

# Cloud Native 架构的演进之路

代闻

亚马逊AWS解决方案架构团队 技术经理



# QCon

全球软件开发大会

## 成为软件技术专家的 必经之路

[北京站] 2018

2018年4月20-22日 北京·国际会议中心

**7折** 购票中, 每张立减2040元  
团购享受更多优惠



识别二维码了解更多



# 极客时间

重拾极客精神·提升技术认知

## 下载极客时间App

获取有声IT新闻、技术产品专栏，每日更新



扫一扫下载极客时间App

# AiCon

全球人工智能与机器学习技术大会

助力人工智能落地

2018.1.13 - 1.14 北京国际会议中心



扫描关注大会官网

# 什么是 Cloud Native?

# 什么是 Cloud Native?

## Answer by CNCF

Cloud native computing uses an open source software stack to be:

### 1. Containerized.

Each part (applications, processes, etc) is packaged in its own container. This facilitates reproducibility, transparency, and resource isolation.

### 2. Dynamically orchestrated.

Containers are actively scheduled and managed to optimize resource utilization.

### 3. Microservices oriented.

Applications are segmented into microservices. This significantly increases the overall agility and maintainability of applications.

<https://www.cncf.io/about/faq/>

# 什么是 Cloud Native?

Answer by Adrian Cockcroft,

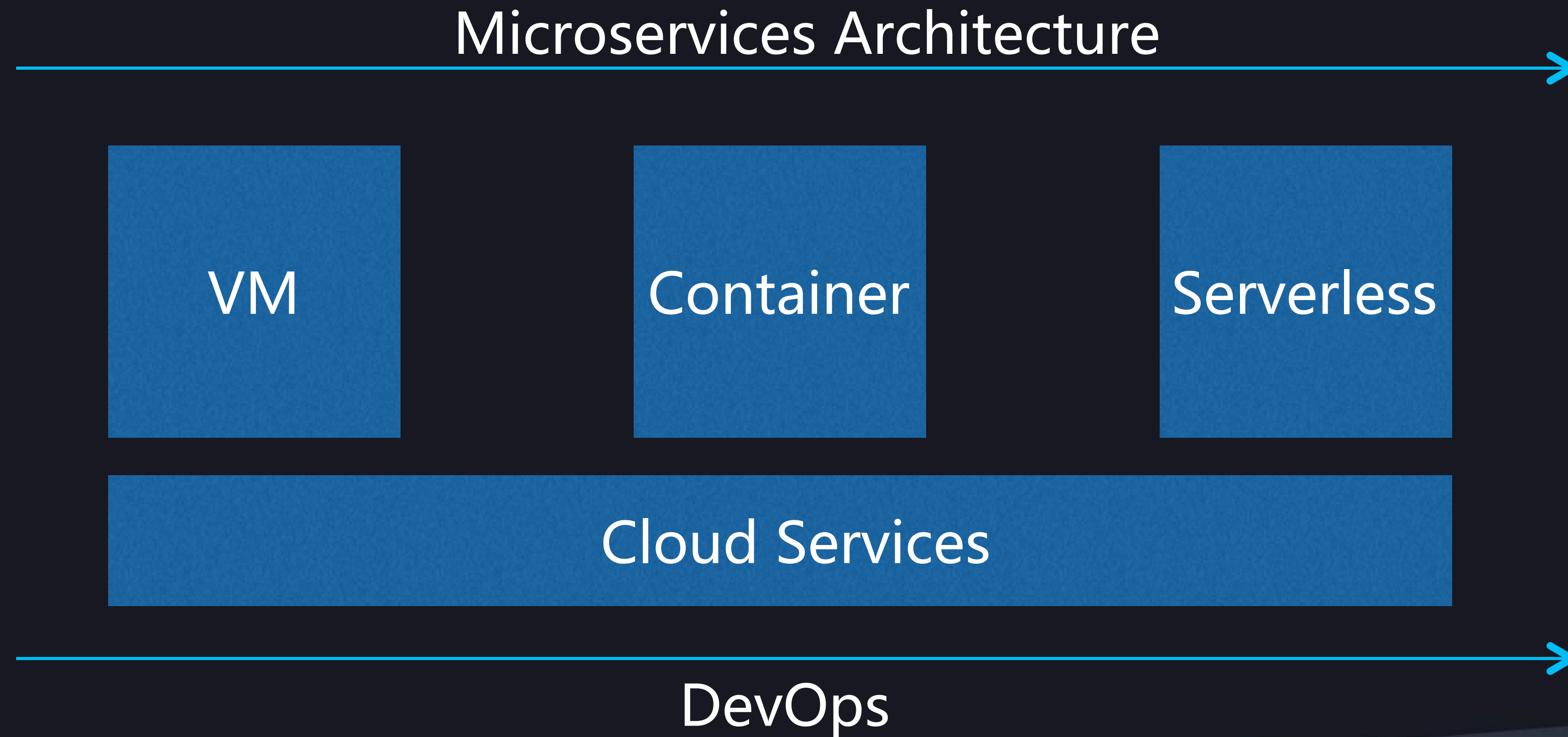
CNCF Board Member, AWS VP of Cloud Architecture, Former Cloud Architect of Netflix

Cloud native architectures take full advantage of *on-demand delivery, global deployment, elasticity, and higher-level services.*

They enable huge improvements in developer productivity, business agility, scalability, availability, utilization, and cost savings.

