



# AWS Summit

AWS技术峰会 2015 · 上海



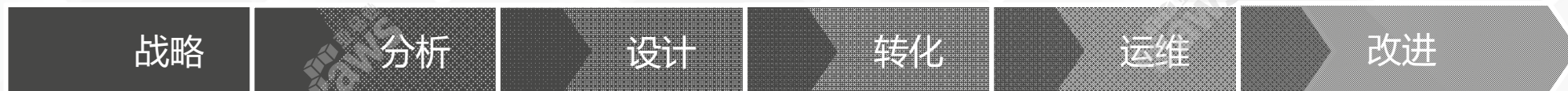


# AWS上的微服务架构

张芸 (Vivian Zhang)  
AWS专业服务咨询顾问



# AWS专业服务内容



- ❖ 上云评估
- ❖ IT规划和路线图
- ❖ 成本收益分析
- ❖ 风险评估和合规审查
- ❖ 组织结构变更评估



- ❖ 云上系统架构设计
- ❖ 业务应用迁移
- ❖ 业务应用优化
- ❖ 大数据分析
- ❖ 安全管理



DevOps

# 议程

- 什么是微服务
- 微服务带来的挑战和改变
- AWS上的微服务
- Gilt案例分享



# 什么是微服务

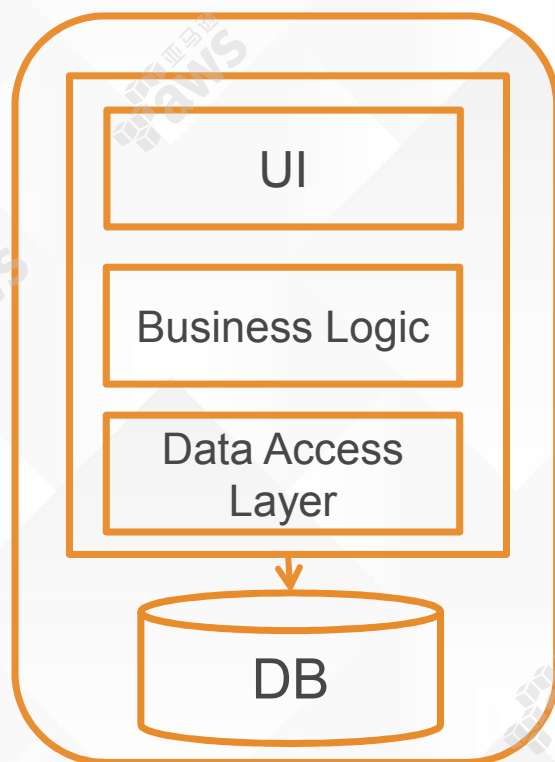


# 什么是微服务？

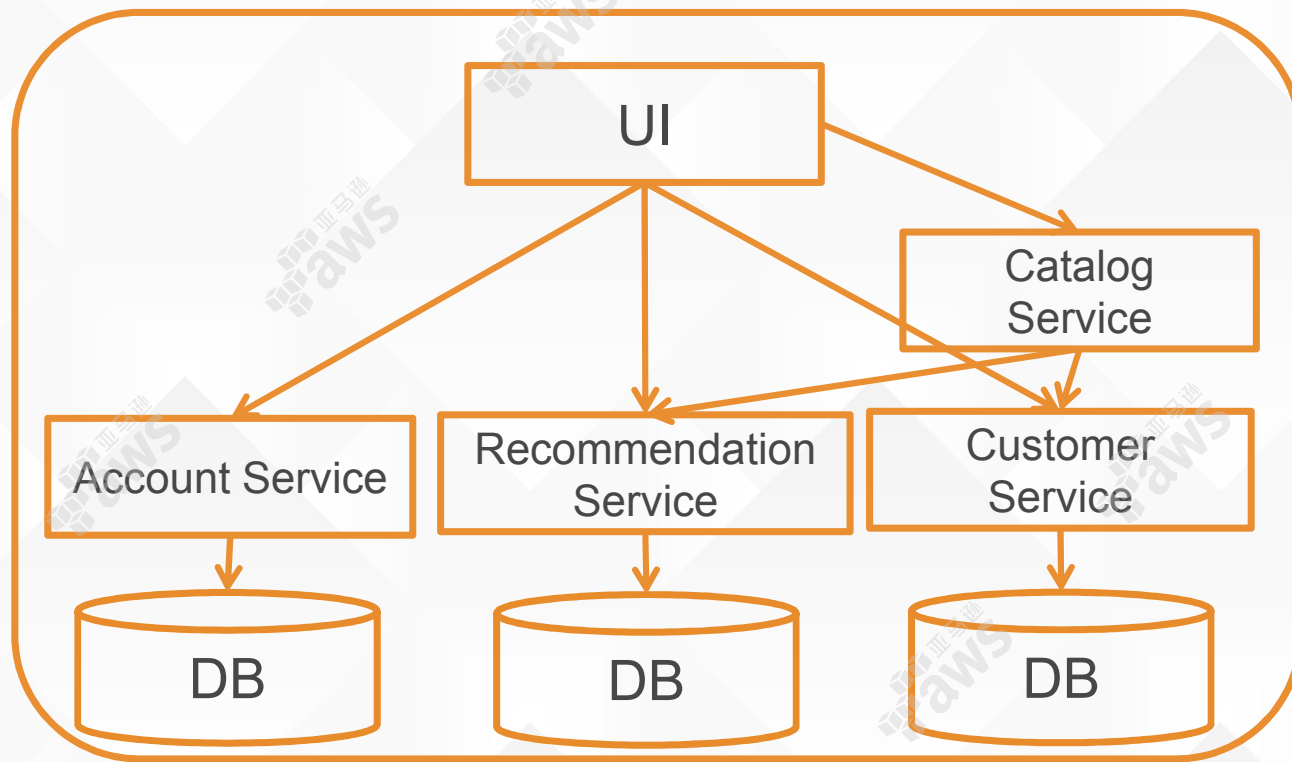
- 微服务是一种软件体系结构类型，复杂的应用程序由许多微小并且相互独立的服务组成：
  - 这些服务相互之间通过与语言无关的API通信；
  - 这些服务是微小的，高度松耦合，并且只关注在一个小的任务；
  - 便于系统构建的模块化方法；
  - 服务是自治并且完整的，控制所有组件，包括UI、中间件、存取和事务。

# 单体应用对比微服务架构

## 单体应用



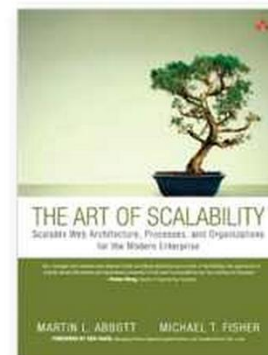
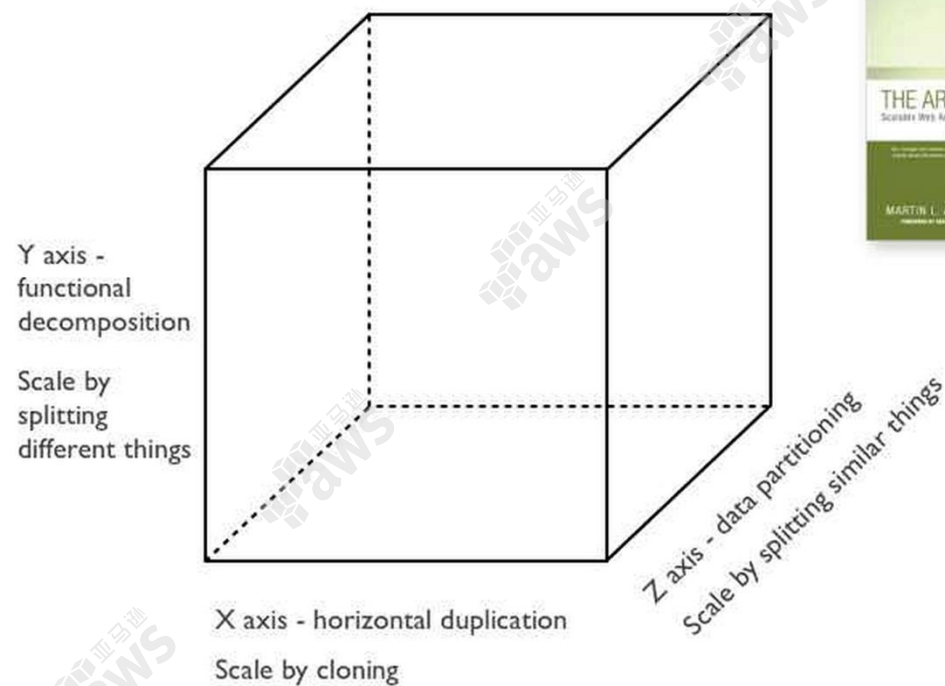
## 微服务架构





