

@MELANIECEBULA / SEPT 2017 / FUTURESTACK

From Monolith to Microservices: How to Scale Your Architecture



What is airbnb?

AN ONLINE MARKETPLACE FOR SHARING HOMES

65k

cities

191

countries

4 mil

listings

A woman with long, wavy brown hair is sitting on a bright red sofa. She is wearing a dark grey, textured zip-up hoodie and blue jeans. She is looking directly at the camera with a slight smile. The background features a large window with a grid pattern, showing a view of a building across a courtyard. There are several large, leafy plants, including a Monstera plant on the left and a palm-like plant on the right. The overall lighting is soft and natural.

Who am I?



100%

Availability



100%

Productivity

Our time together

A Brief History

The origins of our DevOps culture, and how that shapes our tools

Scaling the Monolith

The challenges we face as we scale Monolithic deploys

Moving to Microservices

The best practices for transitioning to microservices (that we learned the hard way)

A high-speed train, likely a Shinkansen, is shown from a low angle, traveling along a curved elevated track. The train is white with a prominent red stripe. The background is a hazy, orange-tinted sky, suggesting either sunrise or sunset. The overall mood is futuristic and dynamic.

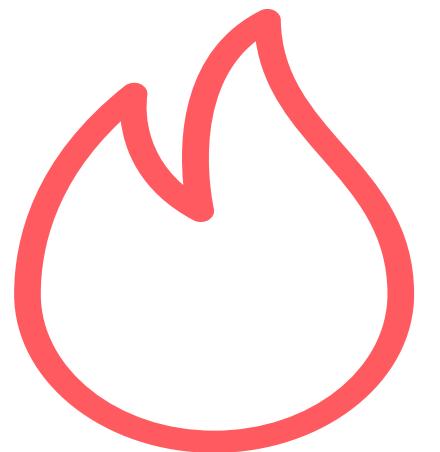
A BRIEF HISTORY

“Monorail”

