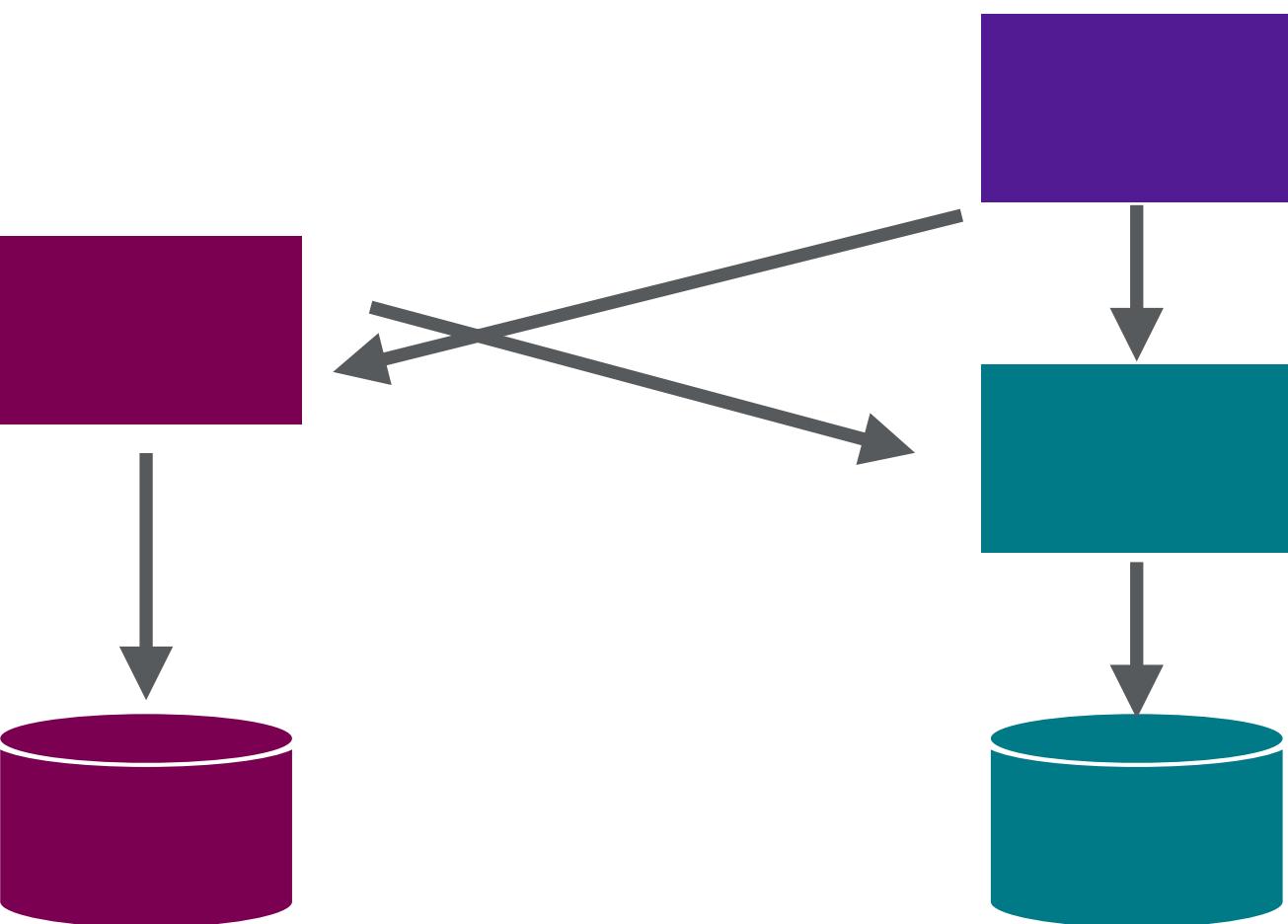
Business logic data service



Microservices

2017 - 2020

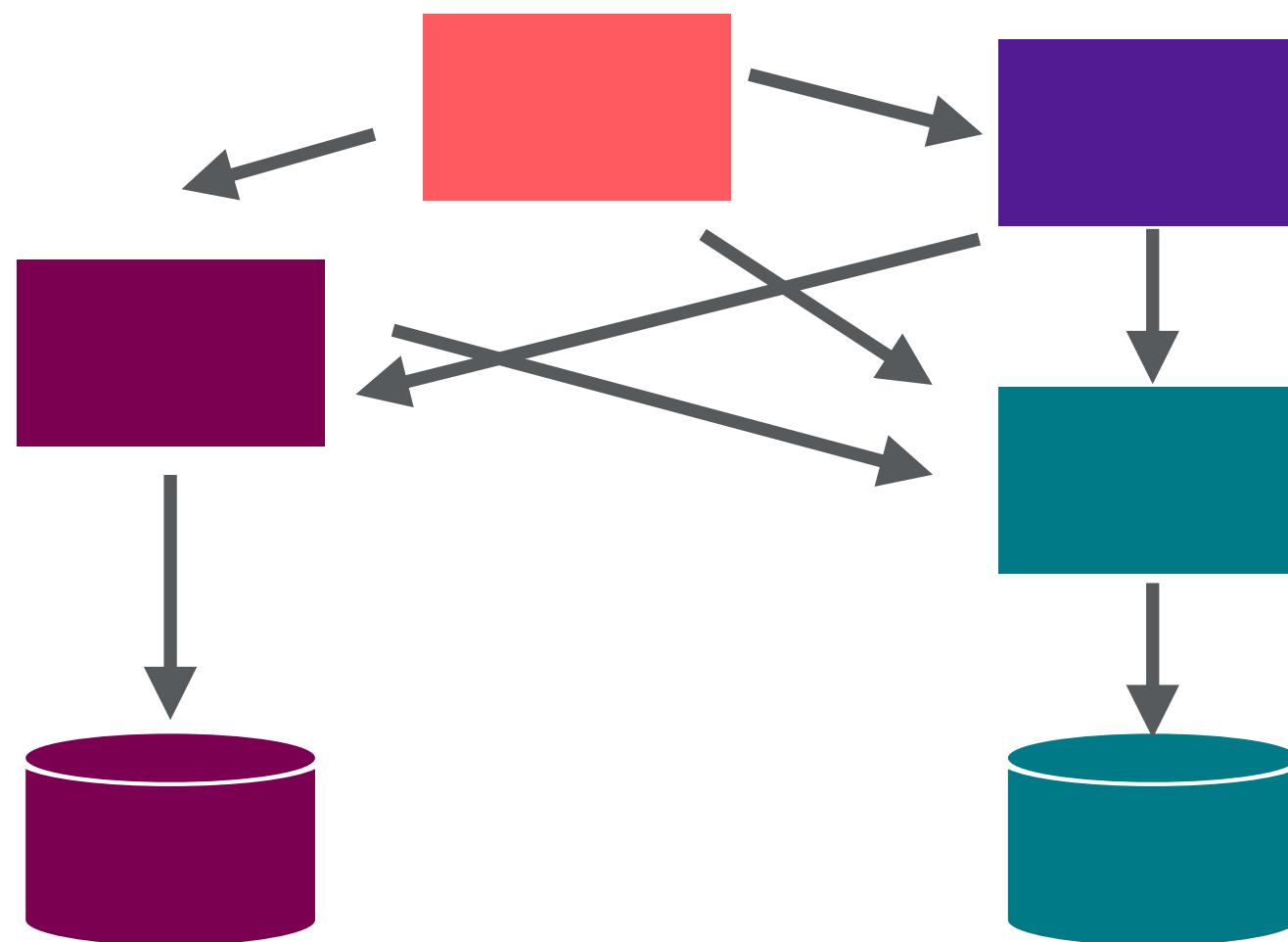
Write workflow service



Microservices

2017 - 2020

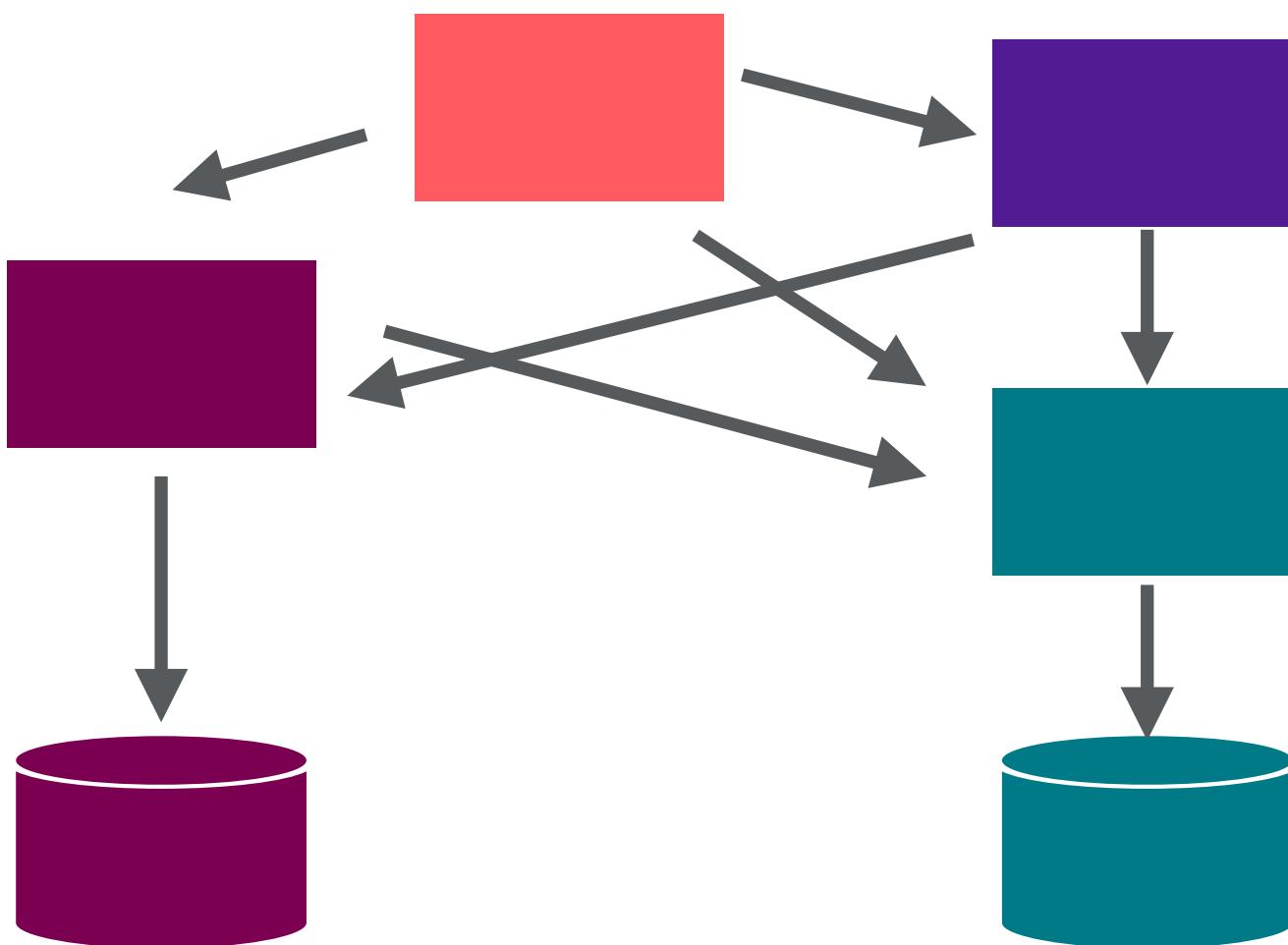
UI aggregation service



Microservices

2017 - 2020

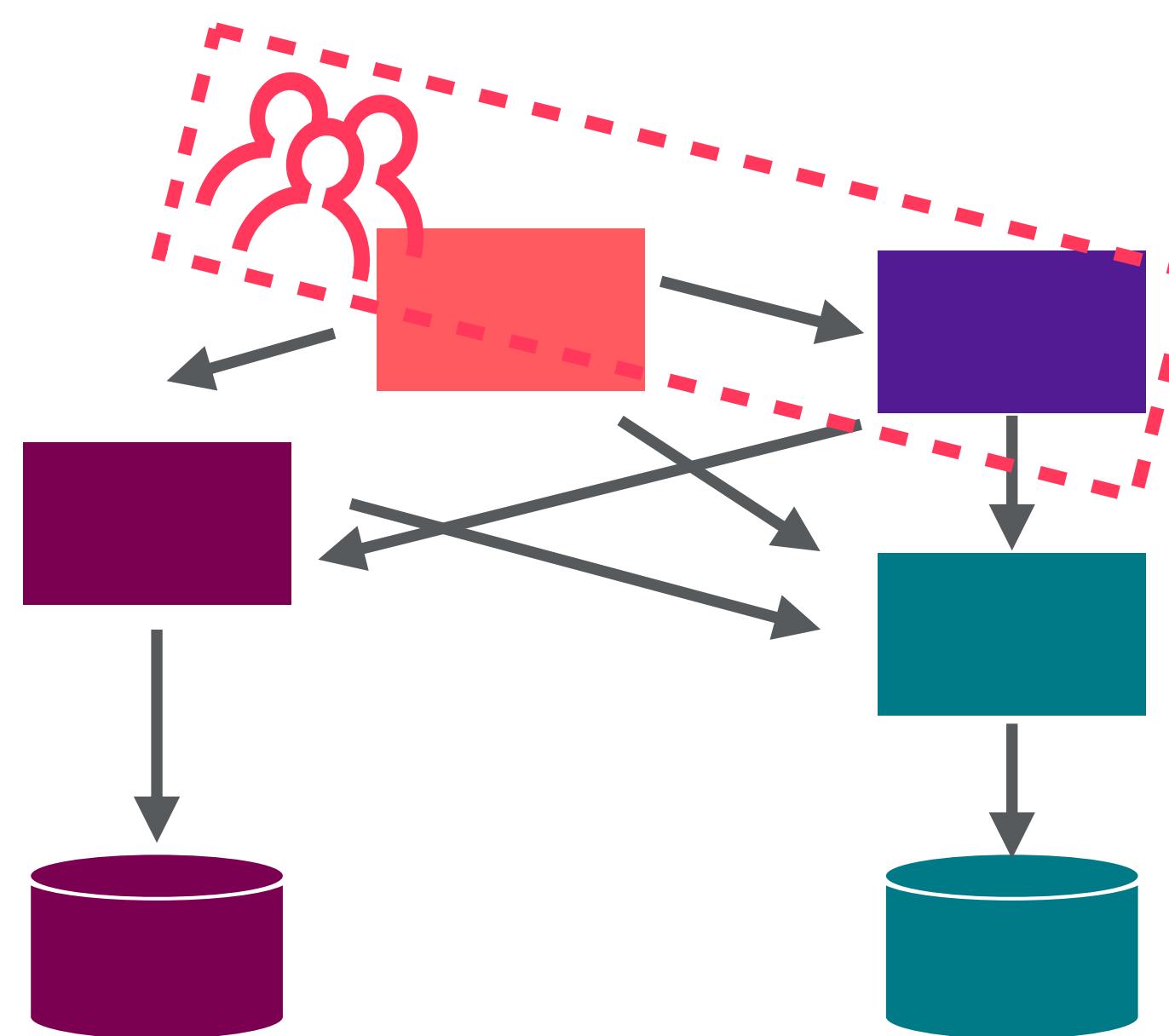
API gateway



Microservices

2017 - 2020

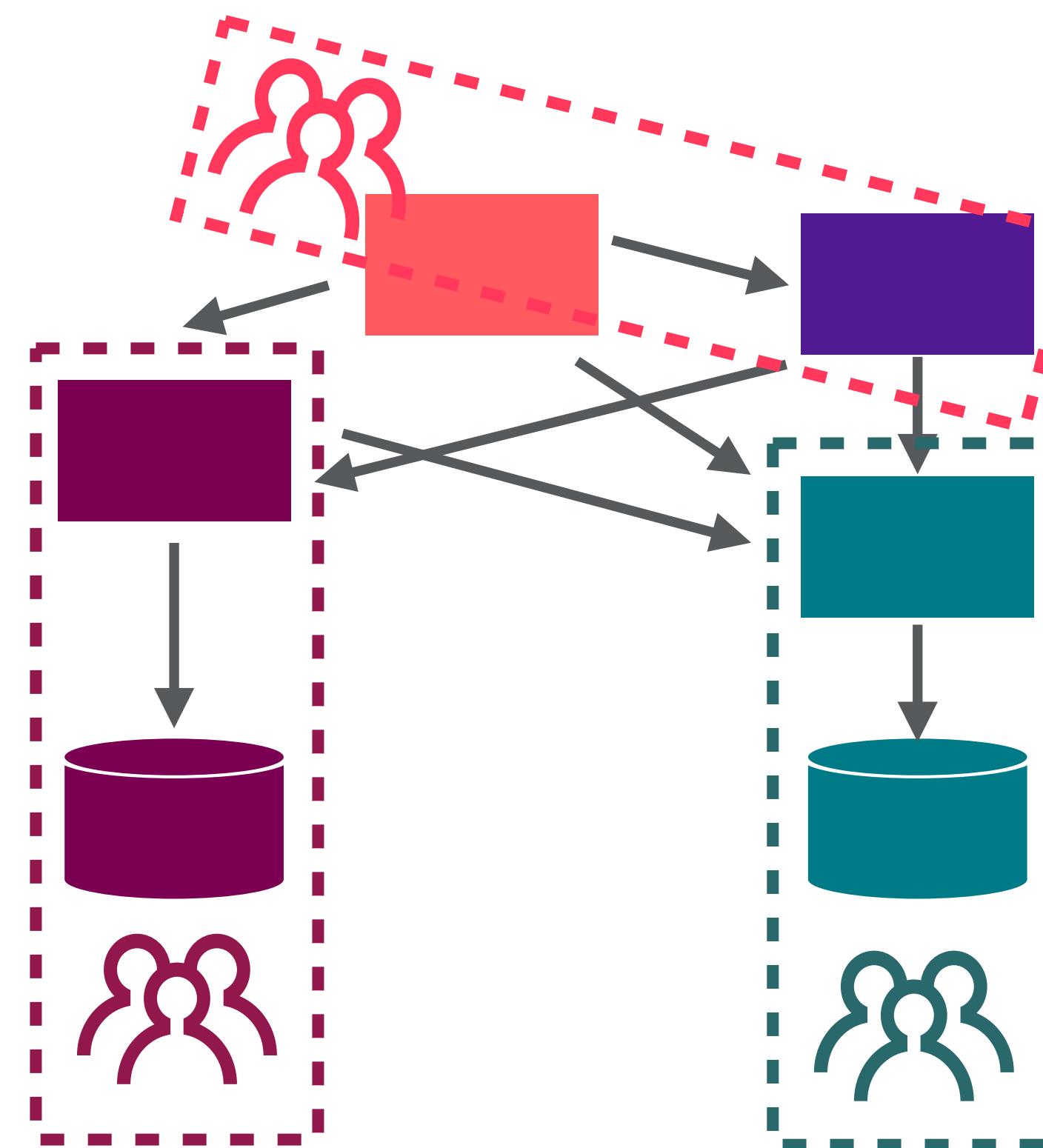
Each service had
one owning team



Microservices

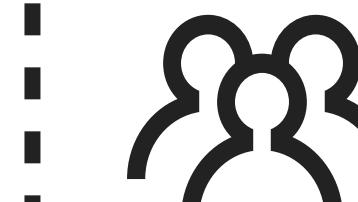
2017 - 2020

More backend only
teams & services

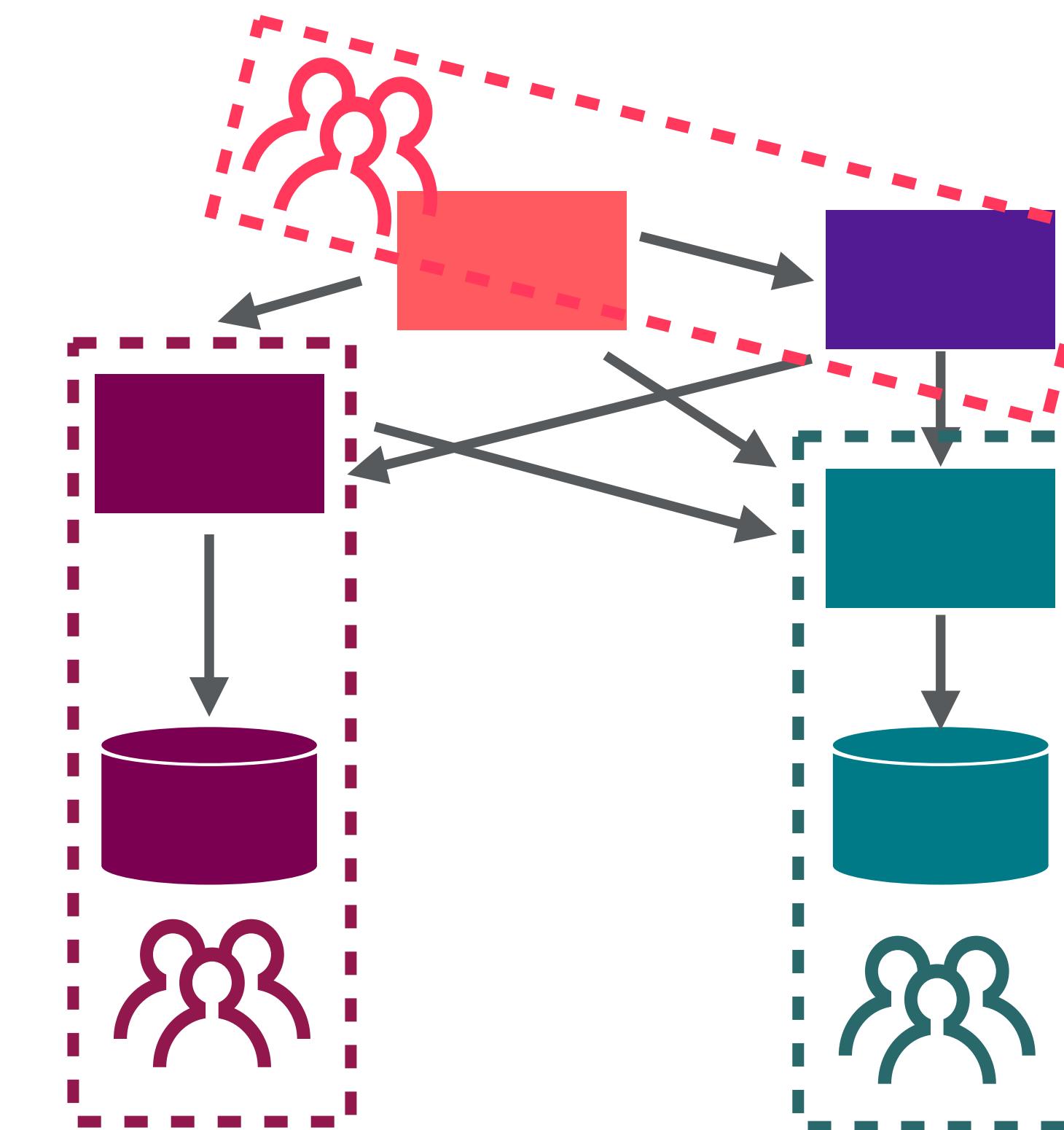


Microservices

2017 - 2020



Dedicated service
migration team

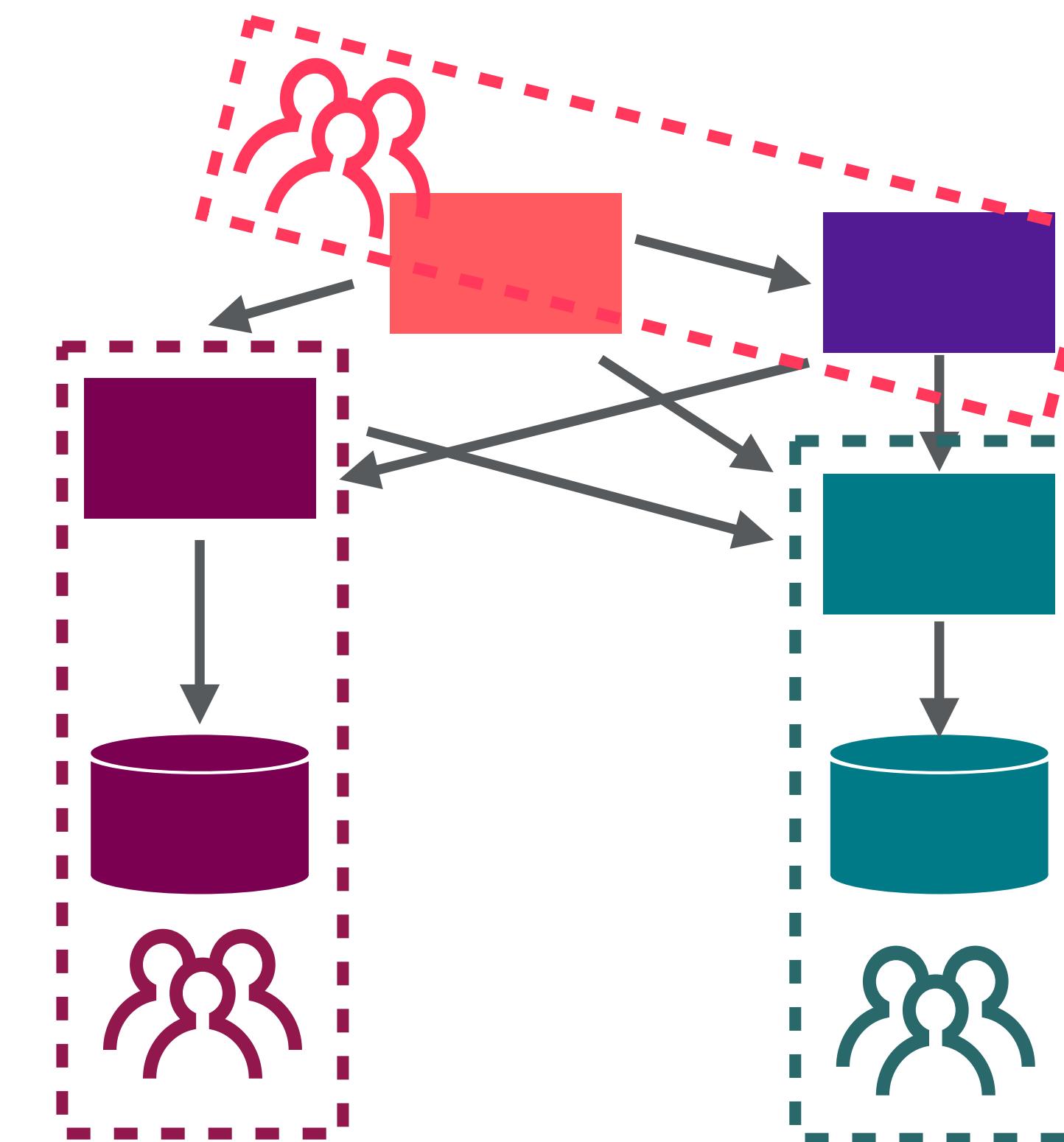


@jessicamtai

Microservices

2017 - 2020

Features required changes
from multiple services & teams



**Hundreds of services and dependencies
were difficult for humans to manage.**

