

Uber's Journey into Microservices

Emily Reinhold, Software Engineer, Uber

JUNE 15, 2016

The Uber logo, consisting of the word "UBER" in white, uppercase, sans-serif font, centered within a black square.

UBER



Background

Our Monolith

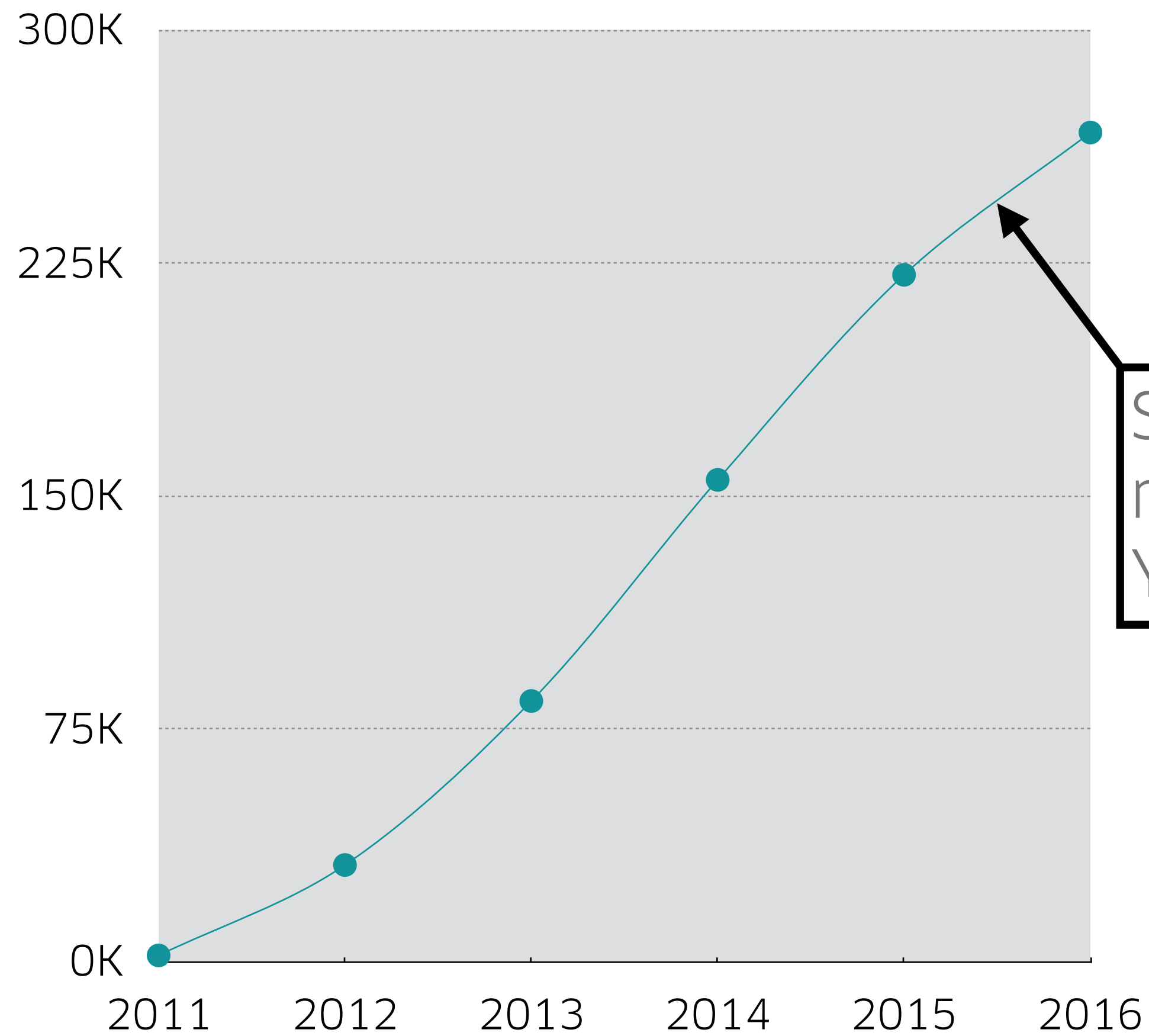
Fun Facts

▶ “API”