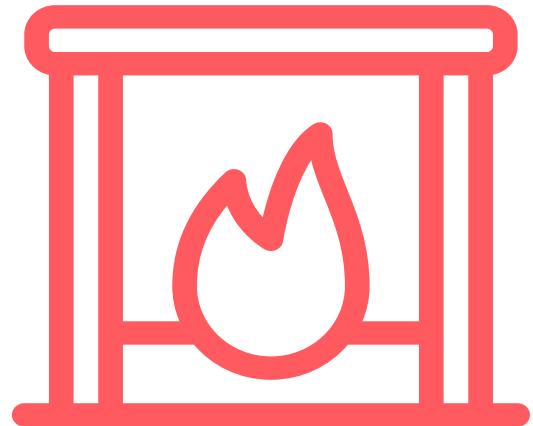
OUR MONOLITHIC RUBY ON RAILS APPLICATION



1 GitHub repo



1 SRE



Sysops

Sysops

INCIDENT RESPONSE

1. triage
2. coordinate
3. communicate

DEVOPS CULTURE

Sysops today

40
on-calls

engineers on
the rotation

2
primaries

sysops buddies

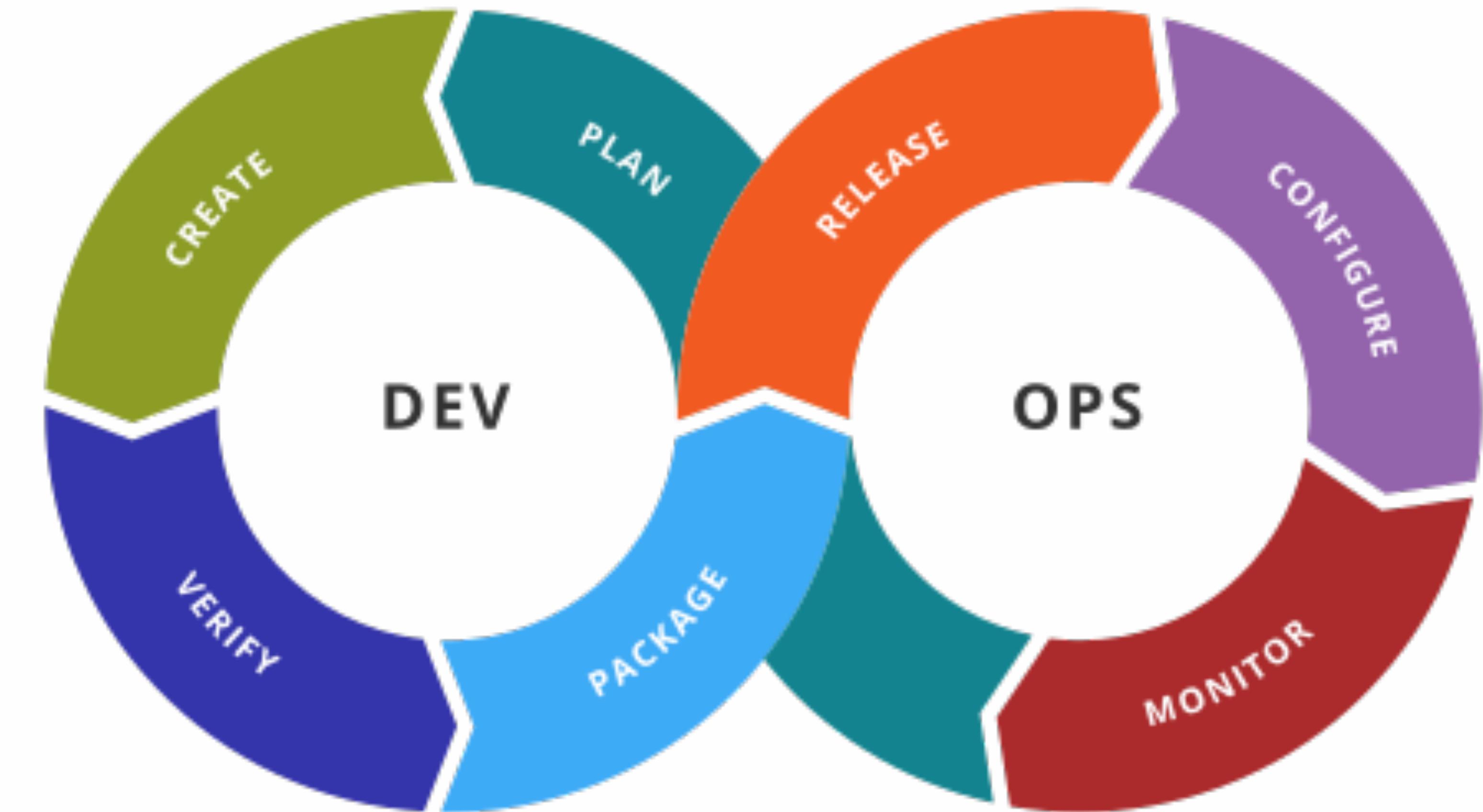
2
shifts

weekdays
weekends

Sysops training

- Sysops Training every quarter
- Sysops training establishes best practices for on-call and incident response
- 33% of engineers have attended a Sysops training
- Service on-calls can escalate to Sysops
- Blameless postmortem culture