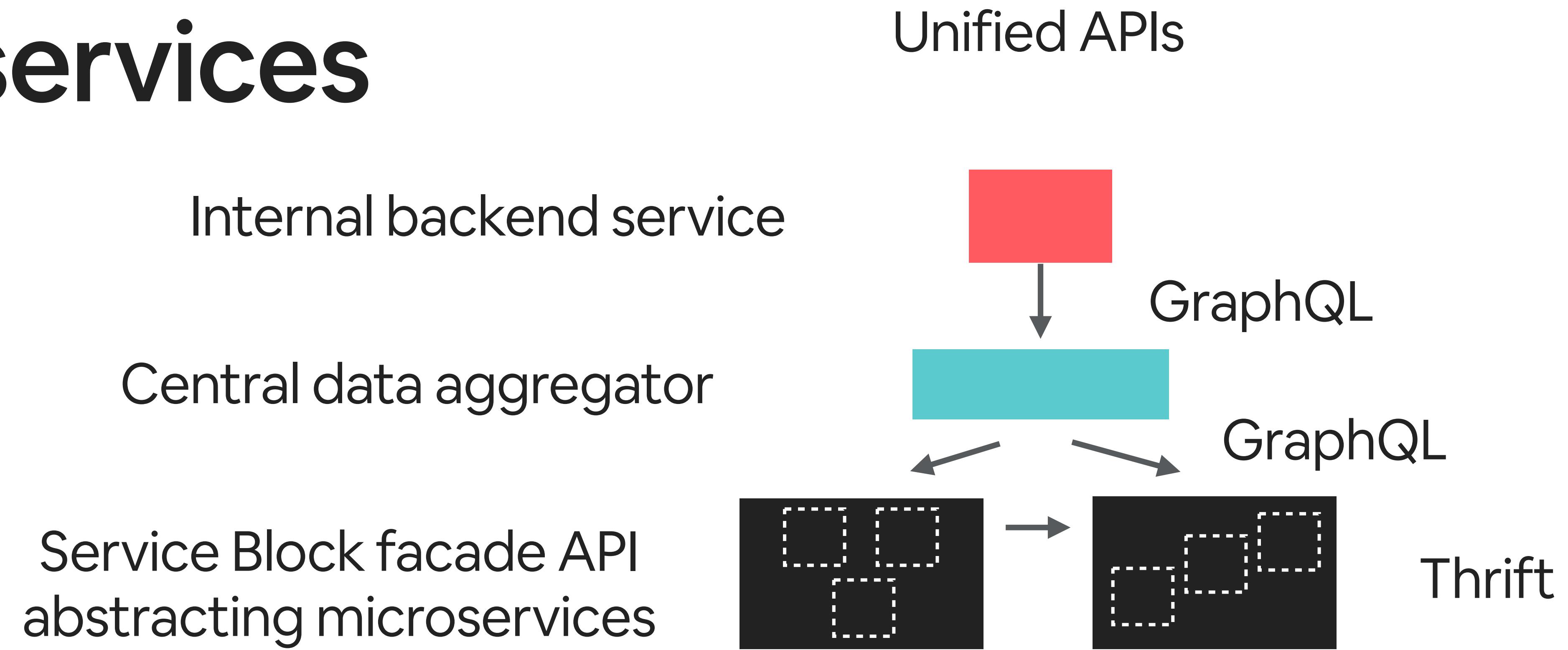
Micro + macroservices

2020 - present

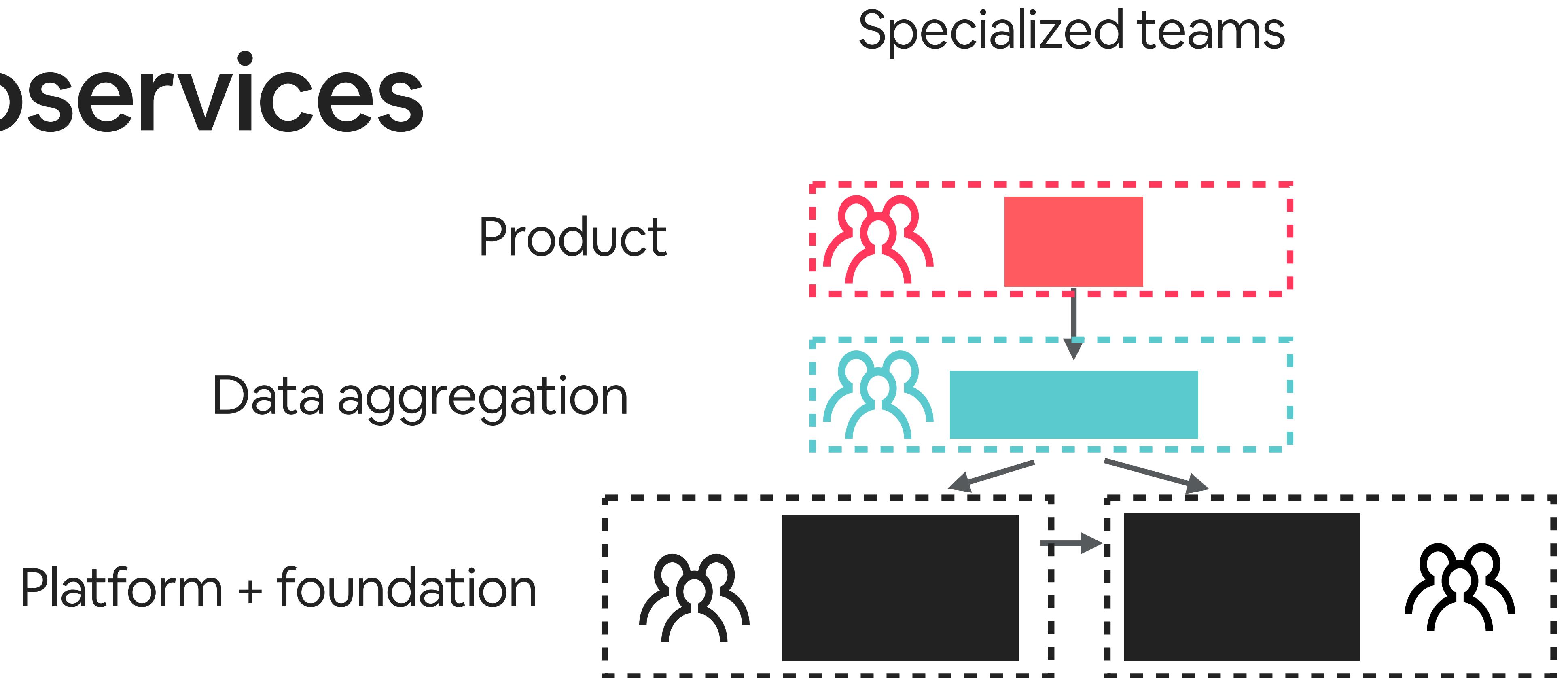
See previous QCon Plus talk for details!

@jessicamtai

Micro + macroservices



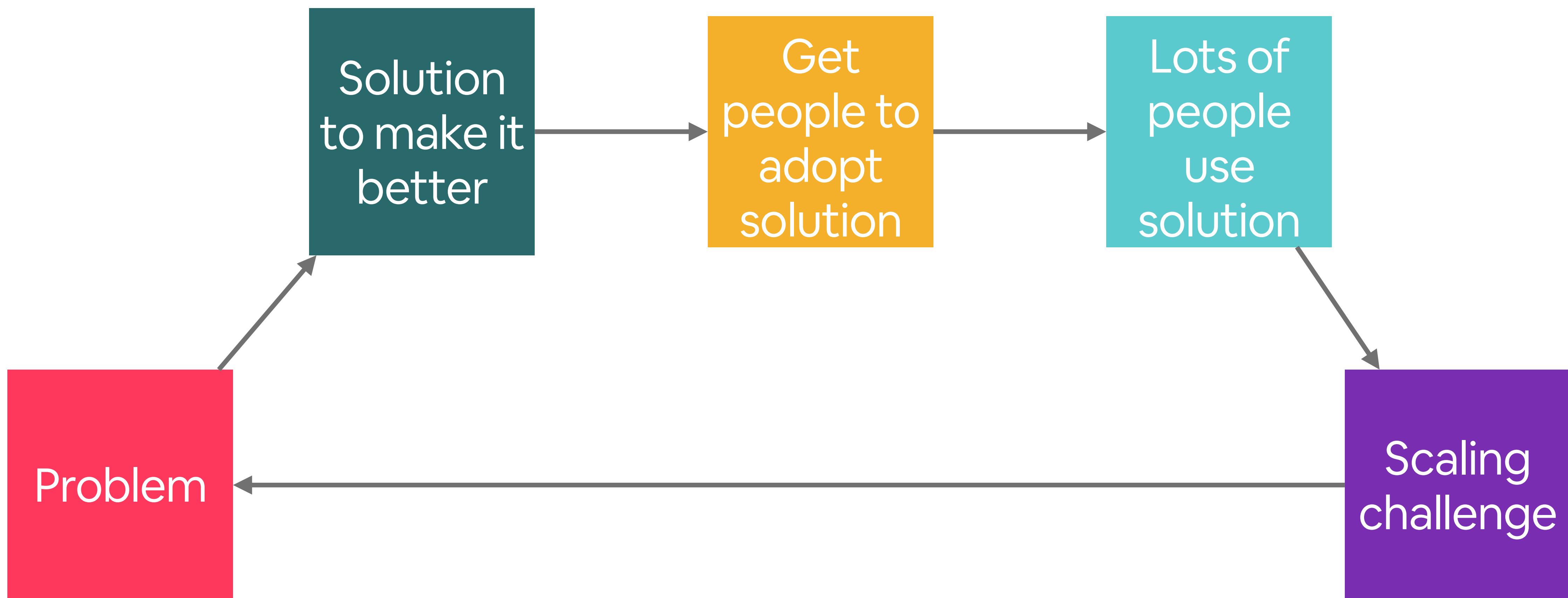
Micro + macroservices



Migration hits and misses

@jessicamtai

Migration challenge journey

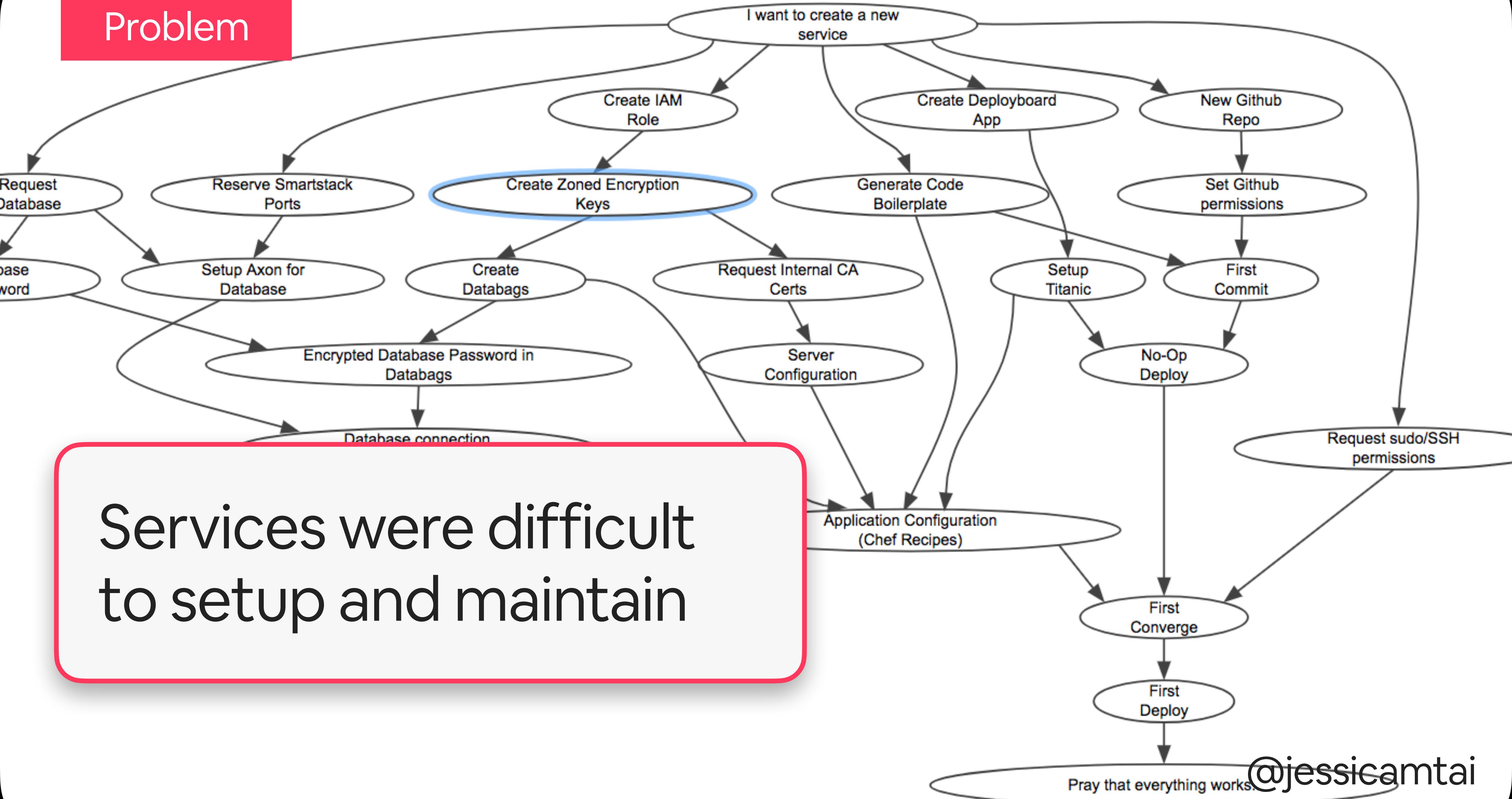


From Monolith to Services

Retrospective

@jessicamtai

Problem



Solution to make it better

Simplify with service infra
as code in single repo

@jessicamtai

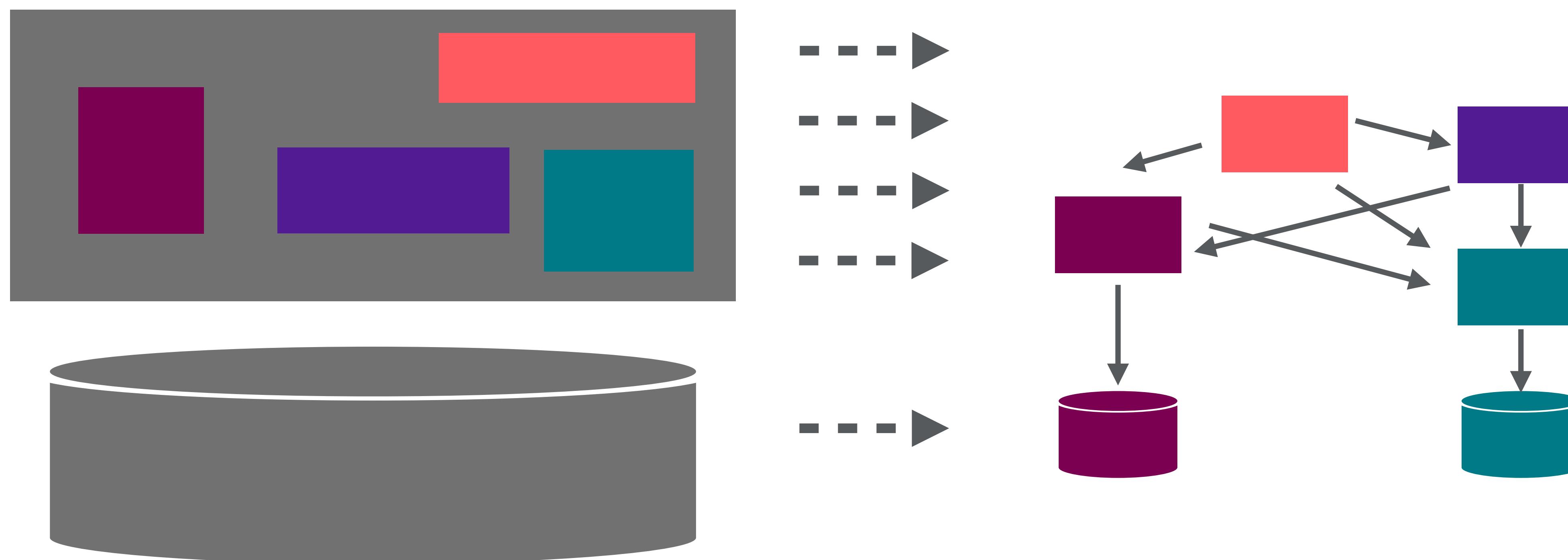
A photograph showing a close-up of a person's lower legs and feet. They are wearing dark green pants and brown boots with white soles. They are walking on a surface with horizontal stripes of yellow and grey. The background is slightly blurred.