# 扩展立方

## 3 dimensions to scaling





# 微服务架构的特性

## —— Martin Fowler

通过服务的组件化

围绕业务能力

产品，而不是项目

聪明的端点和哑管道

分散的治理和数据管理

基础设施自动化 / 考虑到失败的设计

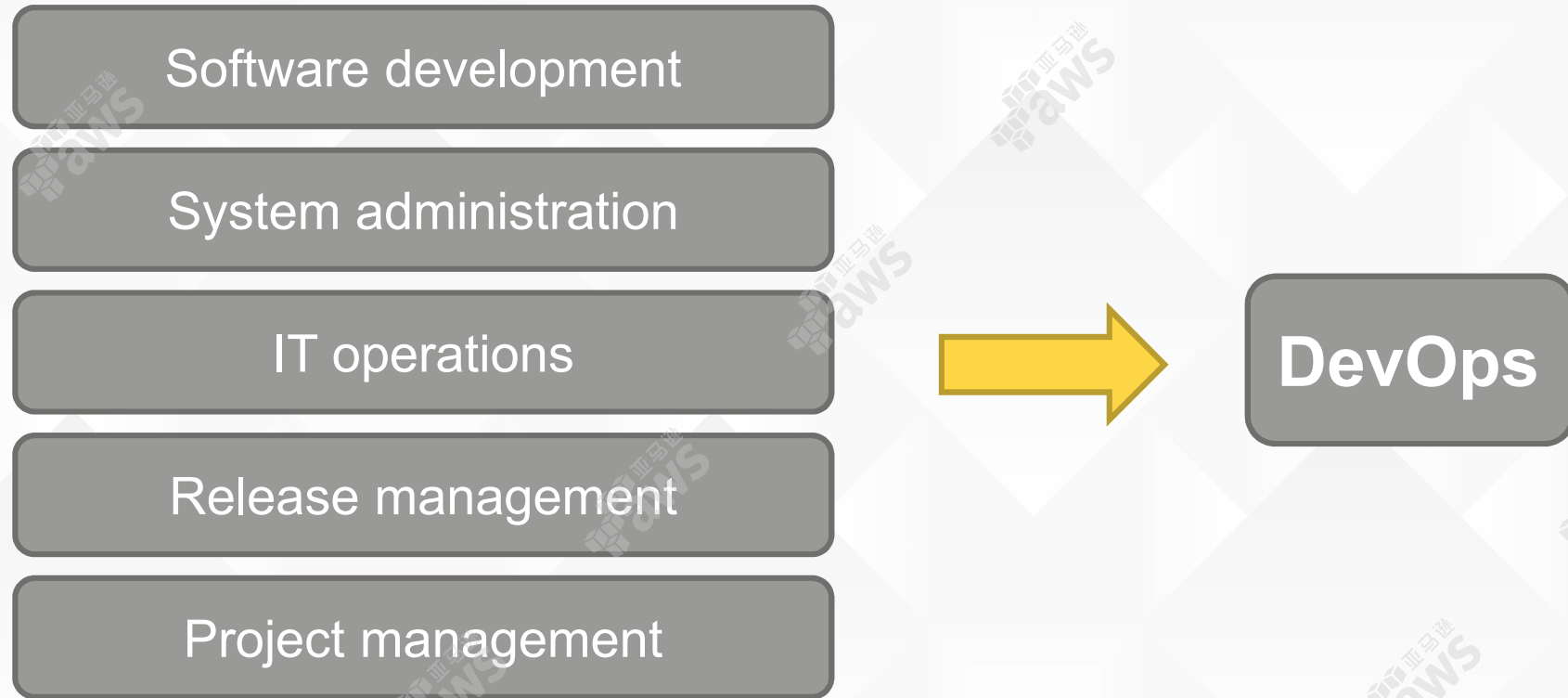
进化的设计



# 微服务带来的挑战和改变



# 文化的改变



# 组织——单体应用

- 按照技术能力组织



UI Team

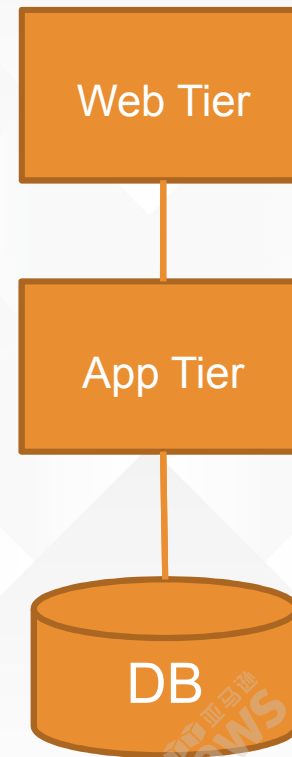


App Logic Team



DBA Team

Organizational Structure



Application Architecture