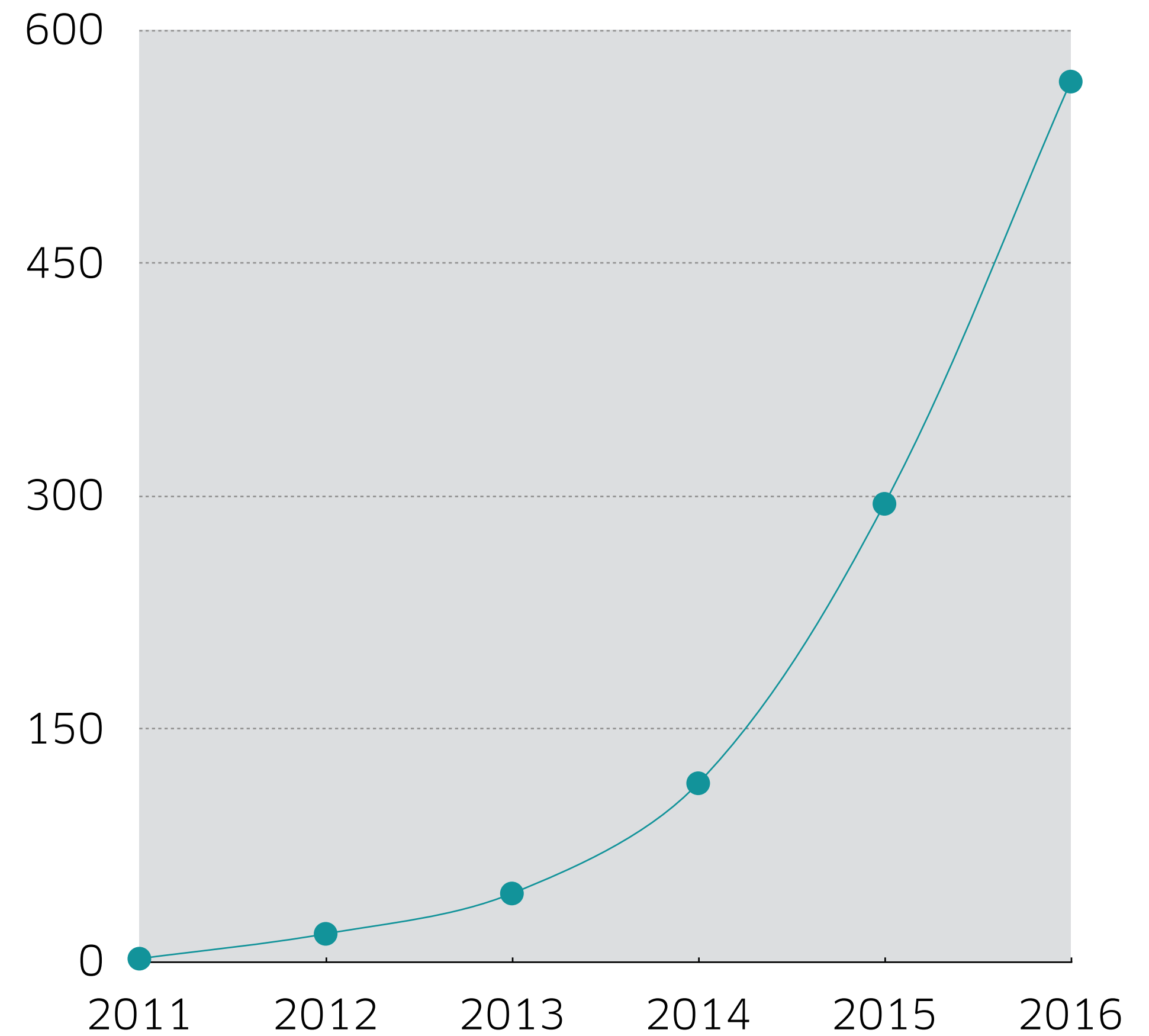
▶ Started January 2011

API Trends

Lines of Code over Time



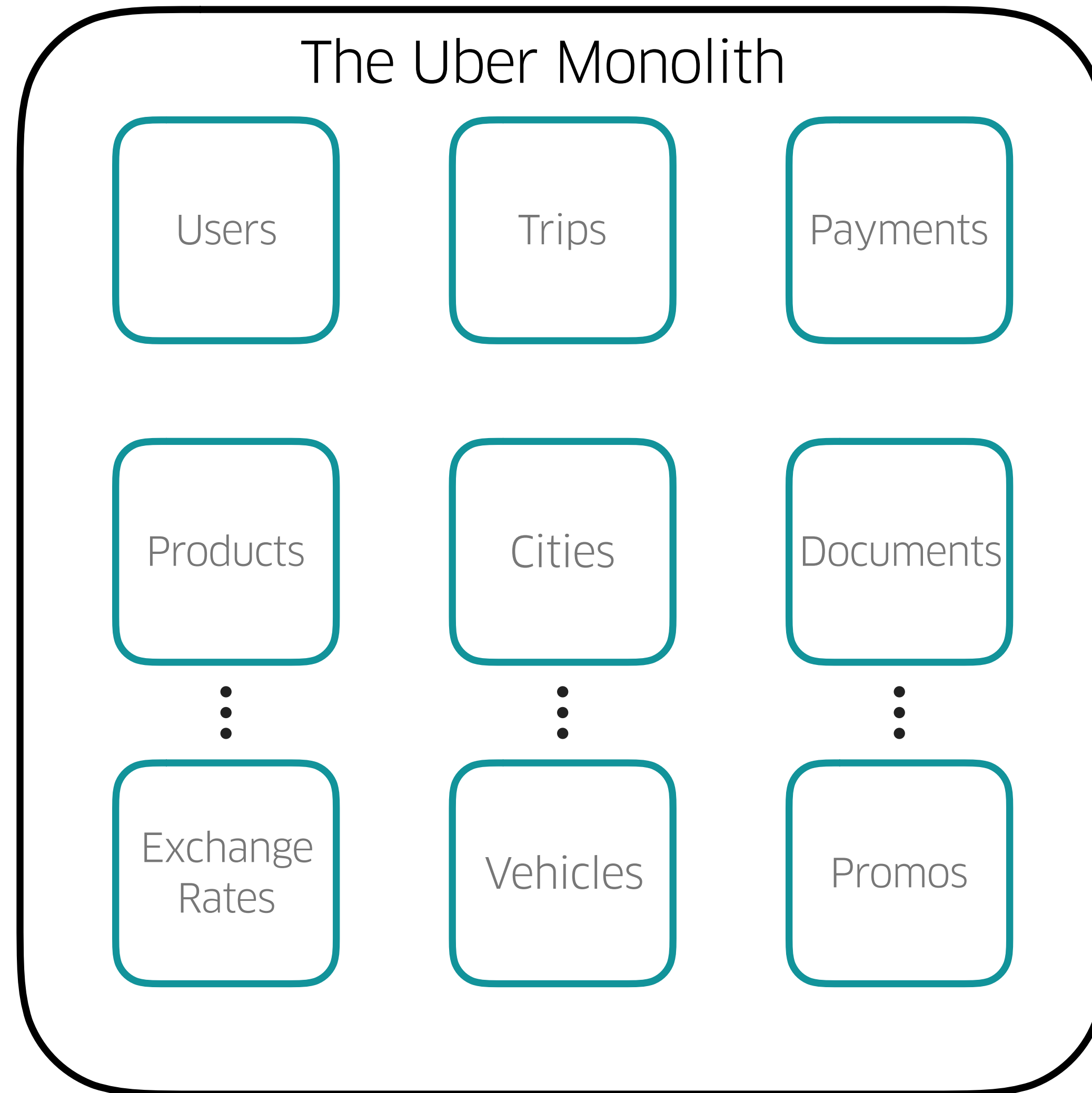
Unique Authors over Time



Fun Facts

- ▶ **“API”**
- ▶ **Started January 2011**
- ▶ **Access to core data models**

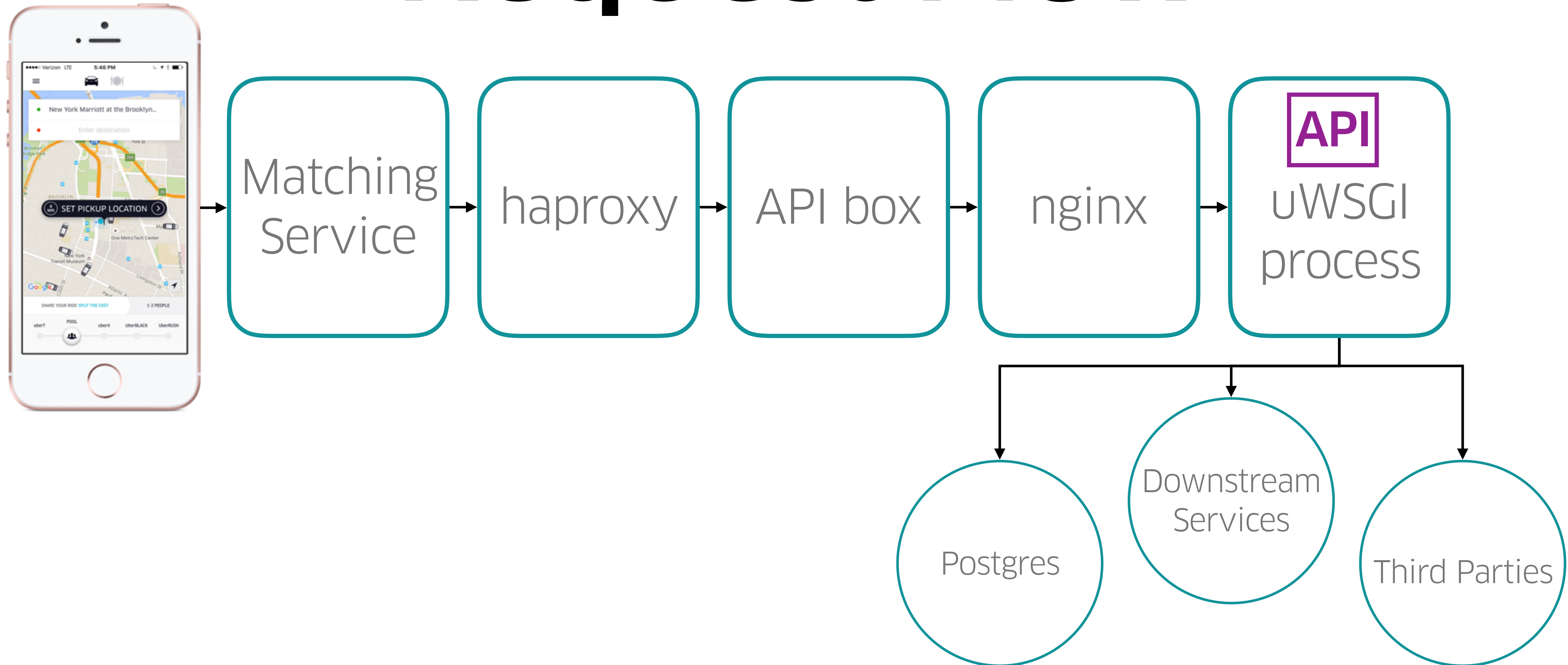
Core Data Models



Technical Details

- ▶ **Postgresql**
- ▶ **Python 2.7**
- ▶ **HTTP/JSON**
- ▶ **uWSGI**

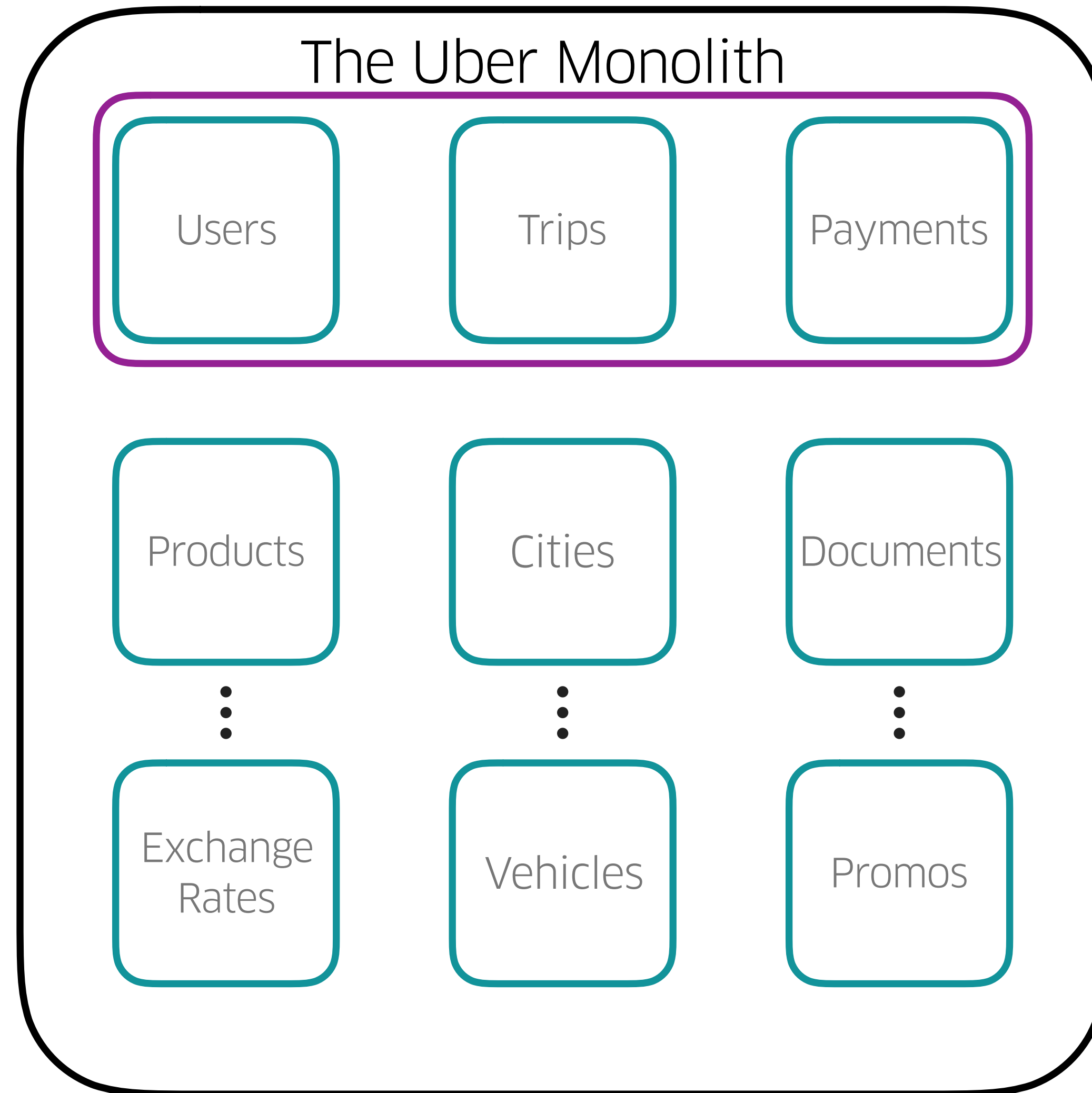
Request Flow



Benefits

- ▶ **Minimal overhead for cross-domain features**

“Cross Domain”



Benefits

- ▶ **Minimal overhead for cross-domain features**
- ▶ **Reuse code across domains**