Get people to adopt solution

Increase adoption with incremental,
parallel migration to microservices

@jessicamtai

Migrate to new services across stack in parallel

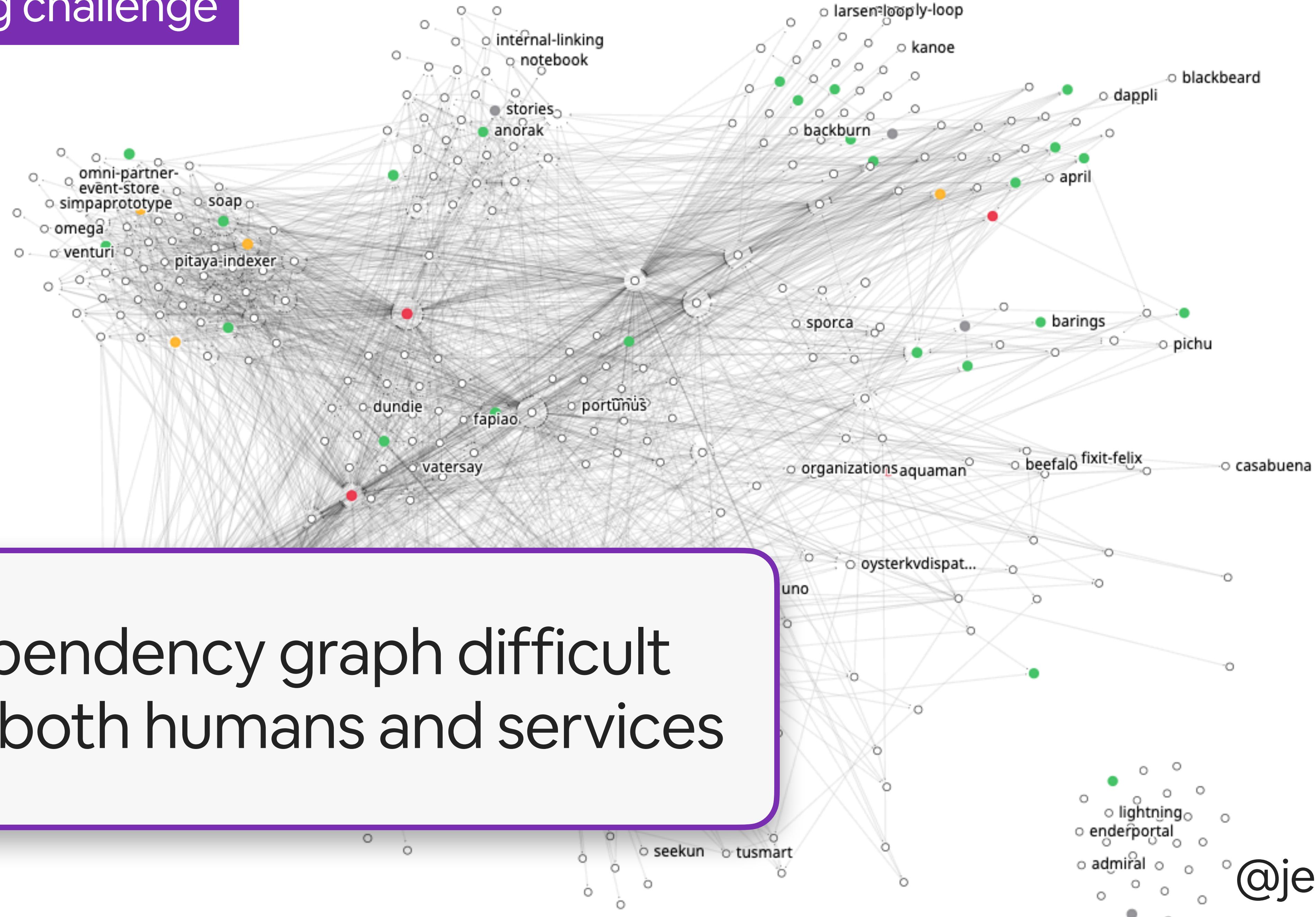


Lots of people use solution

Created more hundreds
more new services

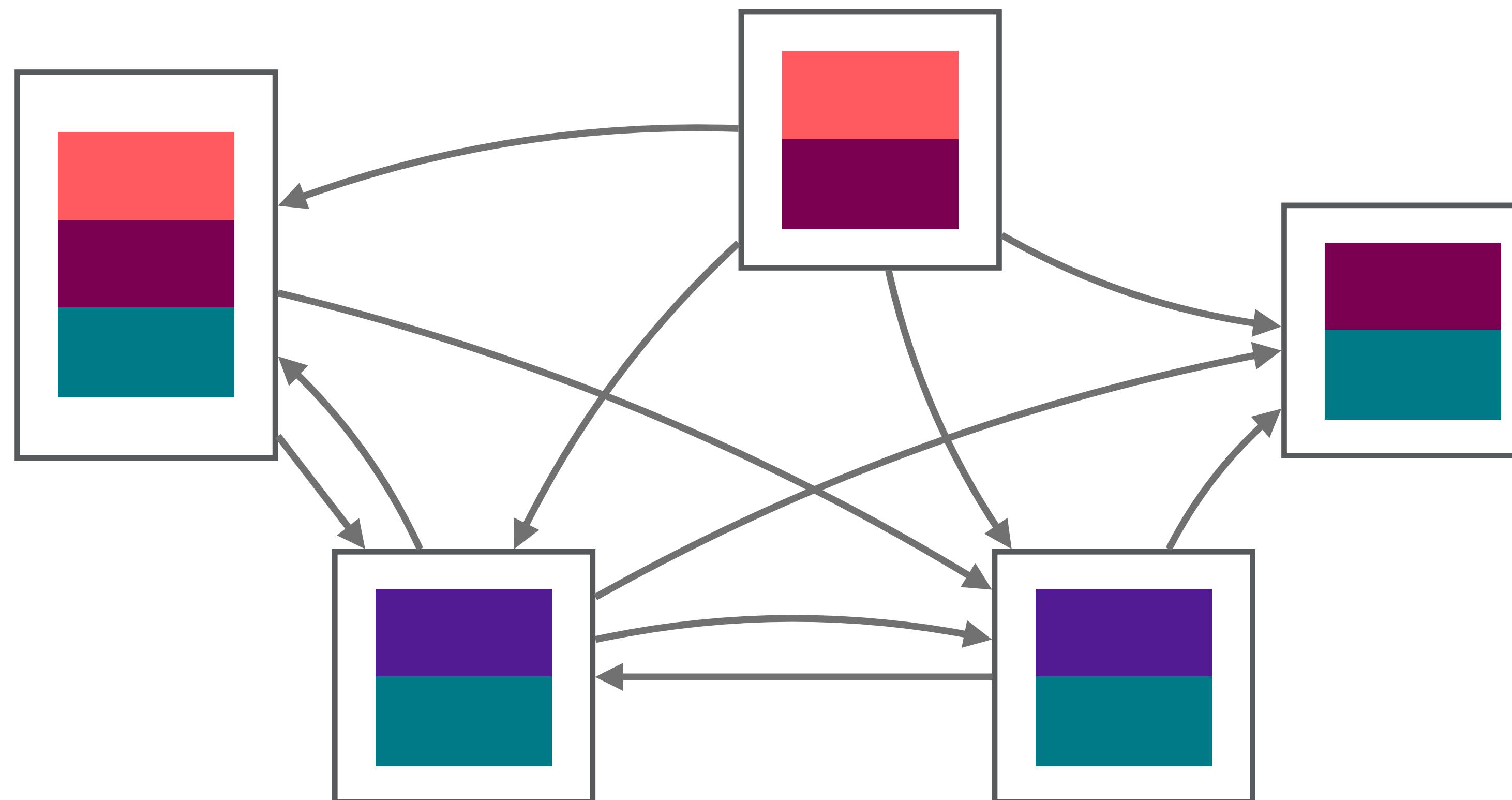
@jessicamtai

Scaling challenge



@jessicamtai

Quick dev velocity led to mixed functionality & APIs



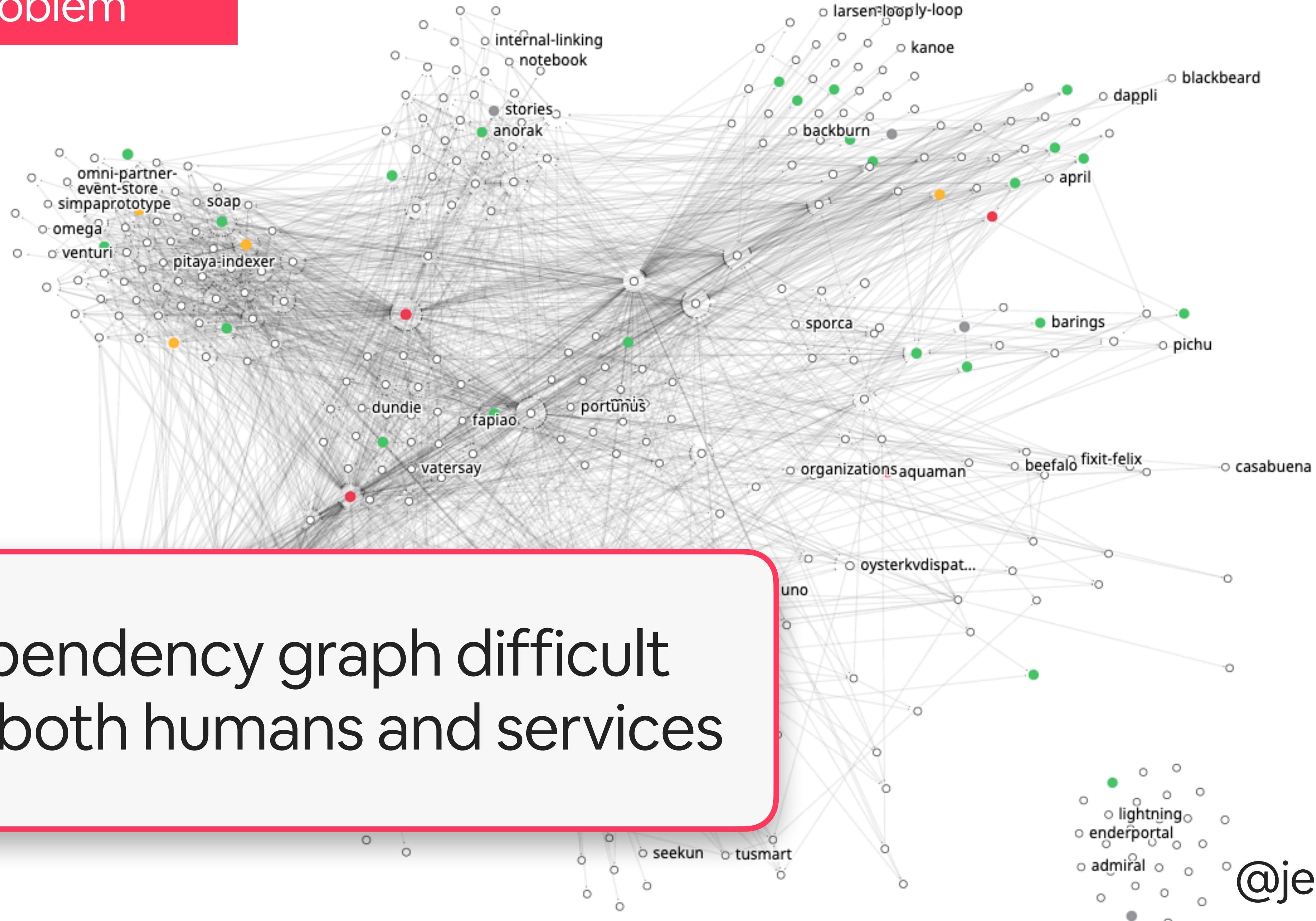
Services at Scale

Retrospective

@jessicamtai

Problem

Dependency graph difficult
for both humans and services



@jessicamtai

Solution to make it better



Improve productivity with ownership,
observability, tools, codegen

@jessicamtai



- Home
- Projects
- Teams
- Roles
- Namespaces
- Resources
- Members

project type primary language project group criticality category criticality tier

Projects

banana

Continuous integration test service for various Service Framework components. Act as server that receives requests. For client please go to: <https://git.musta.ch/airbnb/treehouse/tree/master/projects/dealer>.

Git backend service java public

#service-platform-community

@airbnb.com

banana-portal

APIX

Git backend service javascript

#service-platform-community

banana-stand

Seam based presentation service for banana and plantain

Git backend service typescript

public #node-service-toolkit

codelab-banana

banana

Git backend service other public

@jessicamtai



RENDERED

RAW JSON



demo

Demo Service

JAVA

slack: #demo-goalie**dashboards:** Standard Service Metrics, Standard Client Metrics, Demo Dashboard**doc link:** Demo tech design doc**description:** The Demo Service owns test data in MySQL and Redis.

Thrift annotations



Search Demo schema (51 structs, 478 fields) ...

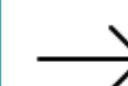


loadSomeData

RPC

/loadSomeData batch endpoint

LoadSomeDataRequest



LoadSomeDataResponse

demo data LoadSomeDataRequest

@jessicamtai

STRUCT

★ Standard Server-Side Metrics ▾

caller

0/7



Search...