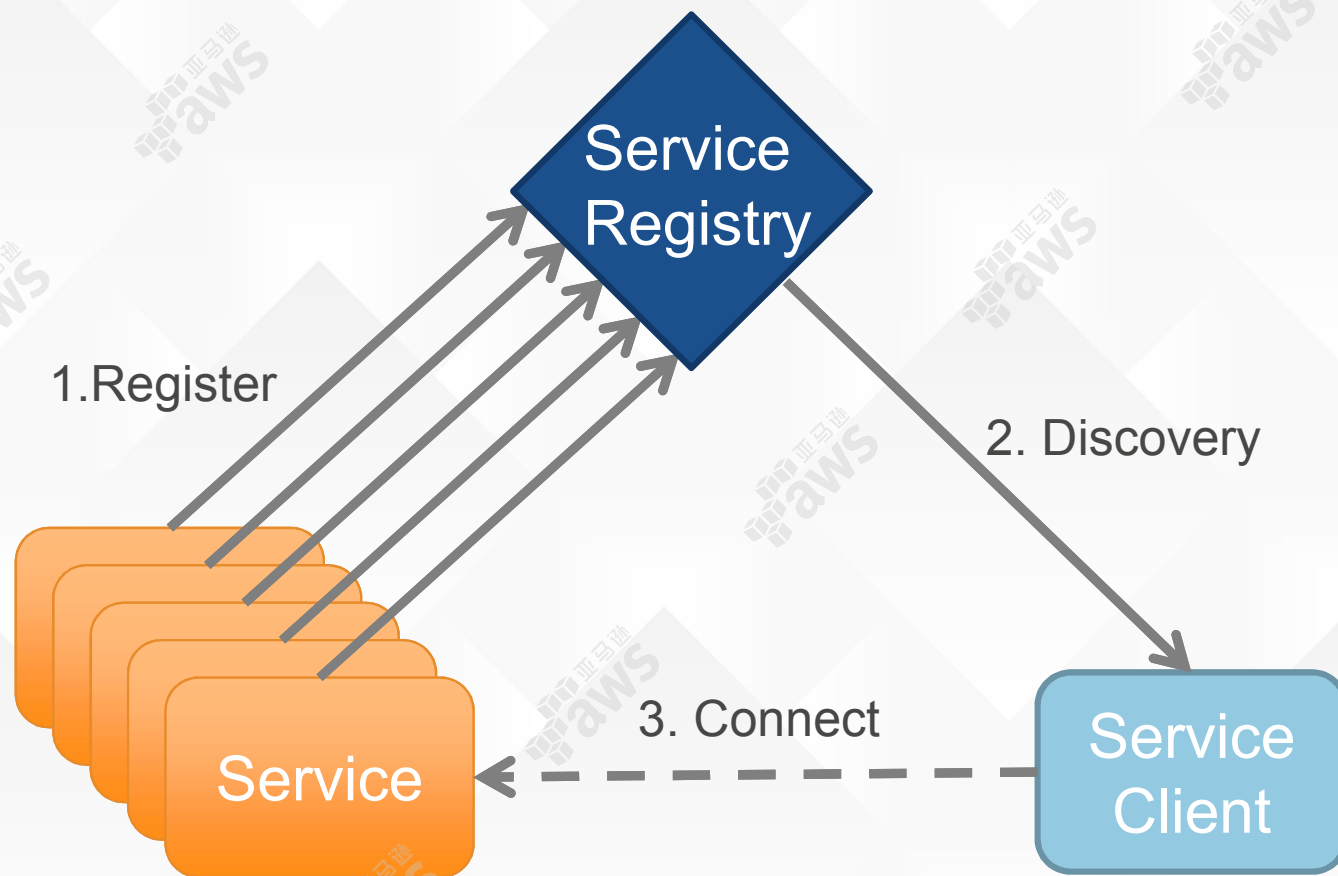
# 组织——微服务

- 按照业务职责组织

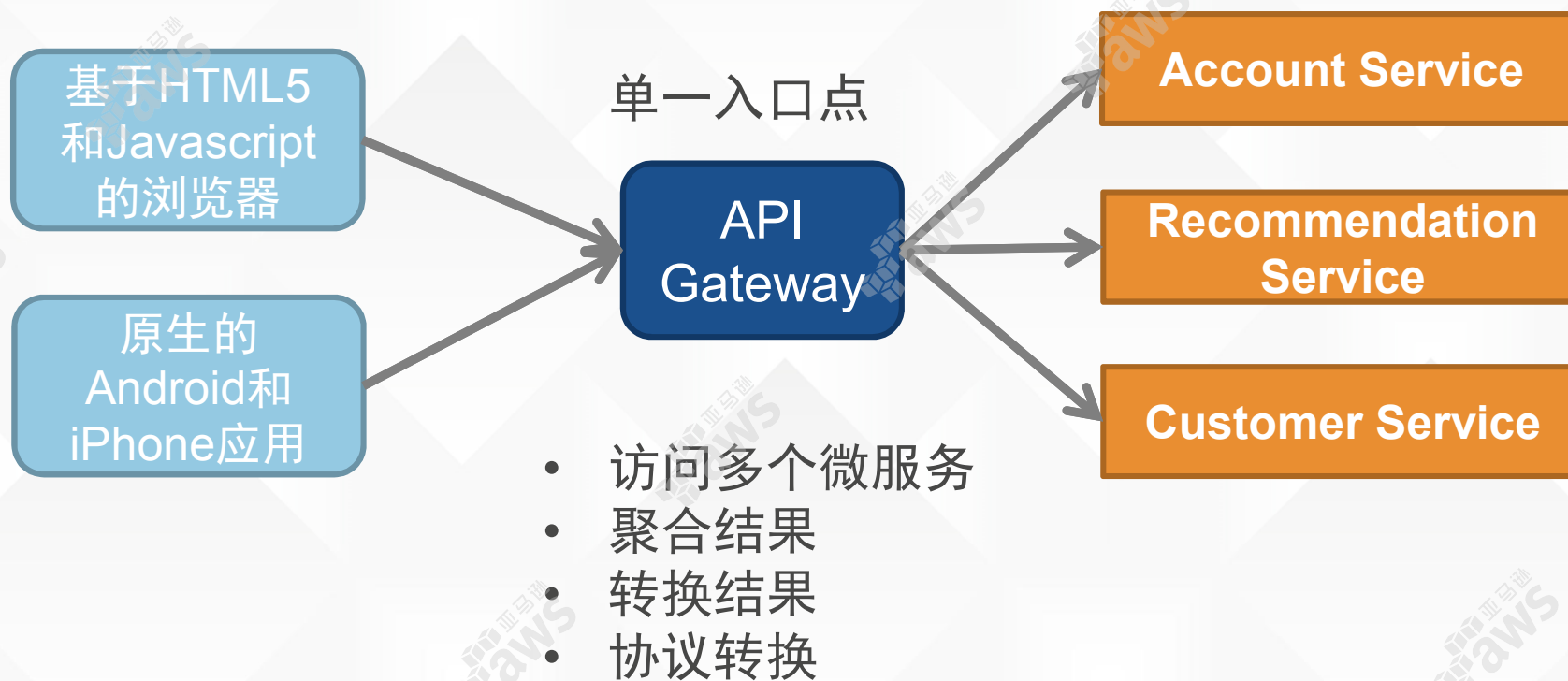


微服务所有权：需求、技术选择、开发、质量、部署、支持

# 服务发现 / 服务注册



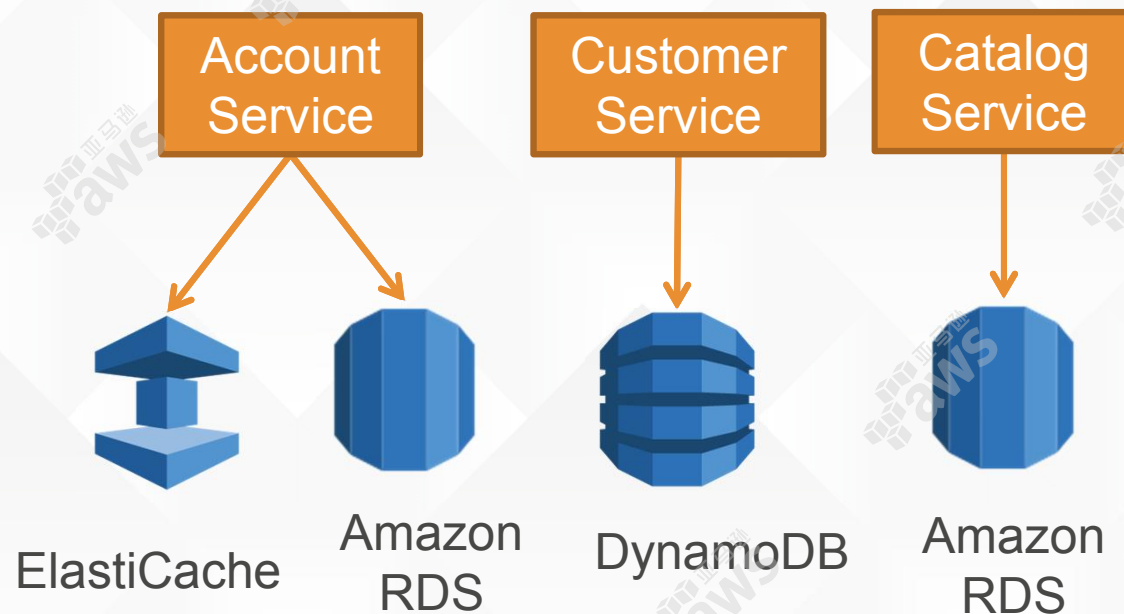
# API Gateway





# 分散的数据存储

- 每个服务选择自己的数据存储技术
- 降低Schema改变的影响
- 独立的可伸缩性
- 数据封闭在服务API之内



# 每个Container/实例中一个服务

- 单独监控
- 单独扩展
- 清晰的所有权
- 不可变的部署

Account  
Service

container or instance

Customer  