Scaling Sysops



Applications

Service maps

Key transactions

Alerts

 APPS
AirbnbTIME PICKER
about 1 hour ending today, 7:55SERVERS
All servers

MONITORING

Overview

Service maps

Transactions

Databases

External services

Ruby VMs

EVENTS

Error analytics

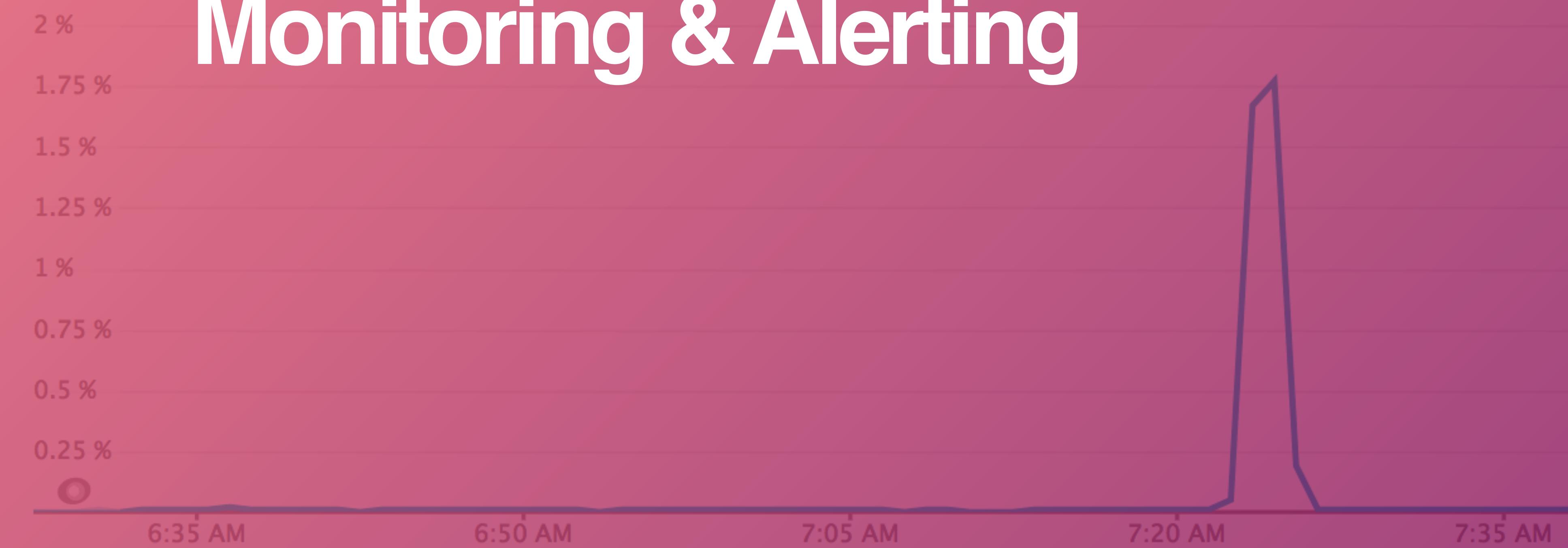
Errors

Alerts

This page is being retired in favor of the Error analytics experience in the future. Check out the new Error analytics page in the left-hand navigation bar for more details.

Error rate (errors per request)

Monitoring & Alerting





Metrics

ALONGSIDE CODE

- STATSD protocol
- Automatically collect system metrics
- Metrics live alongside feature code
- Easy for any developer to write
- Service dashboards with both business metrics and system metrics

Alerts

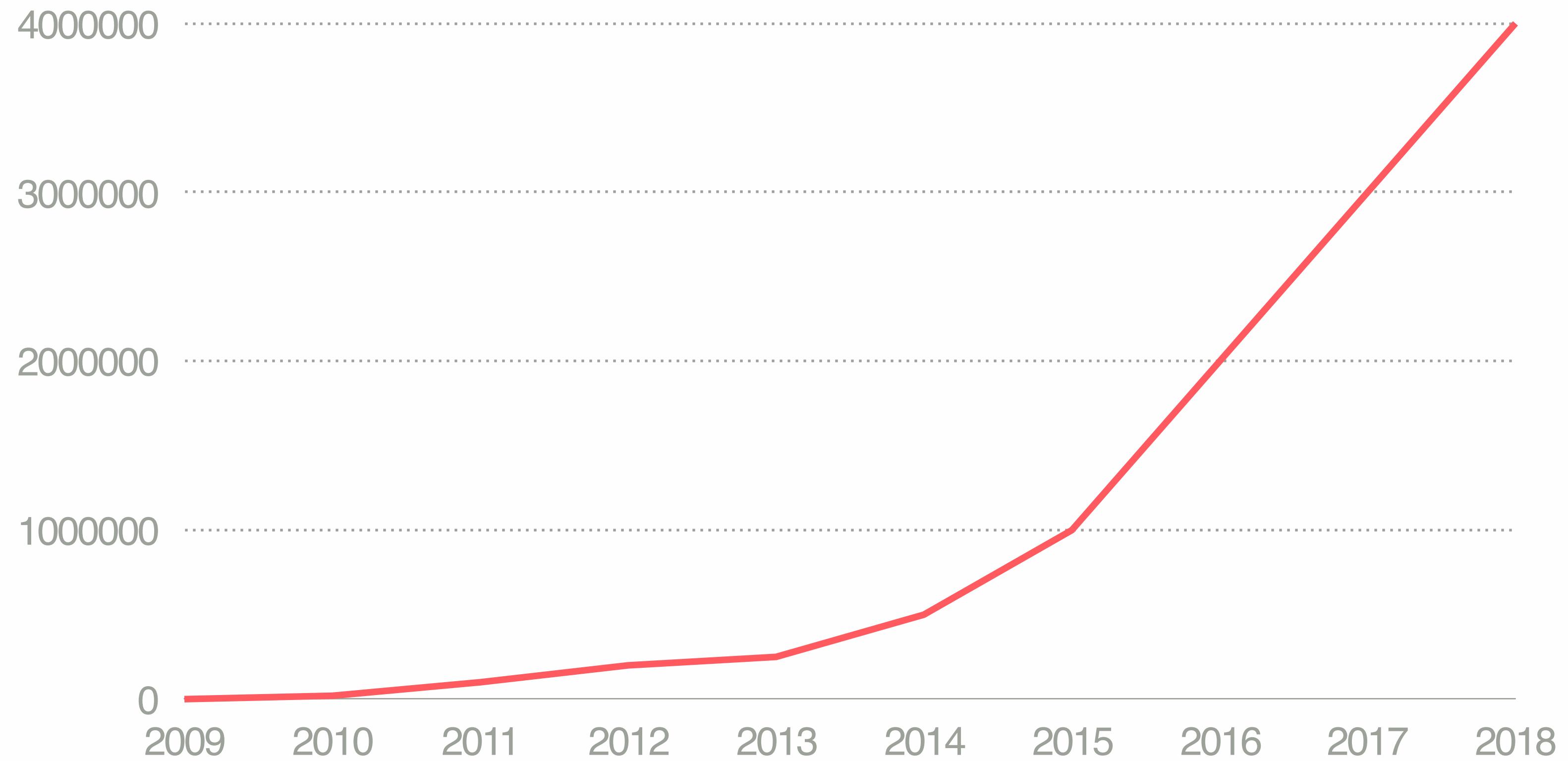
CONFIGURATION AS CODE

```
1 name "Logins dropped"
2
3 message <<END
4 Logins dropped significantly from seasonal forecast.
5 Sysops need to investigate immediately.
6 END
7
8 applies true
9
10 notify.groups ['pagerduty_sysops']
11
12 metric.datadog_query <<E0Q
13 avg(last_30m):
14     anomalies(sum:logins{*}.as_count(),
15                 'agile', 3, direction='below')
16     >= 0.5
17 E0Q
18
19 notify_no_data true
20 no_data_timeframe 10
21
22 silenced false
23 require_full_window true
```

