

IT Home





# Microservices Delivery Platform. Tips and Tricks

# Andriy Trubitsyn Solution Architect @ EPAM

20 years of software development

Senior Solution architect at EPAM

Speaker @ XP Days Ukraine, JEE Conf Ukraine

Speaker @ Z - Java Day, Z-Conference

Speaker @ meetups and events

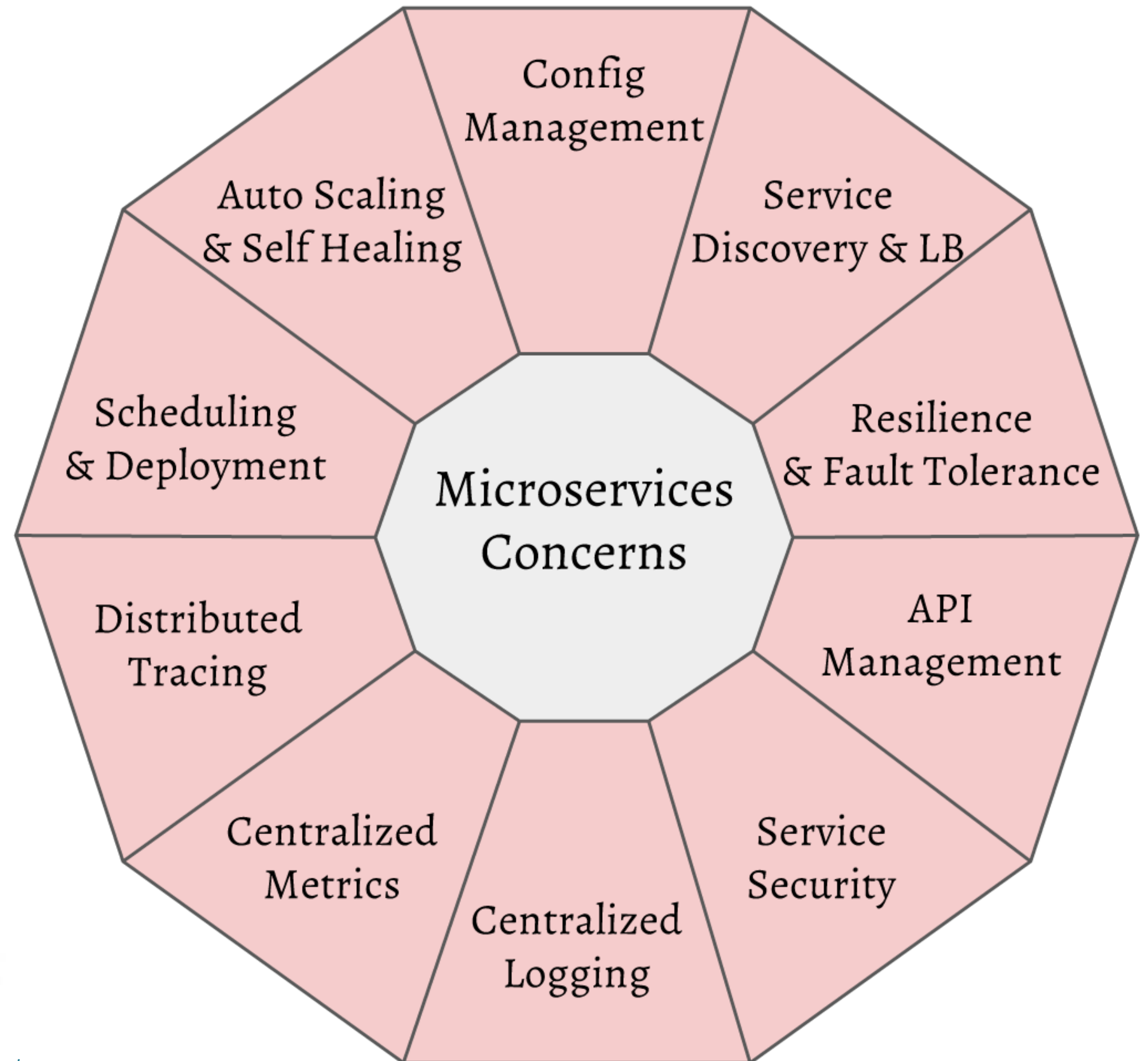




Microservices  
delivery platform  
=  
Microservices  
platform + CI/CD  
+ What?

---

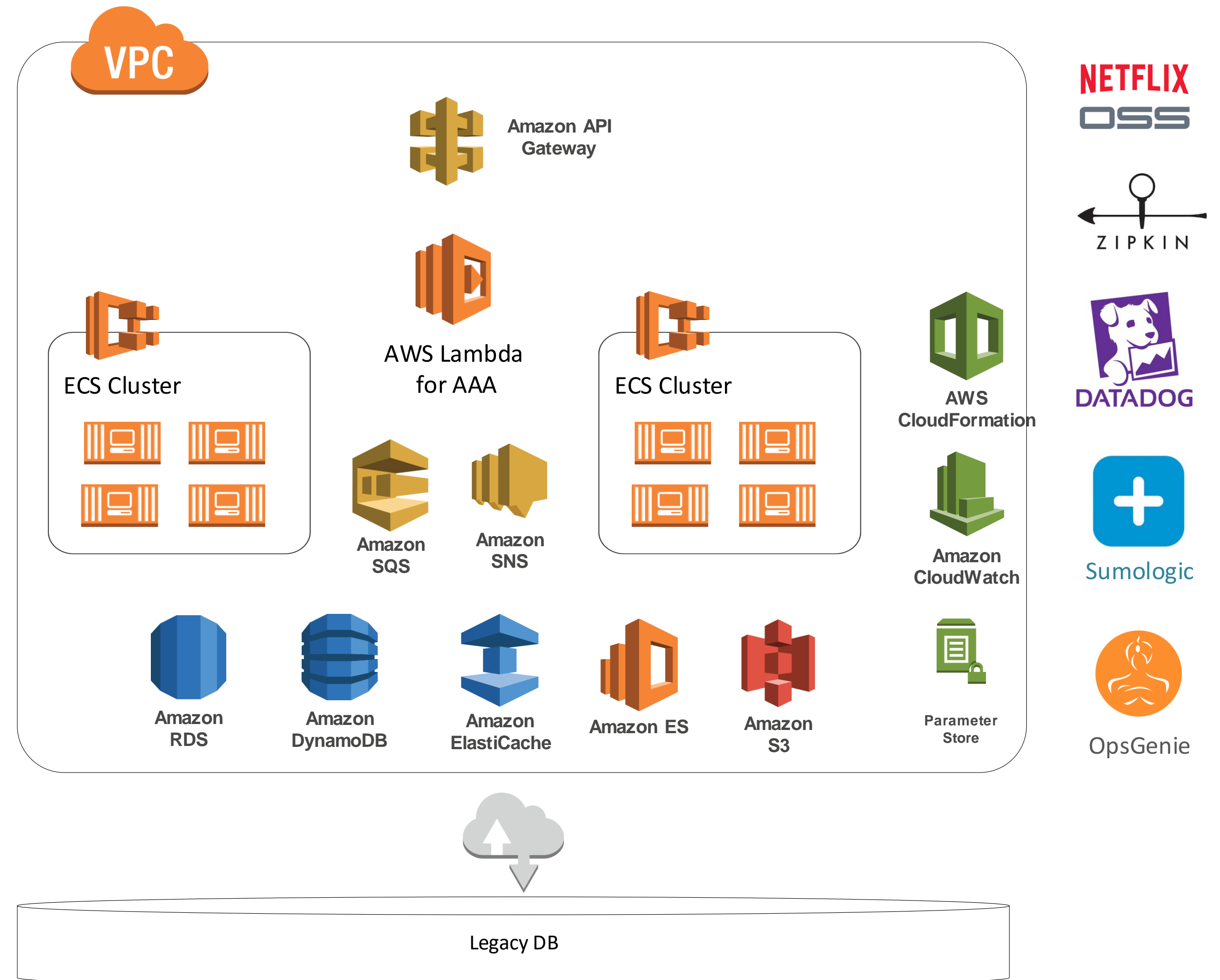
# What Microservices Platform Is ?