- ▶ **Broad, extensible interface**

Historic Outages

- ▶ Honks
- ▶ Halloween 2014
- ▶ Postgresql Master Failure



Image courtesy of Anita Hart. Licensed under CC-BY-SA 2.0.

<https://flic.kr/p/8PG15b>

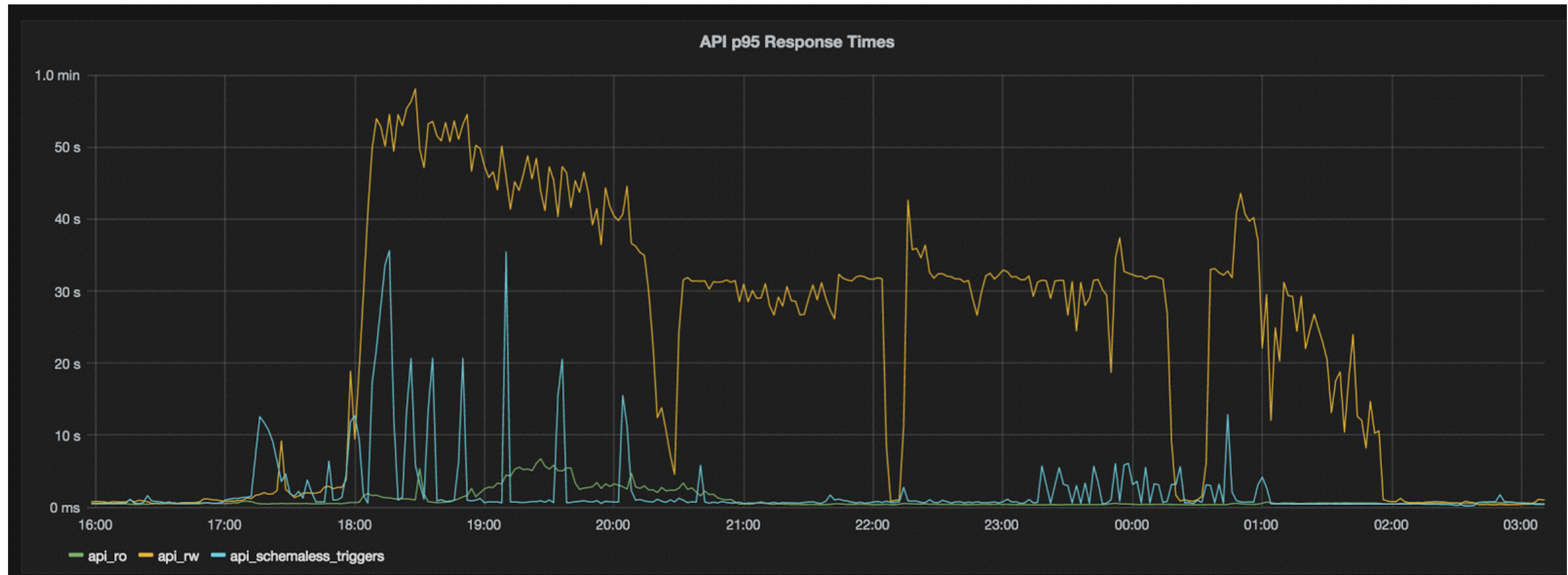
Postgresql Master Failure



Historic Outages

- ▶ Honks
- ▶ Halloween 2014
- ▶ Postgresql Master Failure
- ▶ Kafka Outage

Kafka Outage



Technical Bottlenecks

- ▶ **Deploy burden**
- ▶ **Single Postgres Master**



Image courtesy of Paul Schadler. Licensed under CC-BY 2.0.

<https://flic.kr/p/aDoGkP>

Developability

- ▶ **Rapid hiring**
- ▶ **Slow test suite**
- ▶ **Toe stepping, merge conflicts**