Save or select views

\$service banana

\$chef_role * *

\$k8s_app * *

\$host * *

\$method * *

\$tier production

\$caller * *

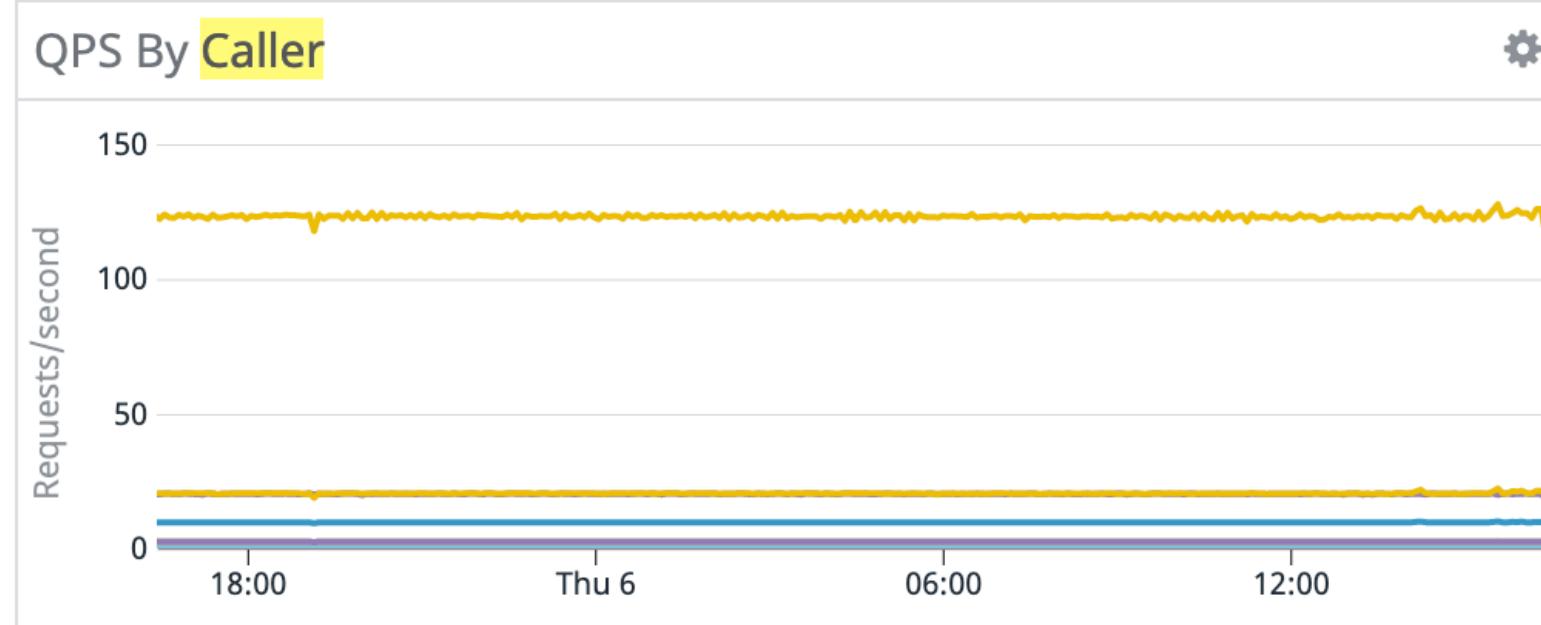
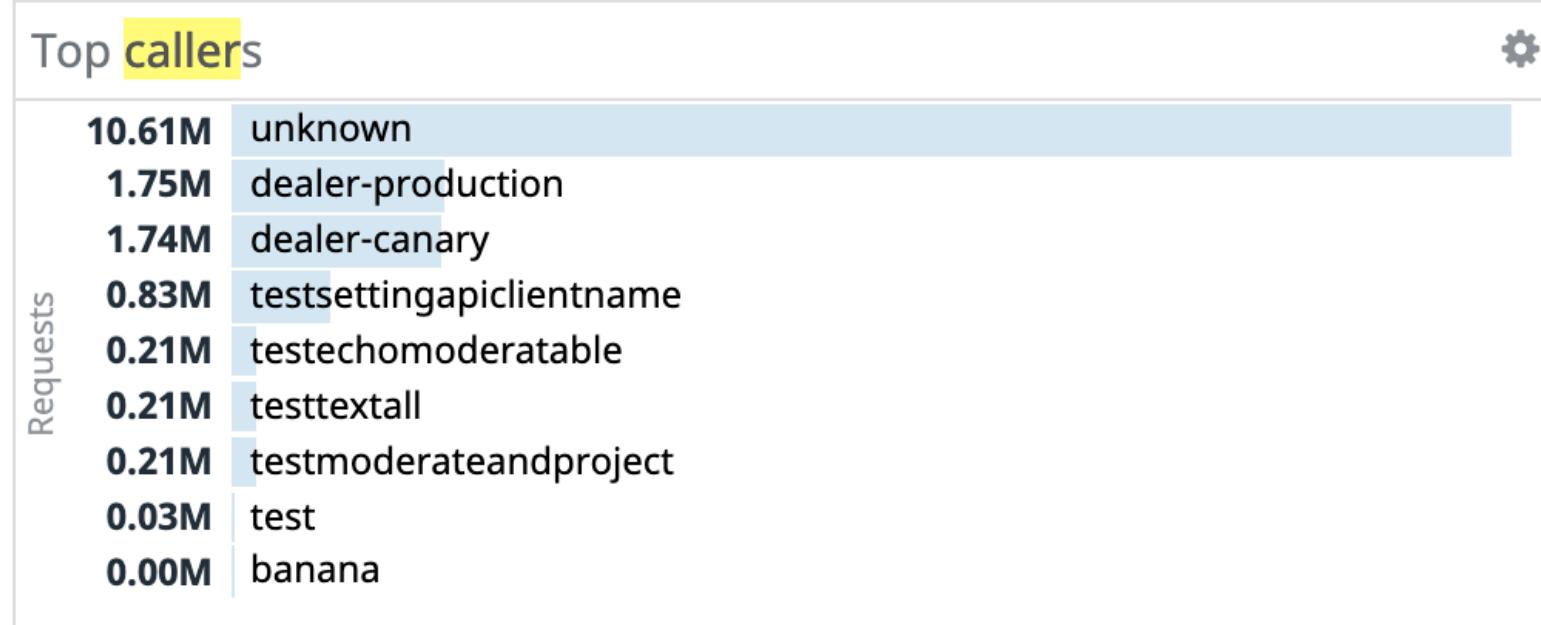
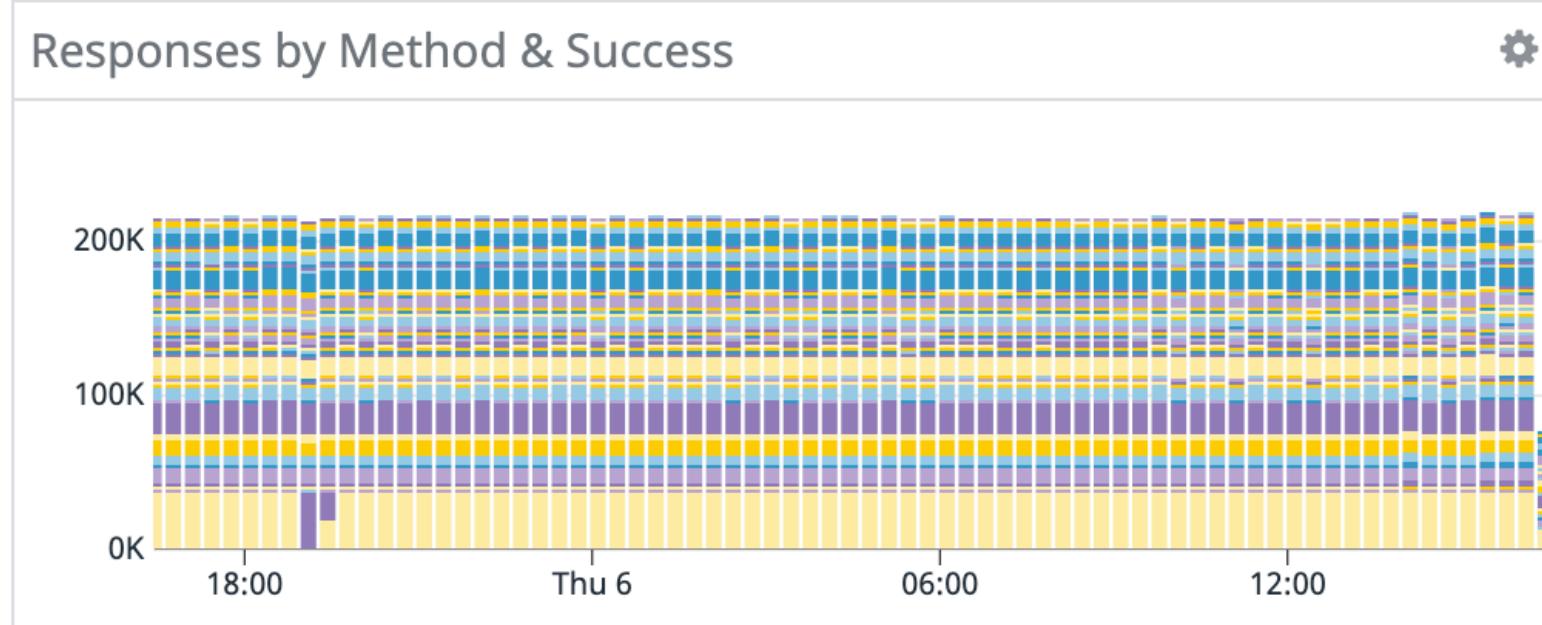