# One Microservices Platform

- Mixed AWS Cloud & OnPrem
- Platforms relies on Netflix OSS
- Dozens ECS clusters
- Hundreds of microservices
- Heterogenous ecosystem: Java, .Net, NodeJS
- Synchronous and asynchronous microservices communication





# Best Practices, Tips and Tricks

---

CI/CD pipelines, Generators, Dedicated Repos,  
Naming and Versioning, 24/7 Support,  
Infrastructure Cost

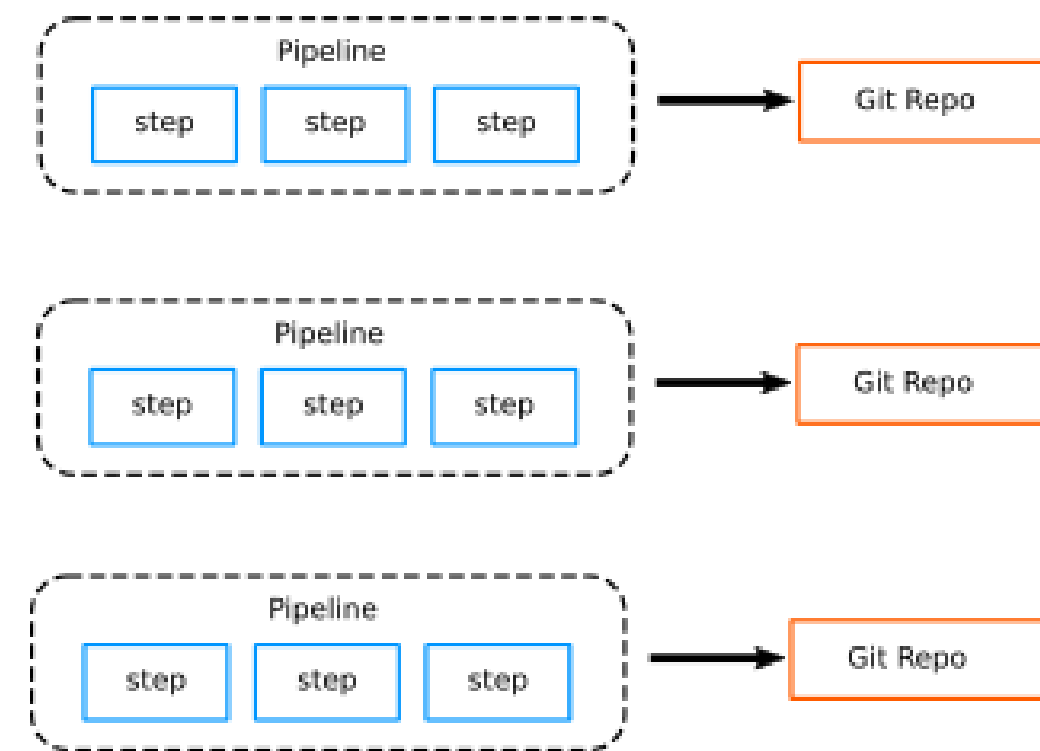
# Microservices

## CI/CD

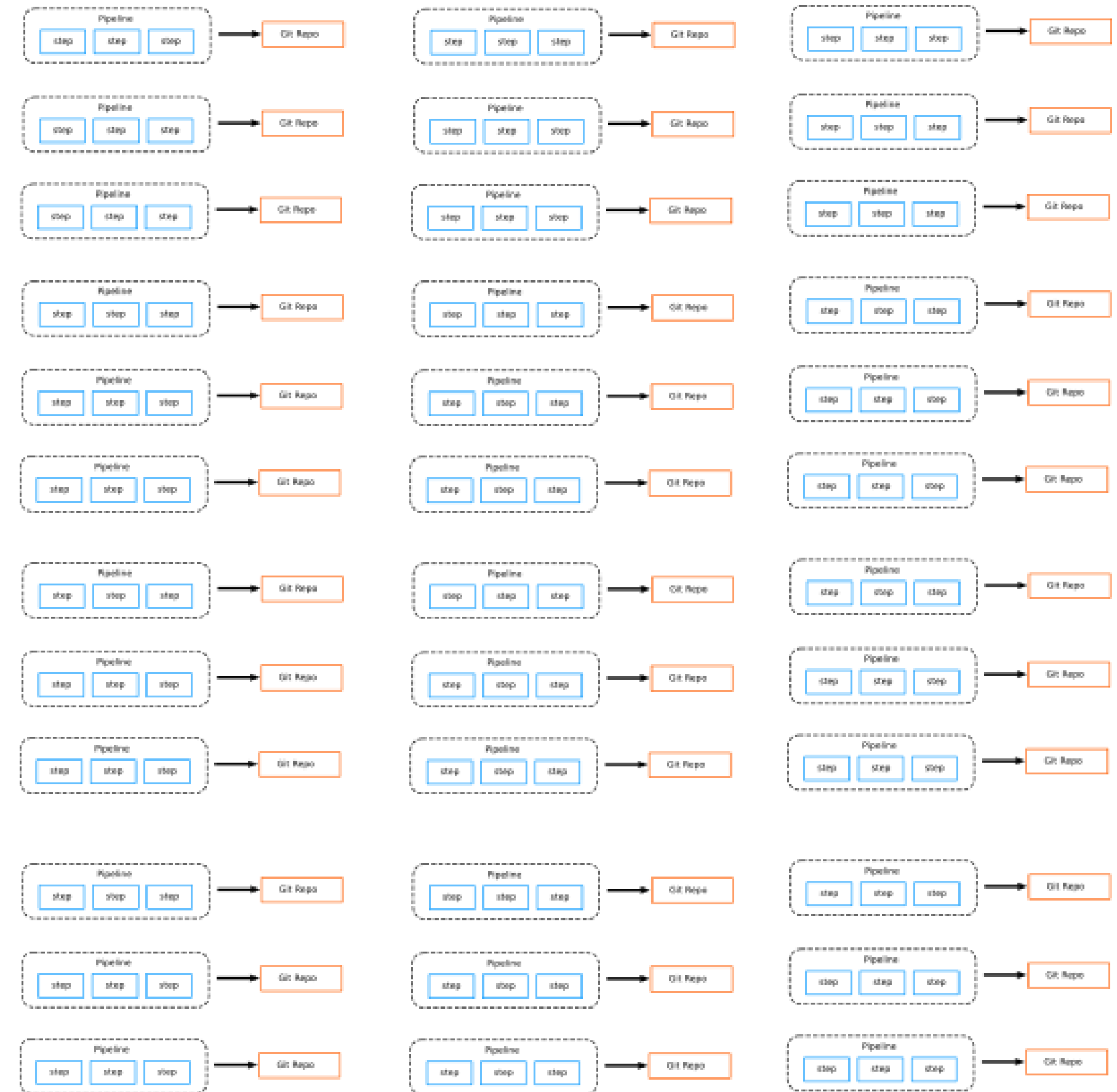


# CI/CD Pipelines Explosion

- Too many pipelines
- Efforts duplication
- Too much variety
- Hard to follow enterprise policies