Service

container or instance

Catalog  
Service

container or instance

# 监控和日志

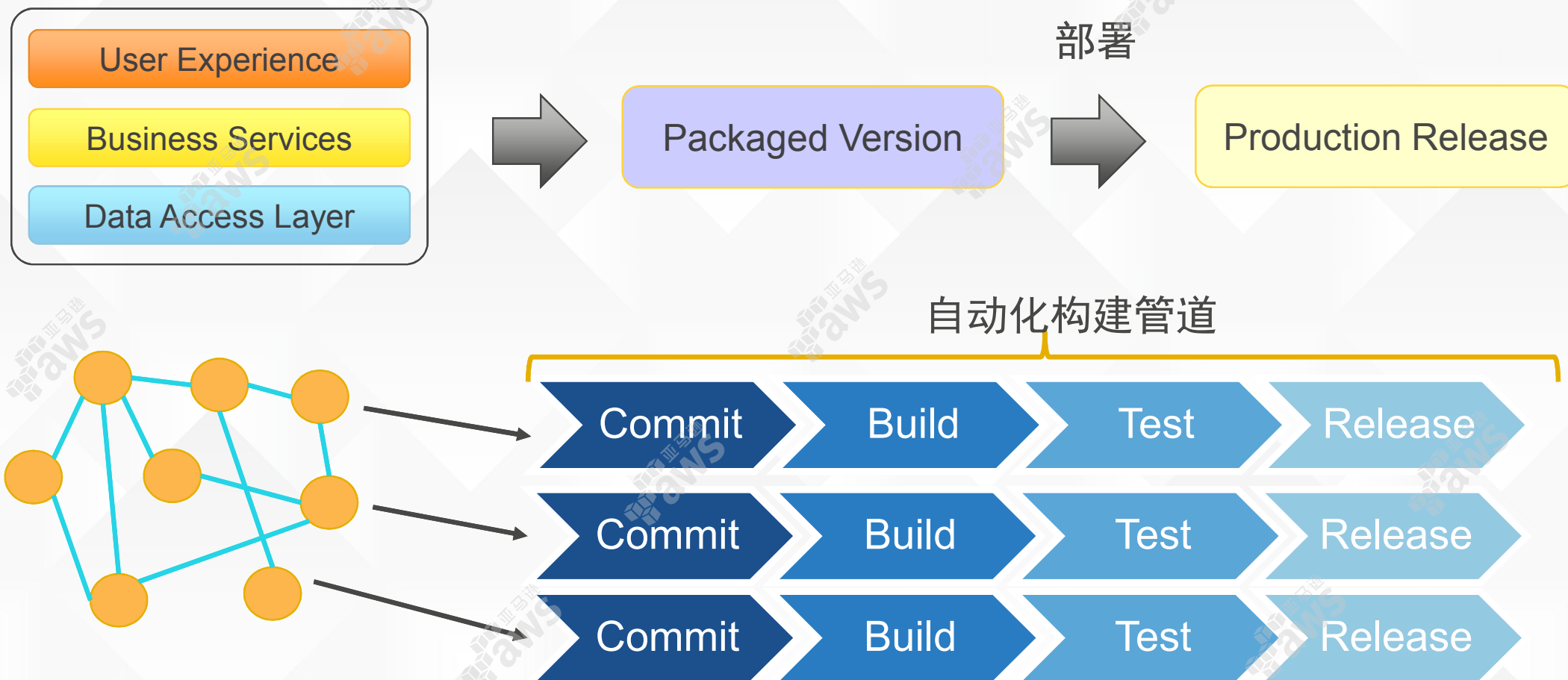
- 监控

- 外部指标：延迟、错误率、响应时间
- 内部指标：基本系统指标、操作系统、应用程序

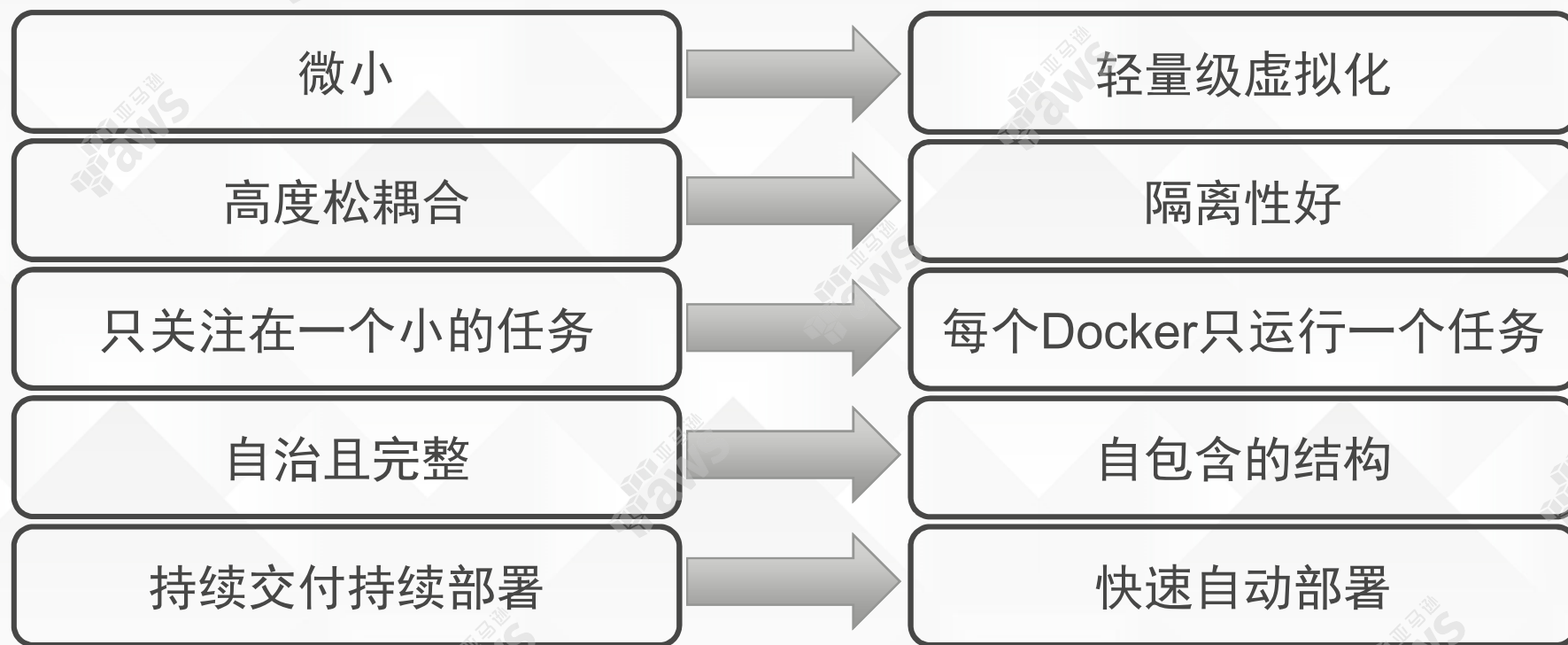
- 日志

- 收集、汇聚、实时分析

# 部署：持续部署和持续交付



# 微服务和Docker





# AWS上的微服务



# AWS对微服务的支持

监控和日志



CloudWatch



Cloud Trail



Kinesis

自动化部署



Elastic  
Beanstalk



Code  