SCALING MONORAIL DEPLOYS

**When it starts to
become a problem**

MONORAIL LOC



Democratic deploys

All developers are expected to “own” their features from implementation to production.

Each developer is expected to:

- deploy their changes
- monitor their changes
- roll back, abort, or revert if necessary

Deployboard ✓ Live Updating ✓ Merges and Deploys Unlocked Monorail

Snapshots Deploy Targets Silhouettes master 65 / 709

Deploy 86441 of Monorail to Production
Started by john du at 1:24 pm.
Deploying snapshot #1574388 (05ad391)
Deploy is running! Please watch metrics!

Roll Back Abort

Ready Preparing Waiting Deploying Finished Failed

Deploy 86442 of Monorail to Next
Started by pengyu li at 1:25 pm.
Deploying snapshot #1574552 (887365c)
Deploy is running! Please watch metrics!

Roll Back Abort

Ready Preparing Waiting Deploying Finished Failed

Airbnb: Error rate (last 30 minutes)
Airbnb: Average response time, by tier (ms) (last 30 minutes)
Airbnb (Resque): Error rate (last 30 minutes)

New Relic Sentry Resque Synapse Dashboards Datadog Dashboards Hide graphs

	Snapshot	SHA	Author	Pushed At (PDT)	Deployed	Compare To	Notes	
<input type="checkbox"/>	#1574636	b0a7345	feng.qian@ai...	Tue, Sep 27, 2016 1:11 PM	<input type="button" value="PREV"/>	<input type="button" value="HEAD"/>	C N P	<input type="button" value="Deploy"/>
<input type="checkbox"/>	#1574564	085dc27	yat.choi@air...	Tue, Sep 27, 2016 1:05 PM	<input type="button" value="PREV"/>	<input type="button" value="HEAD"/>	C N P	<input type="button" value="Deploy"/>
<input type="checkbox"/>	#1574560	8d1f378	nico.moscho...	Tue, Sep 27, 2016 1:04 PM	<input type="button" value="PREV"/>	<input type="button" value="HEAD"/>	C N P	<input type="button" value="Deploy"/>
<input type="checkbox"/>	#1574552	887365c	tony.huang@...	Tue, Sep 27, 2016 1:03 PM	<input type="button" value="PREV"/>	<input type="button" value="HEAD"/>	C N P	<input type="button" value="Deploy"/>

Continuous Delivery

- everything merged into master is deployable to production
- frequent deploys (with smaller batches)
- automate as much as possible

Scaling Deploy Process

- builds, tests, and deploys must be fast and reliable
- ping authors when they need to take action
- automatic (and fast) reverts
- merge queue
- automatic rollbacks for new exceptions

THE MOVE TO MICROSERVICES

What is a microservices architecture?