3 monolithic applications



Each application split to 12 microservices

## CI/CD Pipelines shared library & templates

---

CI/CD Pipelines does a lot of works like:

- Runs code analysis tool (Sonar)
- Builds artifacts
- Publishes artifacts
- Runs unit, functional, integration and other tests
- Runs SAST & DAST
- Custom checks ( deployment to prod from master only)



### Shared Libraries

- Pipeline Building blocks
- API for tools
- Agents injection

### CI/CD Pipeline Templates.

- Default CI/CD pipelines
- Sample book



# Lessons Learned

---

- CI/CD pipelines are very similar
- Makes Sense to diff CI and CD pipelines
- Template pipeline is helpful
- Template pipeline is to be customizable
- Template updates distribution is A disaster





# Microservices Generators



# Microservices Generator Capabilities

---

- Integrates with Service Discovery Eureka
- Sends tracing meta information to a tracing tool Zipkin.
- Sends metrics to a monitoring tool DataDog
- Sends logs to SumoLogic
- Cloudformation script includes ECS Service and TaskDefinition configuration





# Microservices Generator Capabilities

---

- Default scaling strategy by CPU
- Can be build and deployed to all environments  
DEV, INT, LIVE
- Configuration for Cost Optimization function
- Imported CI/CD pipeline in Jenkins with  
configured hooks and integrated with  
SonarQube and JFrog artifactory



# Lessons Learned

---

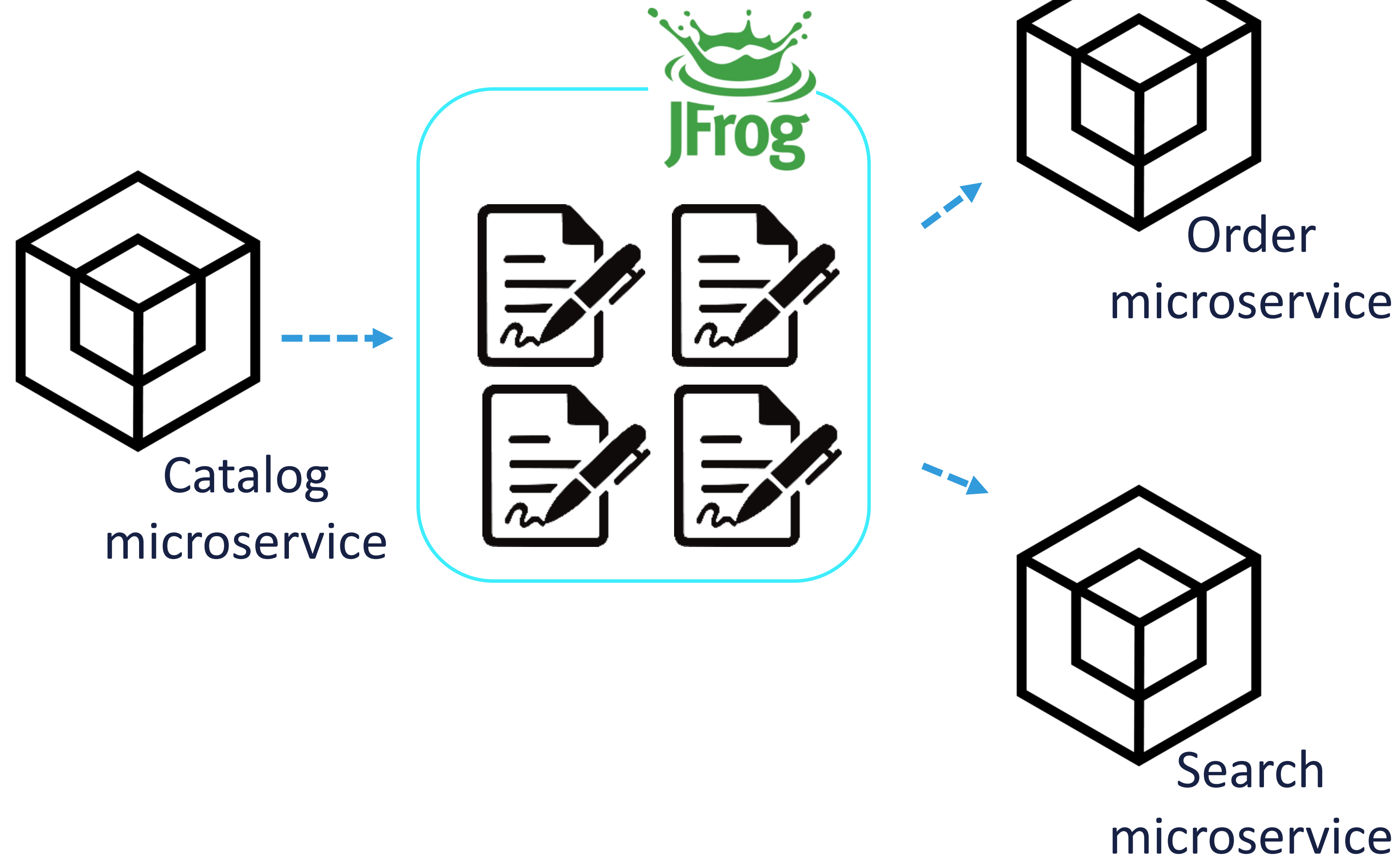
- Generators save incredible amount of efforts
- One generator needs per each language
- Generators bring version implicit conflicts
- Virtual team to own generators