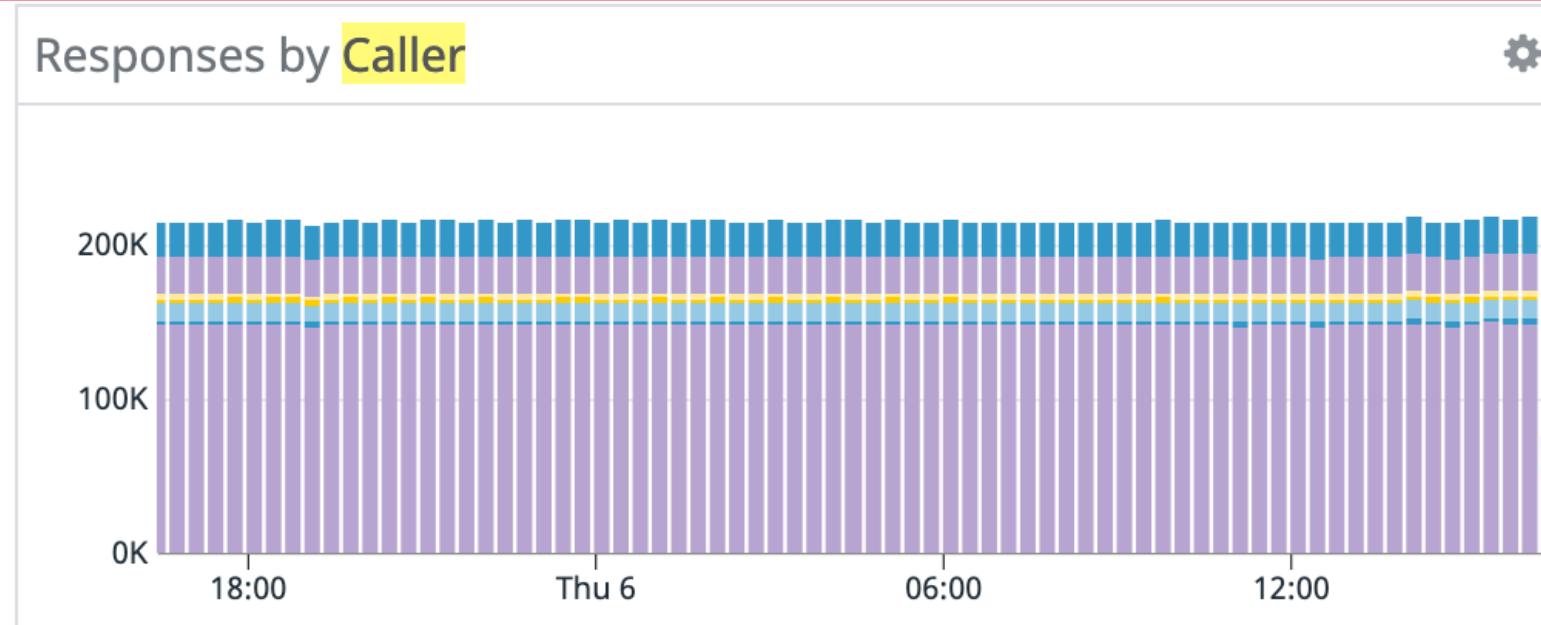
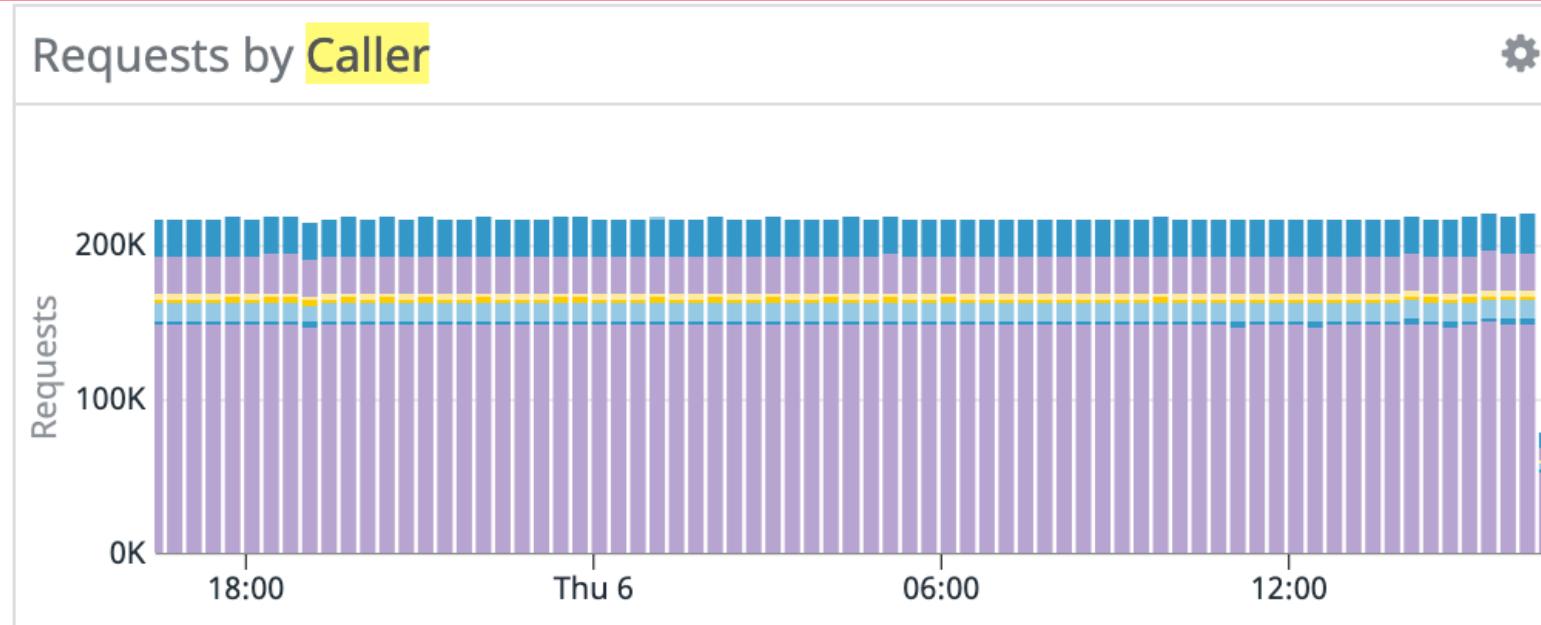
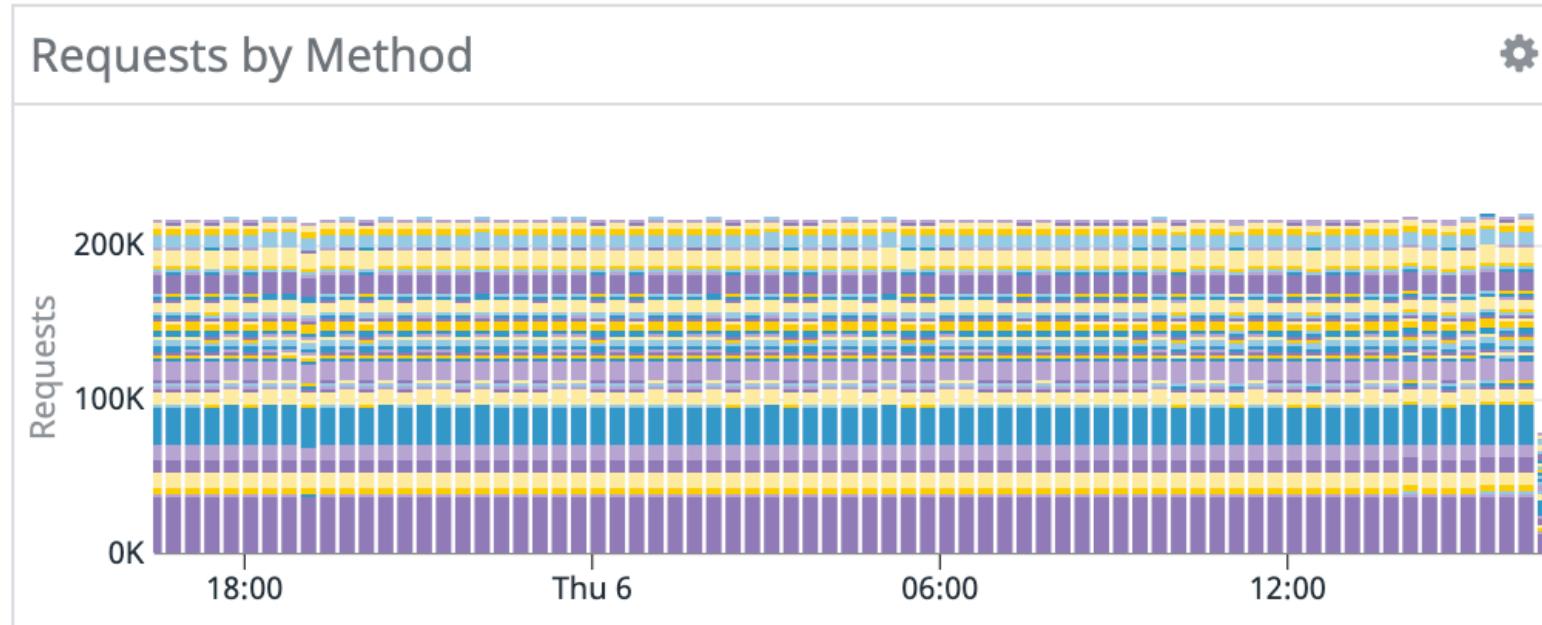
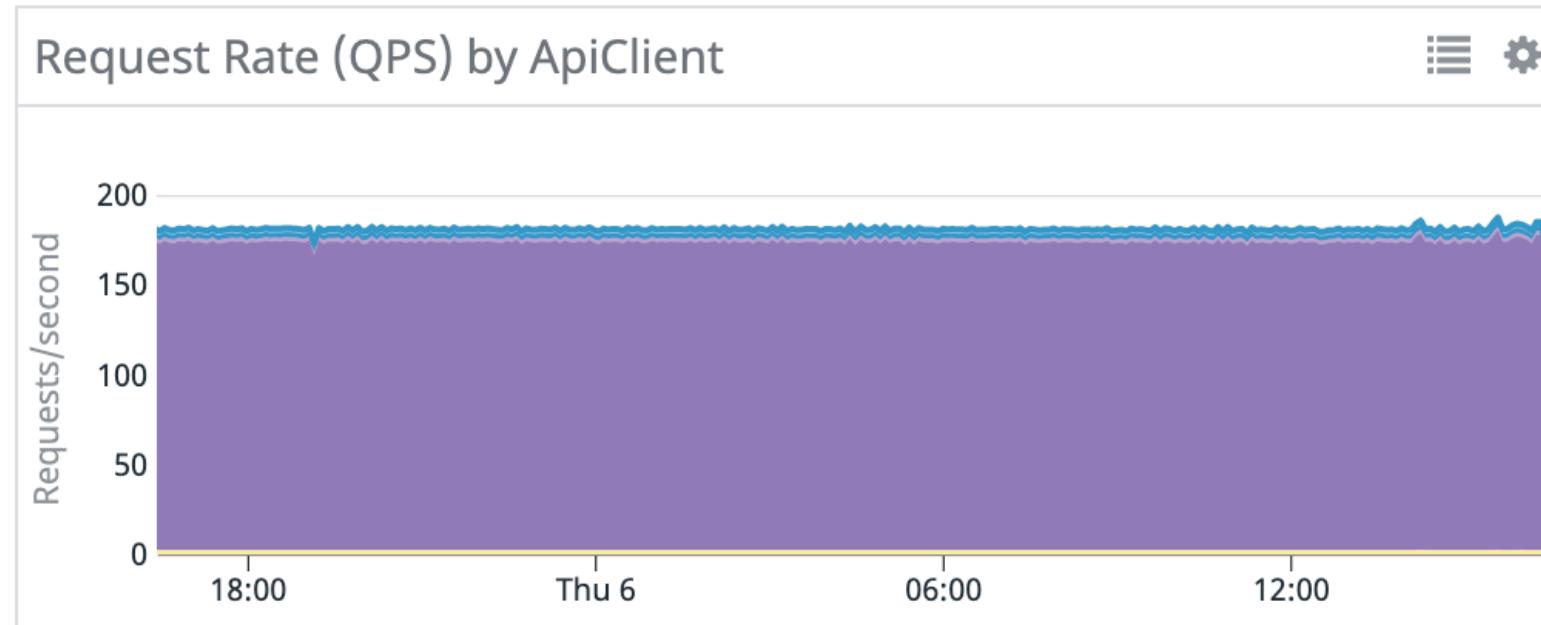
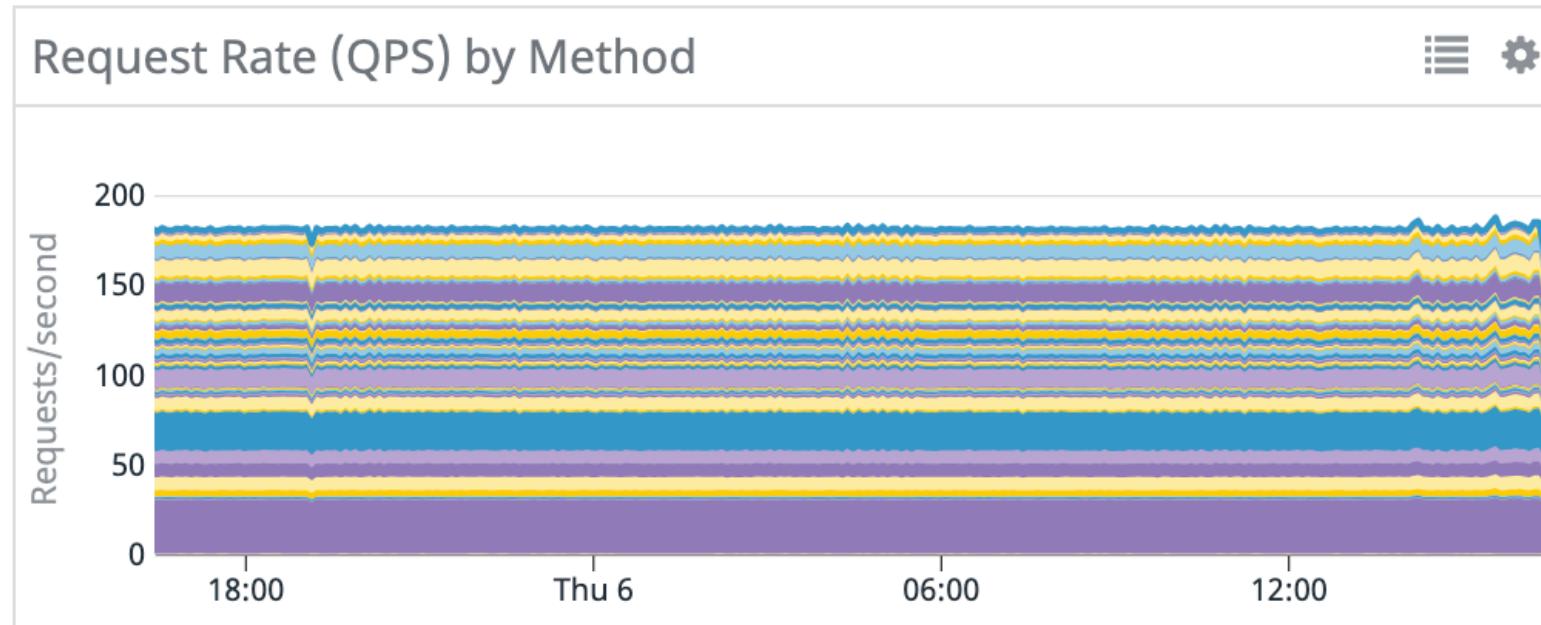
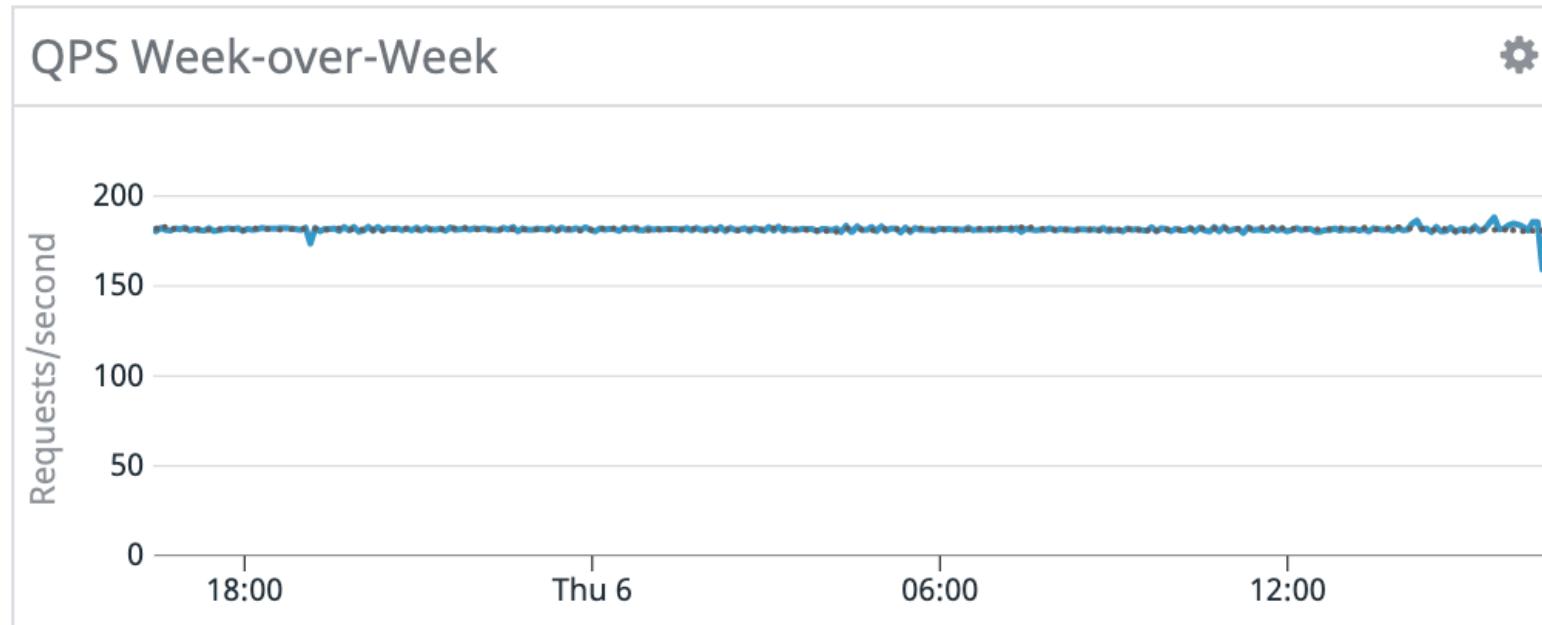
\$api_client_name * *