<https://medium.com/@adrianco/cloud-native-computing-5f0f41a982bf>

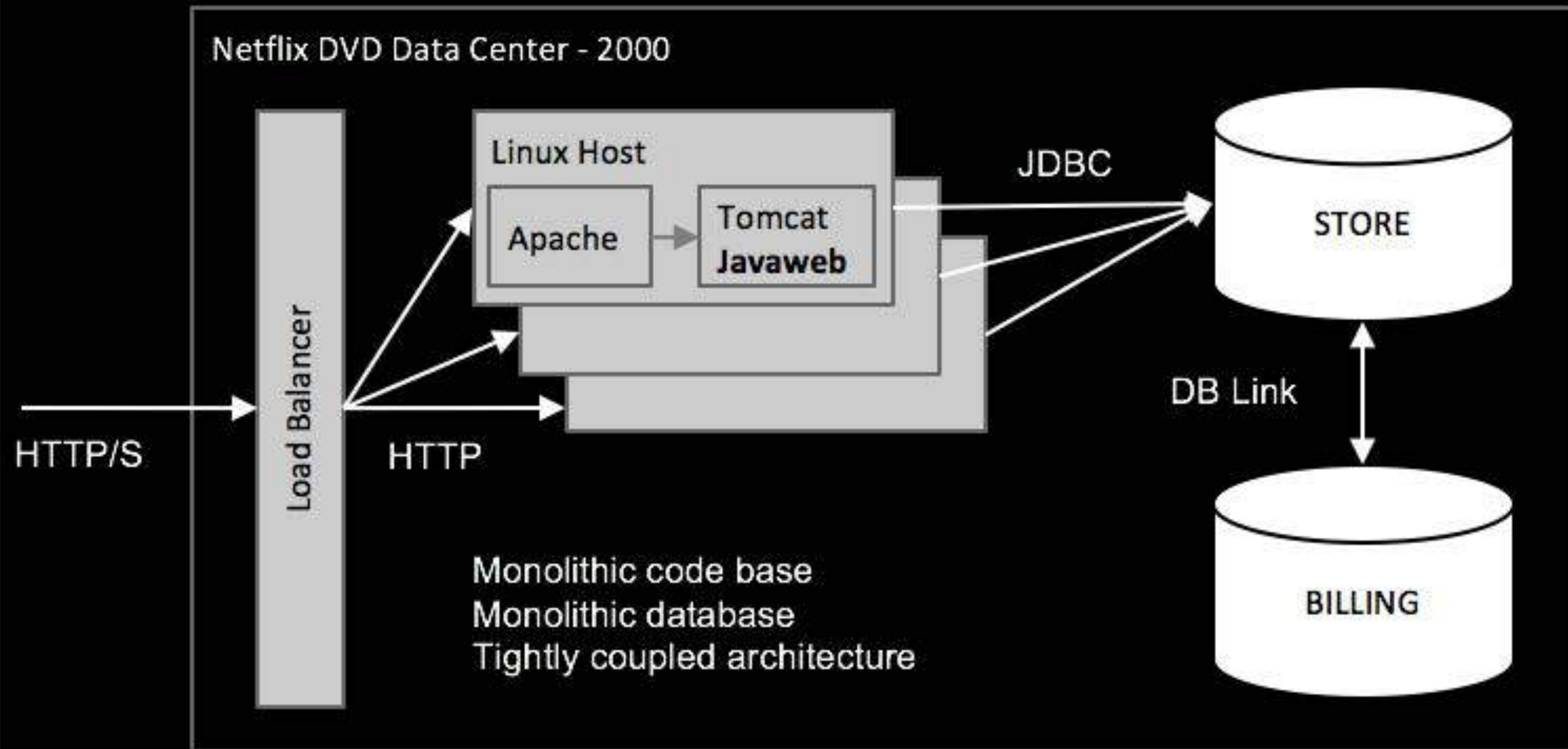
# Cloud Native 架构的演进





# NETFLIX

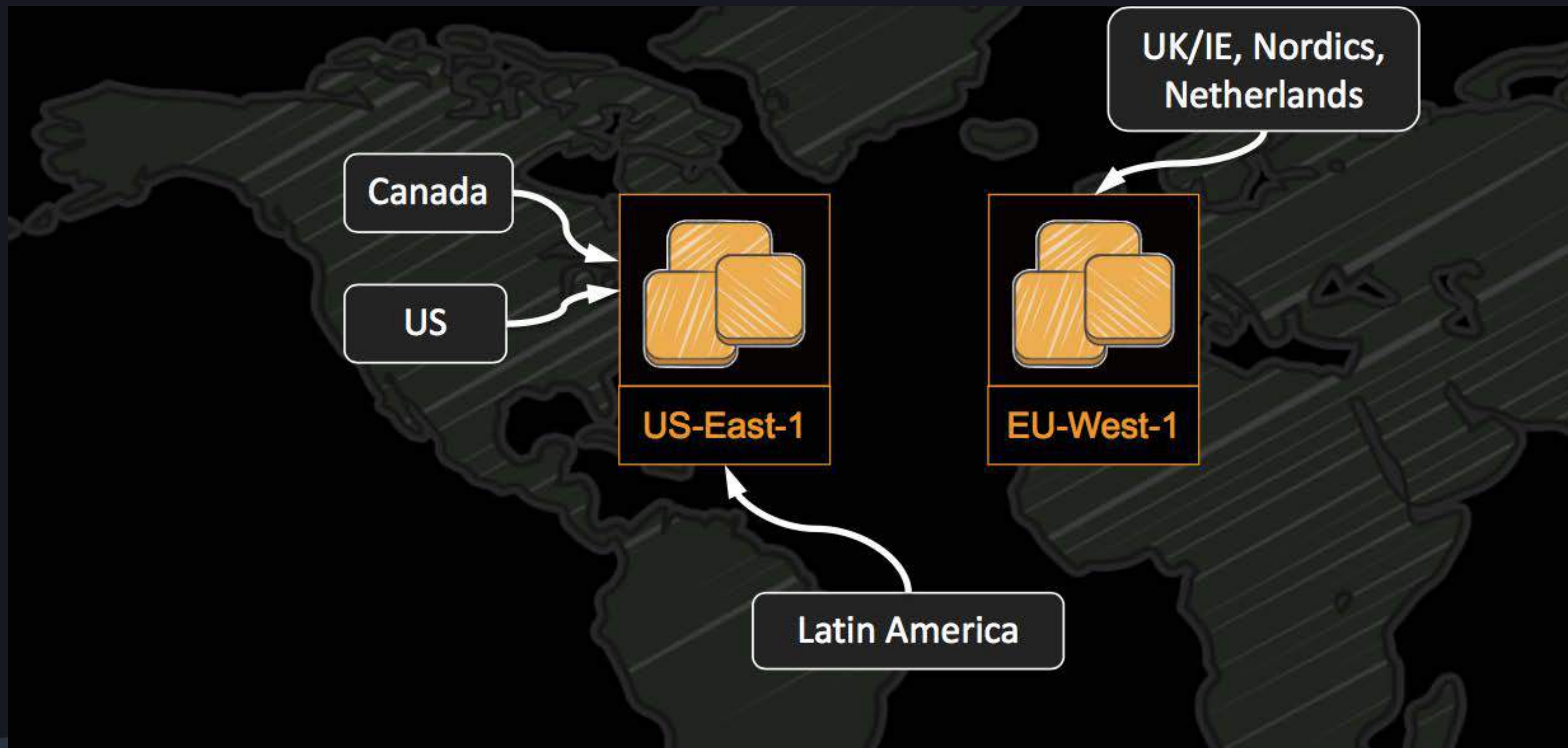
# Netflix Architecture before 2009



# 2010年开始向云上迁移



# 2012年多区域部署



# 2013 – 2014 完成多活



**Survive a large-scale regional service outage**

# Netflix Cloud Native 架构关键点

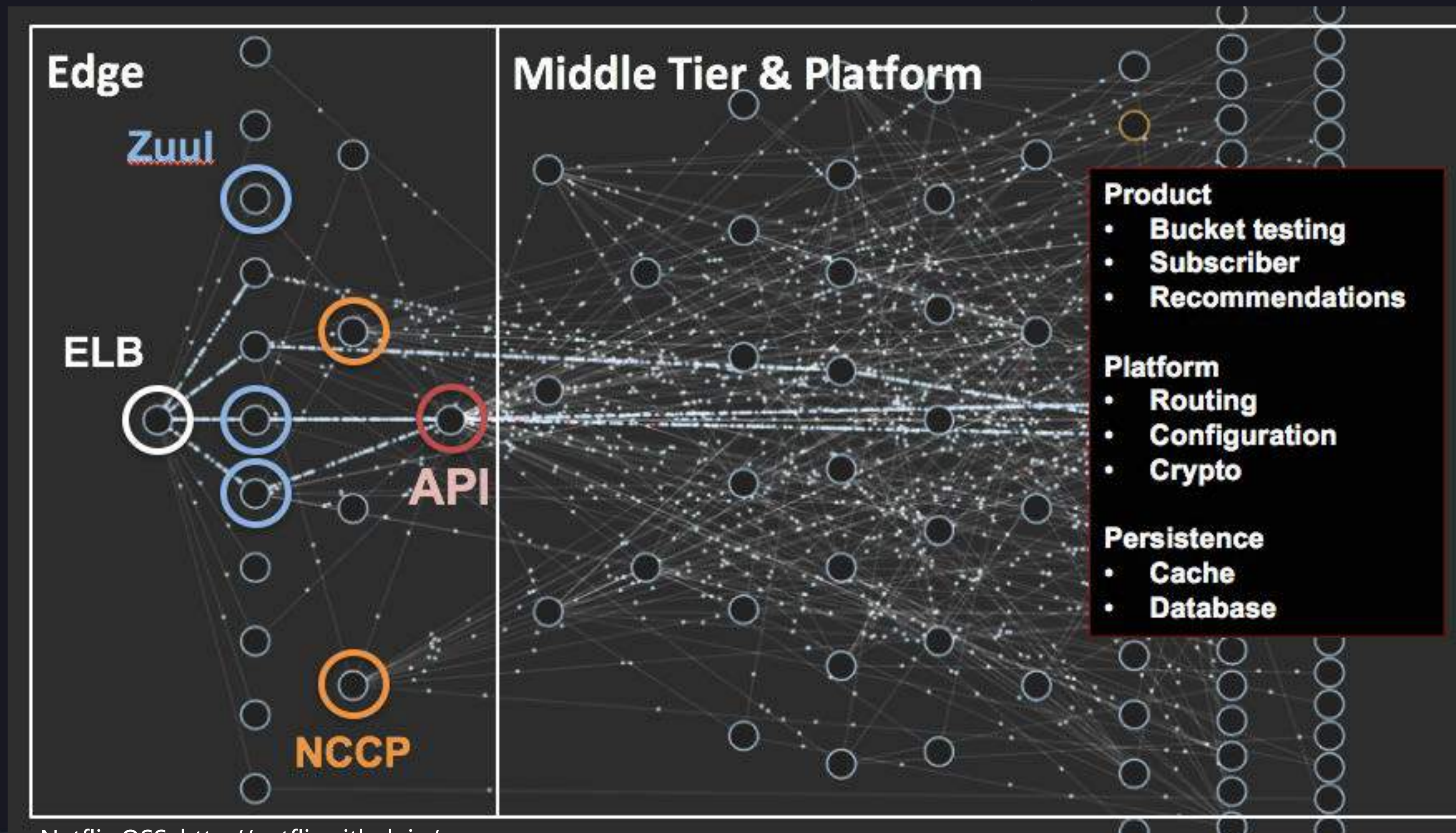
**Microservices**

**Database**

**Cache**

**Traffic**

# Netflix All-In AWS的微服务架构

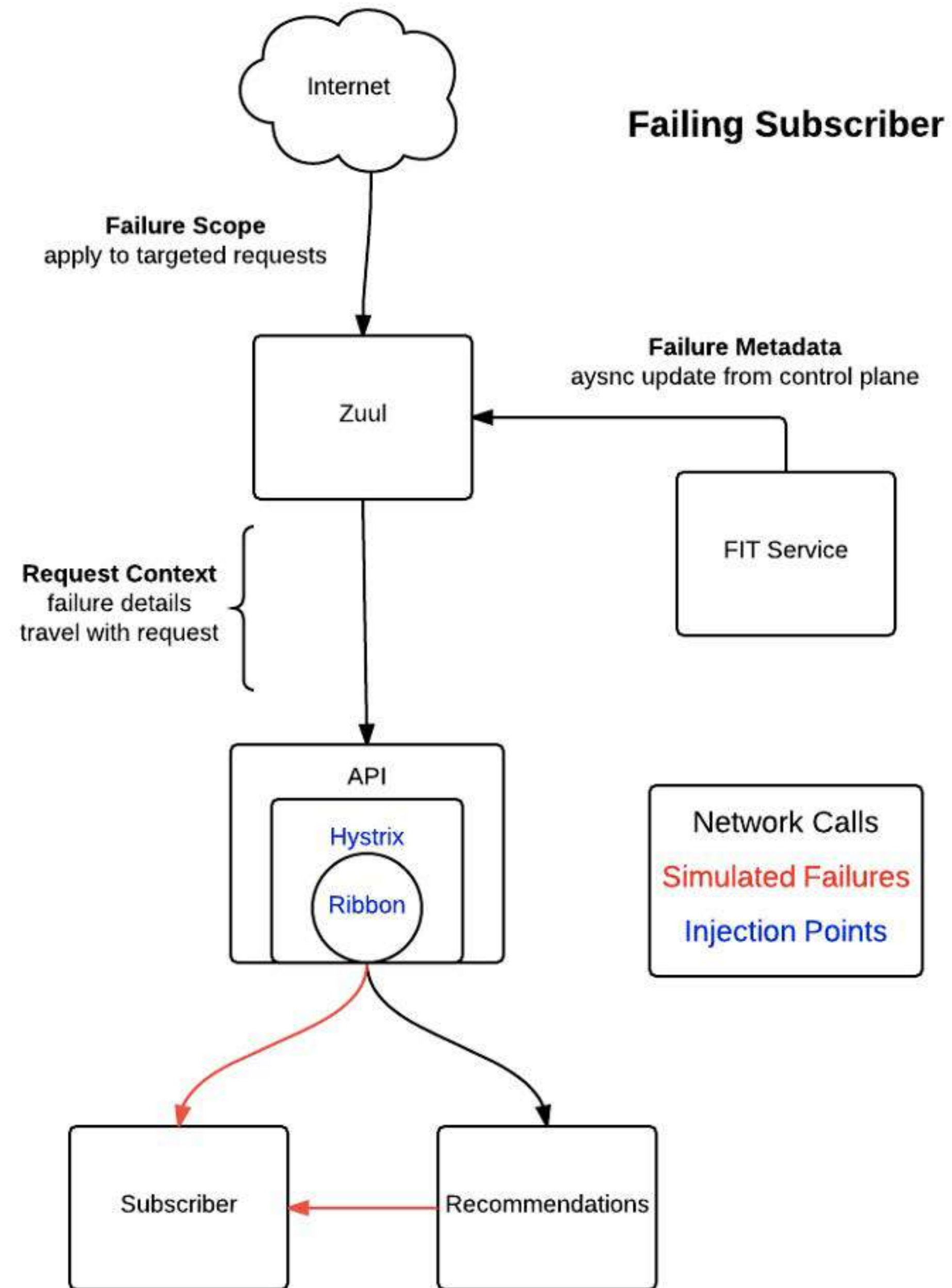