Deploy



Code  
Pipeline



Code  
Commit

应用程序服务



API Gateway



SQS



SWF



SES



SNS

数据库



RDS



DynamoDB



ElastiCache

路由和服务发现、负载均衡



ELB

计算资源



EC2



AMI



ECS



Lambda

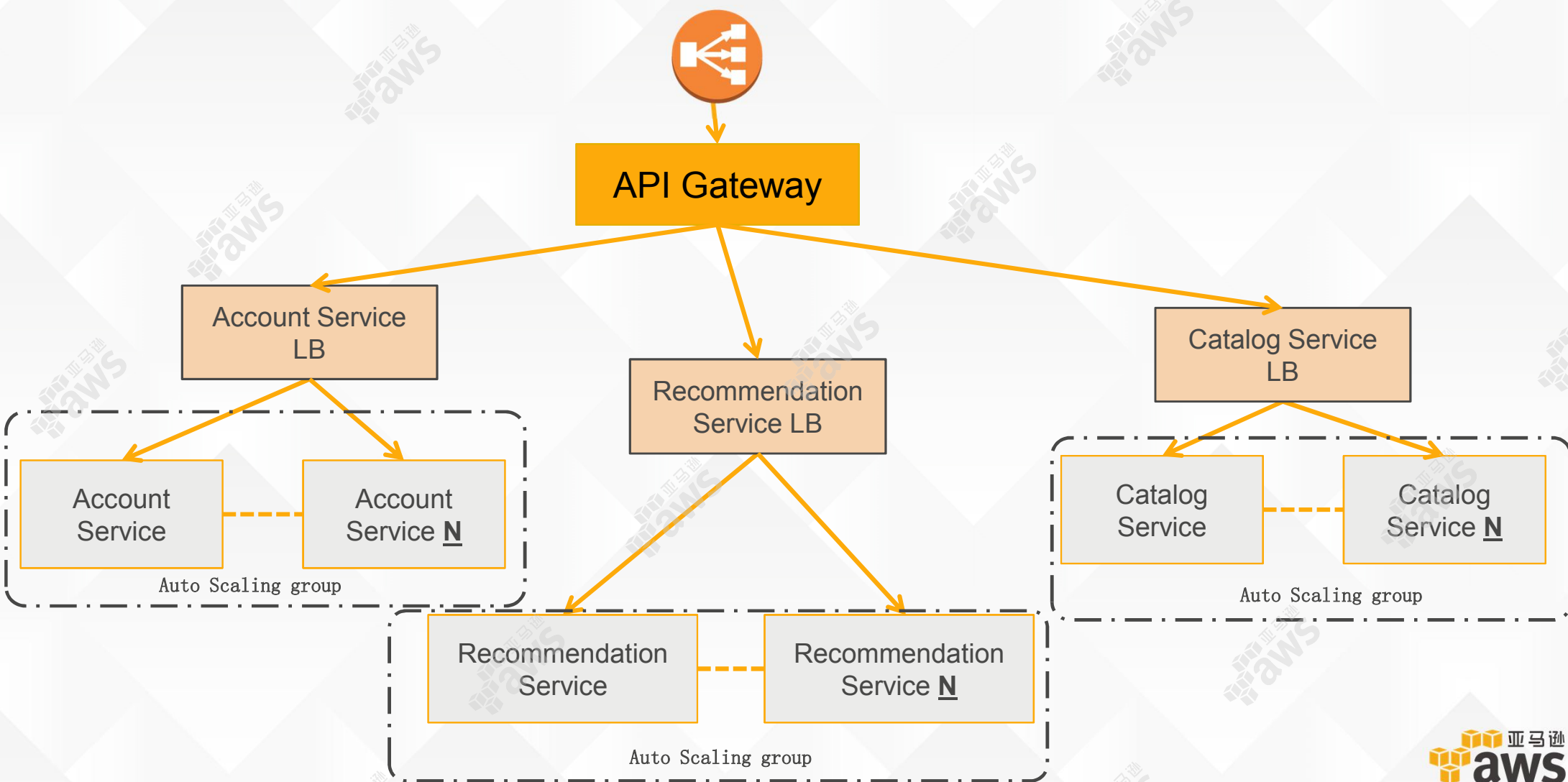


Auto  
Scaling

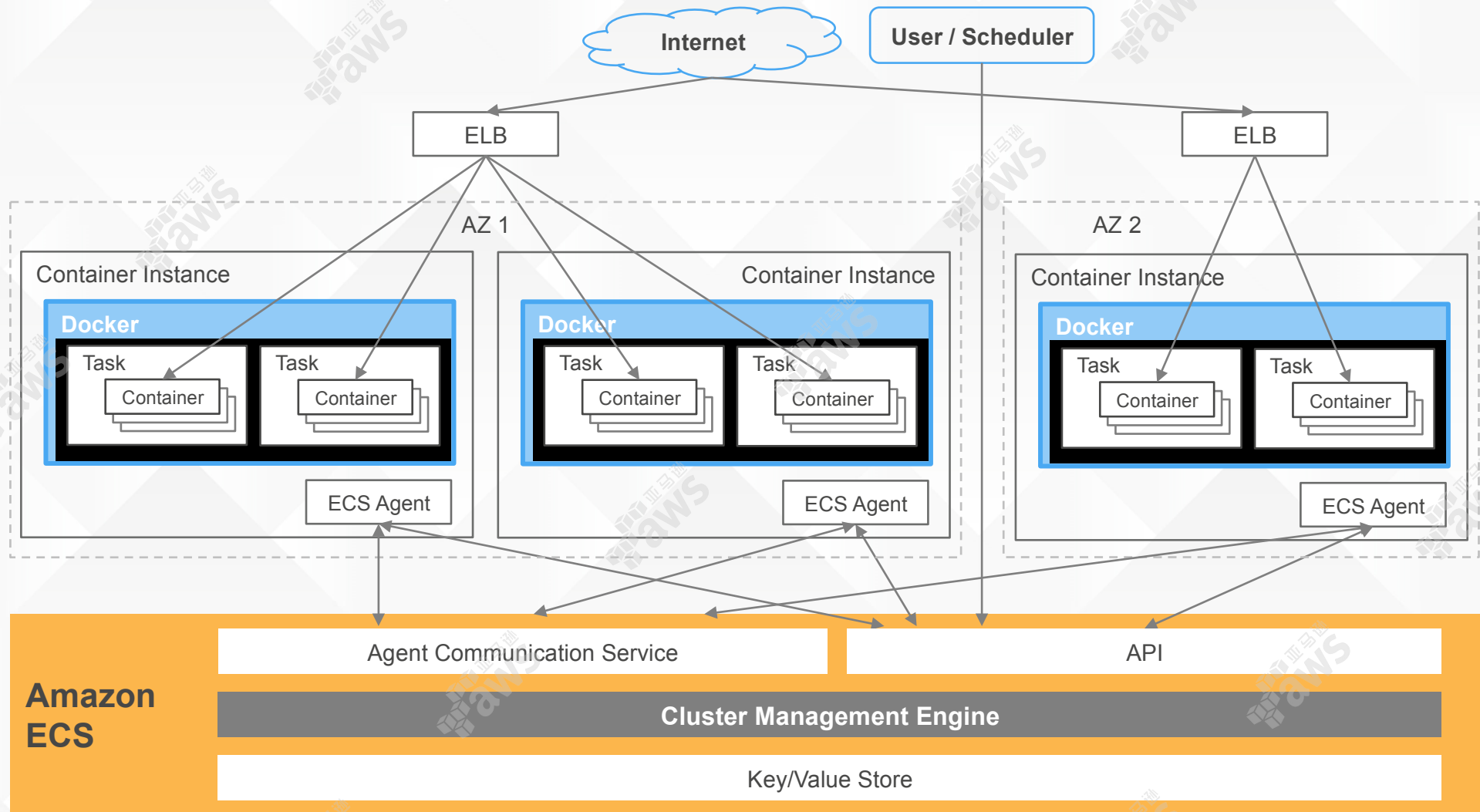




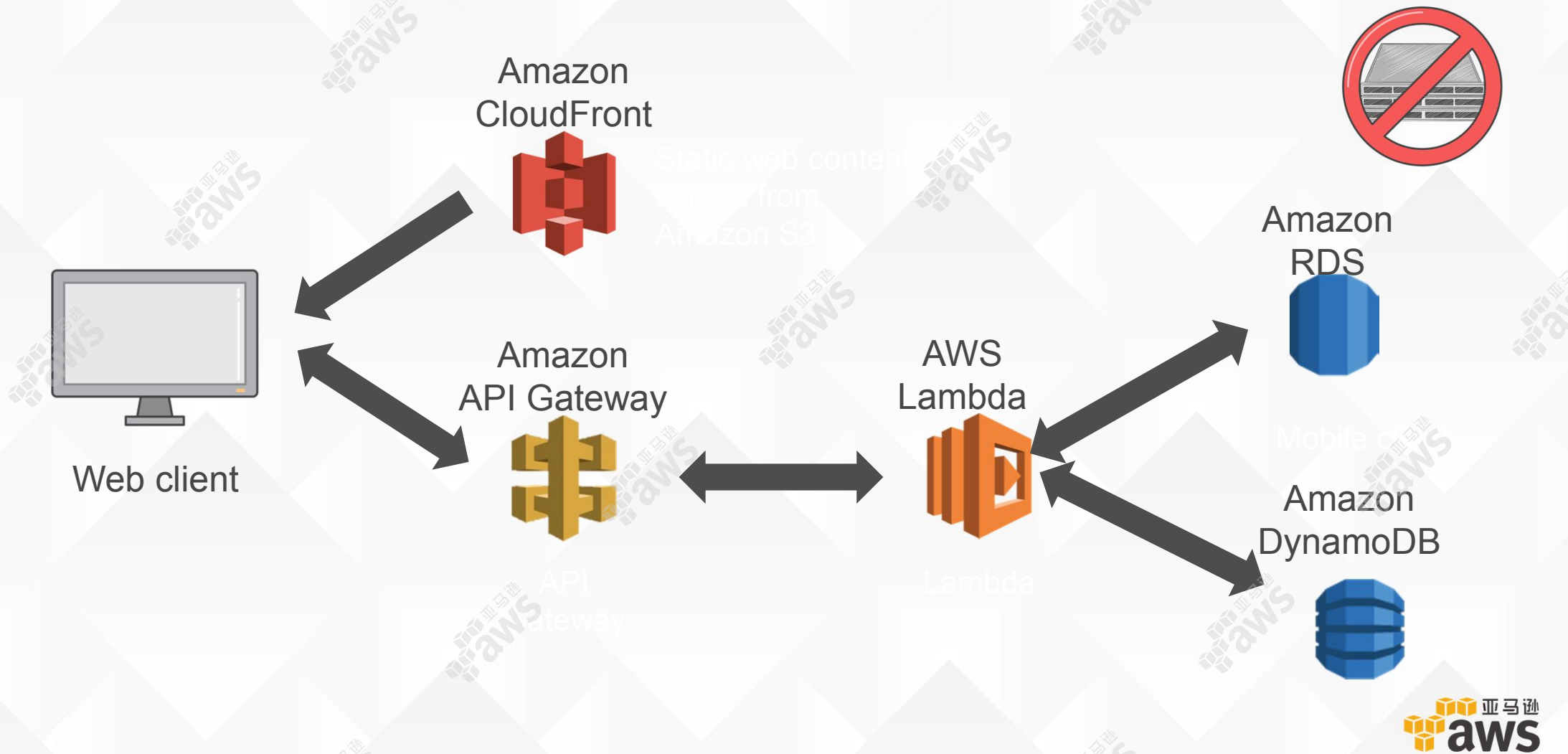
# AWS上的微服务架构一: EC2+ELB+ASG



# AWS上的微服务架构二: EC2 + ECS + ELB

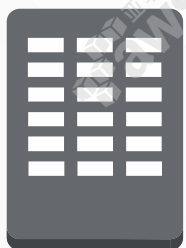