\$aws_account * *

\$pod * *



▼ Requests & Responses

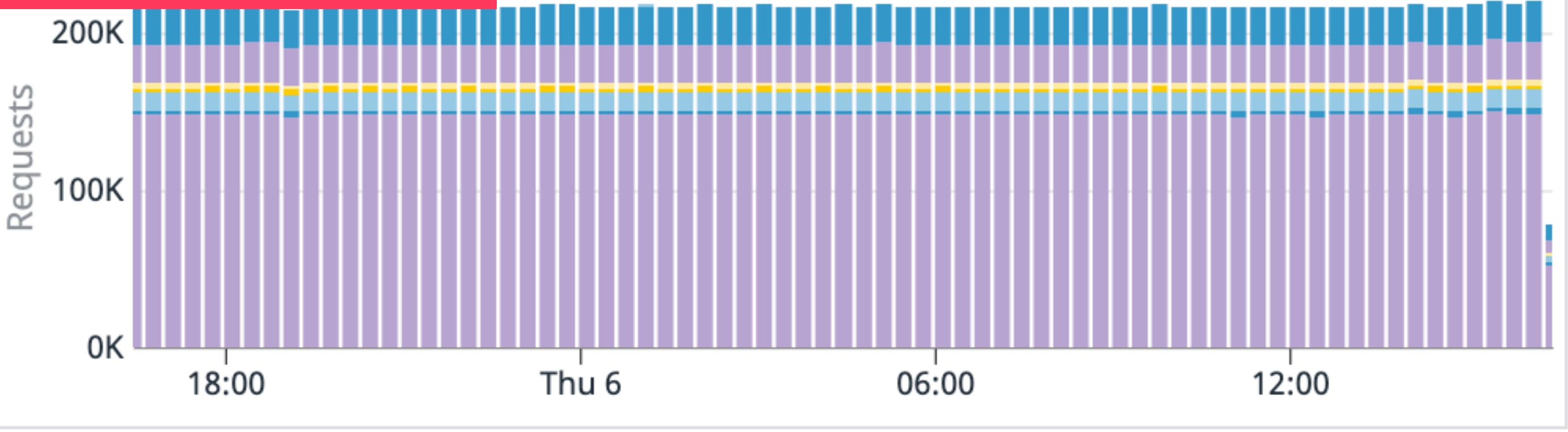
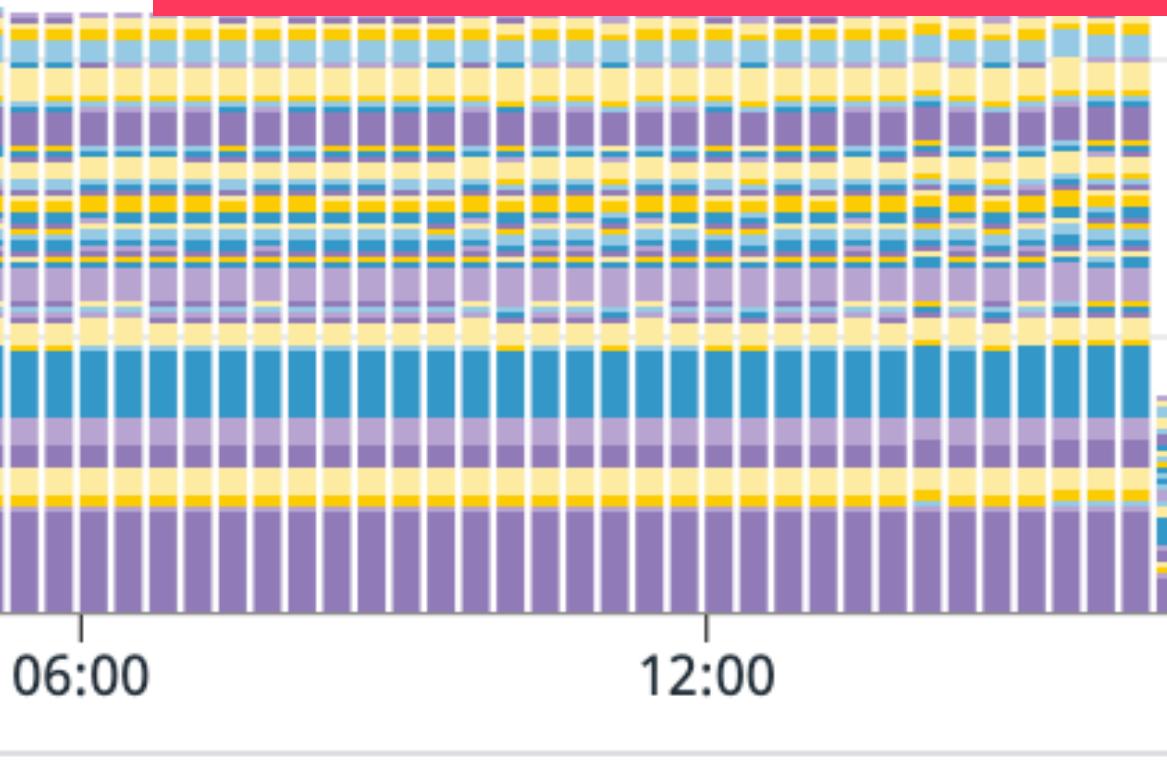