Netflix OSS: <http://netflix.github.io/>

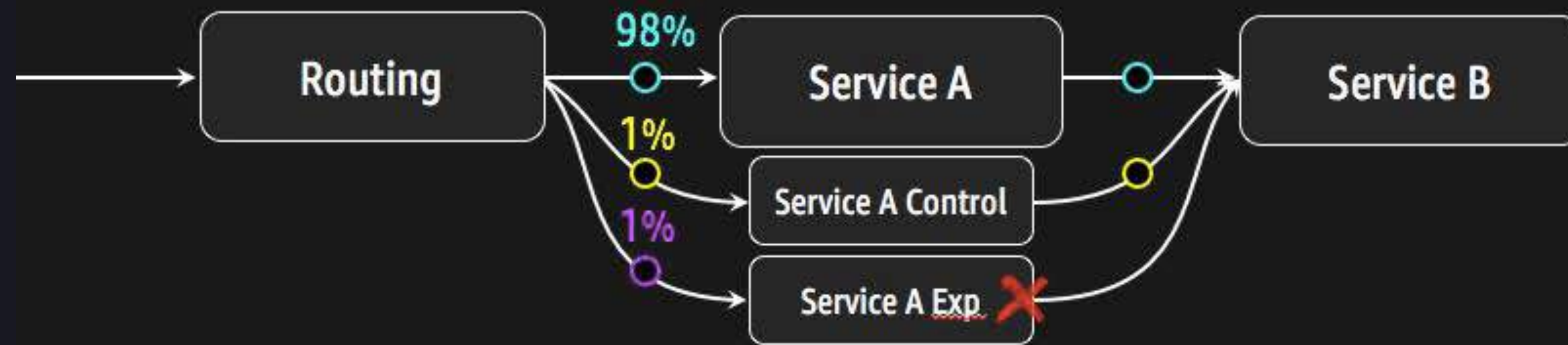
# Microservice 失效测试与恢复







# ChAP: Chaos Automation Platform





CHAOS DOESN'T CAUSE PROBLEMS.  
IT REVEALS THEM.

O'REILLY®

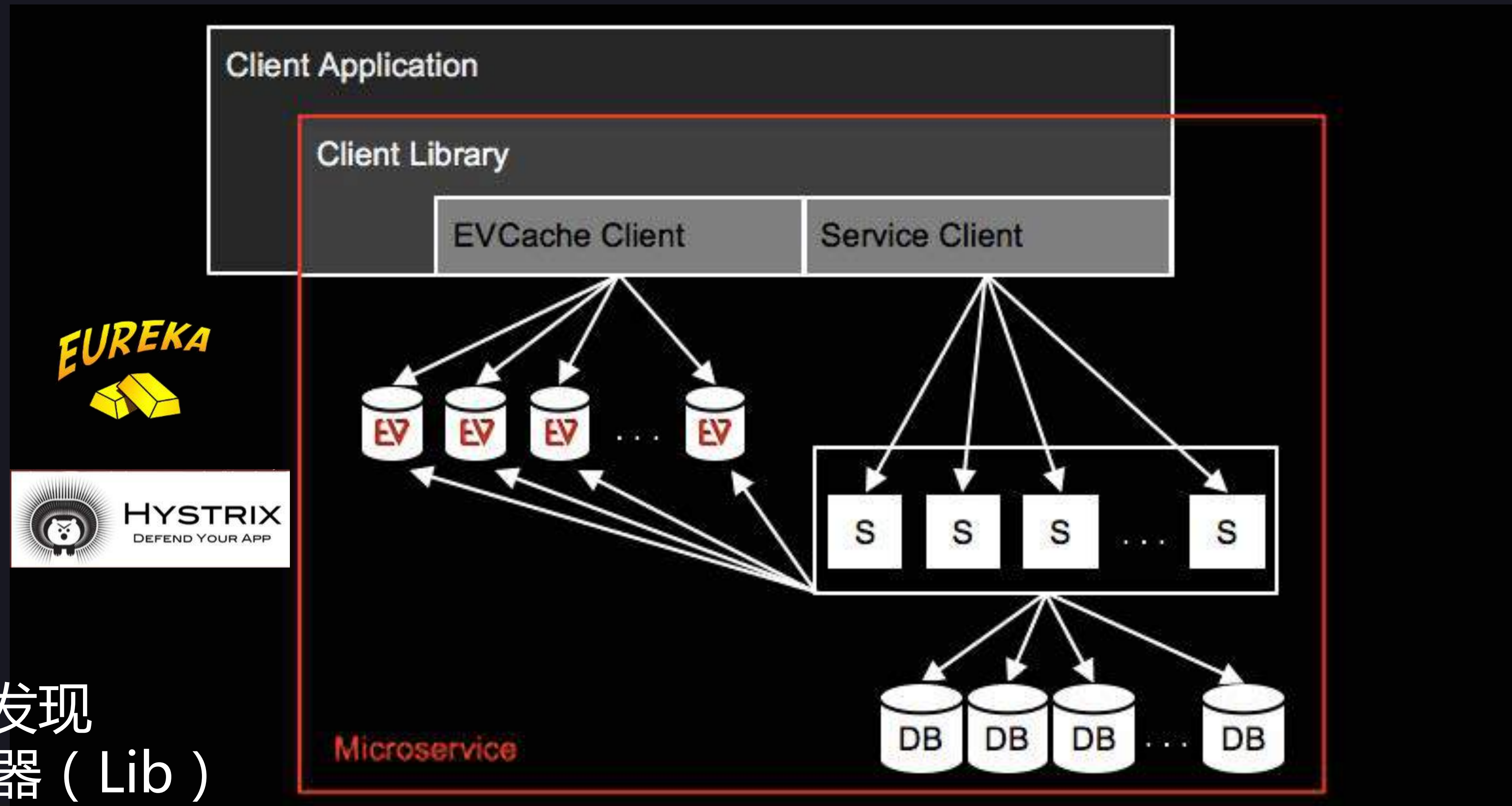
# Chaos Engineering

Building Confidence in System Behavior  
through Experiments



Casey Rosenthal, Lorin Hochstein,  
Aaron Blohowiak, Nora Jones  
& Ali Basiri

# 微服务典型单元

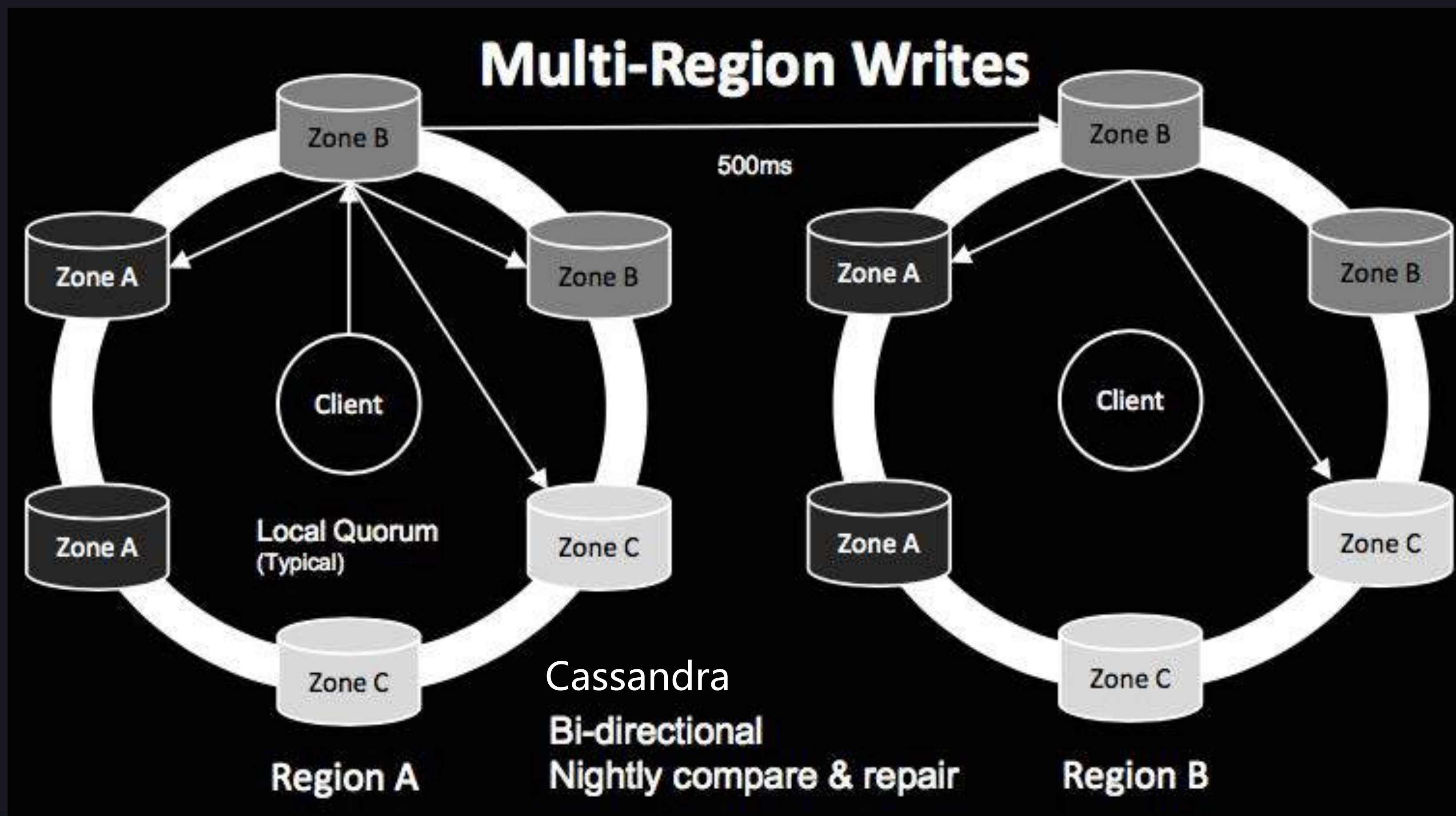


Eureka : 服务发现

Hystrix : 断路器 ( Lib )