“Essentially, a microservices architecture is a method of developing software applications as a suite of **independently deployable, small, modular services** in which each service runs a **unique process** and **communicates through a well-defined, lightweight mechanism** to serve a business goal.”

— someone on the internet

Why Not Microservices?

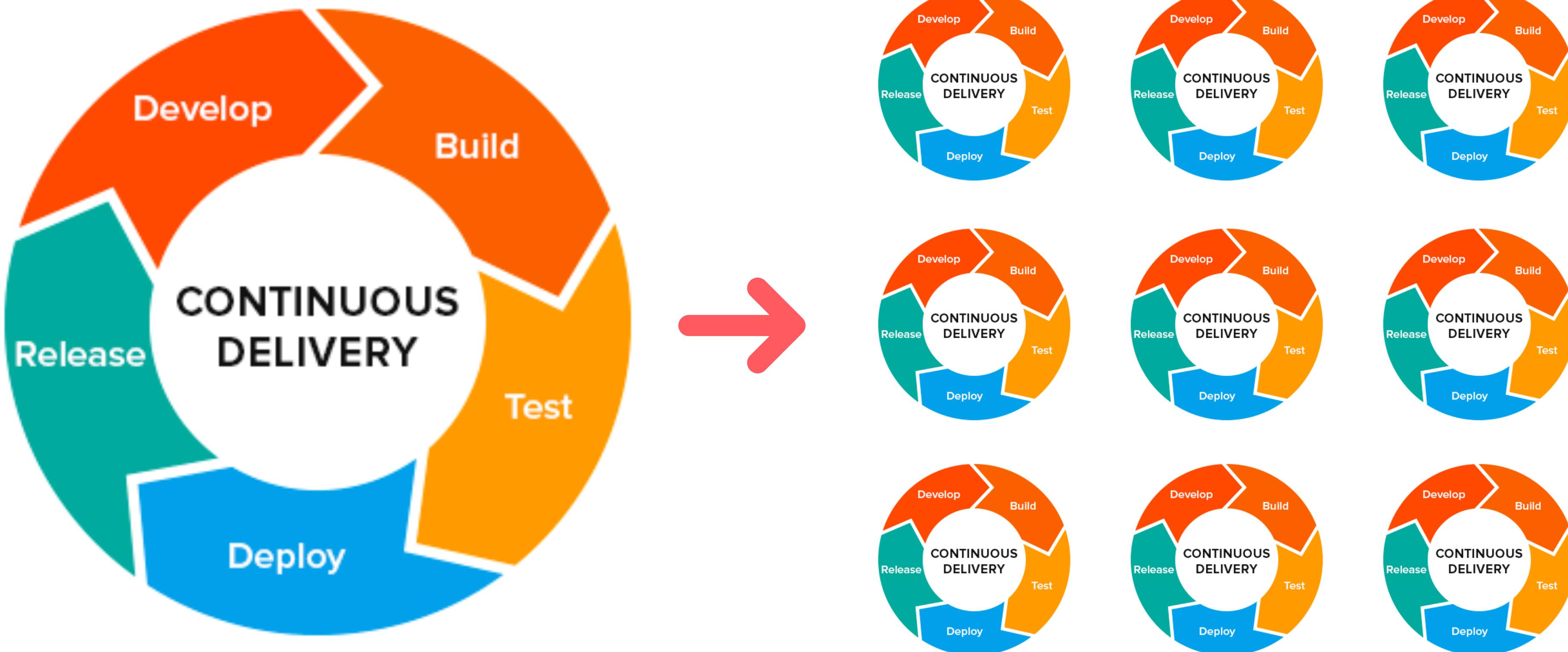
- Distributed systems are hard!
- Naming things is hard!
- Having to support different languages / frameworks

Why Microservices?