# AWS上的微服务架构三: API Gateway + Lambda



# 不断加速的计算平台

On-Premises



几周内部署  
使用数年

Amazon EC2



几分钟之内部署  
使用数周

Amazon ECS



几秒钟之内部署  
使用数分钟 / 小时

AWS Lambda



几微秒之内部署  
使用数秒

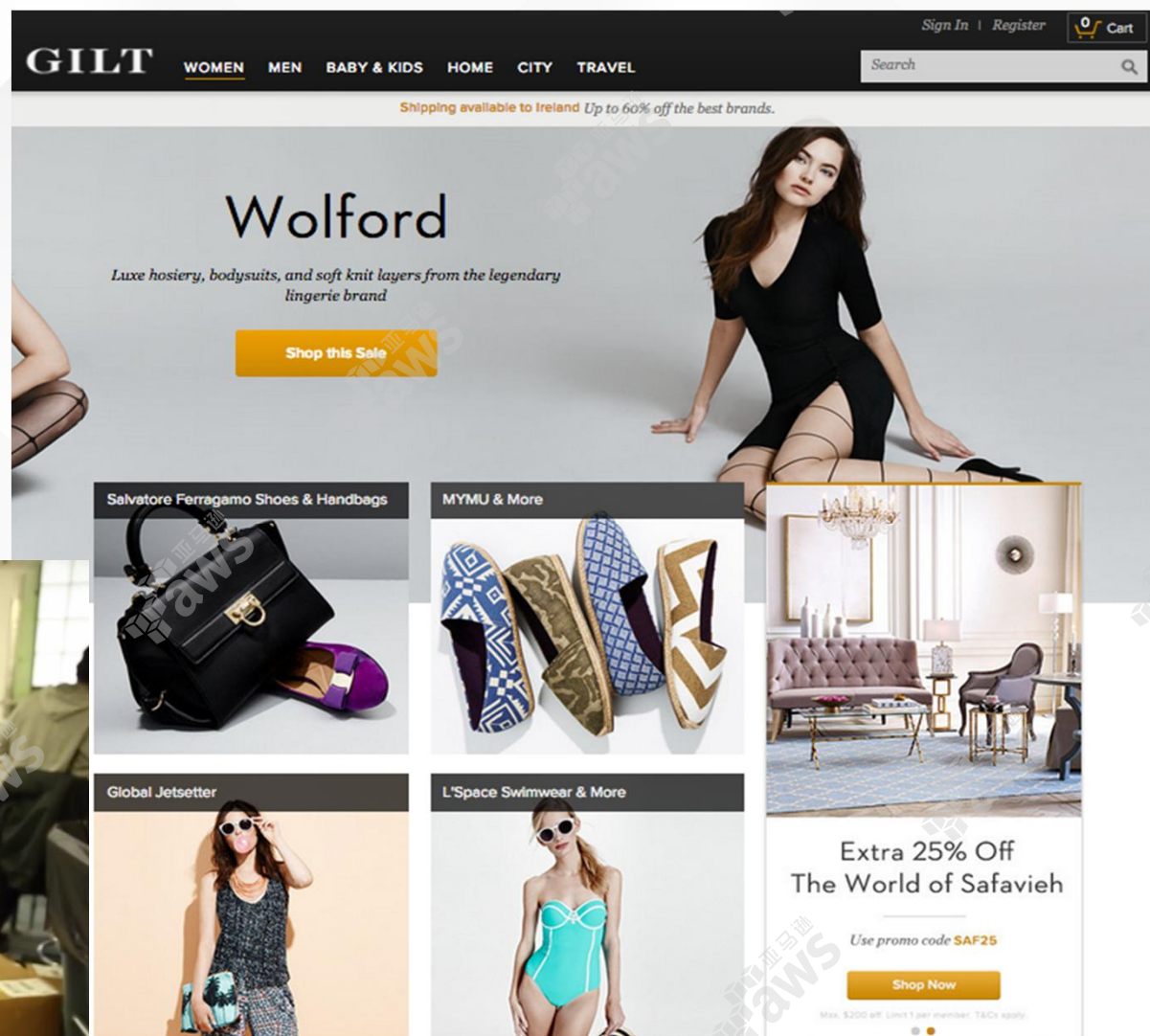


# Gilt案例分享



# Gilt的旅程

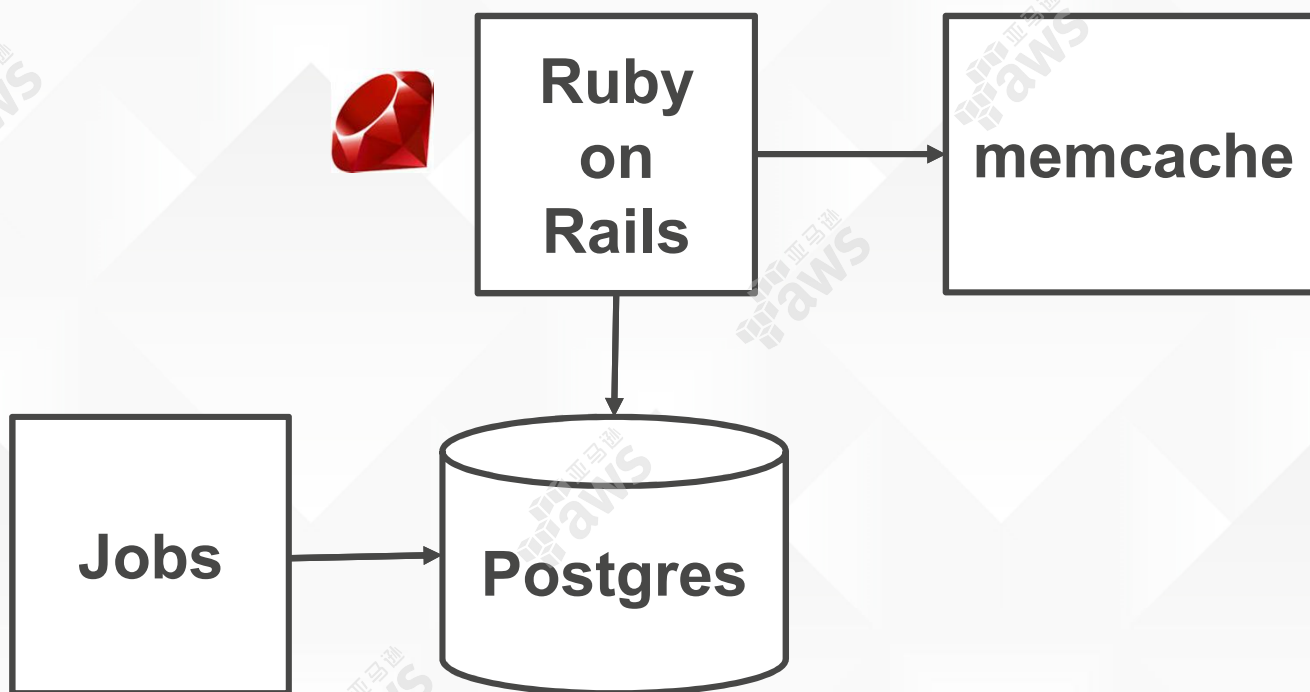
## 从单体应用到微服务架构





# 2007：一个Ruby on Rails的单体应用

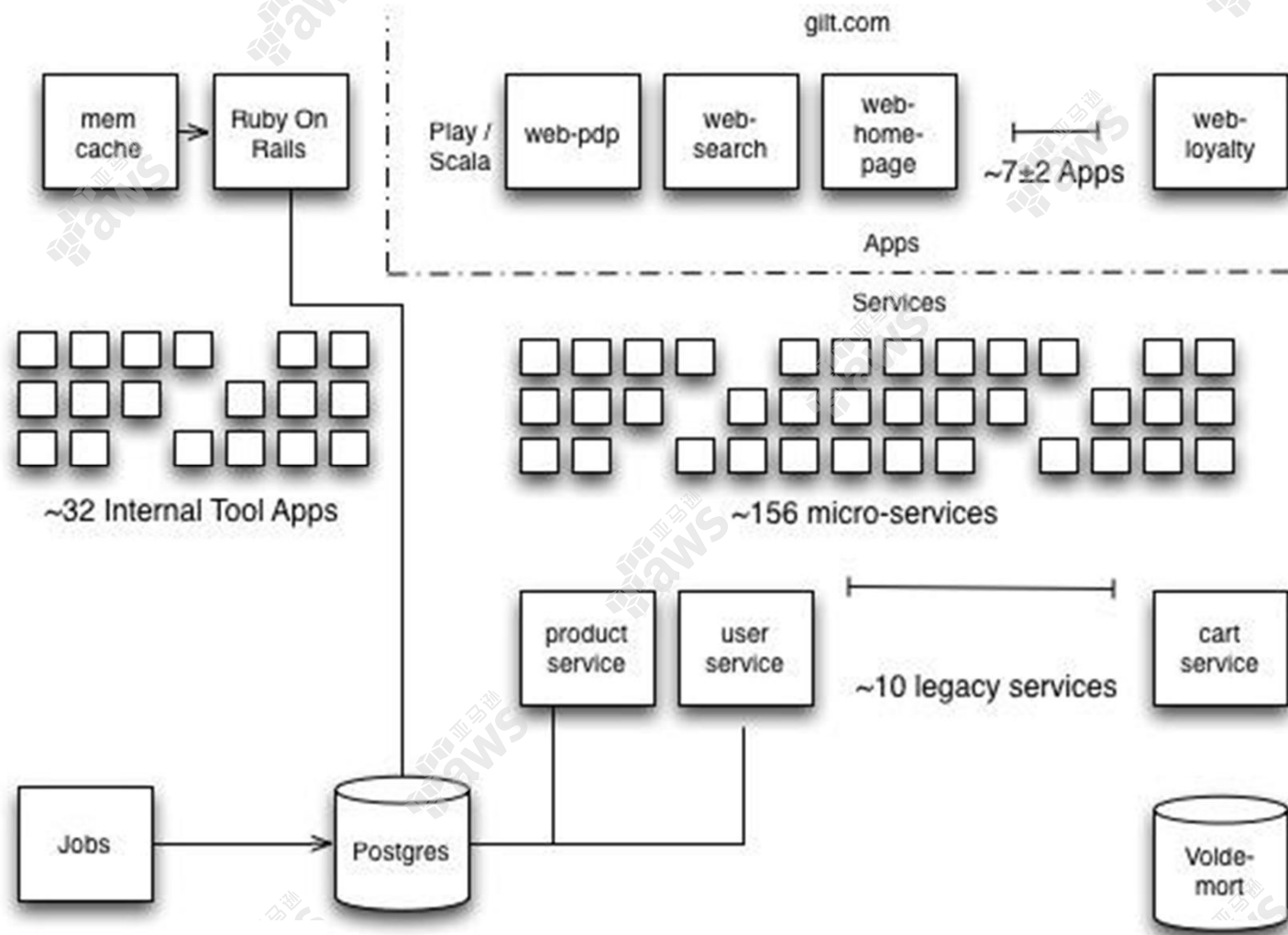