# 数据库的高可用与最终一致性

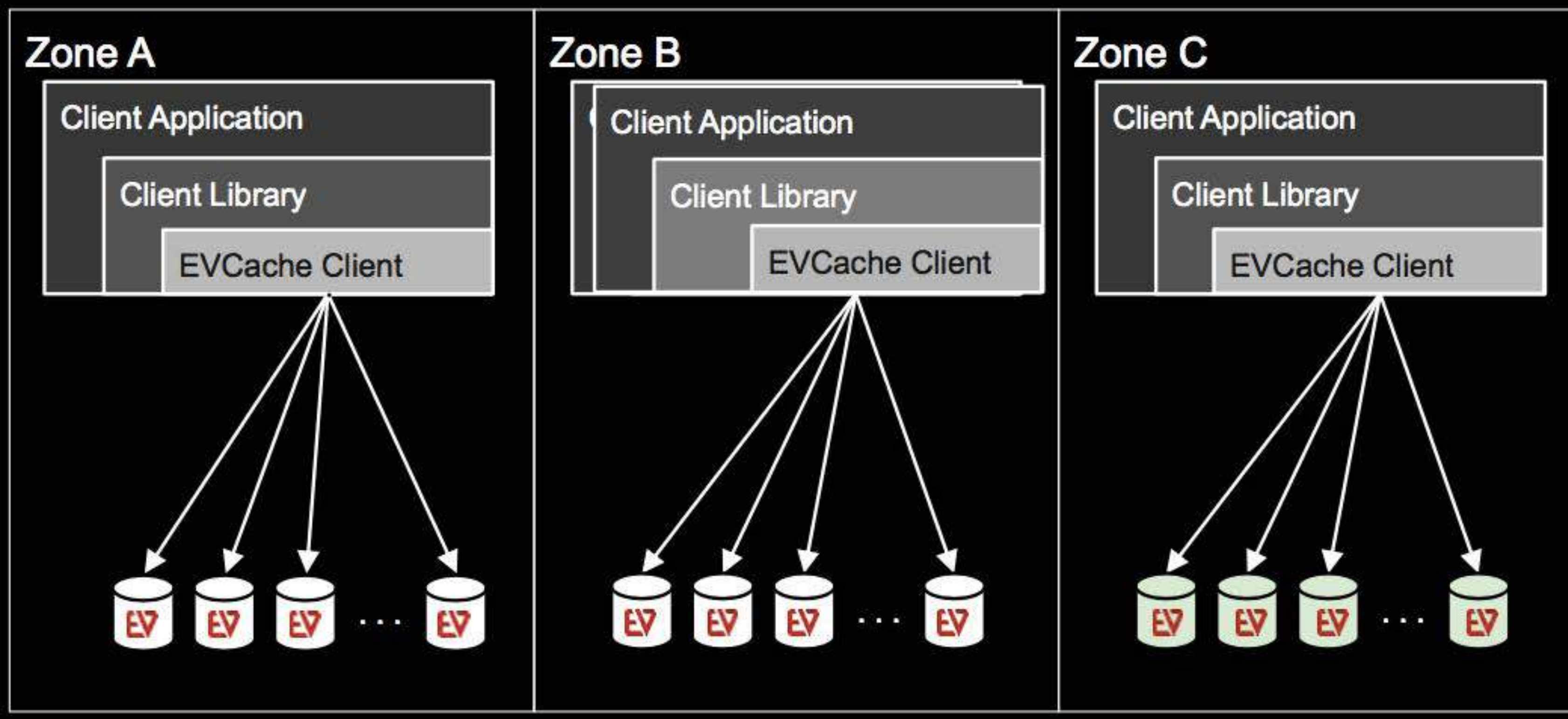
## Cassandra



# Cache的高可用与最终一致

## EVCache

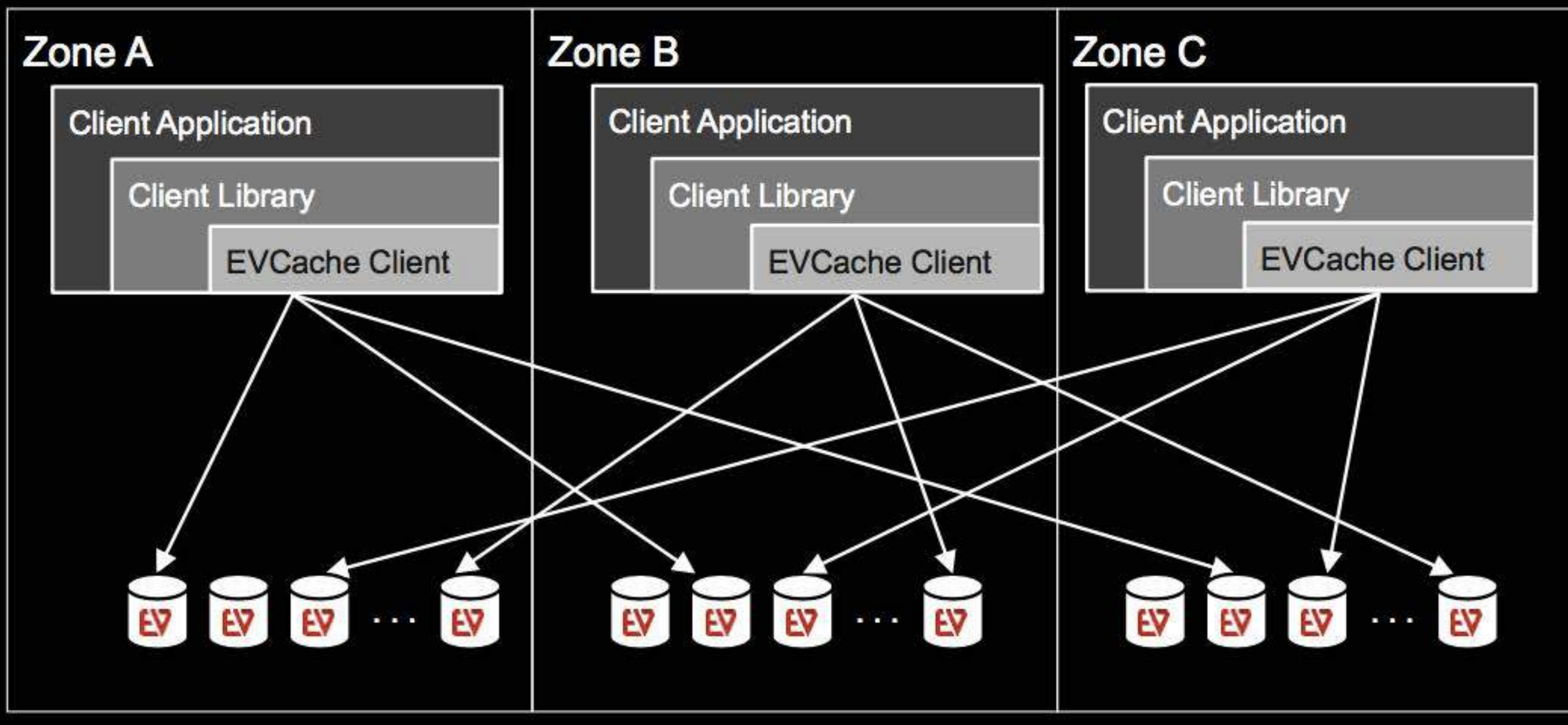
### Reads



# Cache的高可用与最终一致

## EVCache

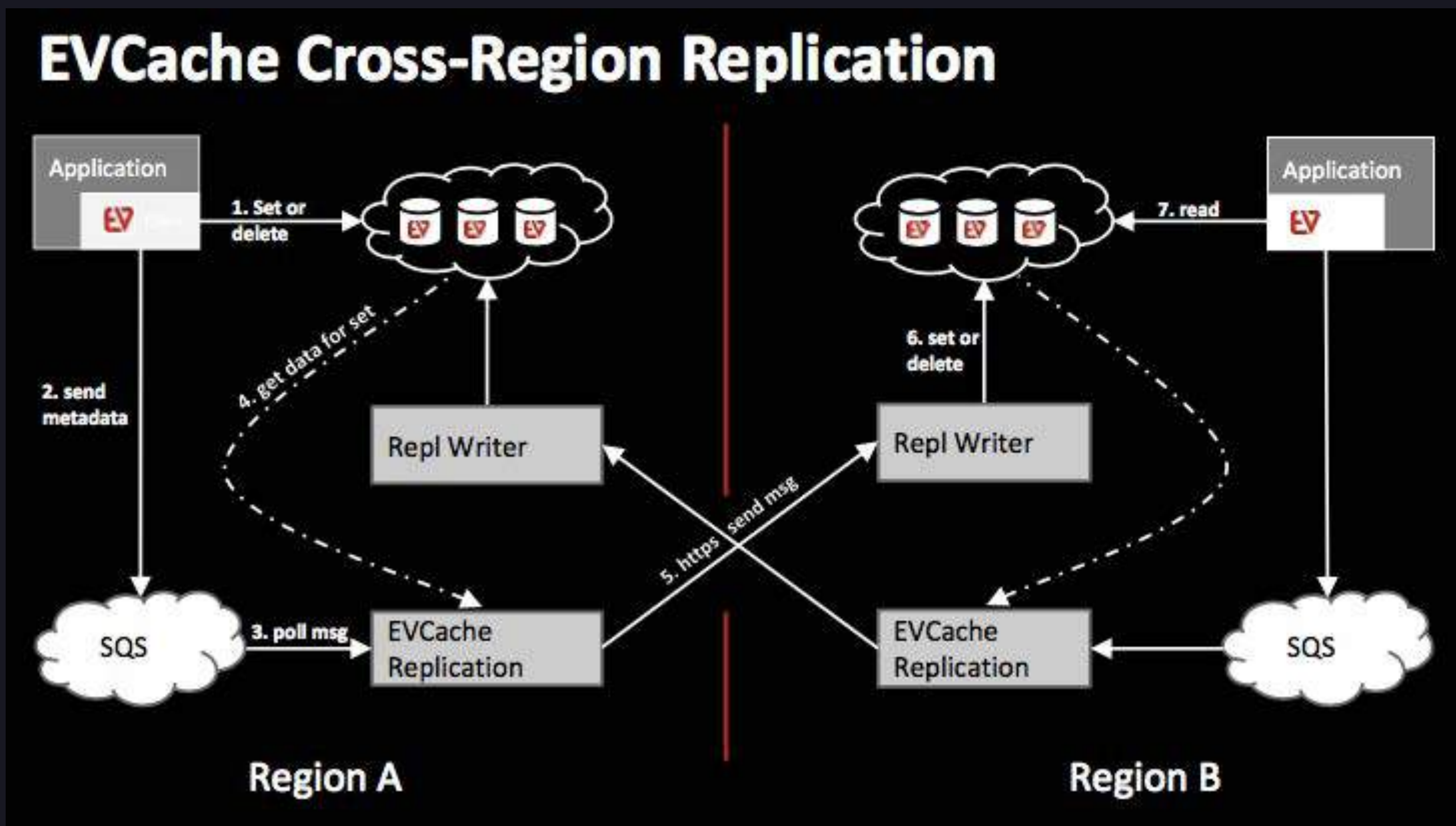
### Writes



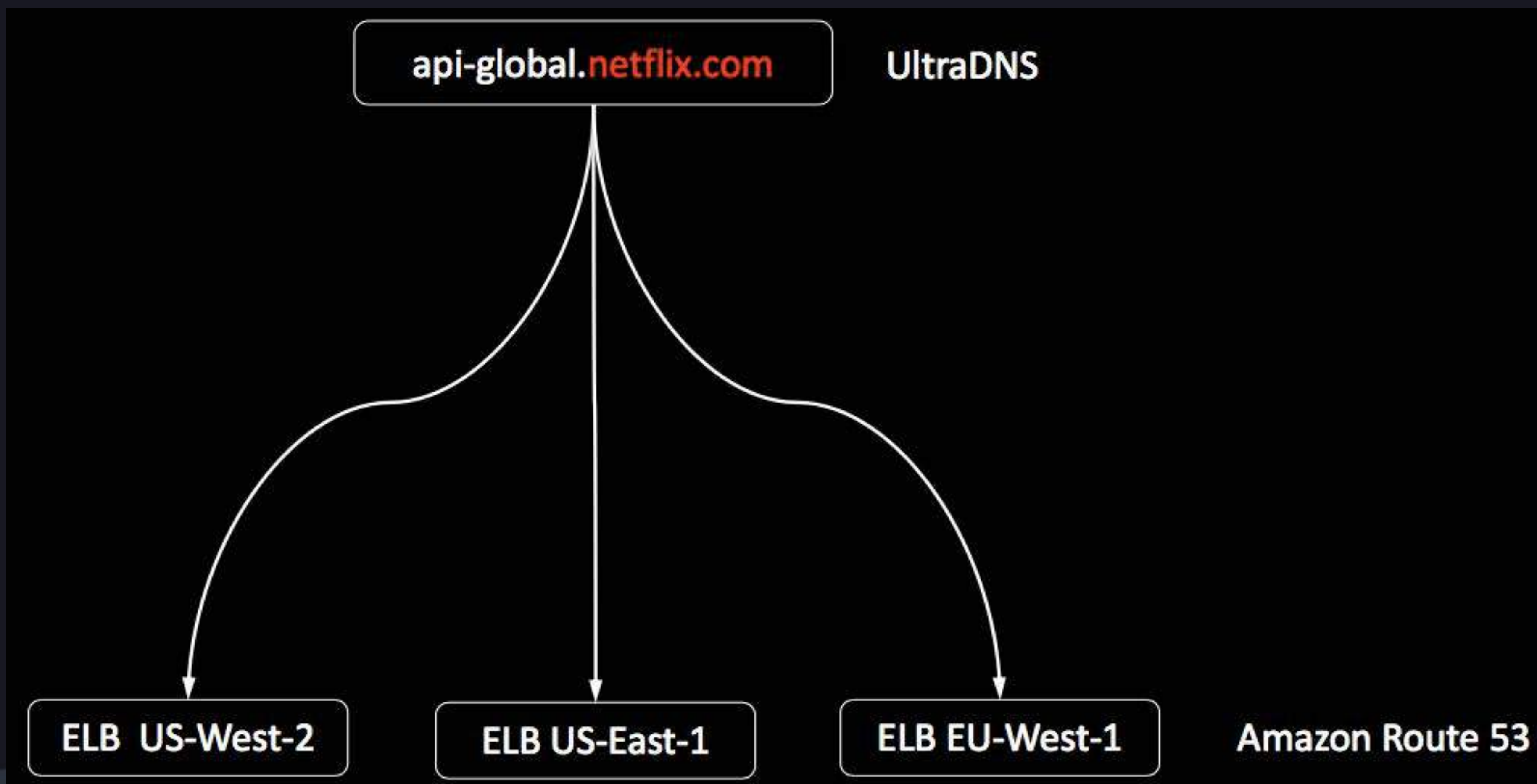


# Cache的高可用与最终一致性

## EVCache

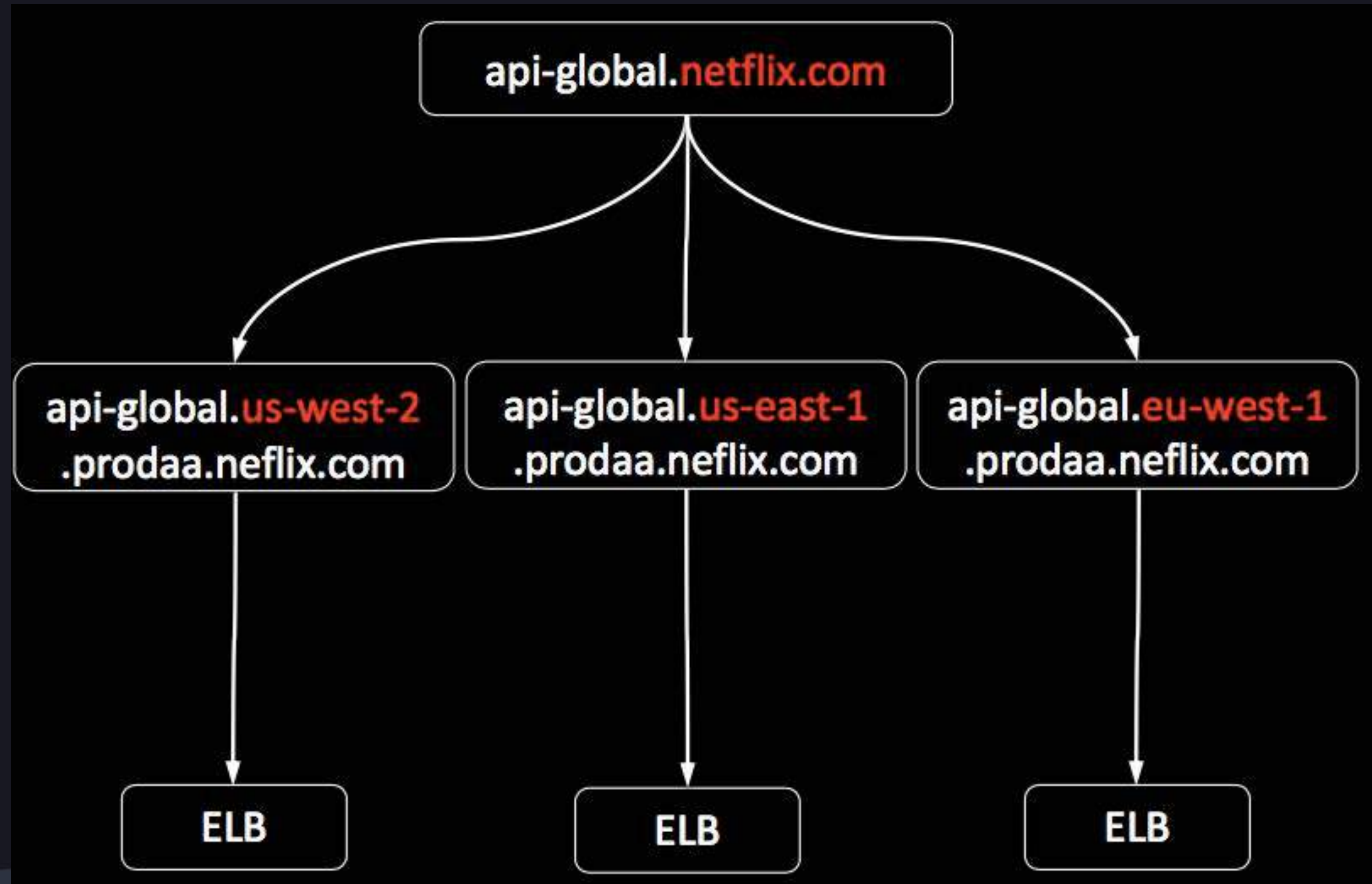


# DNS



# DNS 控制- Denominator

DENOMINATOR



# Netflix A/A 架构关键点

- Services must be stateless—all data / state replication needs to be handled in data tier.
- They must access any resource locally in-Region. This includes resources like S3, SQS, etc. This