- Independent deployment
- Modular structure / Ownership
- Secure / Performant
- Upgrading dependencies

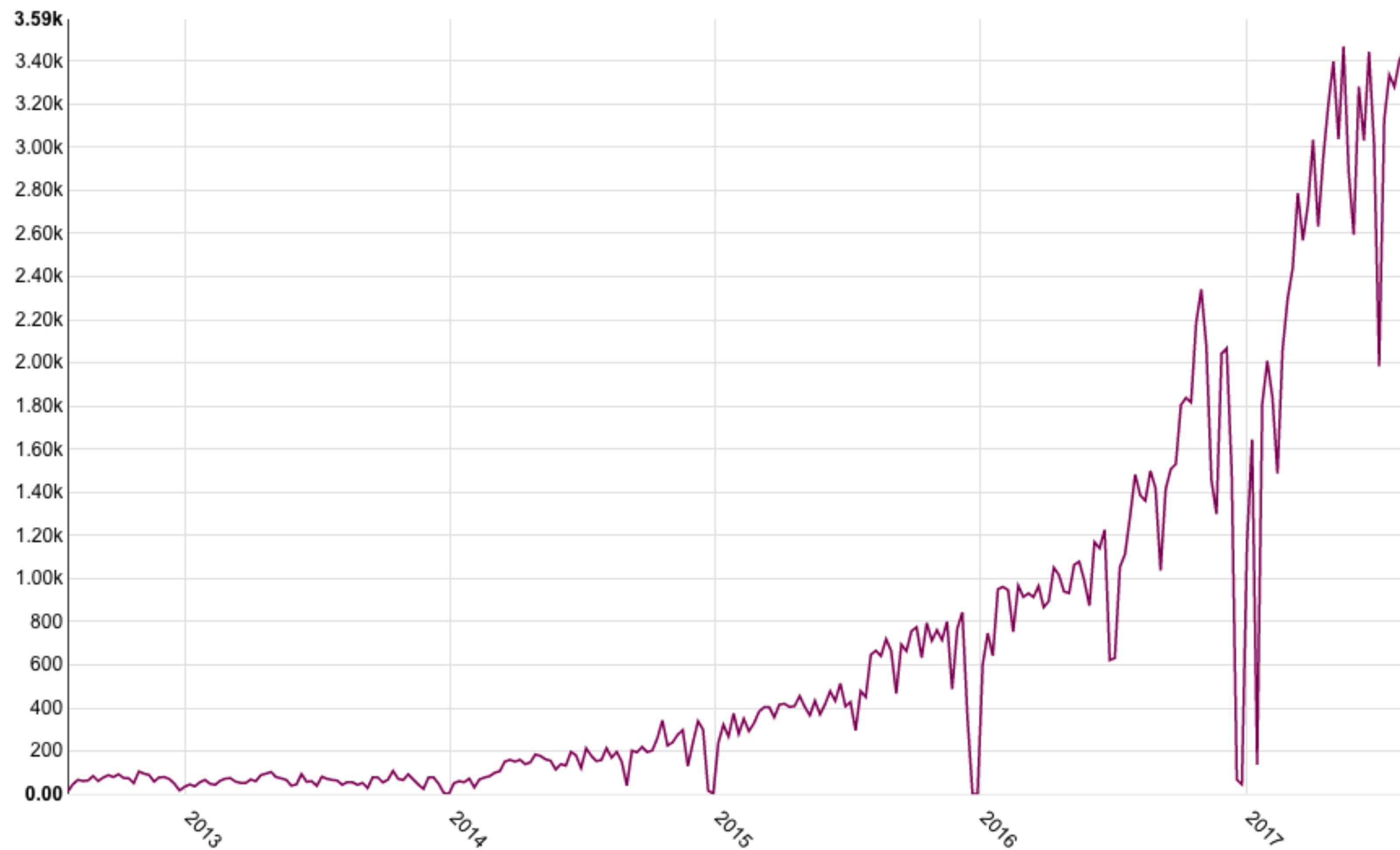
Why Microservices?

SCALING CONTINUOUS DELIVERY



Why Microservices?