# Dedicated Repos For DTO & Contracts

## Dedicated Repo for Contracts

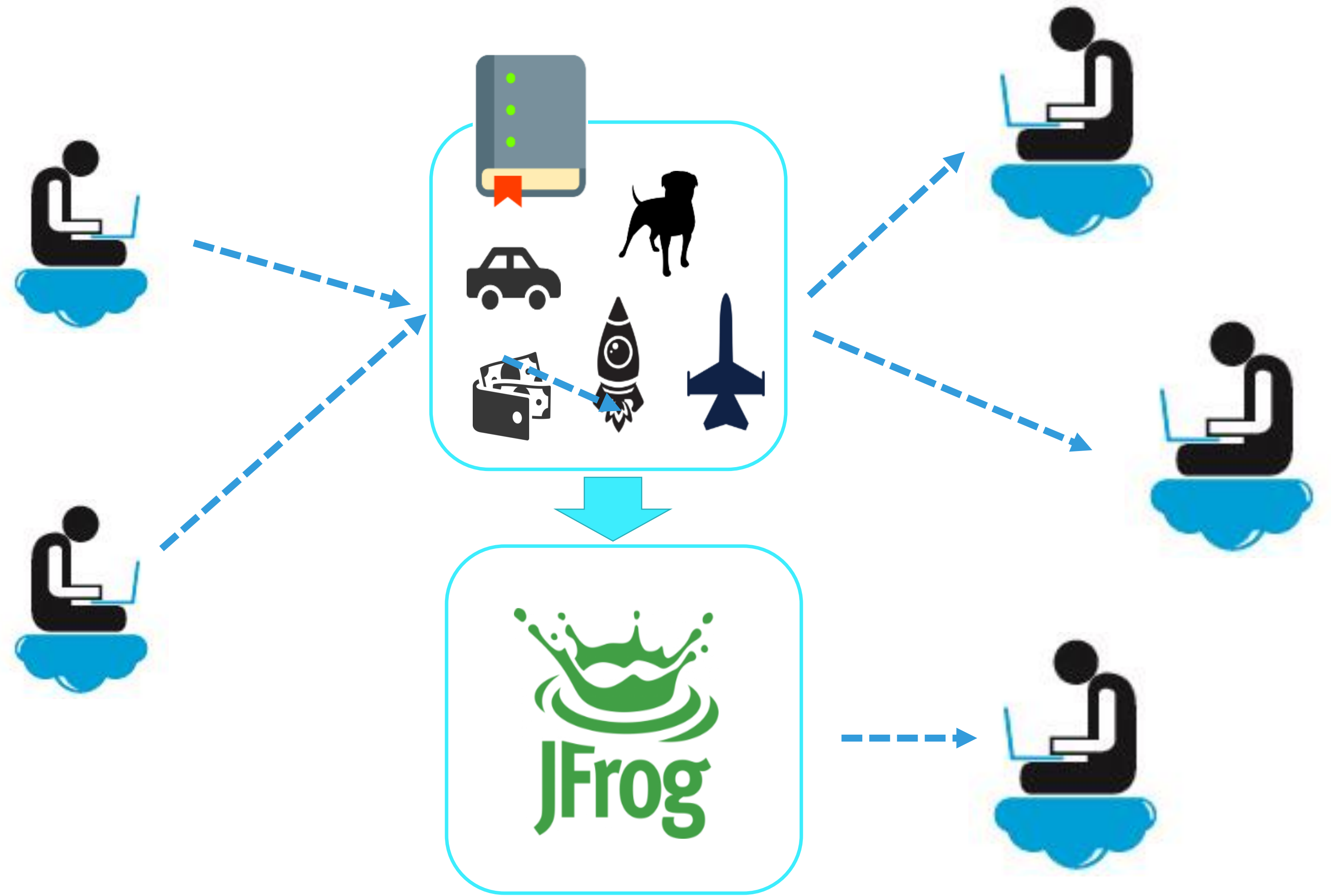
- Single source of truth
- Author published a contract
- Author publishes API stub server
- Author publishes API tests
- Consumers write an API client





# Dedicated Repo for Data Transfer Objects

- Single source of truth
- DTOs in JSON
- Java DTO can be built





# Lessons Learned

---

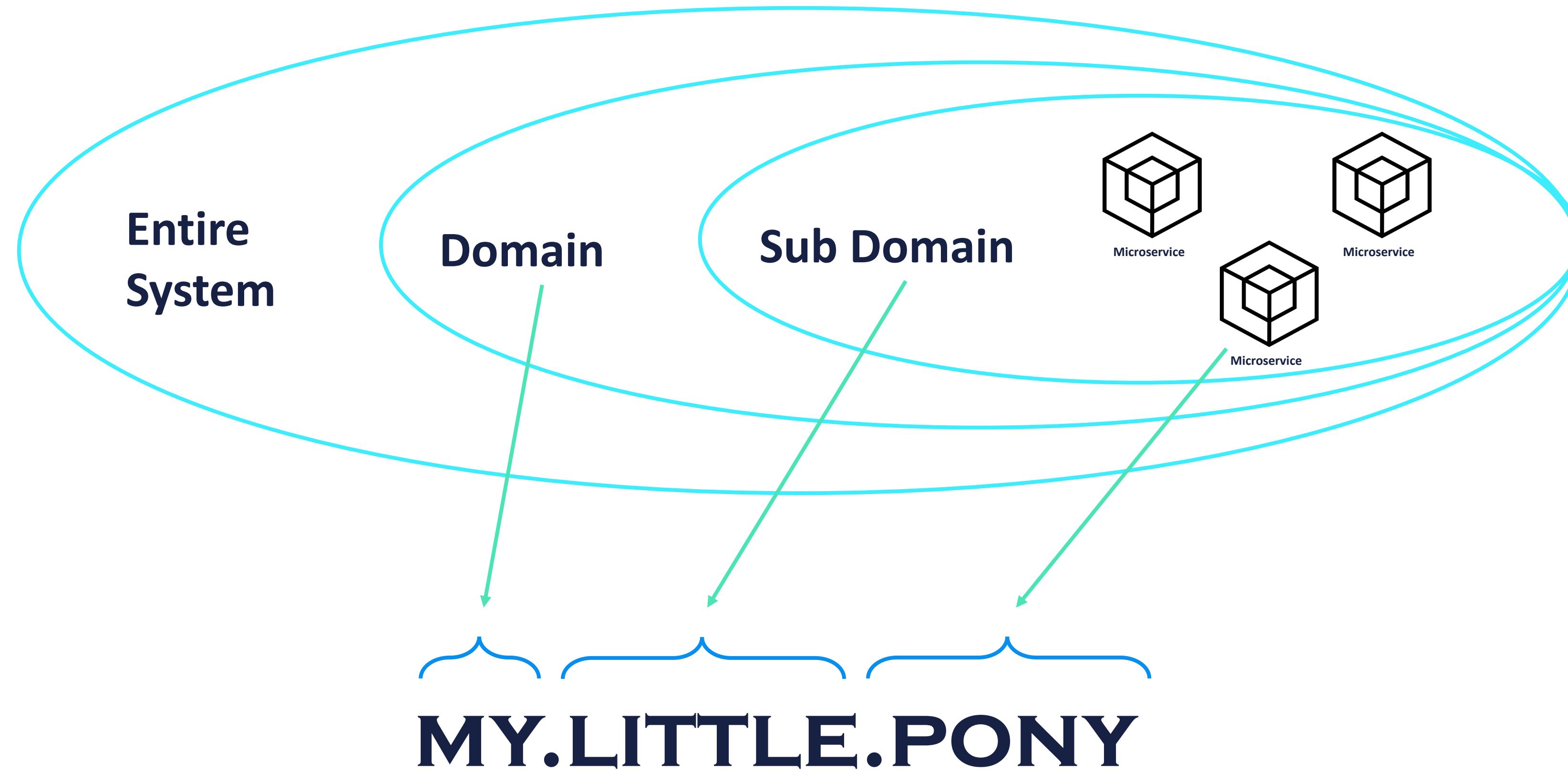
- Single source of truth for all microservices
- Architects govern activity via NFRs
- Architects owns repos
- Needs a channel of schema update notifications