means several applications that are publishing data into an S3 bucket, now have to publish the same data into multiple regional S3 buckets.
- there should not be any cross-regional calls on user's call path. Data replication should be asynchronous.

<https://medium.com/netflix-techblog/active-active-for-multi-regional-resiliency-c47719f6685b>

# Netflix OSS

- <https://netflix.github.io/>

NETFLIX | OSS

# Time to re:Invent the Architecture ...

# 不断发展的云服务 为新用户带来的后发优势

# 容器服务



Amazon Elastic Container Service (ECS)



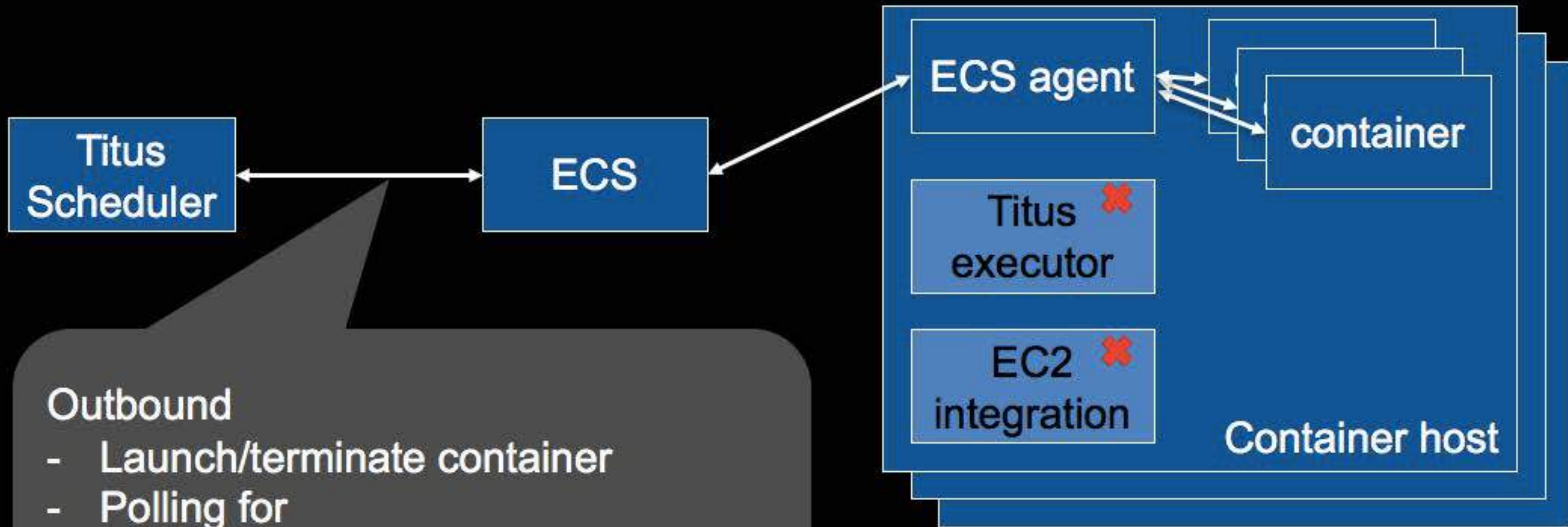
Amazon Elastic Container Service for Kubernetes (EKS)