Deploys per week (all apps, all environments)

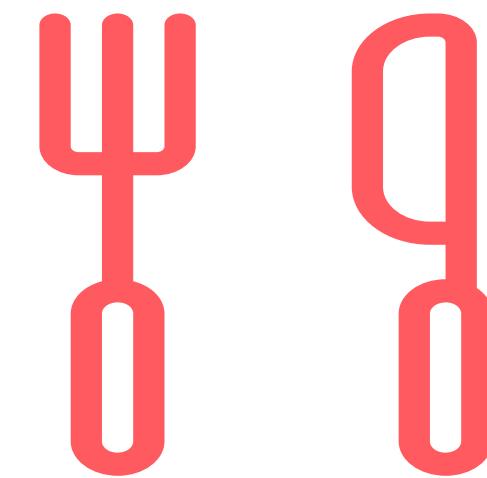


Why Microservices?

75,000
production deploys
per year

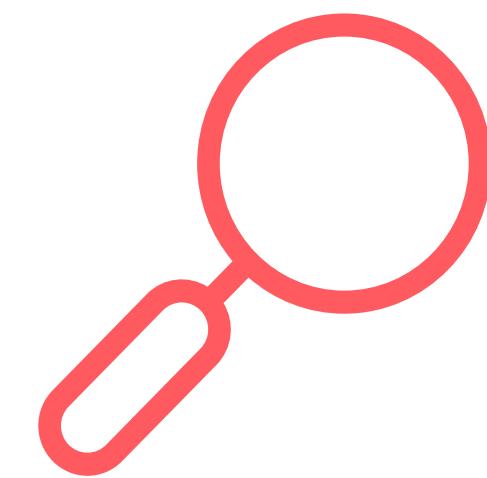
Microservices architecture

THE BARE MINIMUM



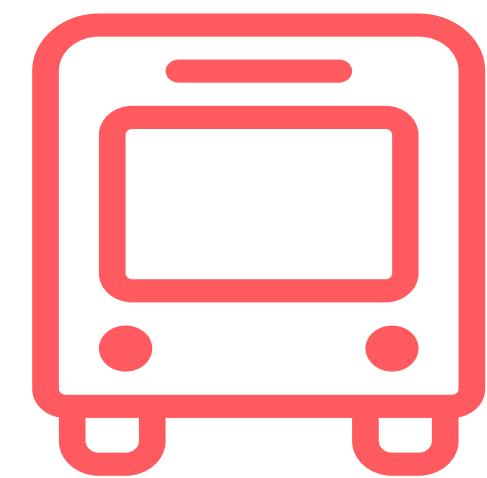
Configuration
Management

Chef
Kubernetes



Service registry
and discovery

Smartstack
Envoy



Message bus

with Kafka

Microservices architecture

AS YOU START TO SCALE

- Rapid provisioning (AWS instances)
- Devops culture
- Basic monitoring / alerting
- Continuous Delivery

Moment of Truth

Are we committed to microservices?