# Microservices Naming & Versioning

# Microservices Naming

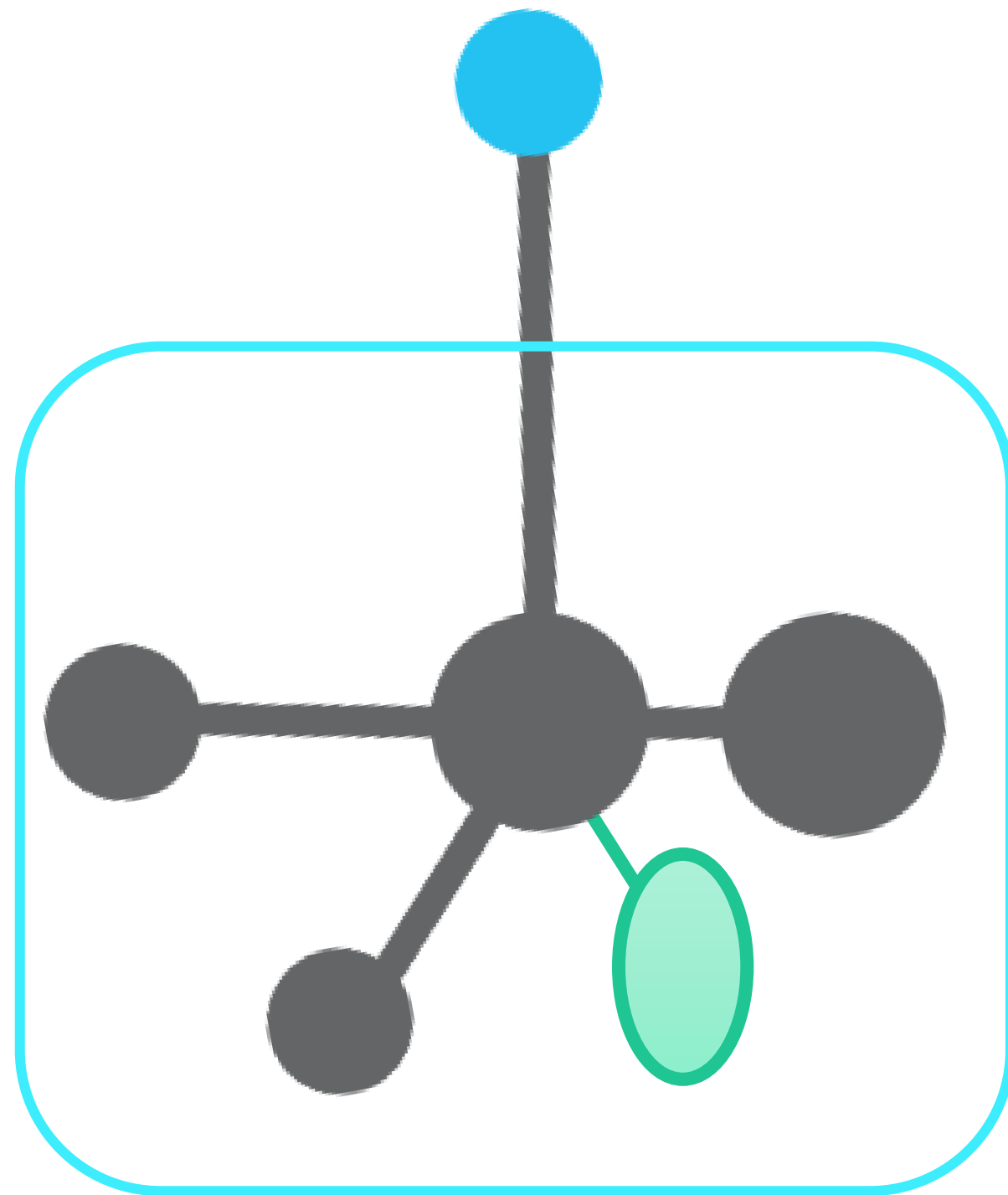
---



# Microservices Versioning

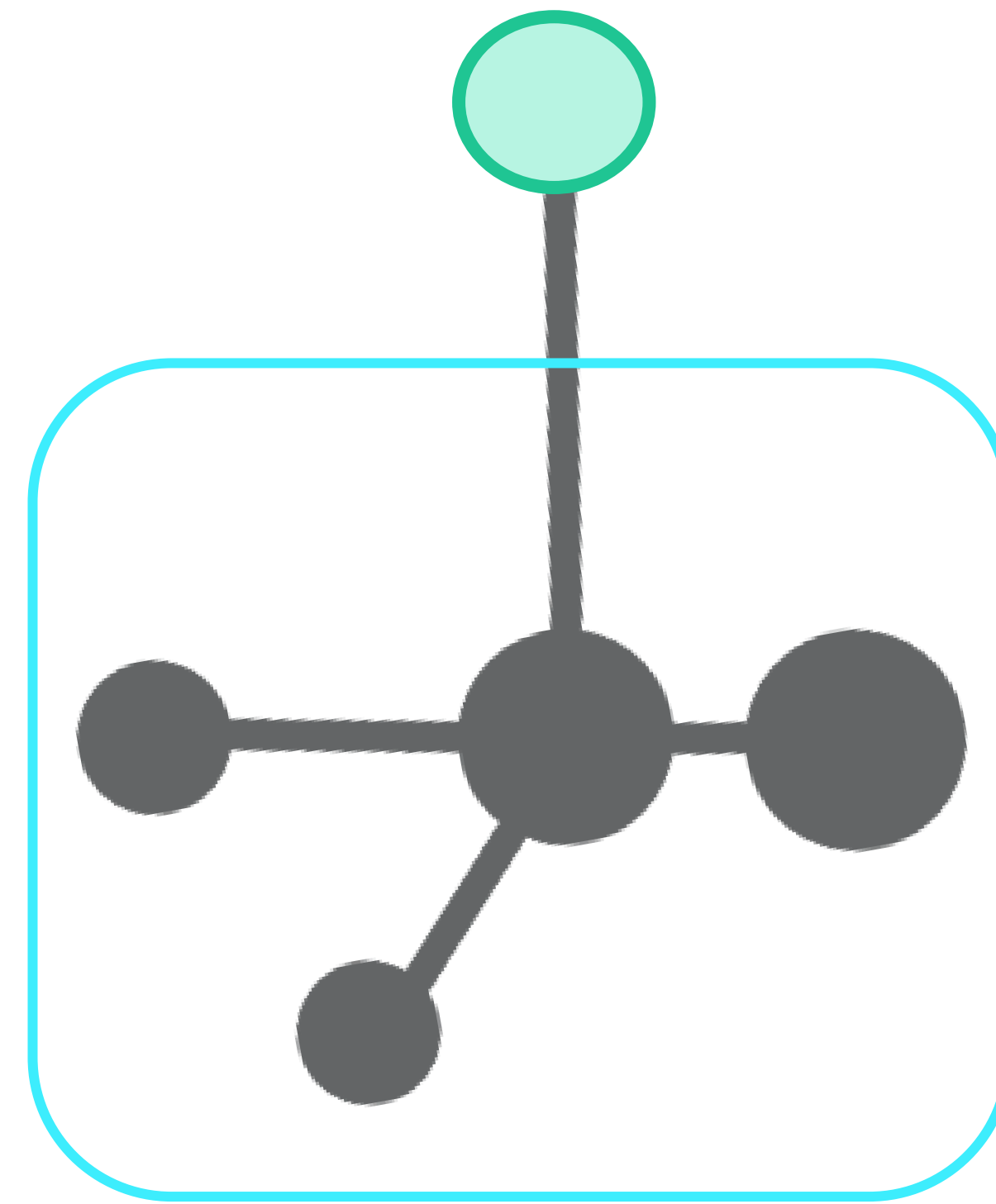
---

Endpoint



Microservice  
Order

Non-Backward  
Compatible Endpoint



Microservice  
OrderExtendedSchema



# Lessons Learned

---

- Simple but important otherwise total mess
- Architects track all new repo & microservices
- Always latest is simple but much duplication
- Still use versions for shared templates/libs
- Hard to invent new names