of callers

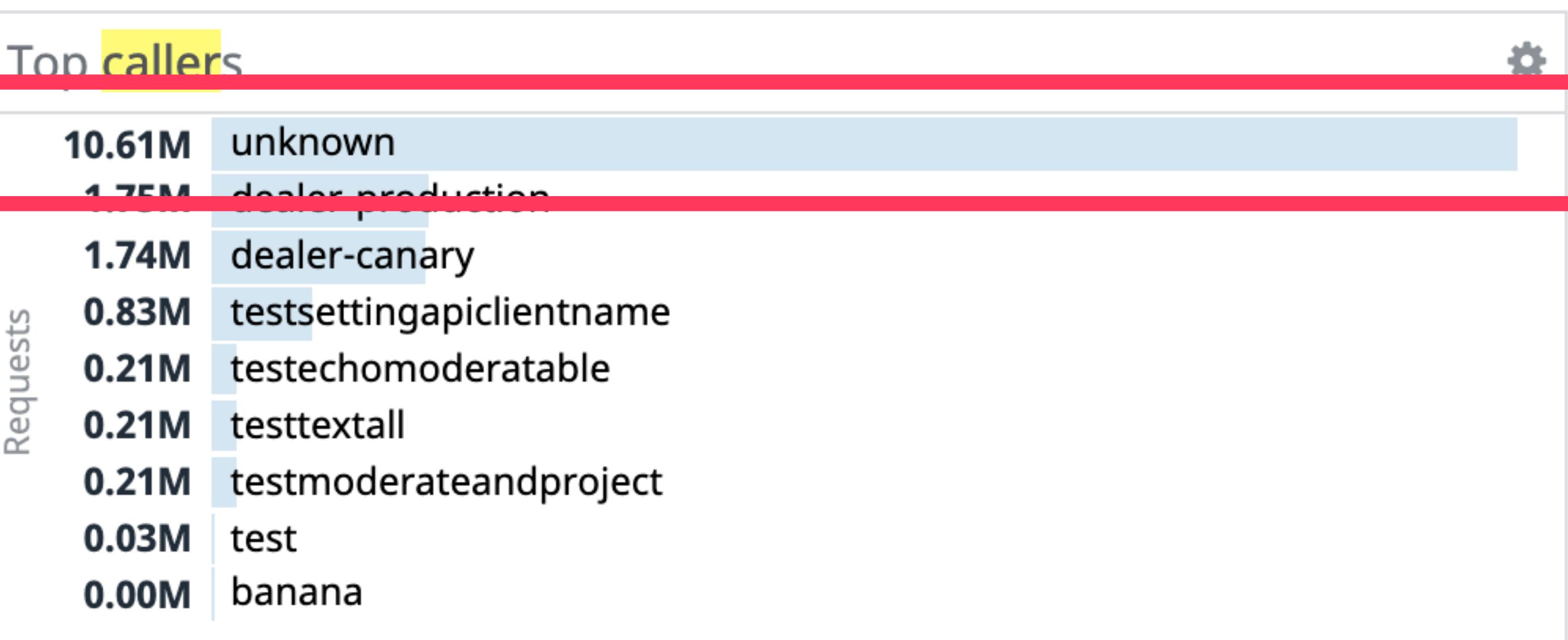
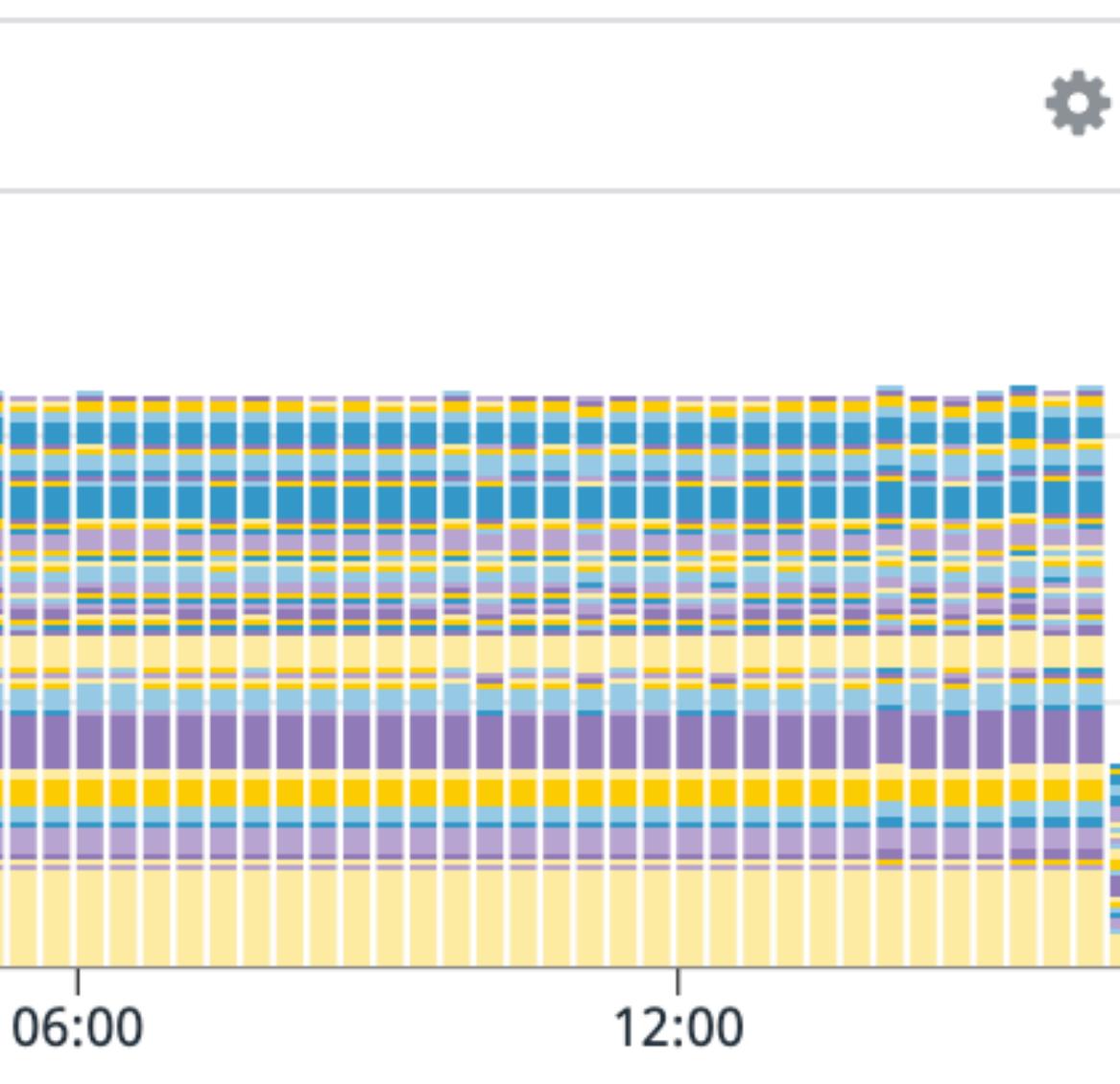
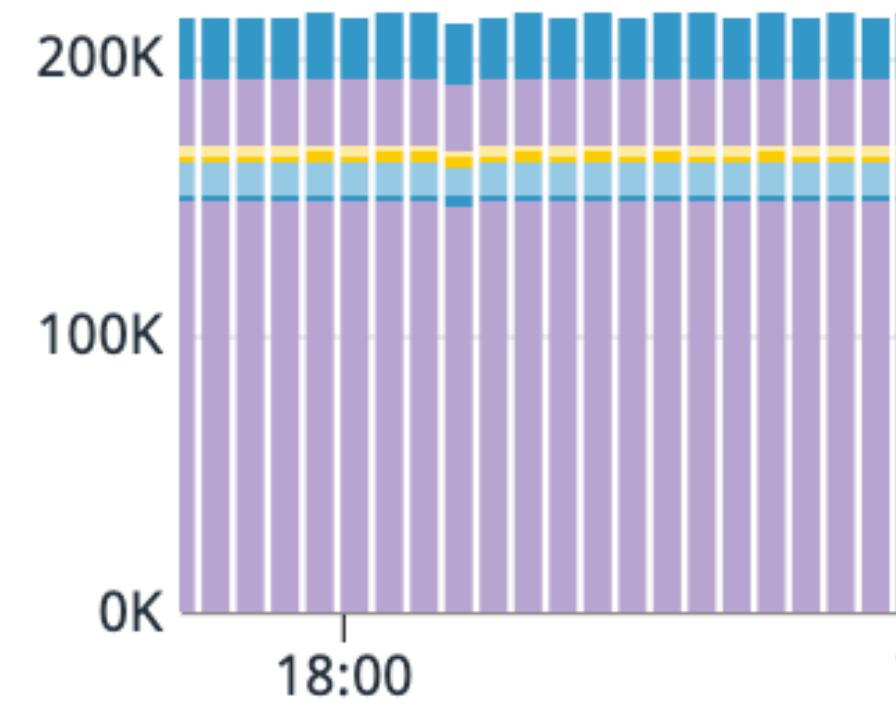


@jessicamtai

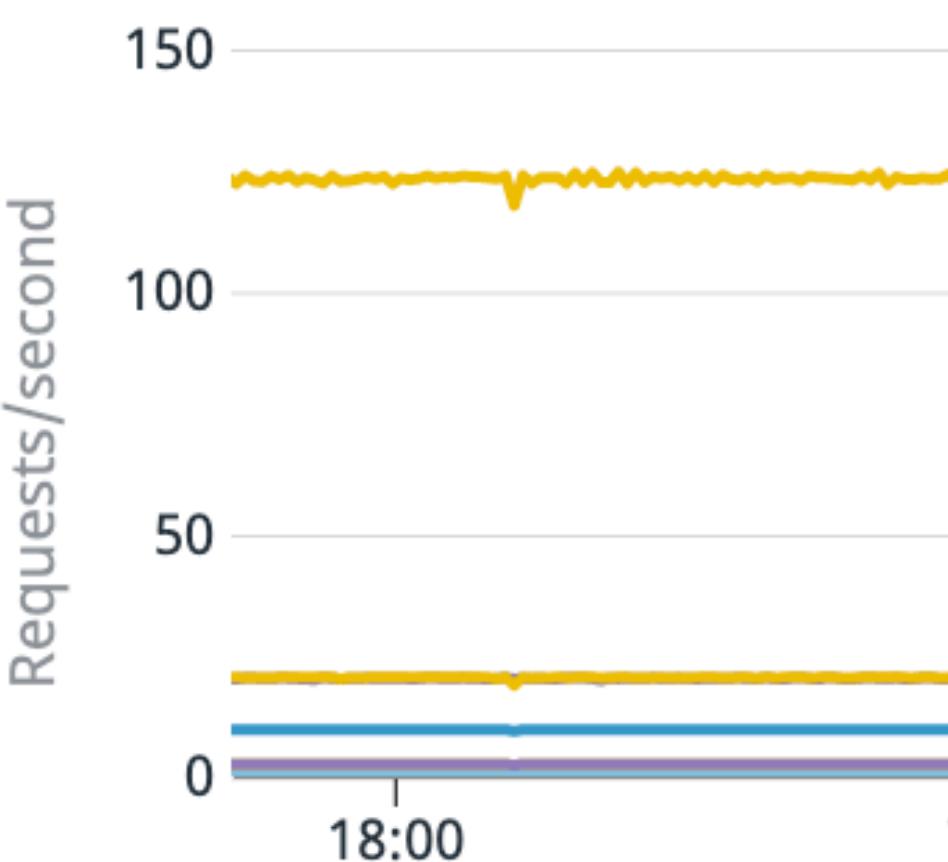
Mini challenge - standardize service request context



Responses by Caller



QPS By Caller

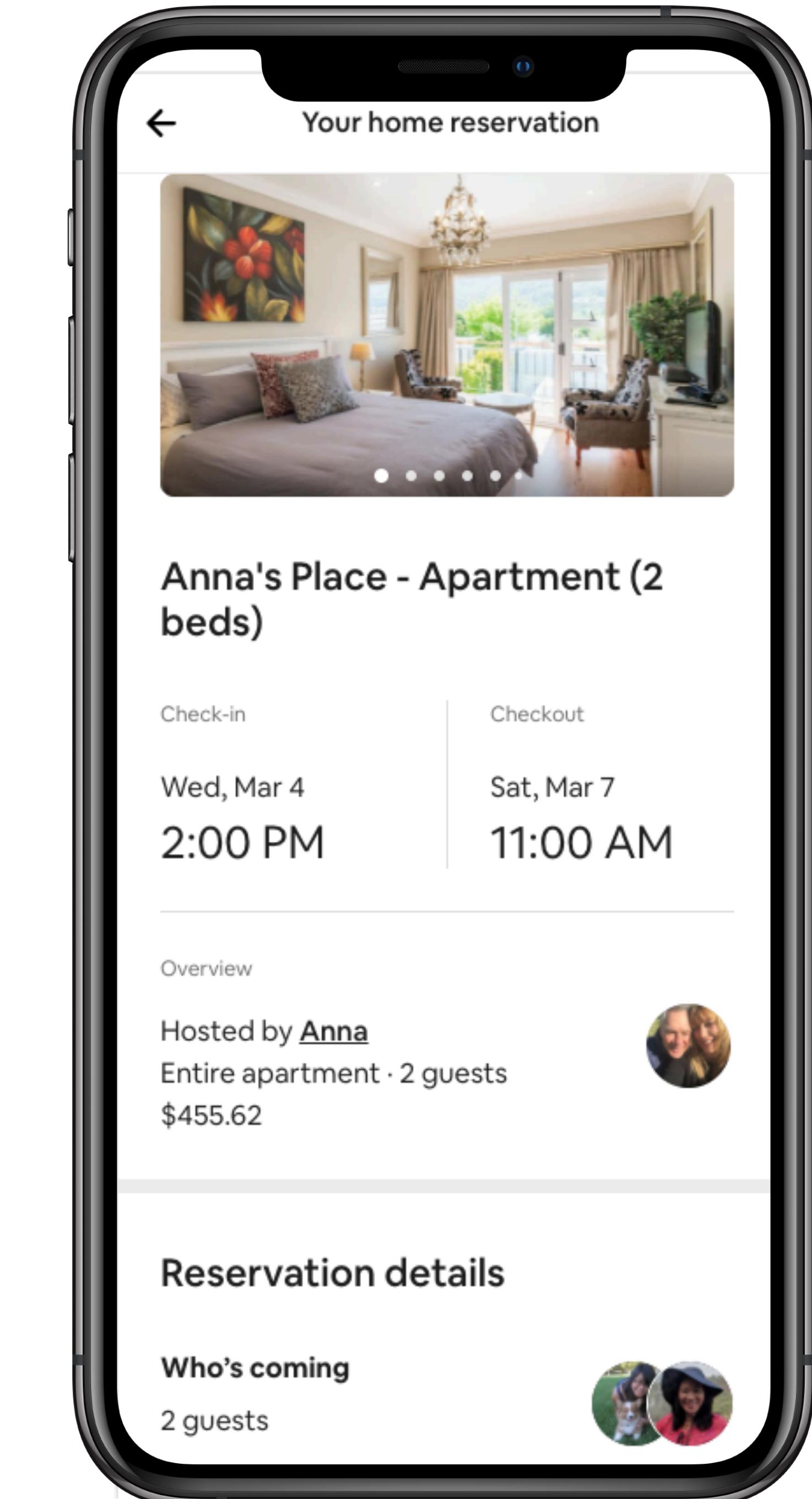


@jessicamtai

Solution to make it better

Replace bespoke Thrift schemas
with unified GraphQL

```
type Reservation {  
    checkInDate: Date  
    numberOfGuests: Int  
    ...  
}
```



Get people to adopt solution

Improve velocity with annotations & codegen

```
type Reservation {  
    @owners(lists: "team-123")  
    @serviceBackedNode(  
        service: "ExampleService"  
        methodName: "/example/endpoint"  
        ...  
    )  
    checkInDate: Date @dataClassification(level: "other_data")  
    numberOfGuests: Int @dataClassification(level: "other_data")  
    @privacy(...)  
}
```

Get people to adopt solution

Improve velocity with annotations & codegen

```
type Reservation {  
    @owners(lists: "team-123")  
    @serviceBackedNode(  
        service: "ExampleService"  
        methodName: "/example/endpoint"  
        ...  
    )  
    checkInDate: Date @dataClassification(level: "other_data")  
    numberOfGuests: Int @dataClassification(level: "other_data")  
    @privacy(...)  
}
```

Alerts, mandatory code reviewers

Get people to adopt solution

Improve velocity with annotations & codegen

```
type Reservation {  
    @owners(lists: "team-123")  
    @serviceBackedNode(  
        service: "ExampleService"  
        methodName: "/example/endpoint"  
        ...  
    )  
    checkInDate: Date @dataClassification(level: "other_data")  
    numberOfGuests: Int @dataClassification(level: "other_data")  
    @privacy(...)  
}
```

**Resolver to call
internal microservice**

Get people to adopt solution

Improve velocity with annotations & codegen

```
type Reservation {  
    @owners(lists: "team-123")  
    @serviceBackedNode(  
        service: "ExampleService"  
        methodName: "/example/endpoint"  
        ...  
    )  
    checkInDate: Date @dataClassification(level: "other_data")  
    numberOfGuests: Int @dataClassification(level: "other_data")  
    @privacy(...)  
}
```

**Permission checks to
internal service**

Lots of people use solution

Migrated hundreds of data entities
to unified GraphQL schema

@jessicamtai

Scaling challenge

Slower build and deploy times
for central data aggregator

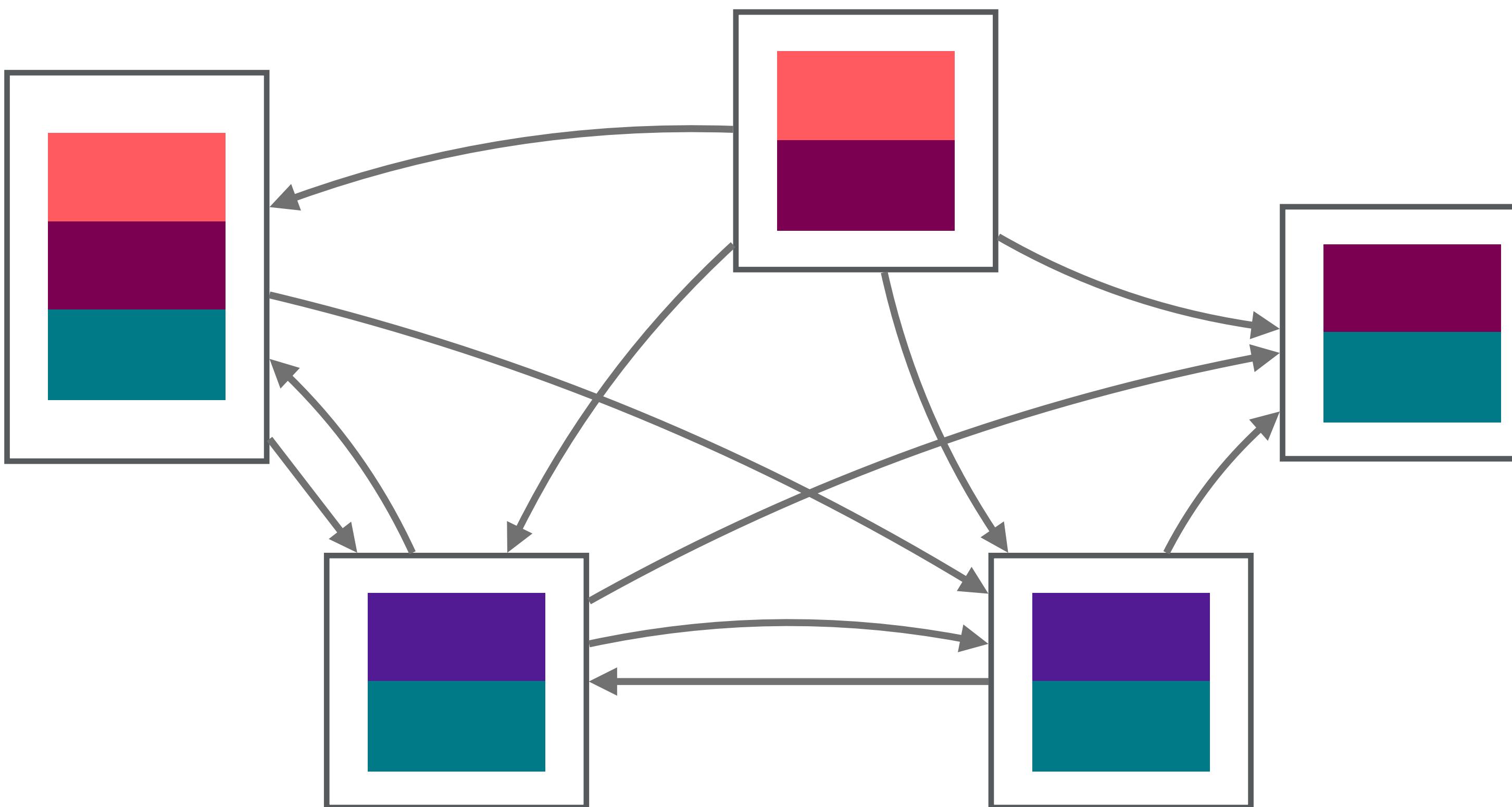


What's next?