Gilt案例





# 2015: 微服务

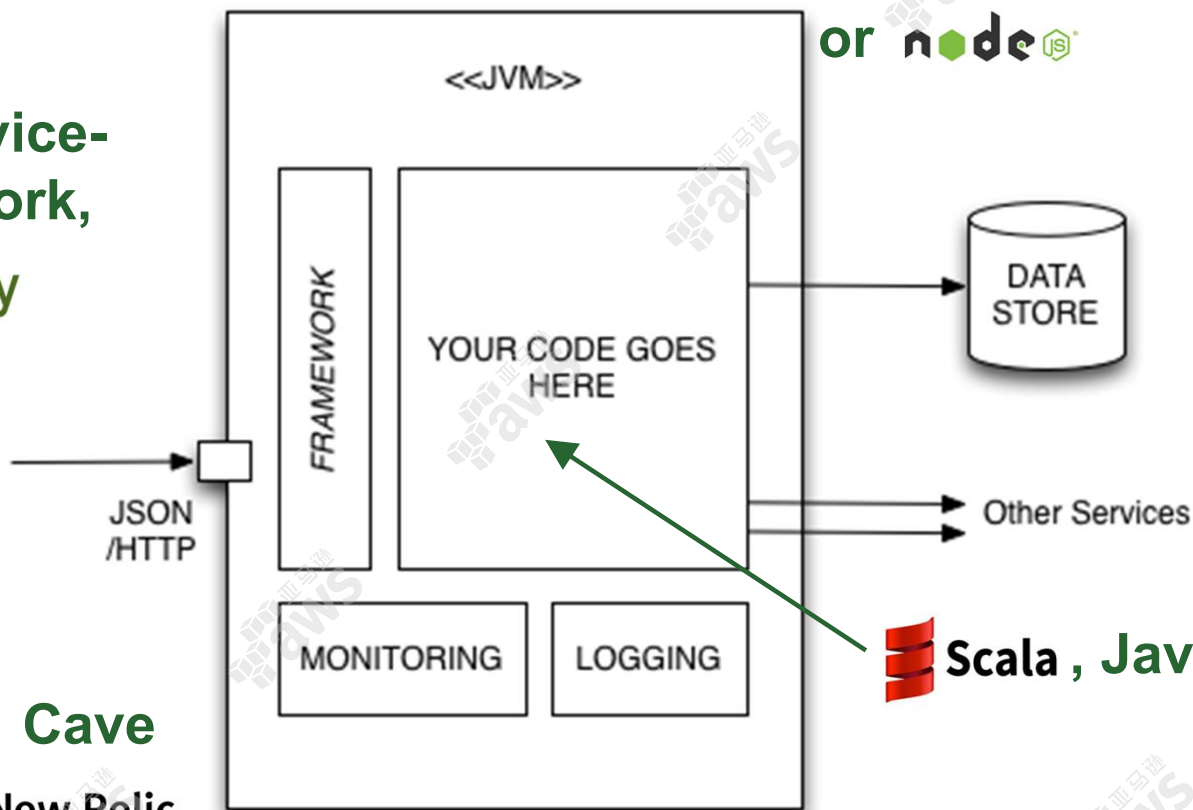
Gilt案例



# 剖析一个服务以及技术选择

Gilt案例

gilt-service-  
framework,



or node

JSON  
/HTTP

FRAMEWORK

YOUR CODE GOES  
HERE

MONITORING

LOGGING

DATA  
STORE

Other Services



Scala, Java, Javascript

Cave



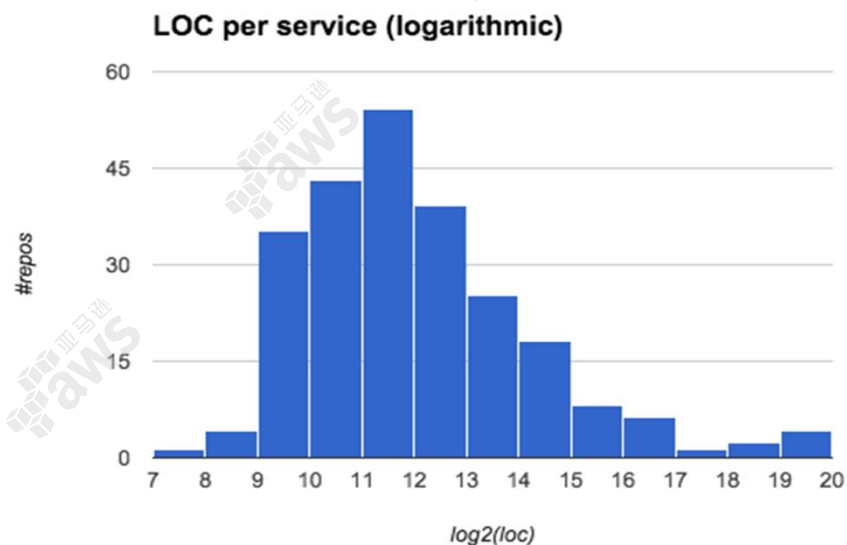
New Relic

Log4j,  
Cloudwatch