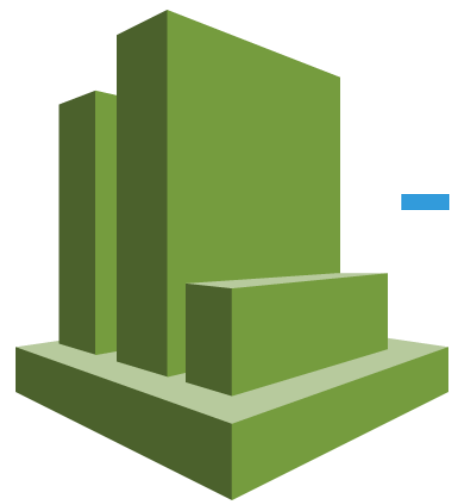




**24/7 Support**

# 24/7 Production Support

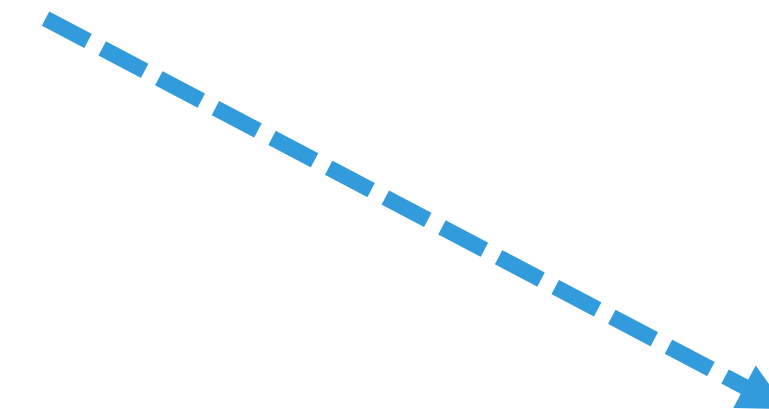
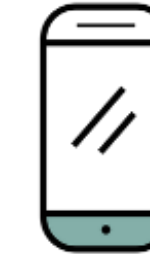
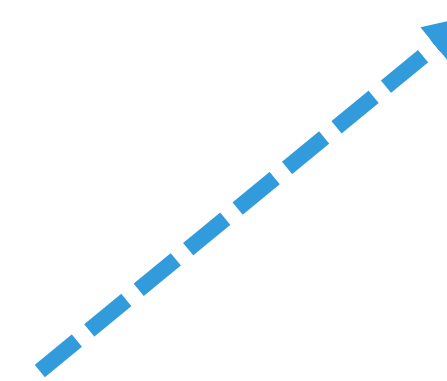
IT Home



Amazon CloudWatch



sumologic®



Boss

# Incident Management Platform OpsGenie



IT Home

opsteam\_schedule (GMT-05:00) America/New\_York EDT

TODAY < > Apr 6 - Apr 12

1 Week 2 Weeks 1 Month

Calendar Timeline

Rotations +

	Mon 6/4	Tue 7/4	Wed 8/4	Thu 9/4	Fri 10/4	Sat 11/4	Sun 12/4
Rot1	John Smith	Jack Brown	John Smith	Jack Brown	John Smith		
Rot2						Bob Wilson	

Overrides + Take on call for an hour

	Mon 6/4	Tue 7/4	Wed 8/4	Thu 9/4	Fri 10/4	Sat 11/4	Sun 12/4

Final Schedule

	Mon 6/4	Tue 7/4	Wed 8/4	Thu 9/4	Fri 10/4	Sat 11/4	Sun 12/4
	John Smith	Jack Brown	John Smith	Jack Brown	John Smith		
						Bob Wilson	

<< Back to schedules

# Lessons Learned

---

- Hard to establish the process
- Code quality becomes much better
- Team Velocity somewhat decreased
- Some people refuse work shifts



# Cloud Infrastructure Cost

A decorative graphic featuring a large, irregular blue circle. Three light blue mathematical symbols are positioned around the circle: a multiplication sign (x) at the top left, a plus sign (+) at the top right, and another plus sign (+) at the bottom center.



# Useless Expenses

<input type="checkbox"/>	Region/AZ	Instance ID	Instance Name	Instance Type	Estimated Monthly Sa...	CPU Utilization 14-Day Average	Network I/O 14-Day Av...	Number of Days Low
<input type="checkbox"/>	us-east-1a	<a href="#">i-02d56c7295c31ea2e</a>	EksJ-VpcA-useast1-Deli...	m5.4xlarge	\$552.96	0.0%	0.13MB	14
<input type="checkbox"/>	us-east-1a	<a href="#">i-0cb1109a1b8d4562d</a>	EksK-VpcA-useast1-Del...	m5.4xlarge	\$552.96	0.0%	0.48MB	14
<input type="checkbox"/>	us-east-2c	<a href="#">i-04bcdbdee9a98f9ce</a>	RR_ImpTRP_3	c5.2xlarge	\$509.76	0.0%	0.08MB	14

⏪ ⏩ 1 to 20 of 20 ⏪ ⏩ View 20 ▼							
<input type="checkbox"/>	Region	DB Instance Name	Multi-AZ	Instance Type	Storage Provisioned (GB)	Days Since Last Connection	Estimated Monthly Savings (...)
<input type="checkbox"/>	us-east-1	<a href="#">vpcardssamlauthdba-medical-...</a>	False	db.r3.large	1	14+	\$208.80
<input type="checkbox"/>	us-east-1	<a href="#">userhistorymiddledba-medical-...</a>	False	db.r4.large	1	14+	\$208.80