Services

900
developers

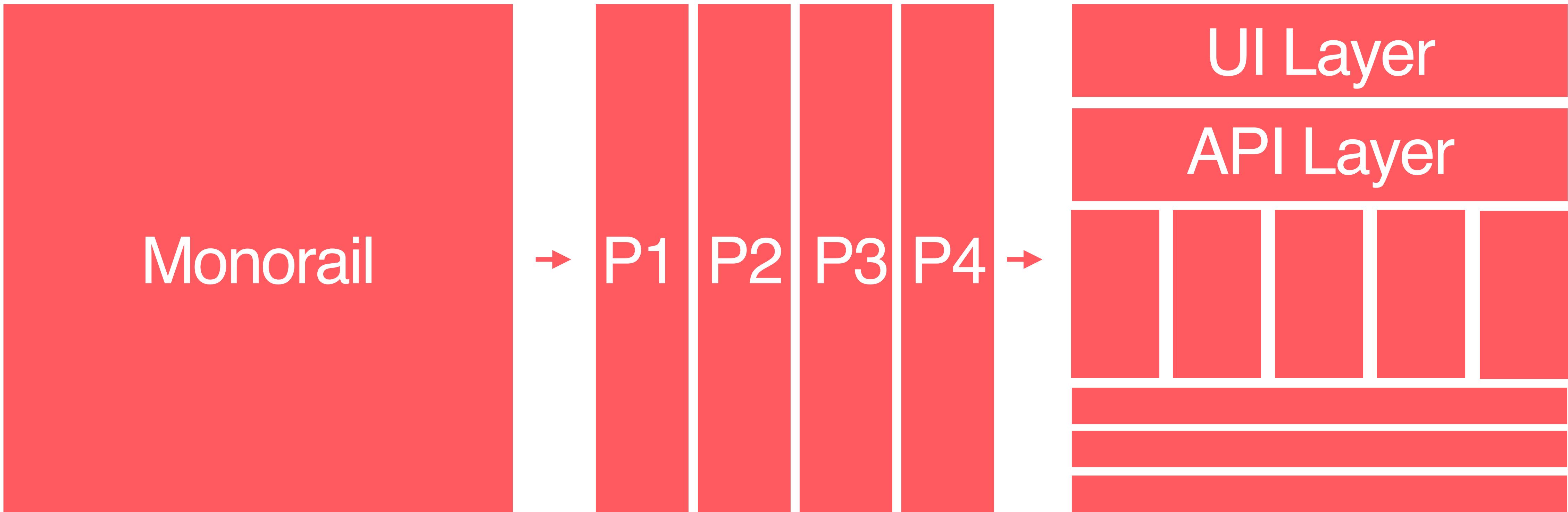
300
services

4k
deploys per week

Service Creation Challenges

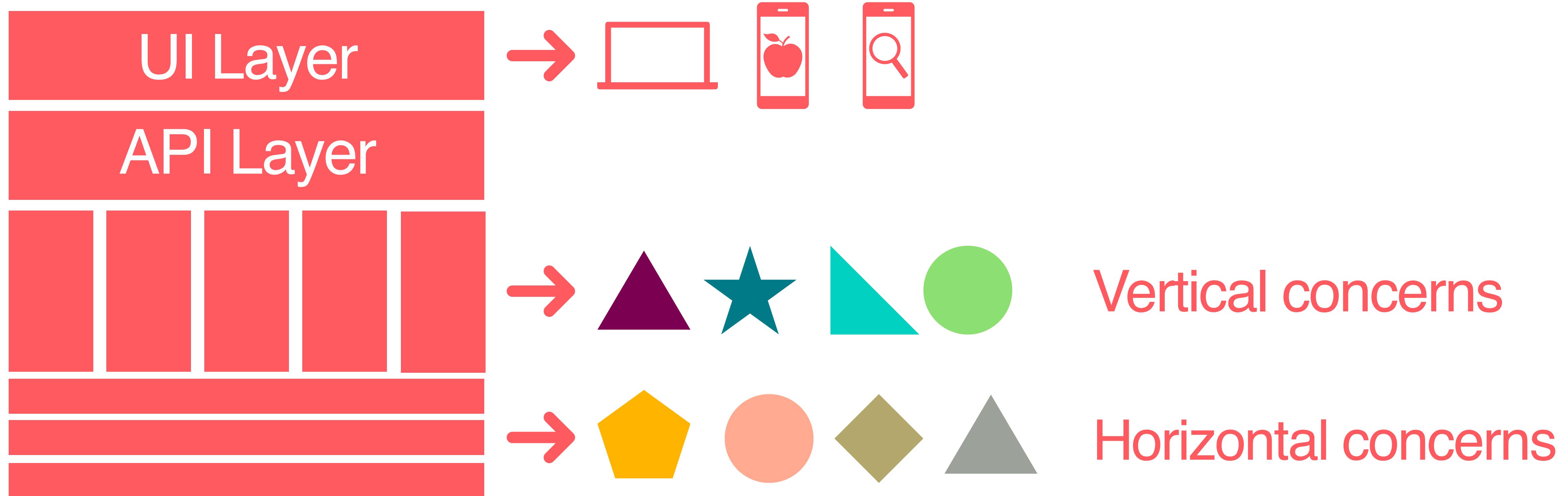
- Supporting different languages / frameworks
- Services own their own data
- Standardized configuration
- Monitoring with microservices
- Splitting the Monolith into services

Splitting the Monolith



Vertical concerns
Horizontal concerns

Splitting the Monolith



Standardize Service Creation

NO SNOWFLAKES



Standardize Service Creation

OKAY BUT HOW?

Standardize and simplify your configuration!

- lives as code, inside application repository
- automatically tested and validated as part of CI
- can deployed and consumed as an artifact
- has a standardized format and defaults

The important part

10 TAKEAWAYS

1. Monolith first
2. Devops culture
3. Configuration as code
4. Monitoring and Alerting
5. Continuous delivery
6. Automate what you can
7. Your first service is going to be bad
8. Services own their own data
9. Break apart the Monolith
10. Standardize service creation