AWS Fargate



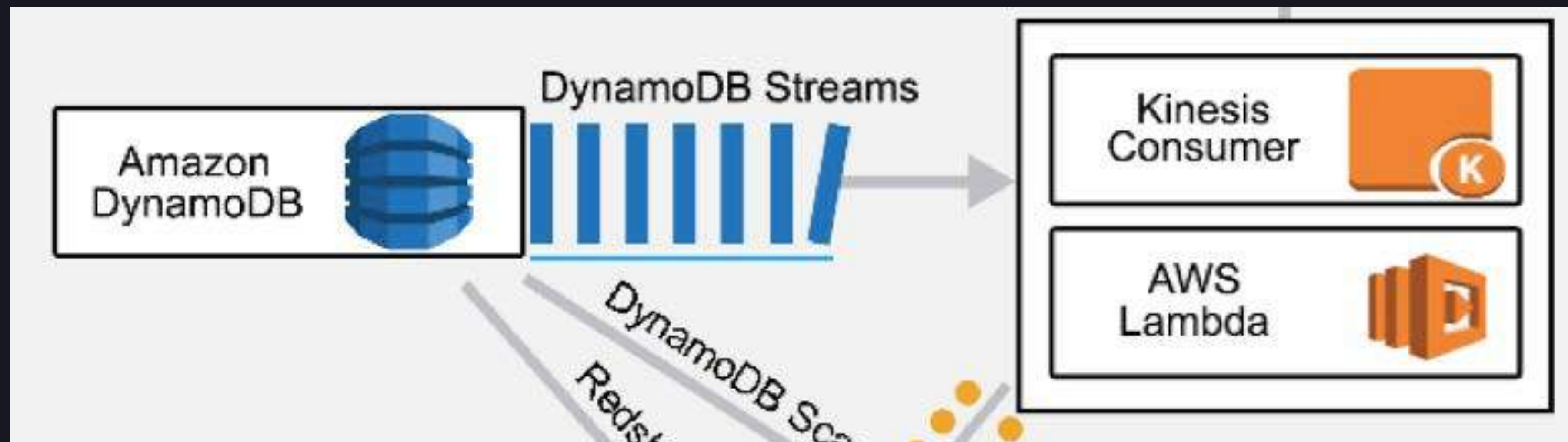
# Netflix使用Amazon ECS部署容器集群



## Outbound

- Launch/terminate container
- Polling for
  - Container host events
  - Container events

# NoSQL: DynamoDB and DynamoDB Stream



# NoSQL A/A: DynamoDB Global Table

First fully managed, multi-master, multi-region database

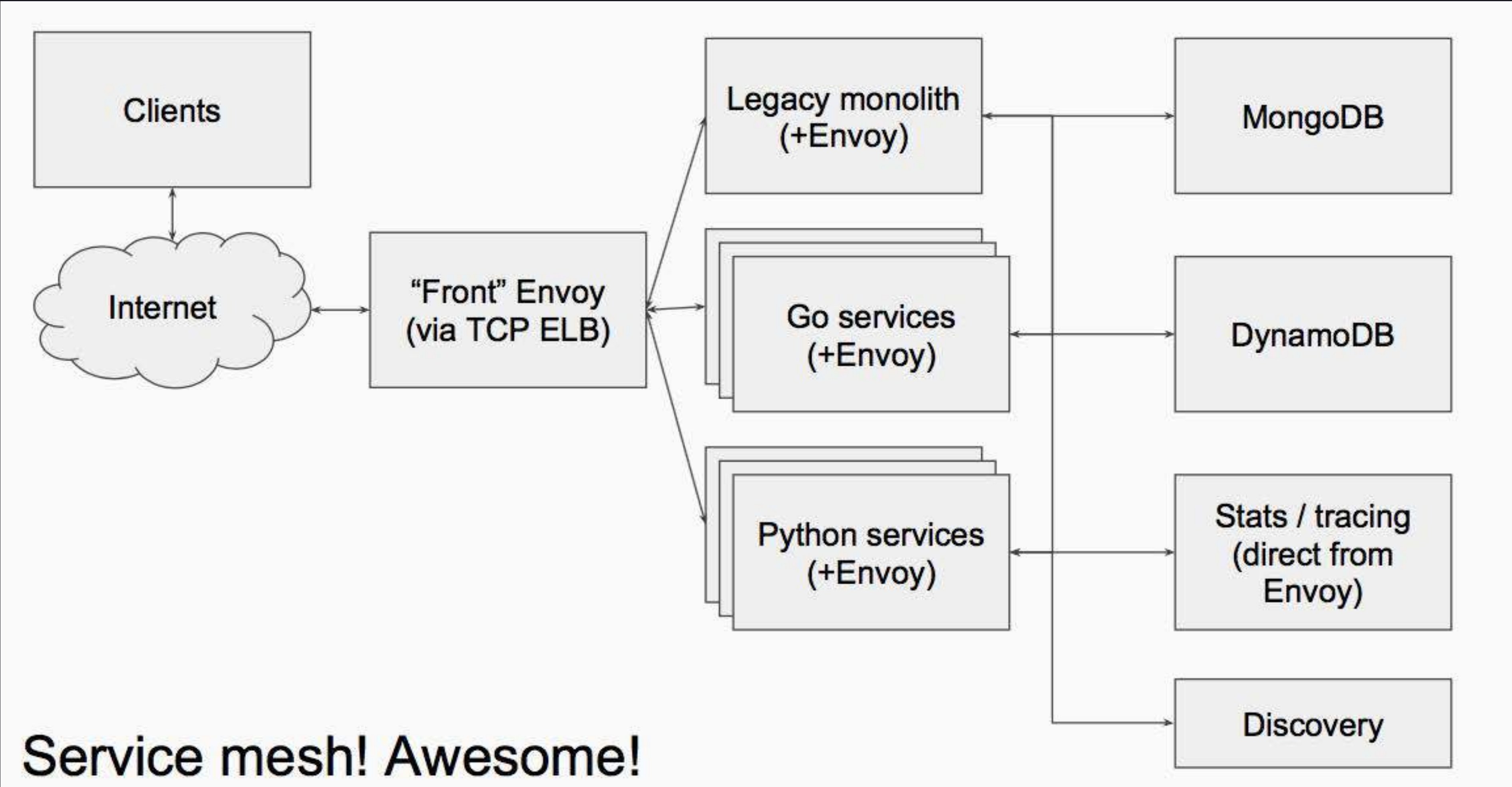


Build high performance, globally distributed applications

Low latency reads & writes to locally available tables

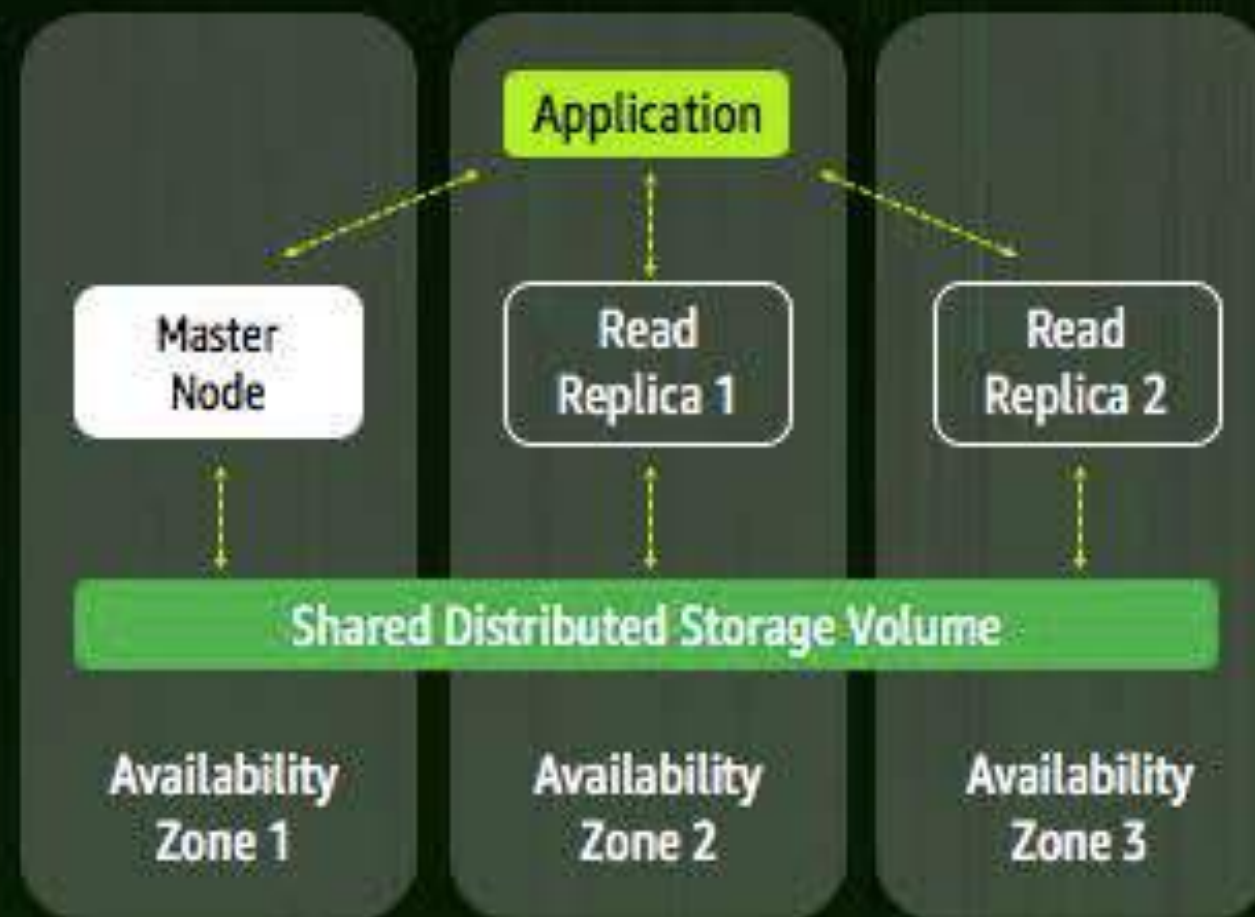
Disaster proof with multi-region redundancy