Scalability 3 Ways

- ▶ Growth of business
- ▶ Growth of features
- ▶ Growth of organization

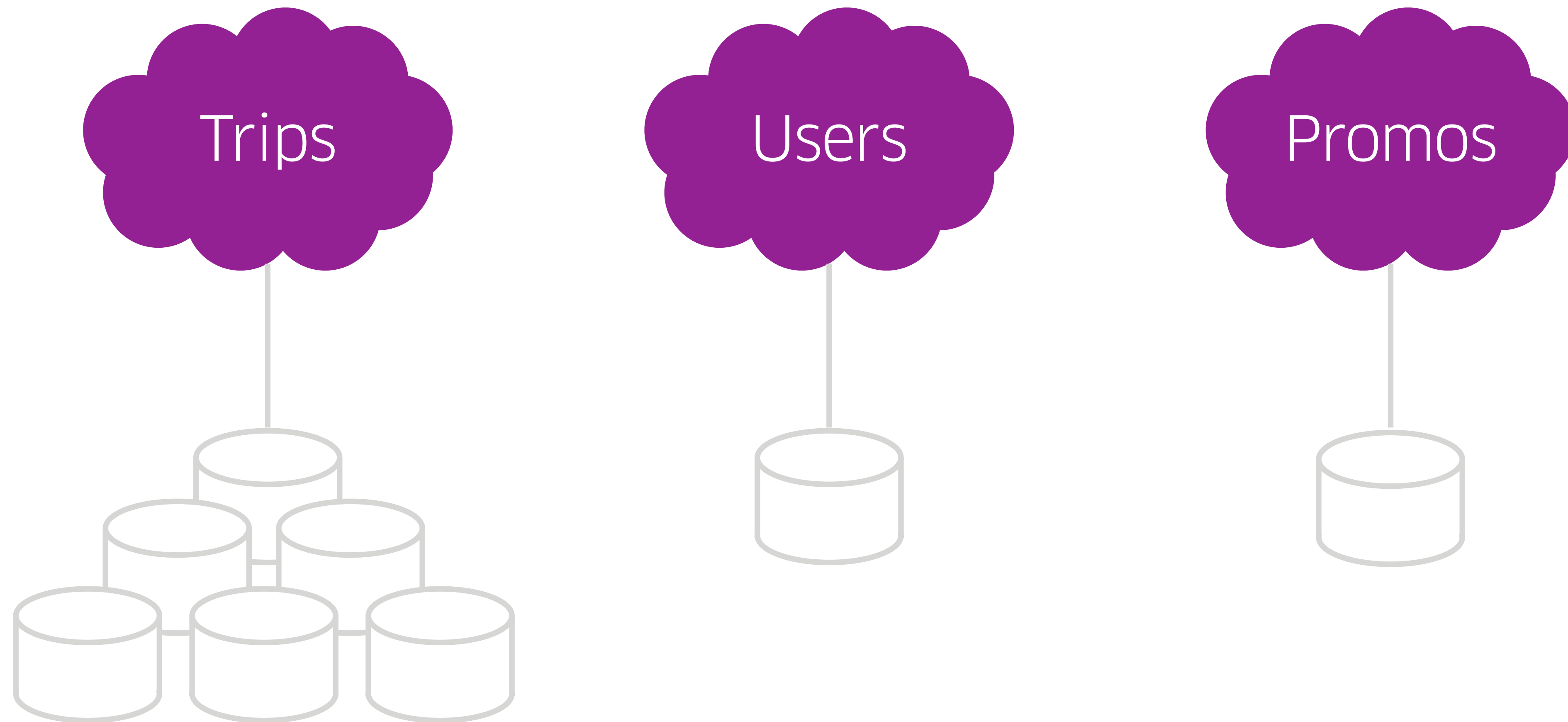
Our Future

In Microservices

Benefits

- ▶ **Horizontally scalable**


Scalability




Benefits









- ▶ **Horizontally scalable**
- ▶ **Improved onboarding**
- ▶ **Clear ownership**

Ownership



SERVICES TECHNOLOGIES TEAMS

SERVICE | 

Owning Team	money_riders
Organization	Money
Team Email	
Team Contact Email	
	-
	
On Call Engineer	  

No hiding!

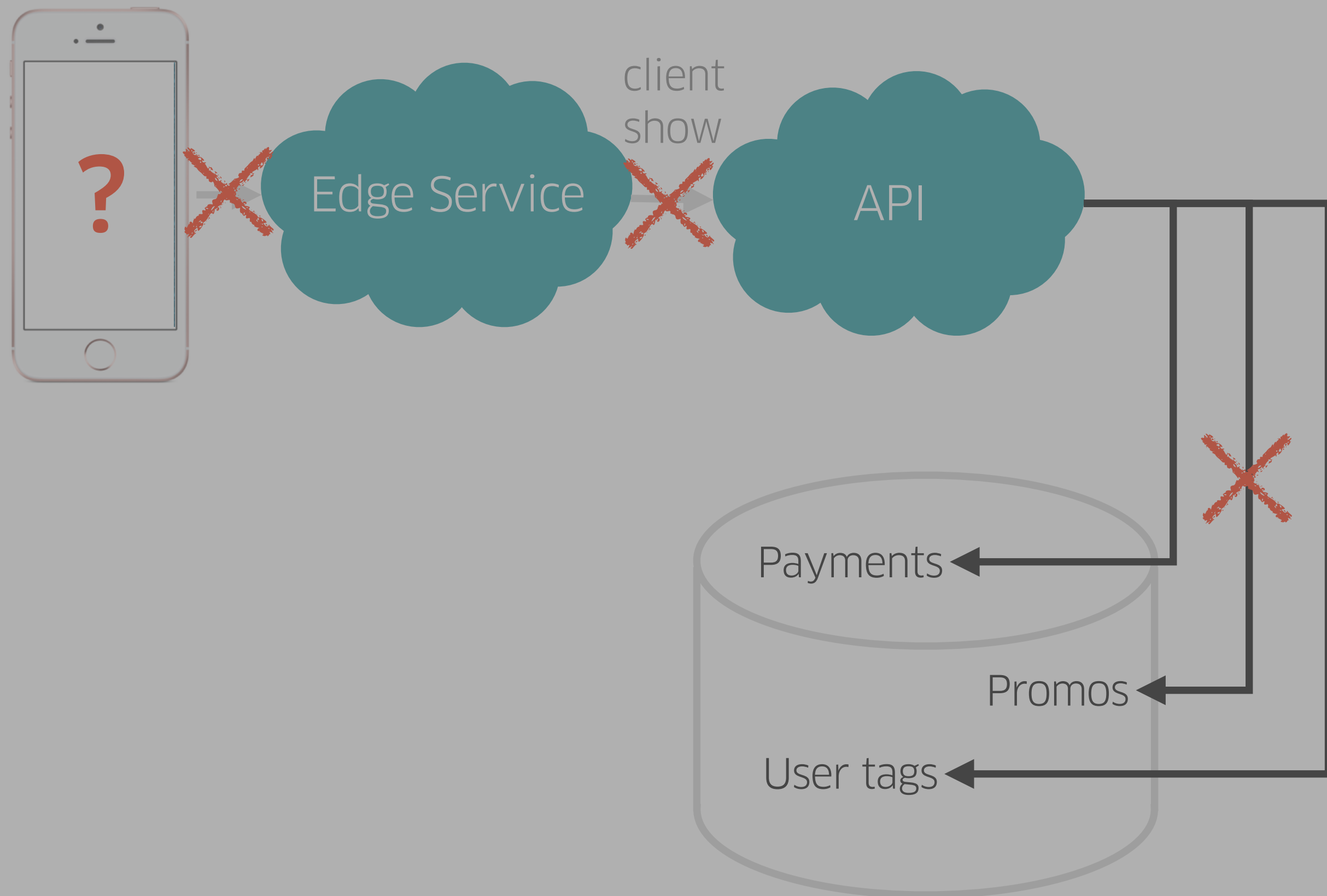
You're on call!

Benefits

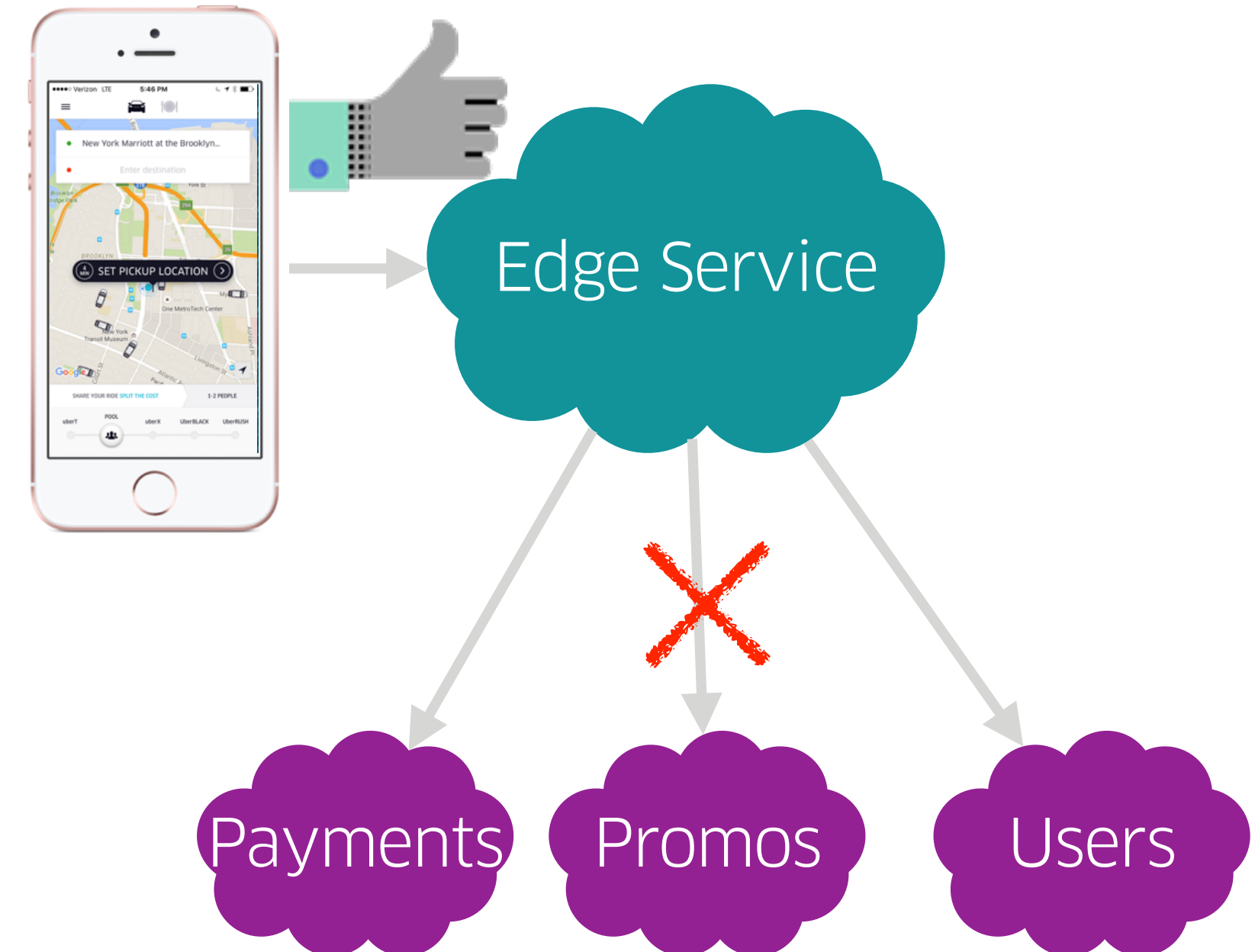
- ▶ **Horizontally scalable**
- ▶ **Improved onboarding**
- ▶ **Clear ownership**
- ▶ **Failure isolation**