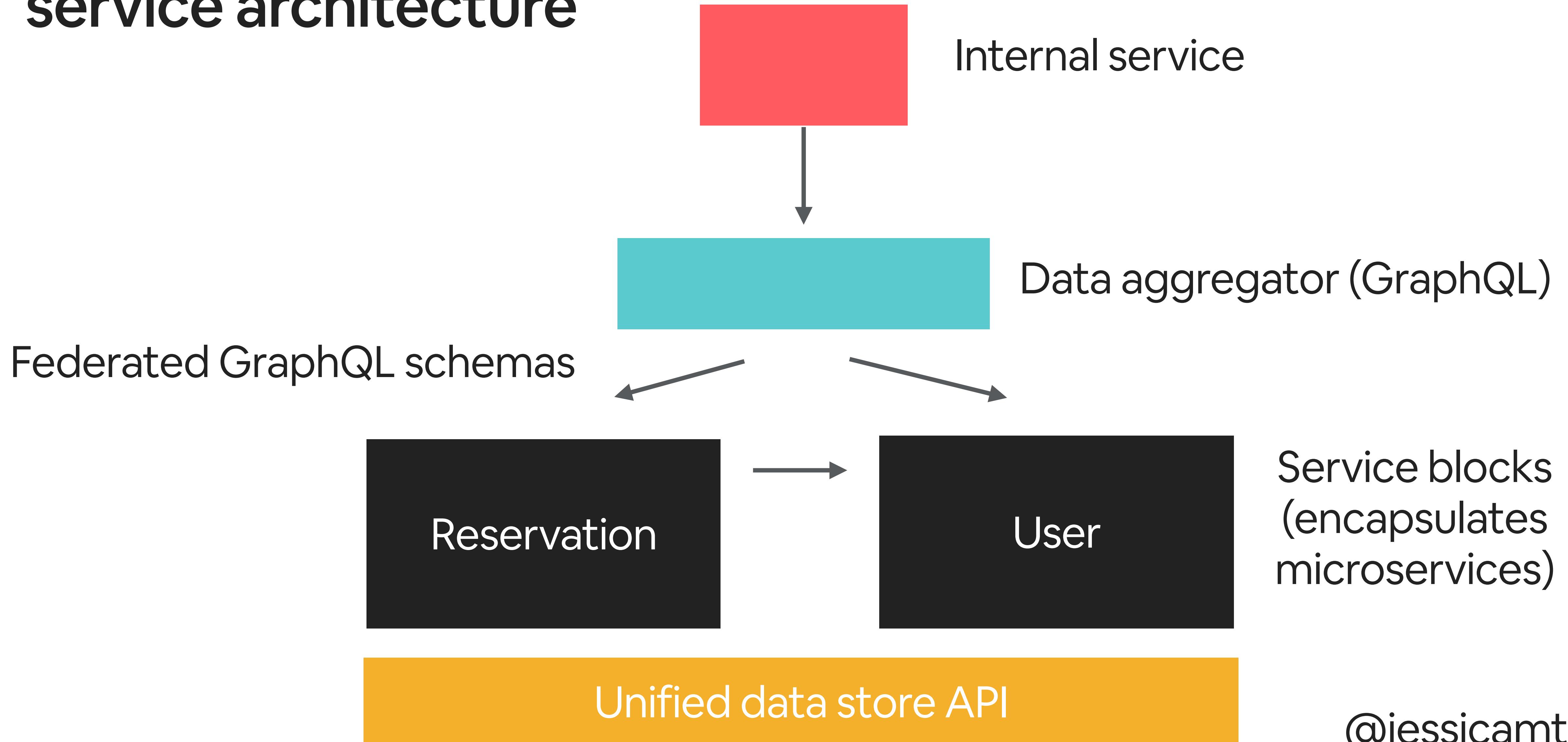
@jessicamtai

Airbnb is moving towards a hybrid mode -
aggregators abstracting microservices.

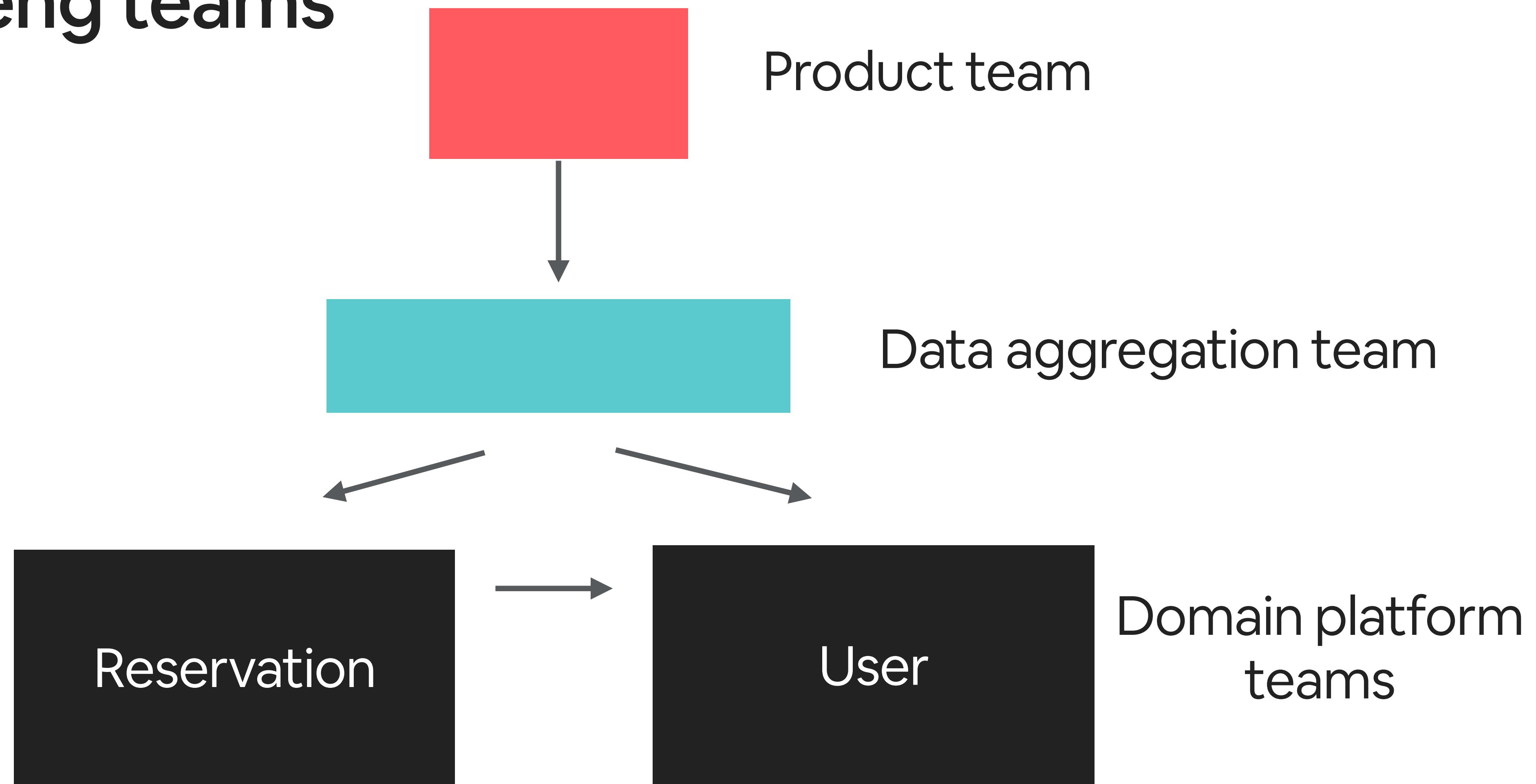
Migrate away from hundreds of free-range services



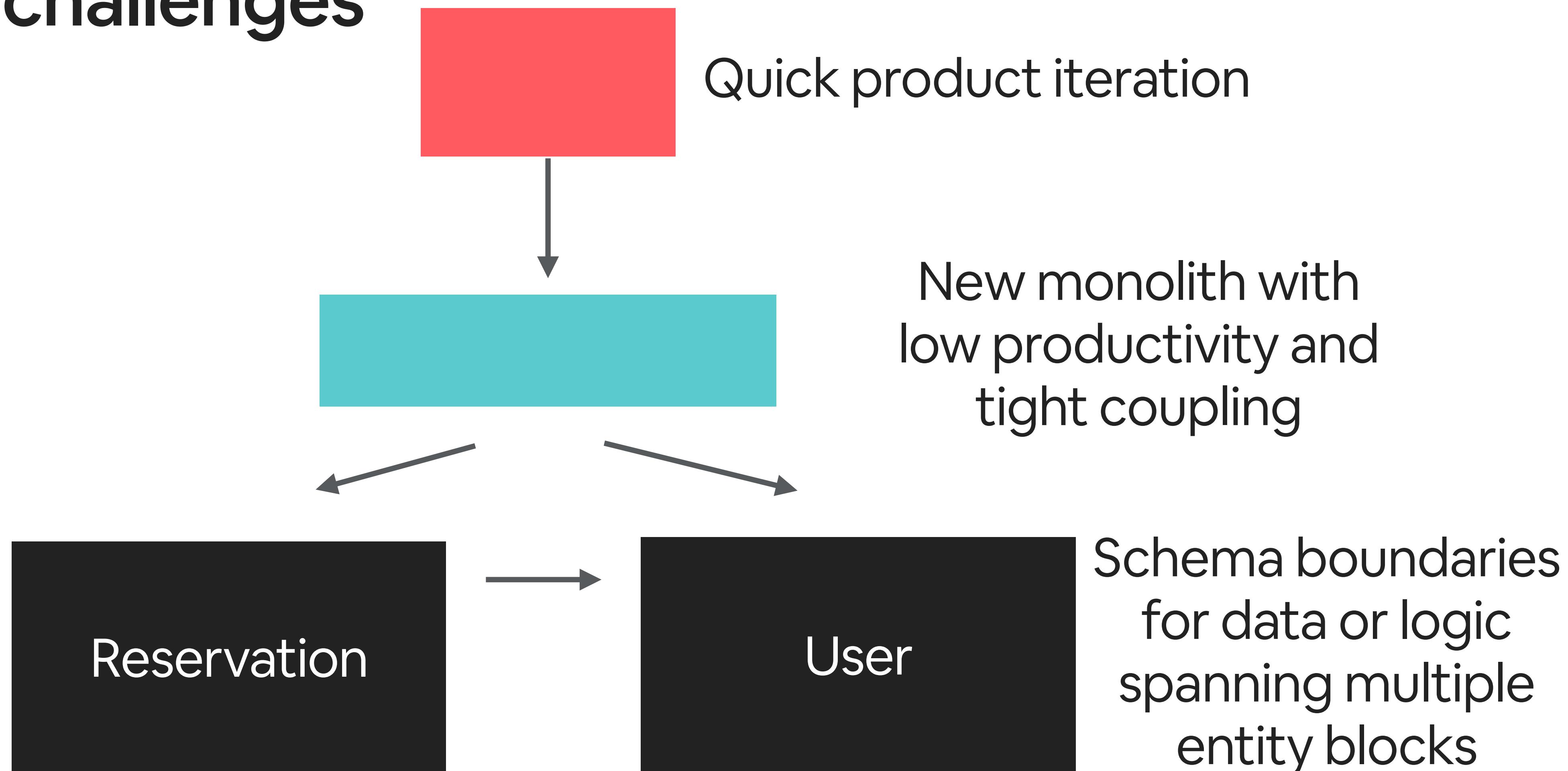
Enforce paved path for micro+macro service architecture



Clarify responsibilities with specialized eng teams



Get ahead of foreseeable challenges





Migrations on track, but lots
of parallel tracks

**Deprecation of the monolith takes
a very long time.**

Deprecation working group

Mobile app depreciation >12 months

Elevate & track monolith tech debt

Sunset low usage endpoints

Long-term ownership

Recognize depreciation as impactful



@jessicamtai

Takeaways

Improve human productivity

Restructure teams to collaborate in new architecture

Invest in developer productivity through tools, automation

Increase focus with central domains and platform teams

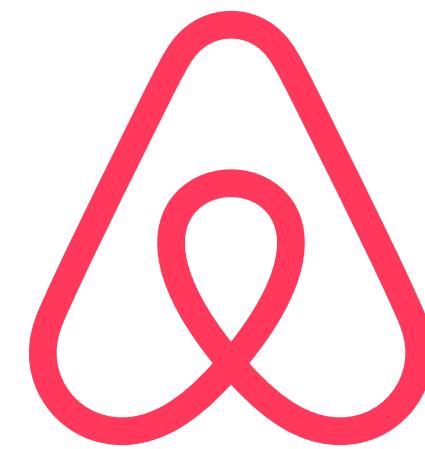
Expect new challenges to appear

@jessicamtai

Decentralizing into
microservices created
both new technical and
human challenges...

But many of the growing
pains were necessary
parts of our evolving
journey.

QCON PLUS 2021



@JESSICAMTAI

GITHUB®, the GITHUB® logo design, OCTOCAT® and the OCTOCAT® logo design are exclusive trademarks registered in the United States by GitHub, Inc.
Slack is a registered trademark and service mark of Slack Technologies, Inc.

Kubernetes® is the trademark of The Linux Foundation.

AWS is a trademark of Amazon.com, Inc. or its affiliates in the United States and/or other countries.

All trademarks are the properties of their respective owners. Any use of these are for identification purposes only and do not imply sponsorship or endorsement.