Easy to setup and no application re-writes required



# RDBMS: Aurora

AURORA TODAY: SCALE OUT FOR MILLIONS OF READS PER SECOND



Up to 15 read replicas across 3 availability zones

Auto-scale new read replicas

Seamless recovery from read replica failures

# RDBMS A/A: Aurora Multi-Master

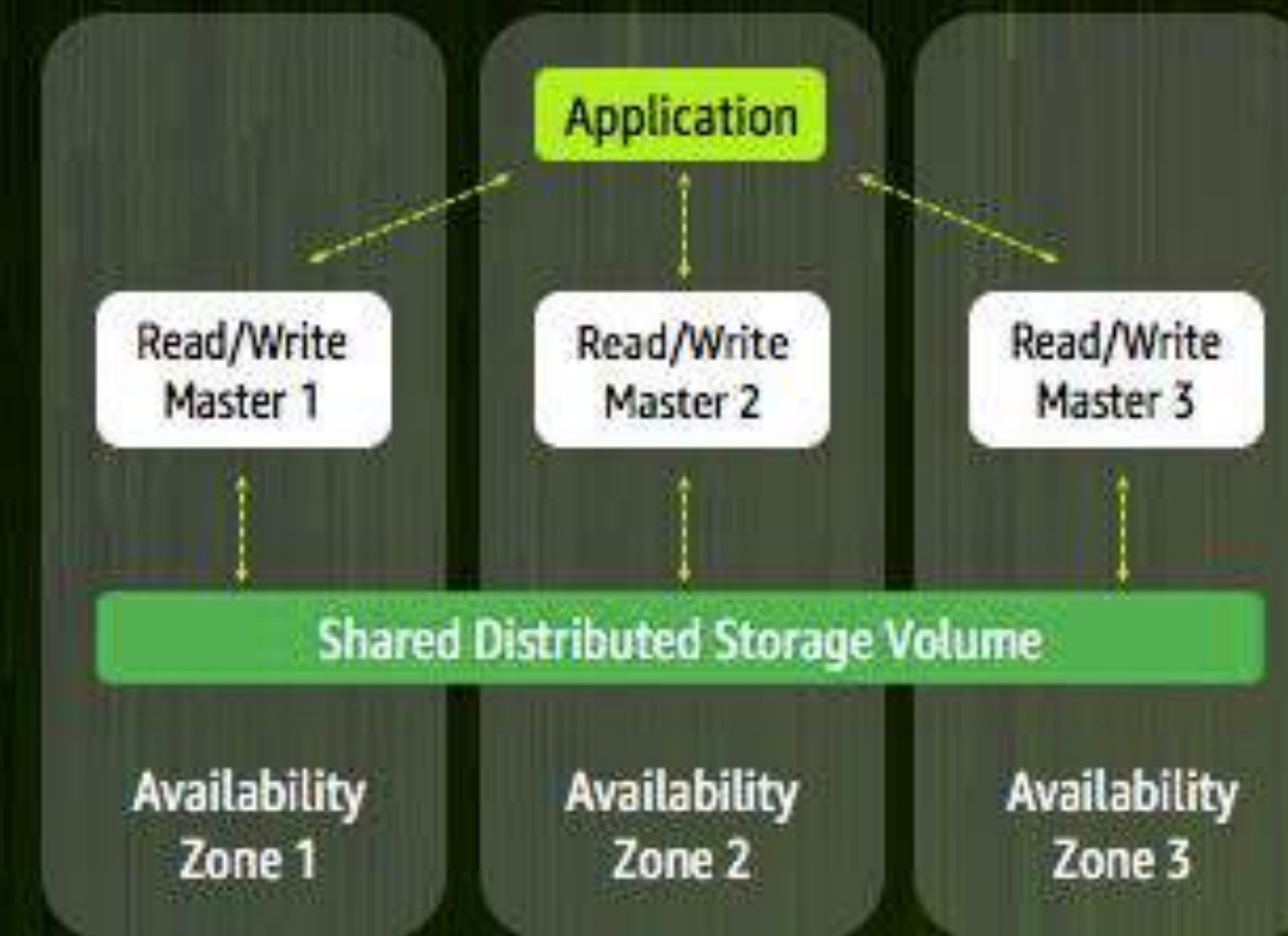
First relational database service with scale-out across multiple datacenters

Zero application downtime from ANY node failure

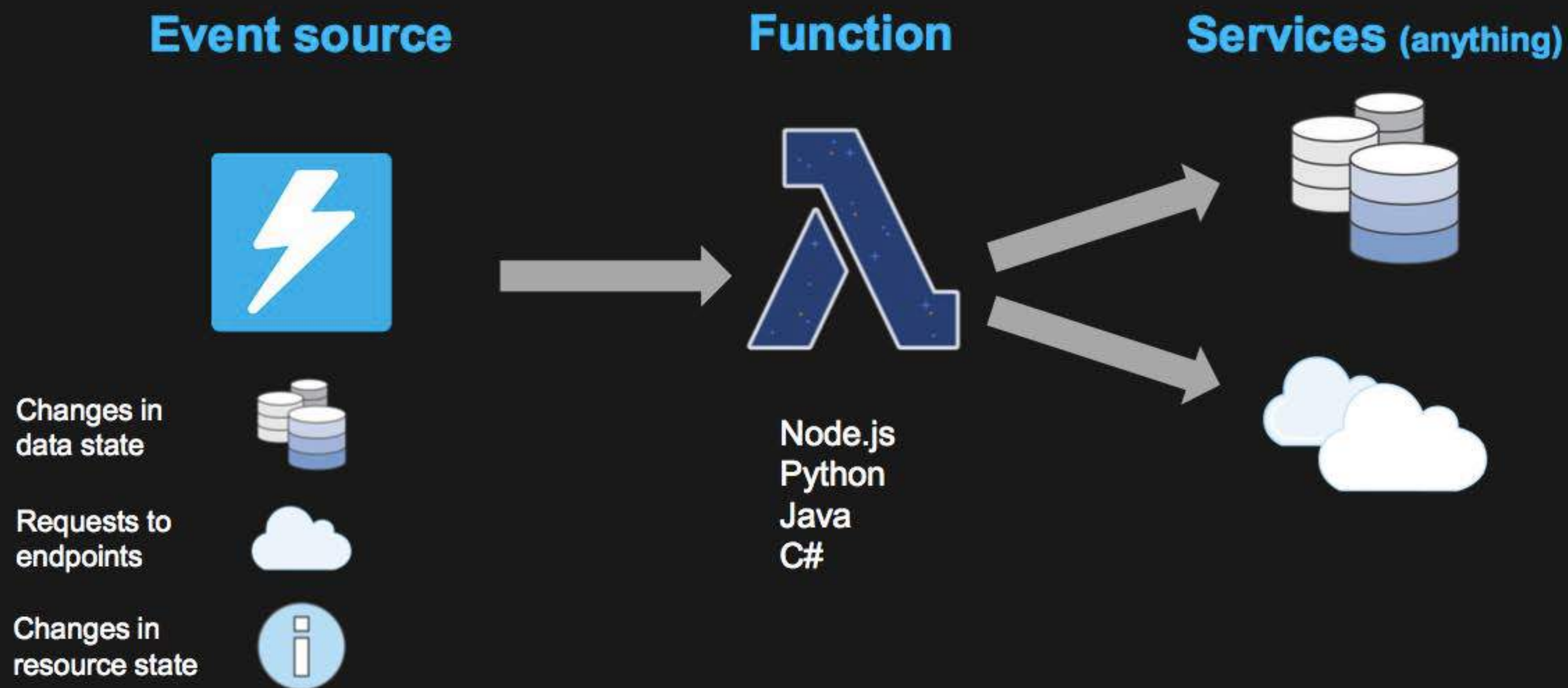
Zero application downtime from ANY AZ failure

Faster write performance

Multi-region coming in 2018



# Serverless



# 事件驱动

## Event-driven services

-  AWS Lambda
-  Amazon API Gateway
-  AWS Step Functions
-  AWS X-Ray

## Event sources

-  Amazon S3
-  Amazon DynamoDB
-  Amazon Kinesis Streams
-  Amazon Kinesis Firehose
-  Amazon SNS
-  Amazon SES
-  Amazon Cognito
-  AWS IoT
-  Amazon CloudFormation
-  Amazon CloudWatch Logs
-  Amazon CloudWatch Events
-  AWS CodeCommit
-  AWS Config
-  Amazon Lex
-  Amazon CloudFront