Granular Failures

Old Way



New Way



Lessons Learned

Limit Scope

- ▶ Drastically changing tech stack

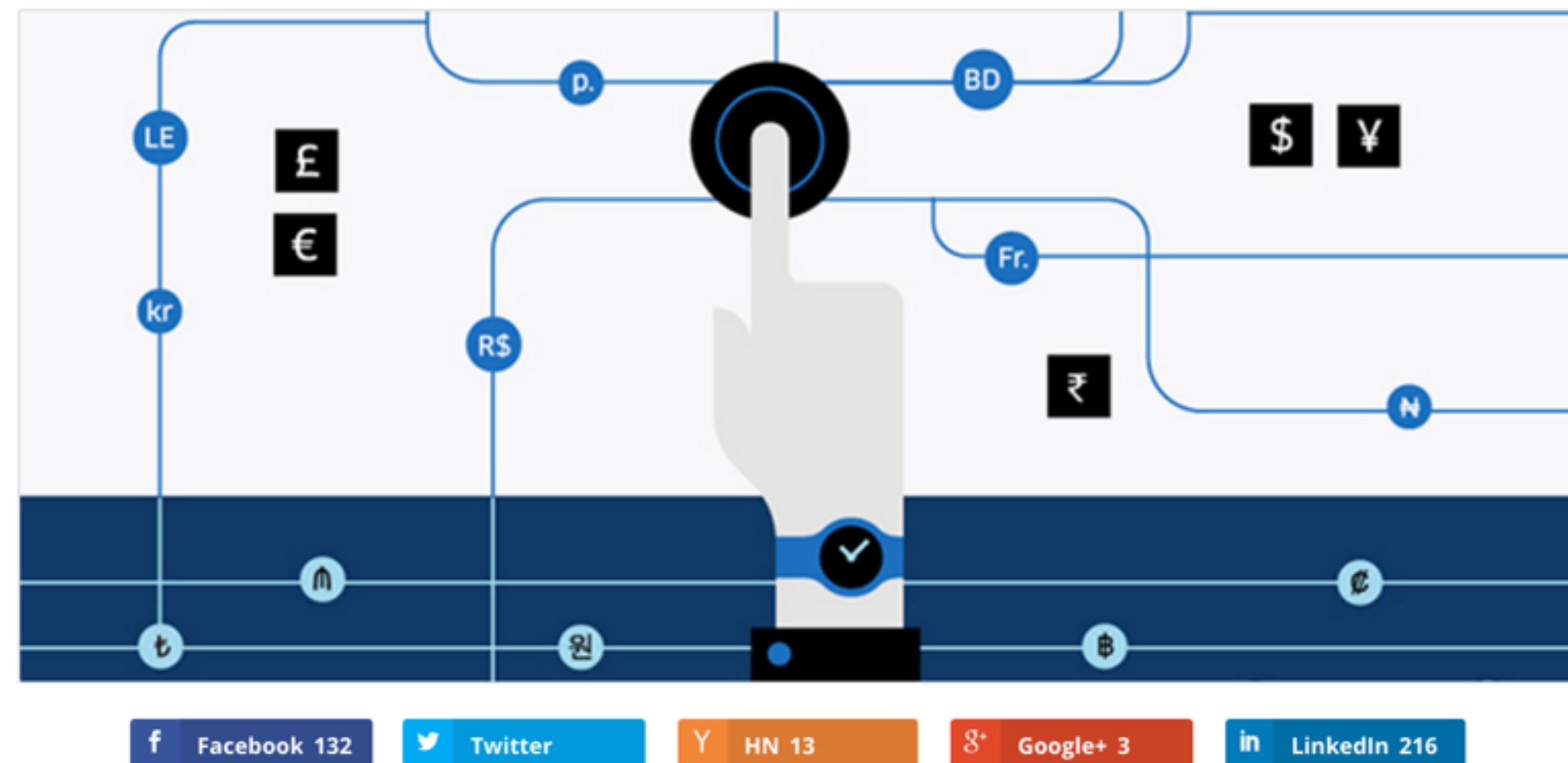
Tech Stack

ARCHITECTURE

REWRITING UBER ENGINEERING: THE OPPORTUNITIES MICROSERVICES PROVIDE

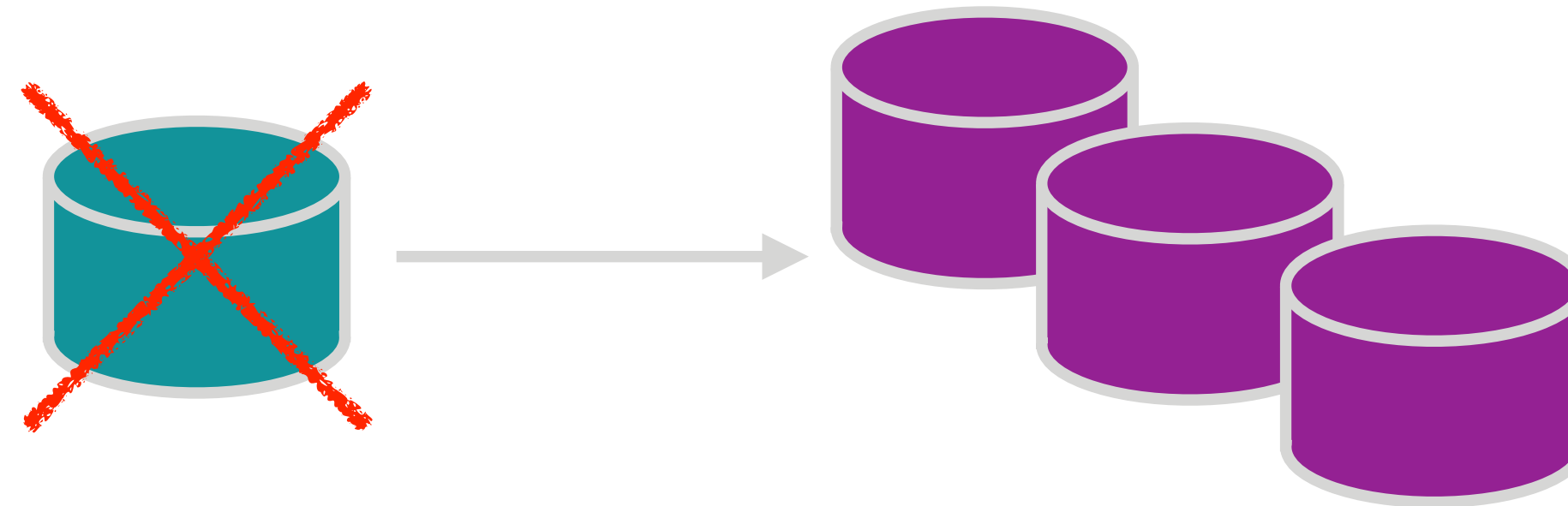
APRIL 20, 2016
BY EMILY REINHOLD

That's me!