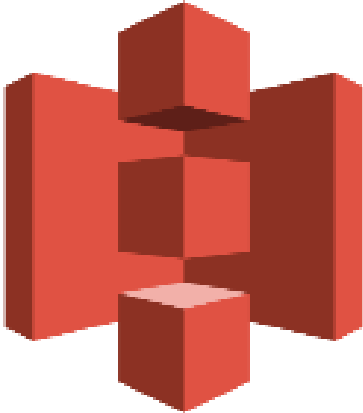
# Infrastructure Cost Function



- Tagging in CF:
- Domain
  - SubDomain
  - Team



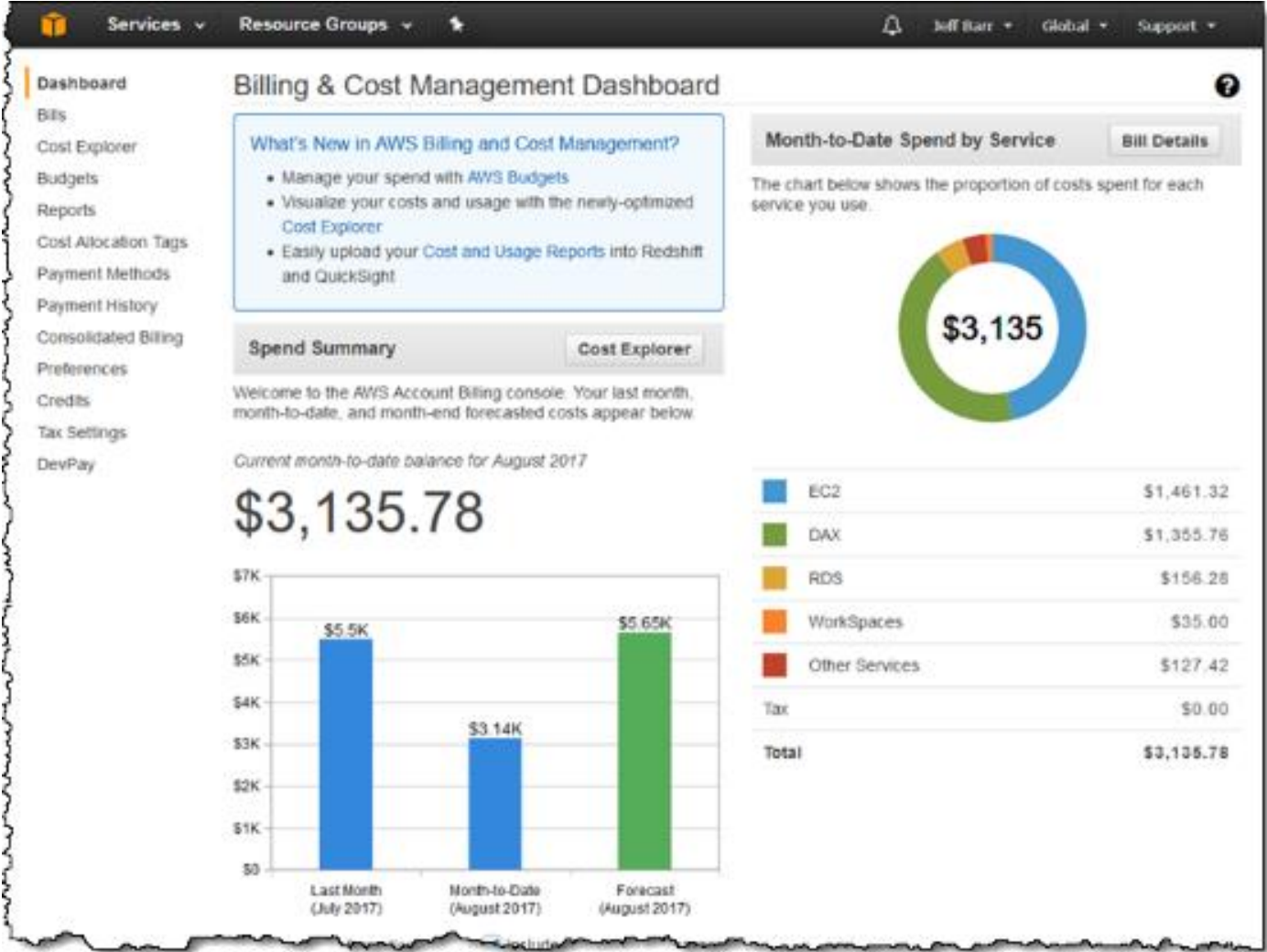
Amazon  
RDS



Amazon  
S3



Amazon  
ElastiCache



# Lessons Learned

---

- Tag your resources
- Track infra cost of any changes
- Clean resources automatically
- Developers does not use resources efficiently