## Lambda inside

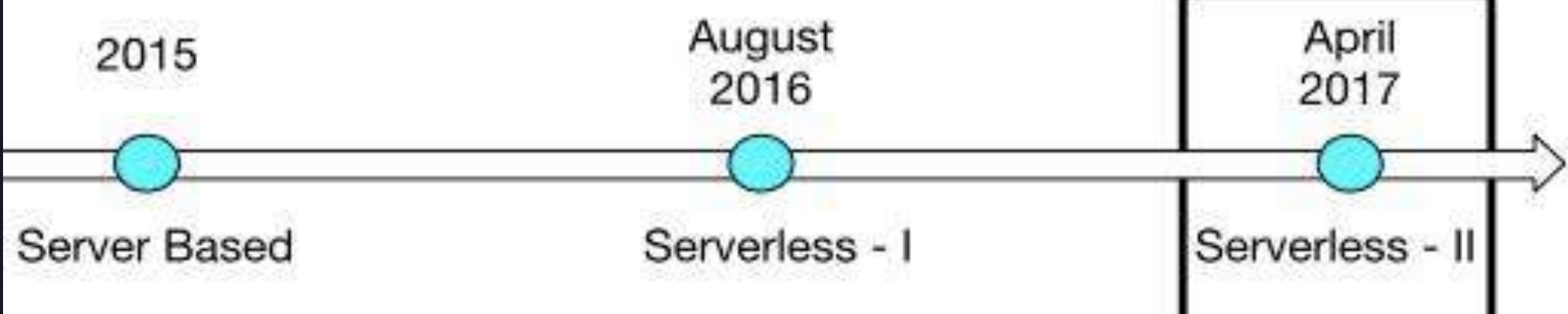
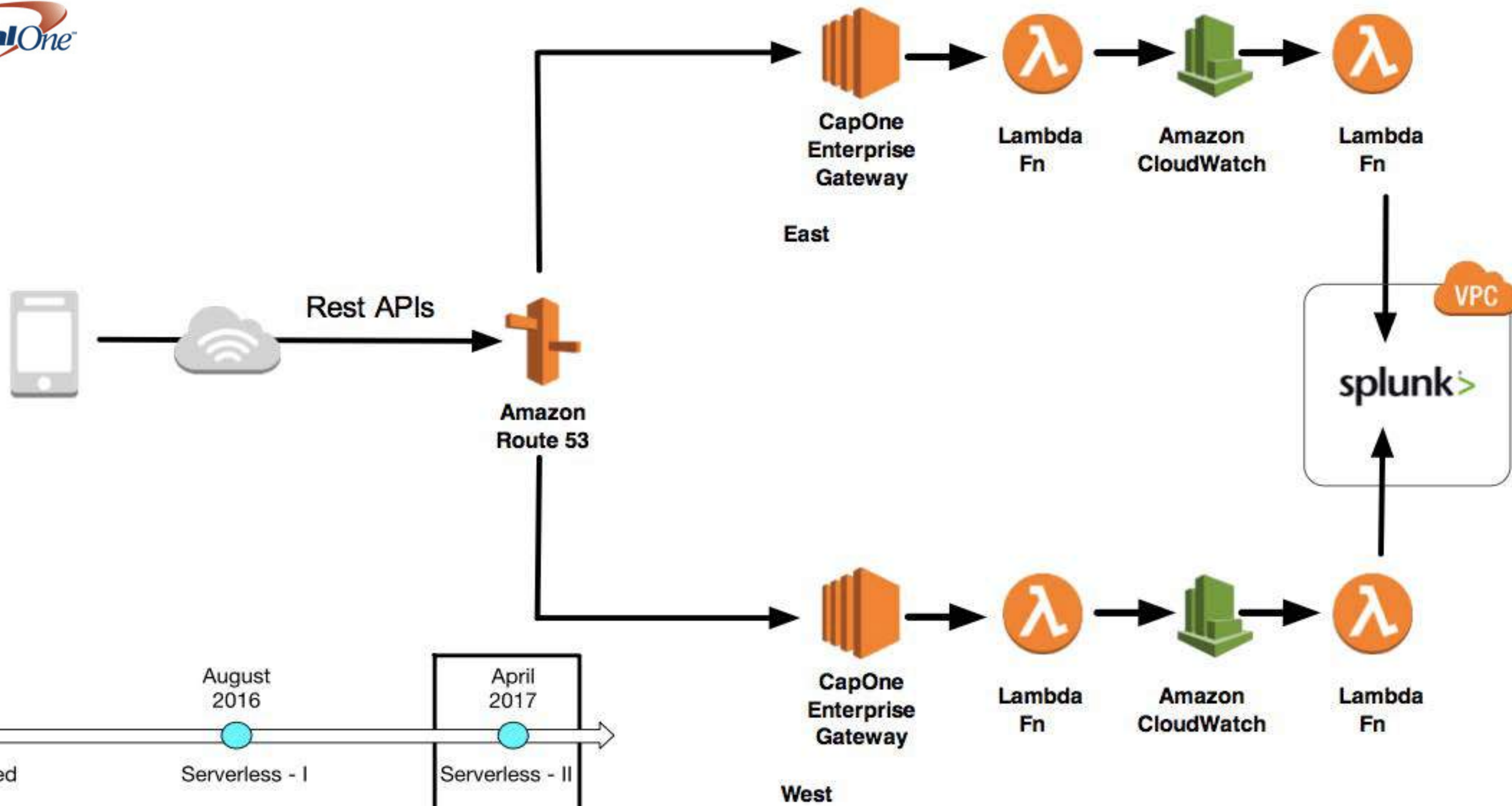
-  AWS IoT
-  AWS IoT Button
-  AWS Greengrass
-  AWS Snowball Edge
-  AWS Lambda@Edge

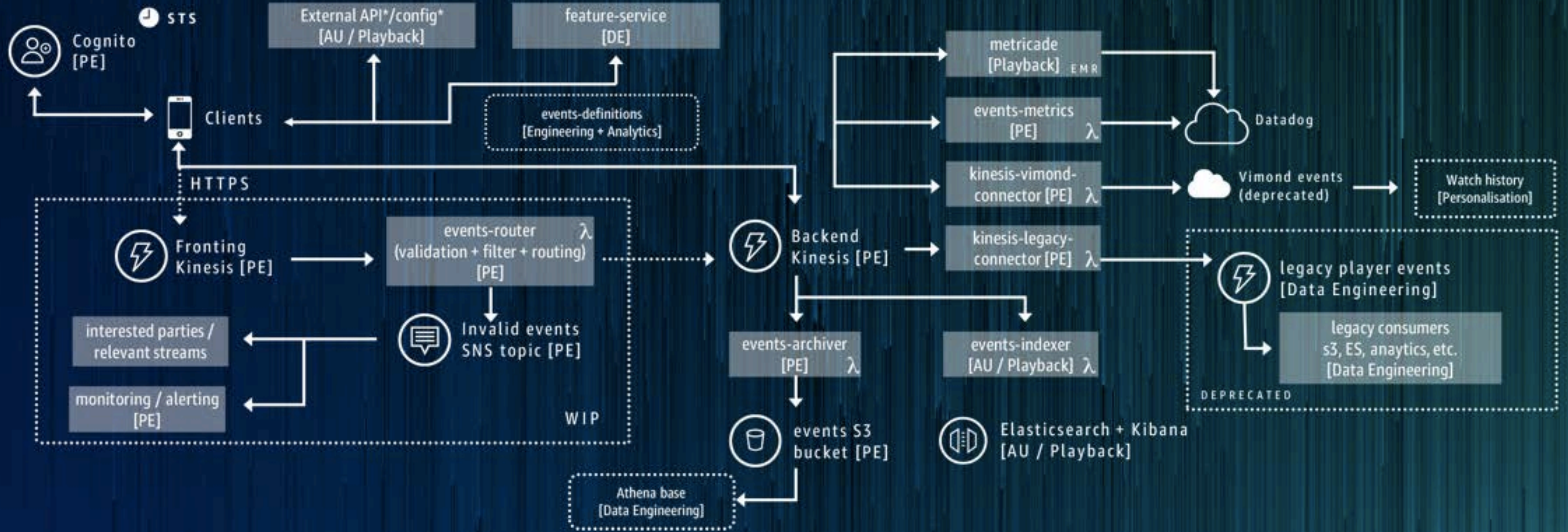


AWS Lambda









# Cloud Native开发测试环境



AWS  
CodePipeline



AWS  
CodeDeploy



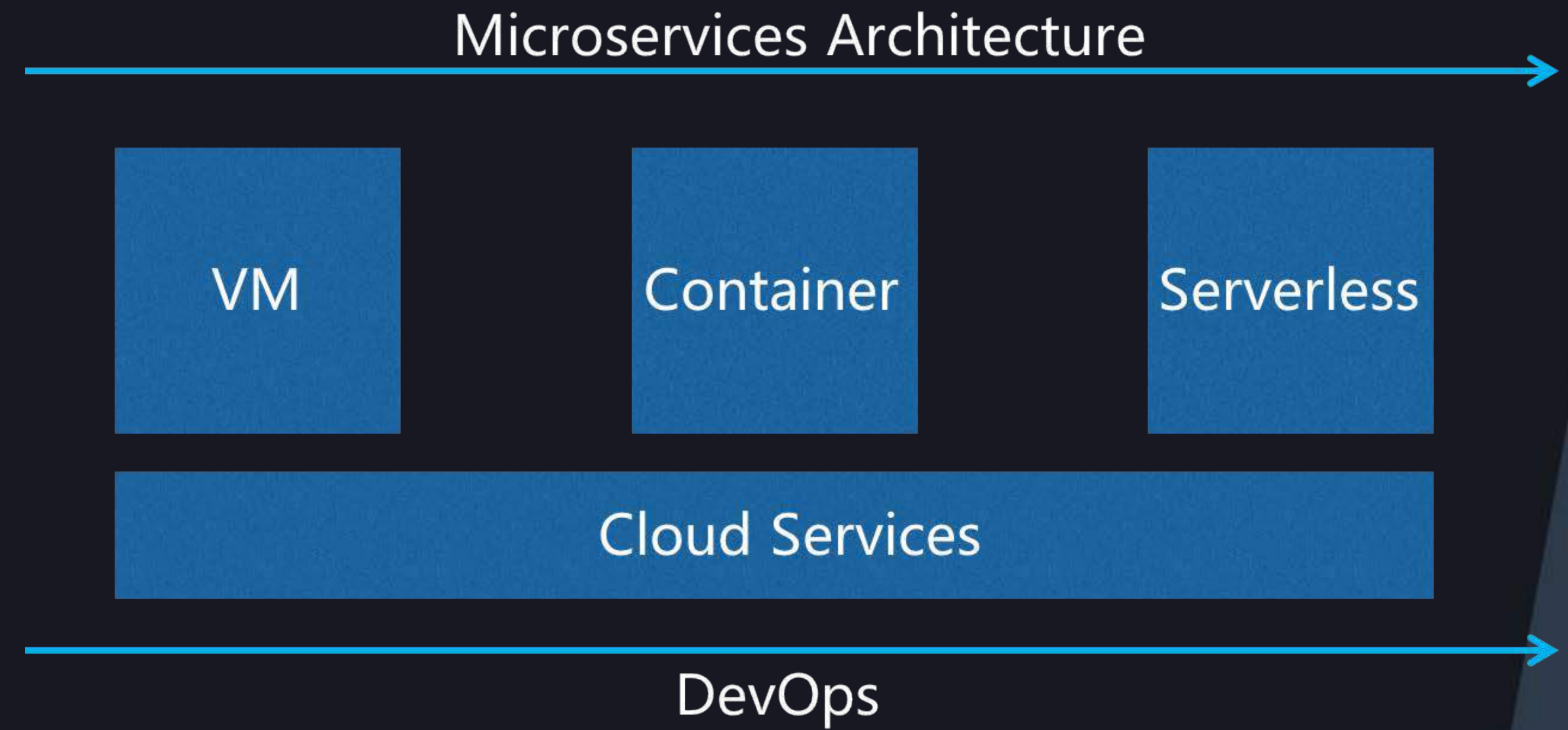
AWS  
CodeCommit



AWS  
CodeBuild

Containerized,  
Dynamically Orchestrated,  
Microservices Oriented.

On-Demand Delivery,  
Global Deployment,  
Elasticity, and Higher-Level Services.



# THANK YOU

---

如有需求，欢迎至 [ 讲师交流会议室 ] 与我们的讲师进一步交流