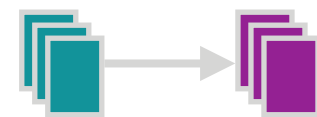
eng.uber.com/building-tincup/

Data Migration

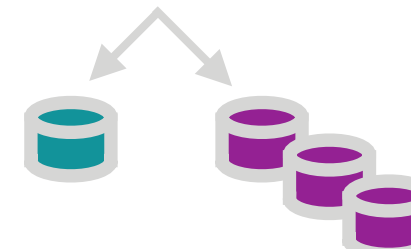


1. Develop **new** schema

2. Implement converter



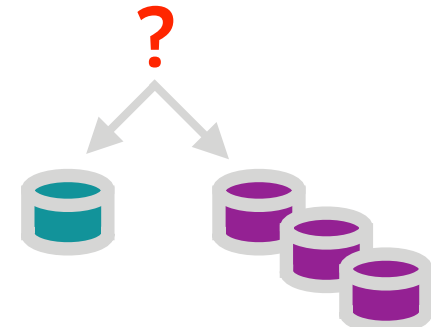
3. Dual write



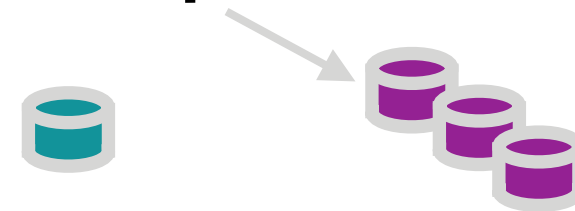
4. Backfill



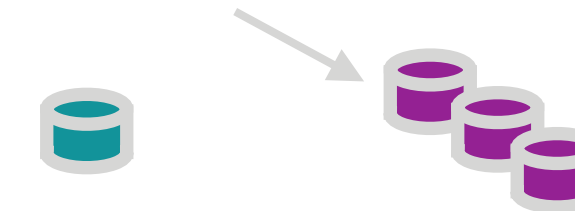
5. Validation



6. Move read queries



7. Remove old writes



Limit Scope

- ▶ **Drastically changing tech stack**
- ▶ **Infrastructure not ready**
 - ▶ Understaffed
 - ▶ Lacked tooling

Things Happen



Image courtesy of Kim Scarborough. Licensed under CC-BY-SA 2.0.

<https://flic.kr/p/23HUAK>

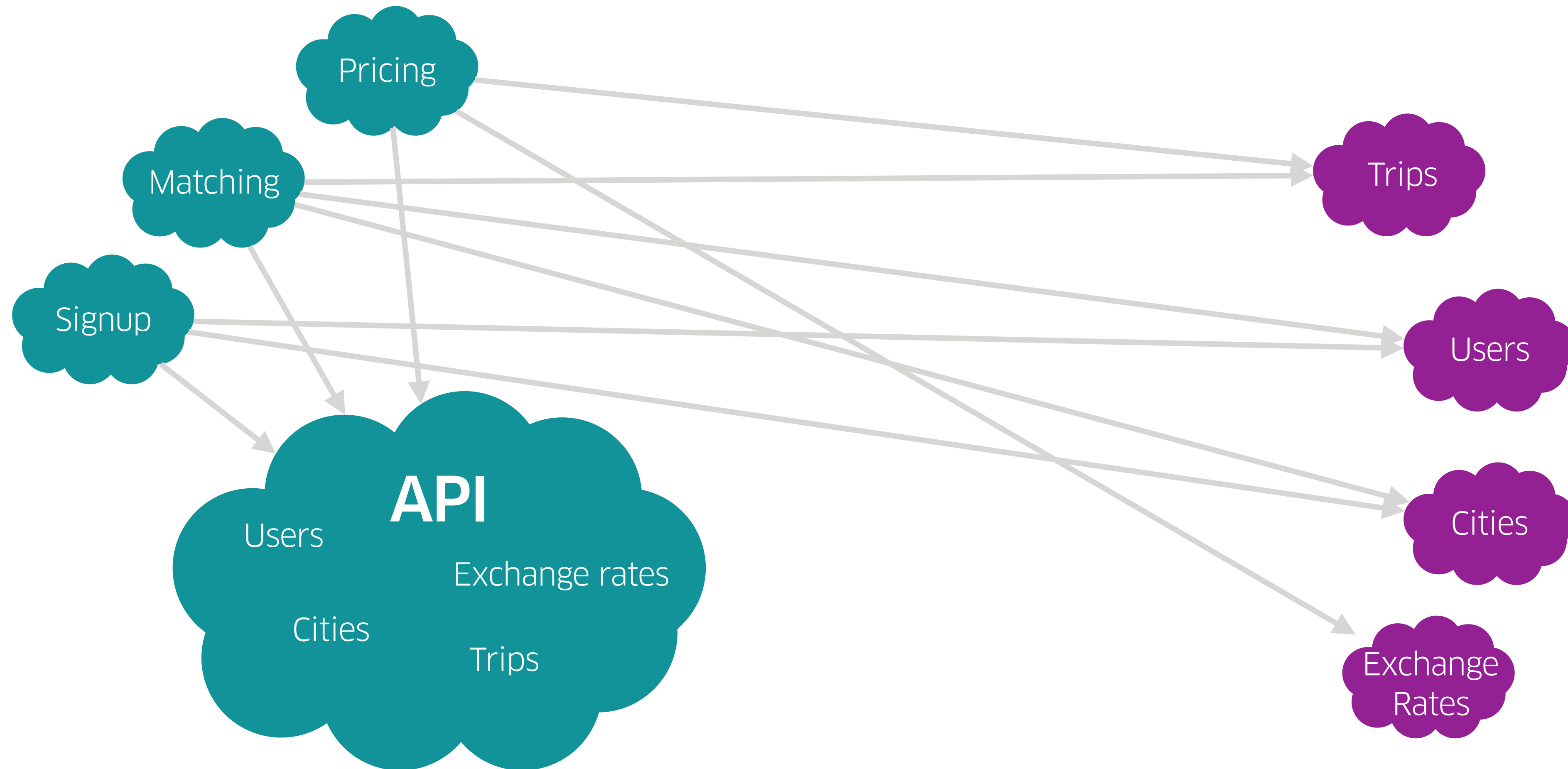
Limit Scope

- ▶ **Drastically changing tech stack**
- ▶ **Infrastructure not ready**
 - ▶ Understaffed
 - ▶ Lacked tooling
- ▶ **Understanding timelines**

Aligning with Consumers

- ▶ Monolith's consumers
directly impacted

Consumers



Aligning with Consumers

- ▶ Monolith's consumers directly impacted
- ▶ Importance of migrating not well communicated
- ▶ Constant interruption