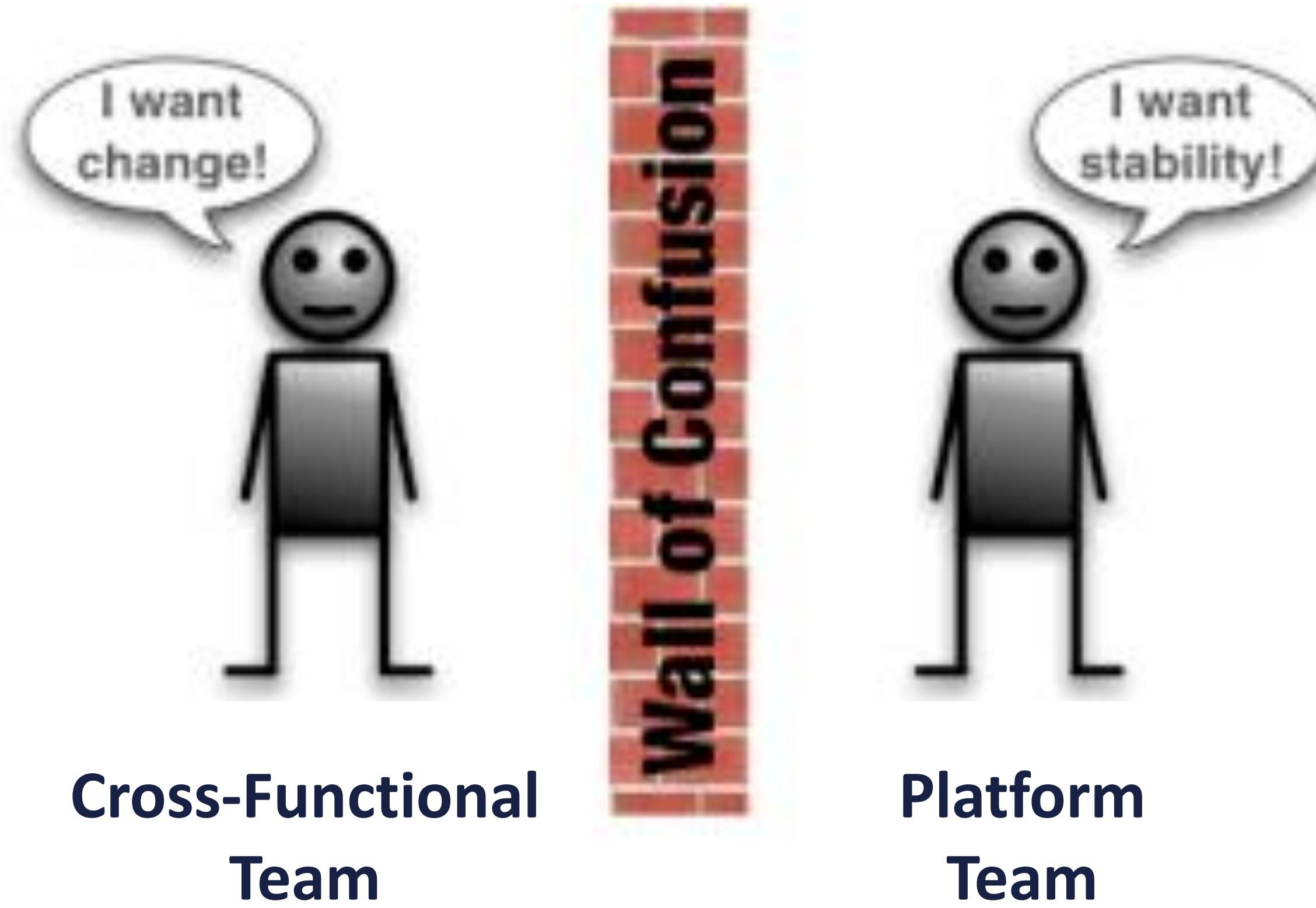


# Dedicated Platform Team



## Existed Communication Pattern

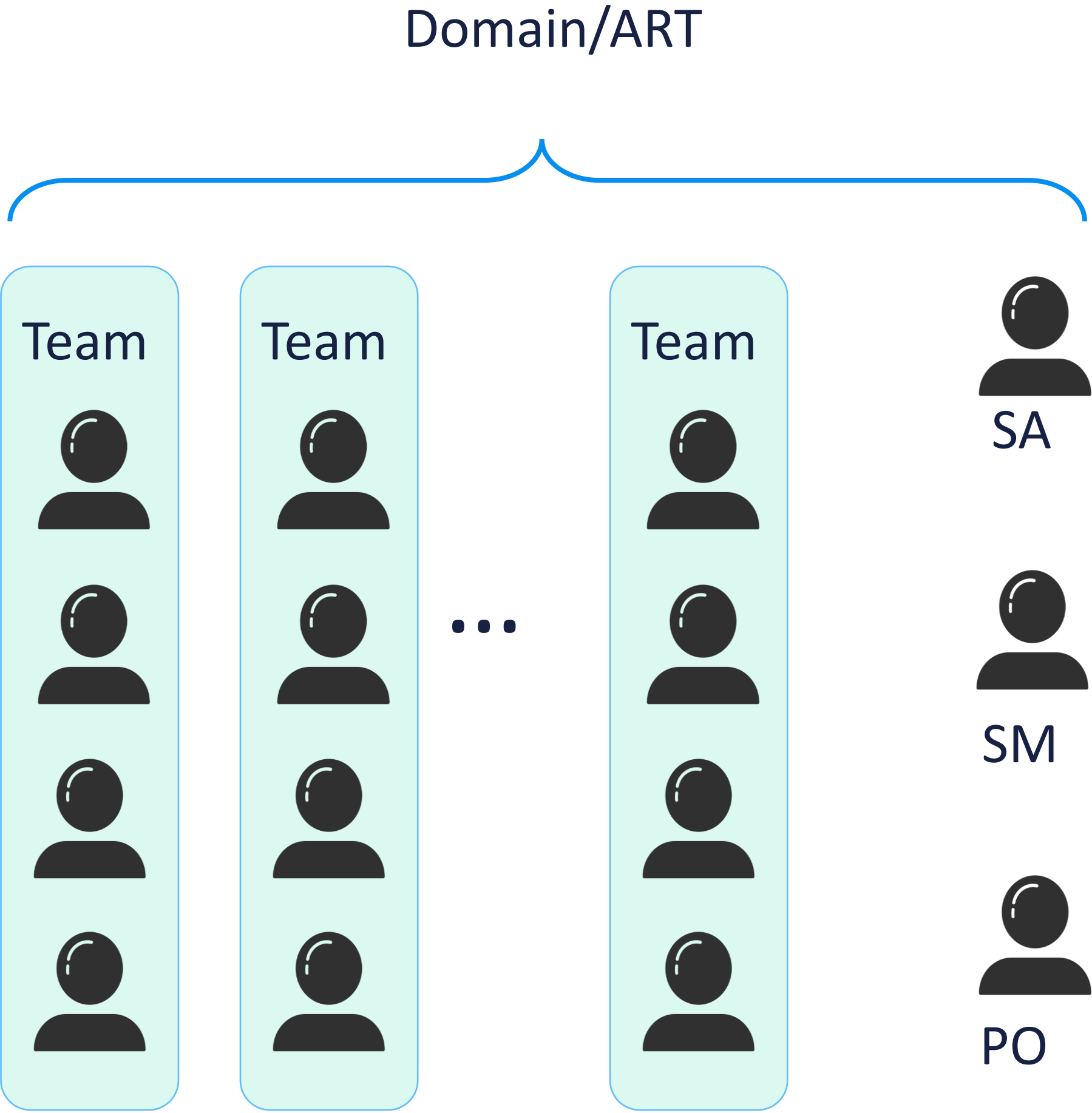
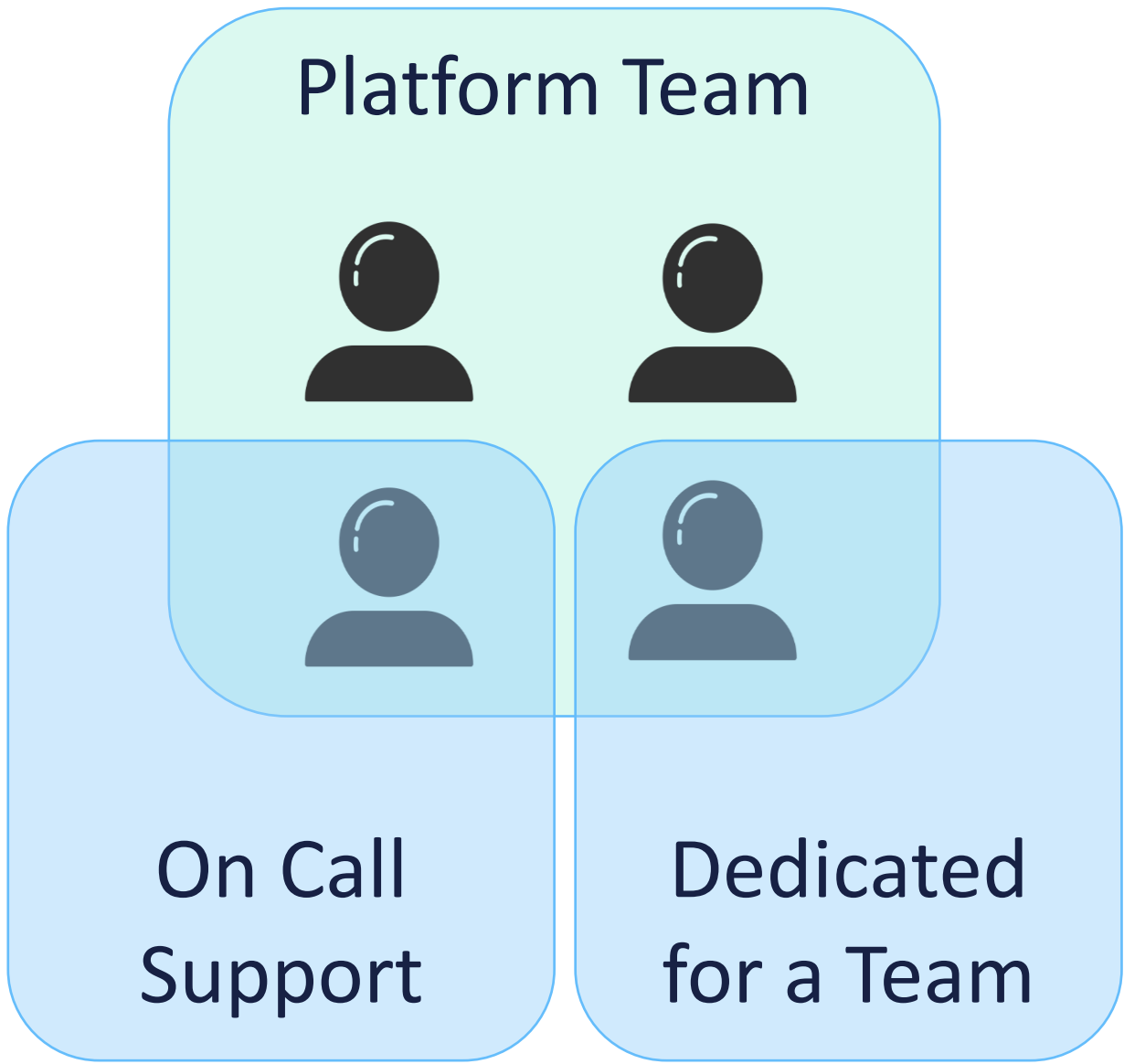
---





# Existed Communication Pattern

IT Home



# Lessons Learned

---

- There is work not for cross functional teams
- Cross functional teams are not fully independent
- Direct involvement works





**THANK  
YOU!**