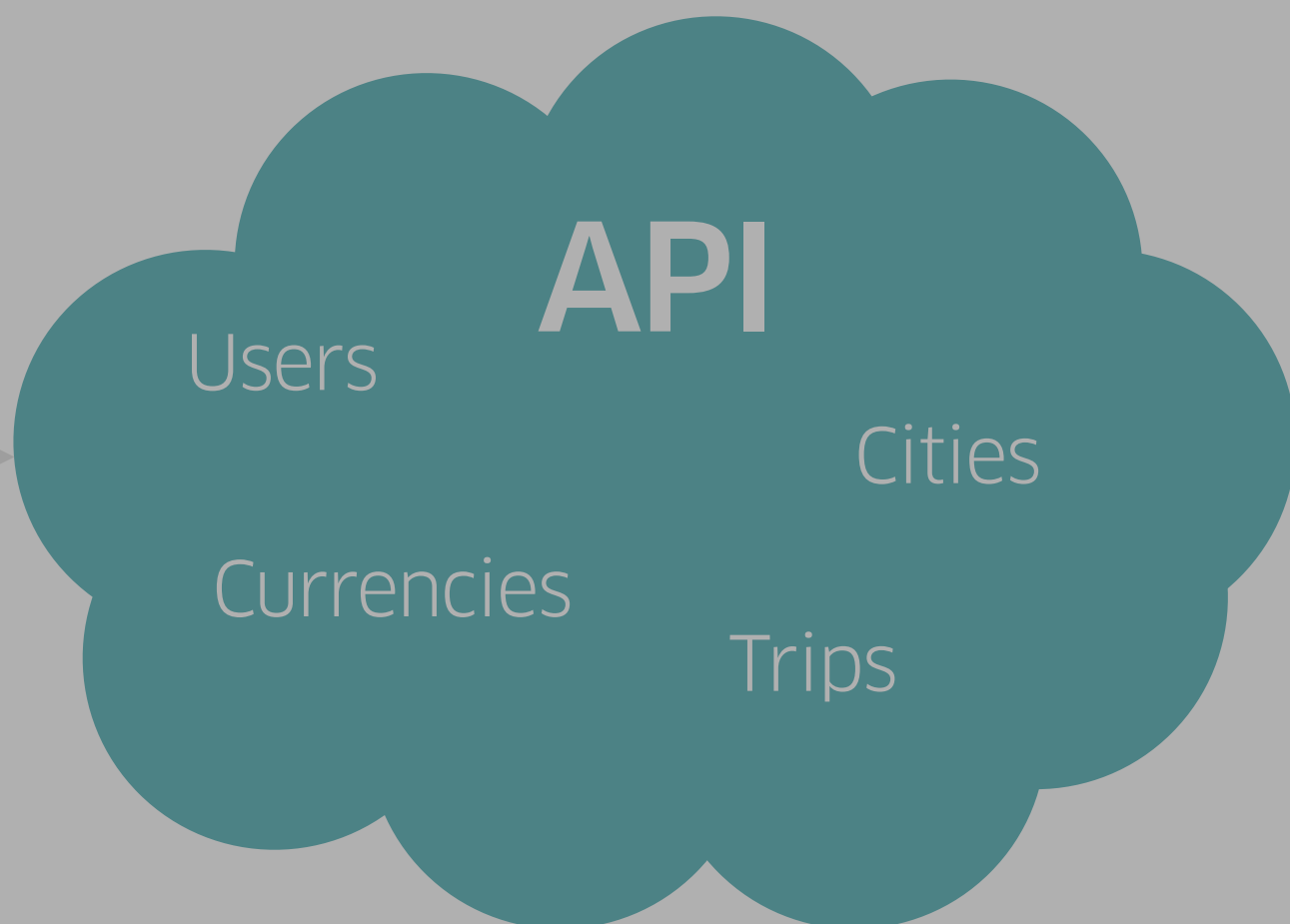
Build Tools Initially

- ▶ For service developers and migrating consumers
- ▶ Tool to split up request

Request Splitter

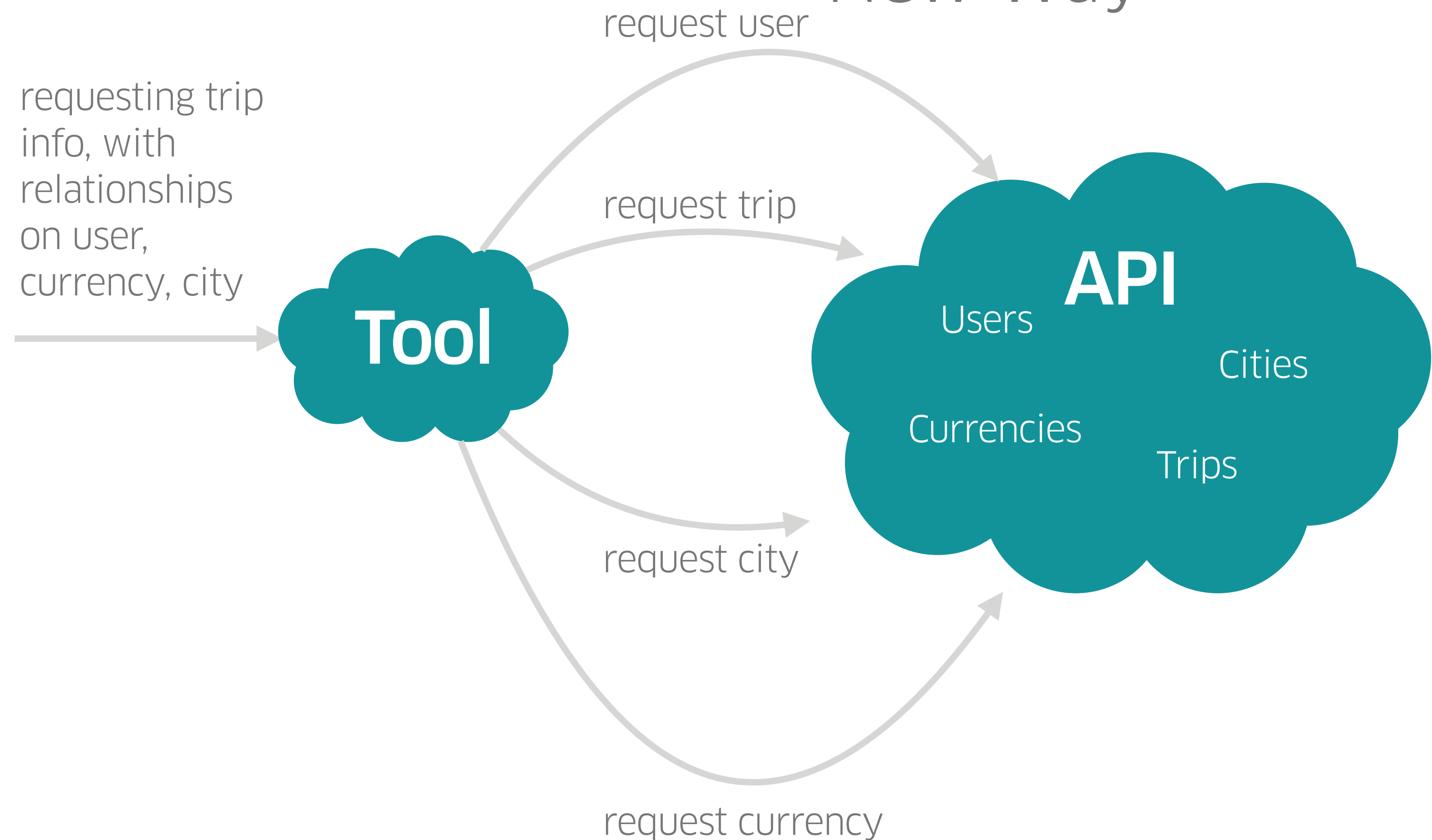
Old Way

requesting trip
info, with
relationships
on user,
currency, city



New Way

requesting trip
info, with
relationships
on user,
currency, city



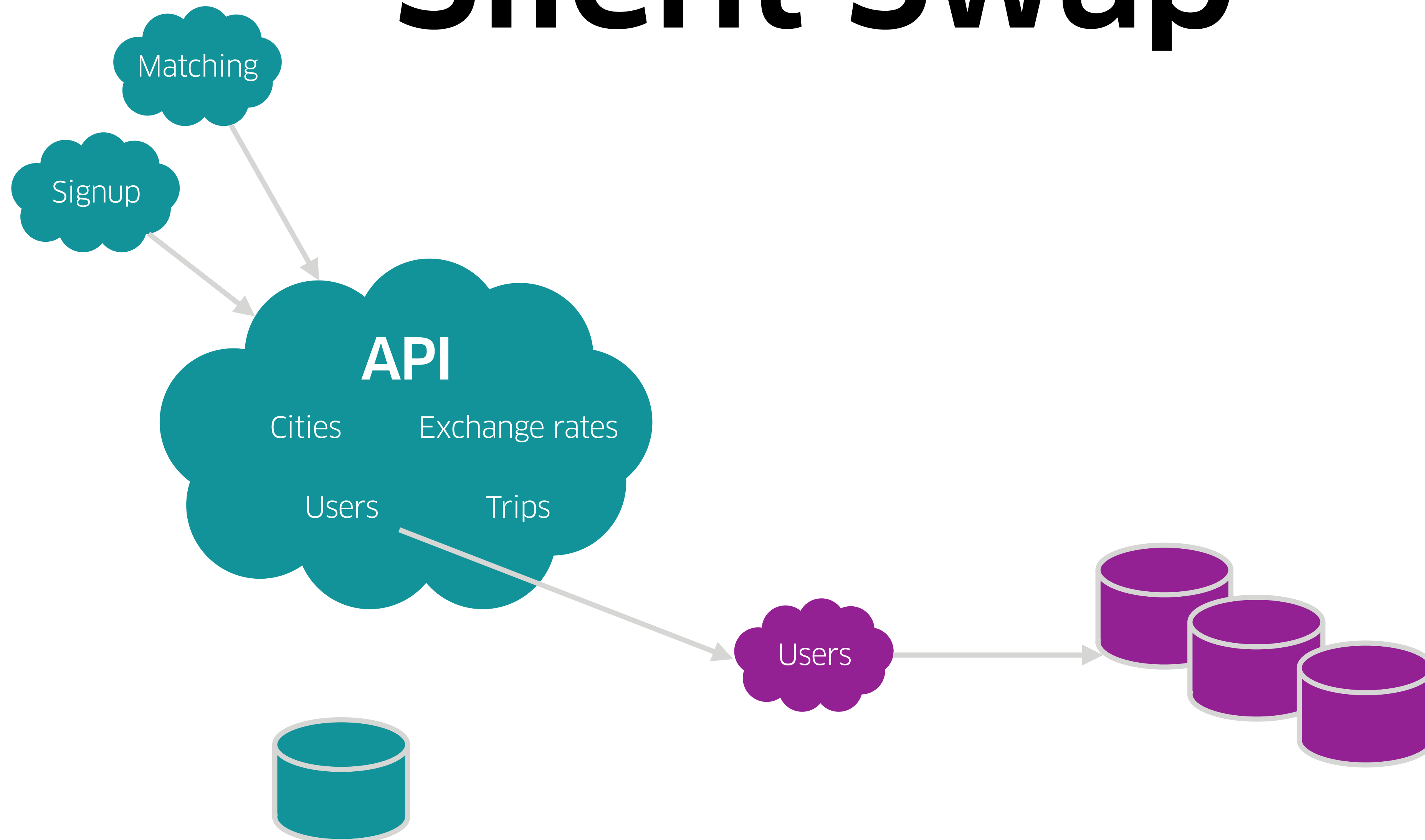
Build Tools Initially

- ▶ For service developers and migrating consumers
- ▶ Tool to split up request
- ▶ Tool to validate new response
- ▶ Tool to gain visibility into consumers

Consumer Migration Tips

- ▶ Most difficult aspect of migration
- ▶ Silently call new system

Silent Swap



Consumer Migration Tips

- ▶ Most difficult aspect of migration
- ▶ Silently call new system
- ▶ Most successful migration kept interface the same

Love your Monolith

- ▶ Tendency to avoid changes
- ▶ Tackling core (User) directly
required refactor

Migration Blockers

- ▶ Likely to impact other teams
- ▶ Teams shift focus

Migration Successes

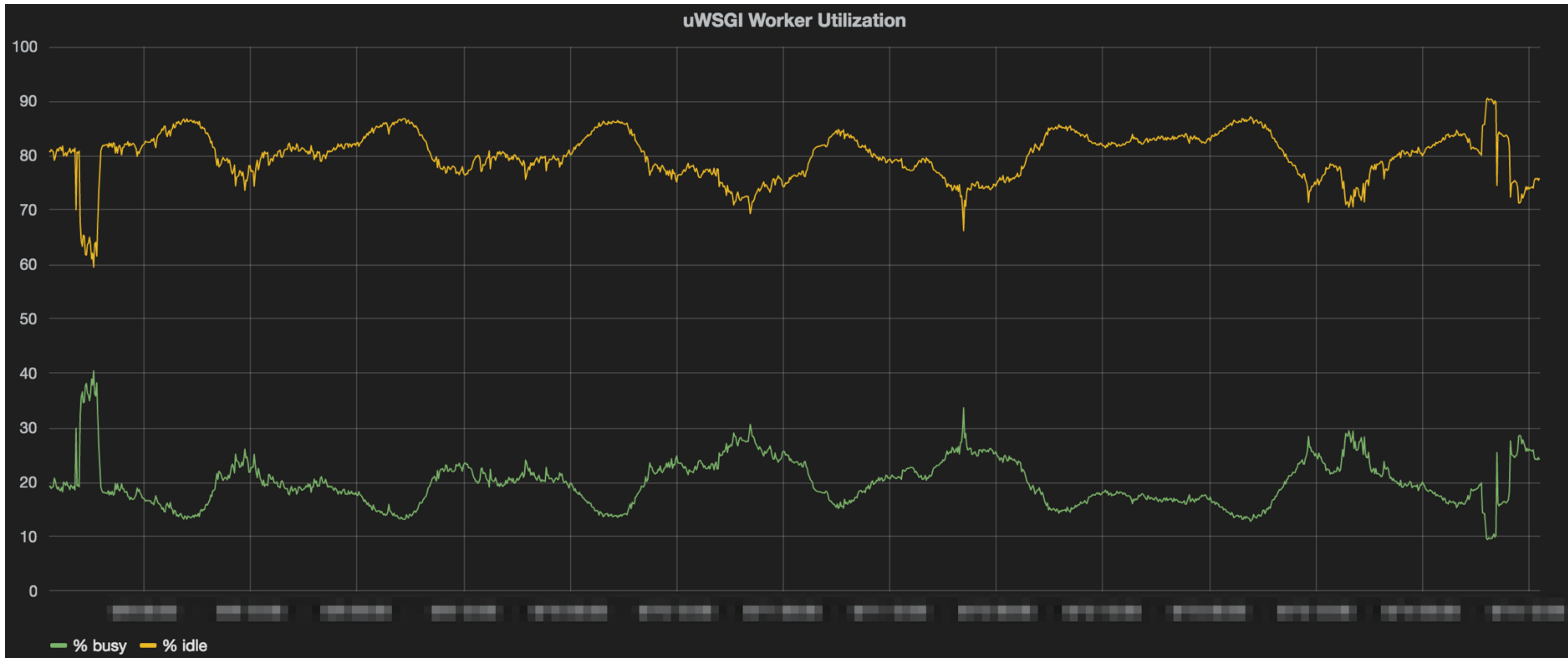
Collaboration

- ▶ Shared accomplishment
- ▶ Relationships formed
- ▶ New understanding

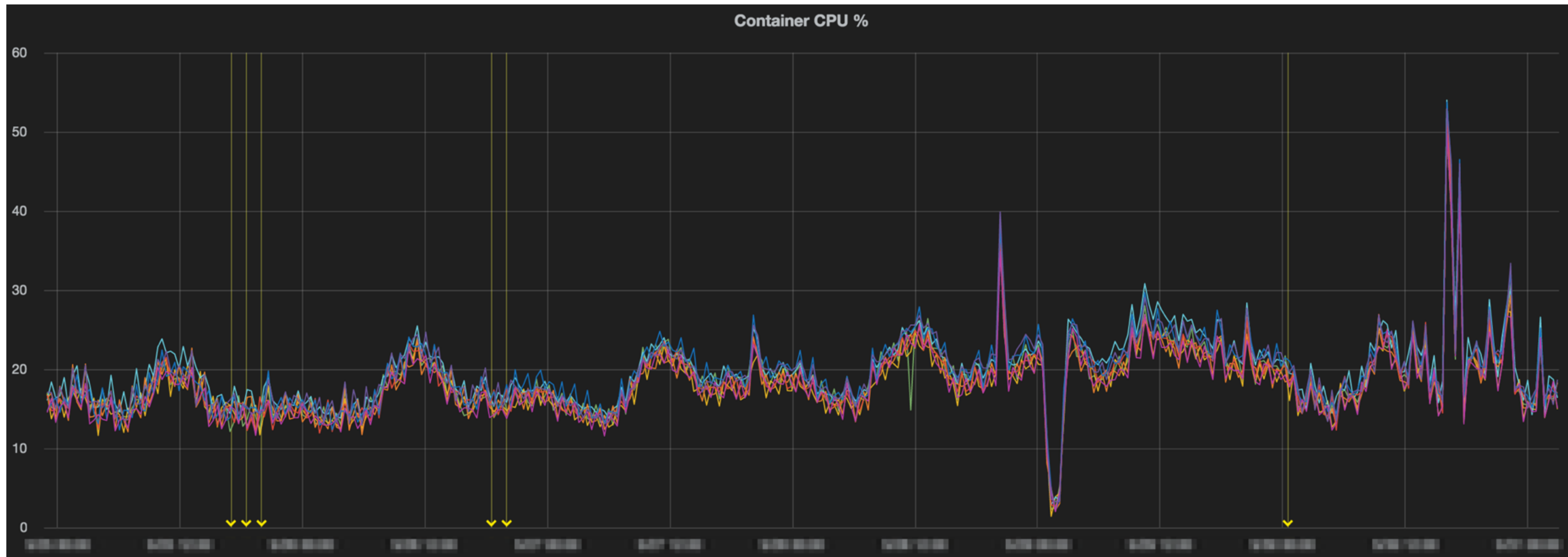
Hardware efficiency

- ▶ New tech stack: more efficient
- ▶ No more uWSGI!

uWSGI Utilization



CPU Usage



Hardware efficiency

- ▶ New tech stack: more efficient
- ▶ No more uWSGI!
- ▶ Calls asynchronous

Deeper Understanding

- ▶ How our systems work
- ▶ Design for stability
- ▶ Evenly spread ownership

Thank you

Email: emilyr@uber.com

Engineering Blog: eng.uber.com

Twitter: [@UberEng](https://twitter.com/UberEng)

The Uber logo, consisting of the word "UBER" in a bold, black, sans-serif font, centered within a white square. The square is positioned in the bottom right corner of the slide, overlapping a teal background that features a subtle, repeating geometric pattern of interlocking squares.

UBER

UBER