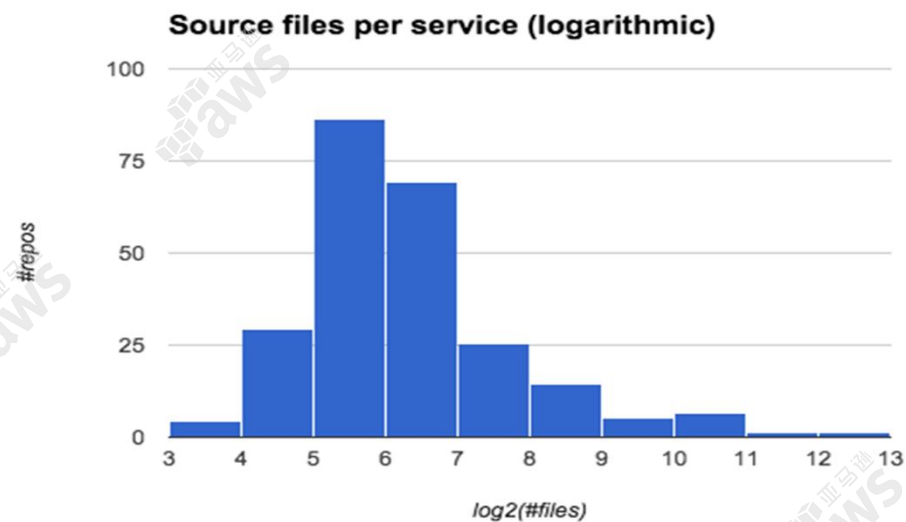


# 服务的规模

Gilt案例



每个服务的代码行数（对数比例）

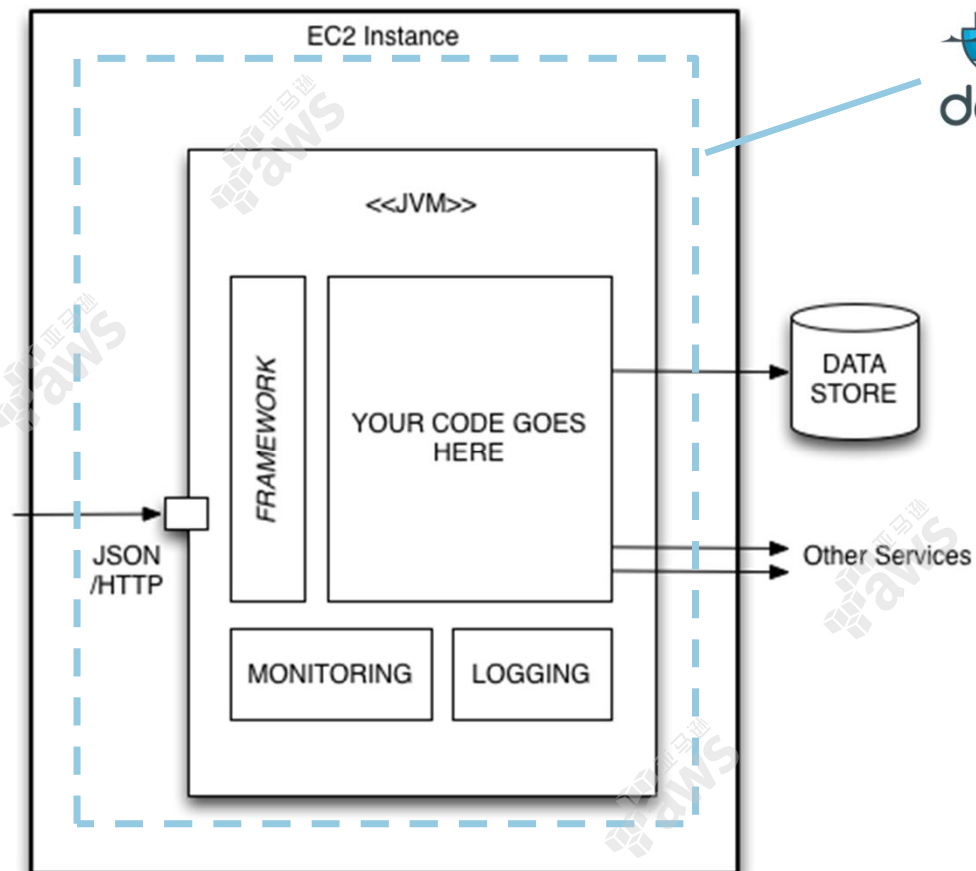
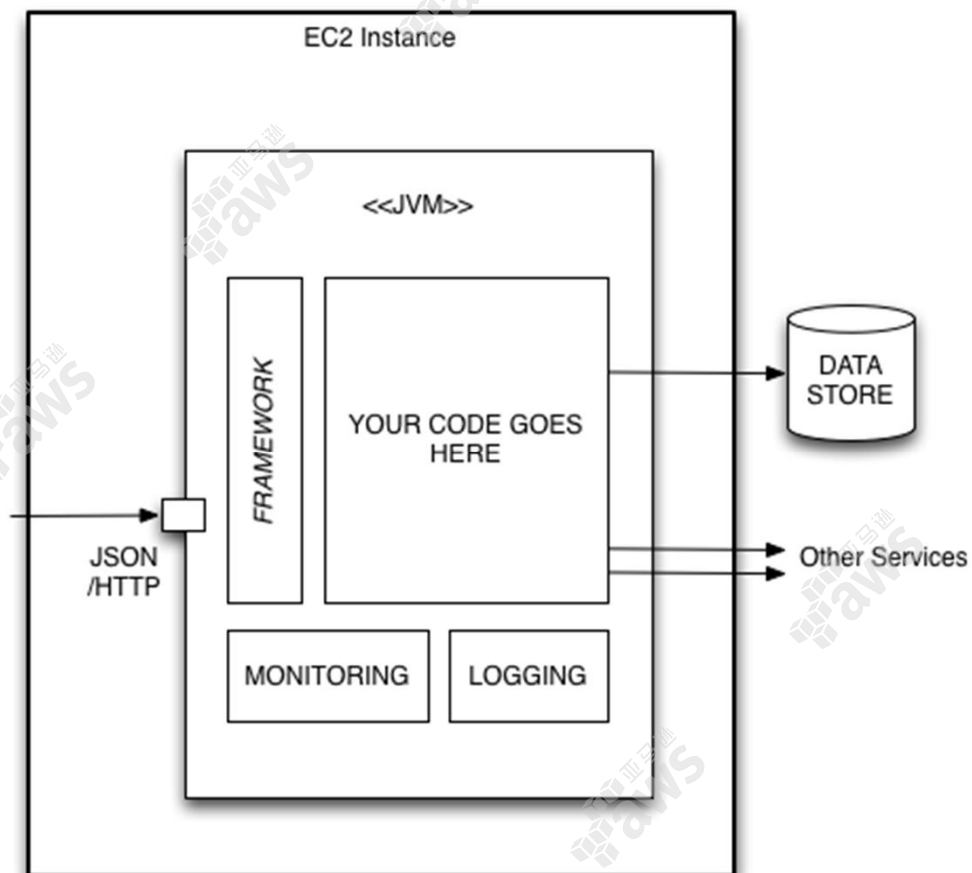


每个服务的源文件数量  
——（包括build、config、xml、  
Java、Scala、Ruby）

# 服务部署

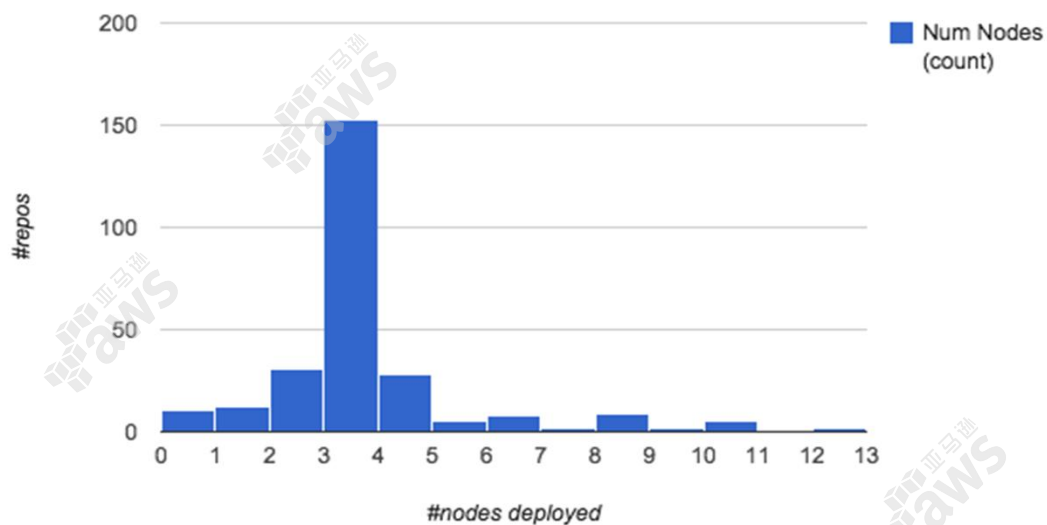
## ——EC2实例和Docker

Gilt案例



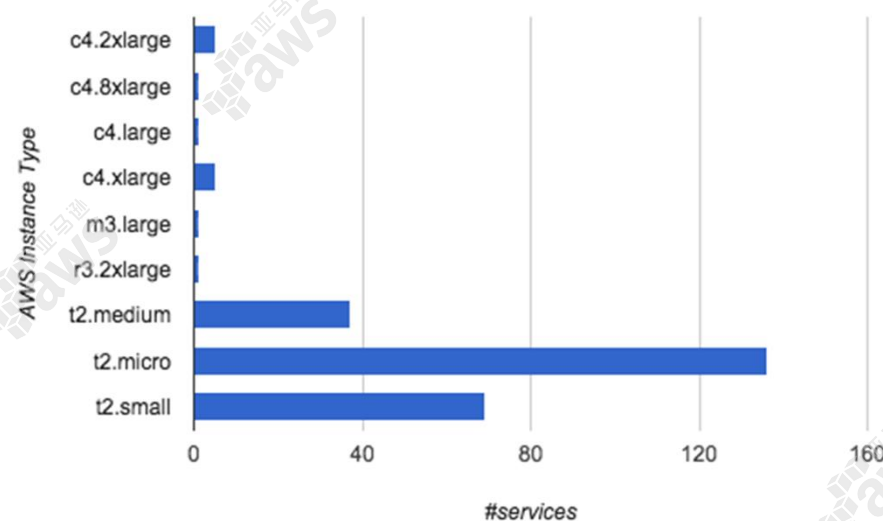
# EC2实例选择

## Instances per service



每个服务运行的实例数量

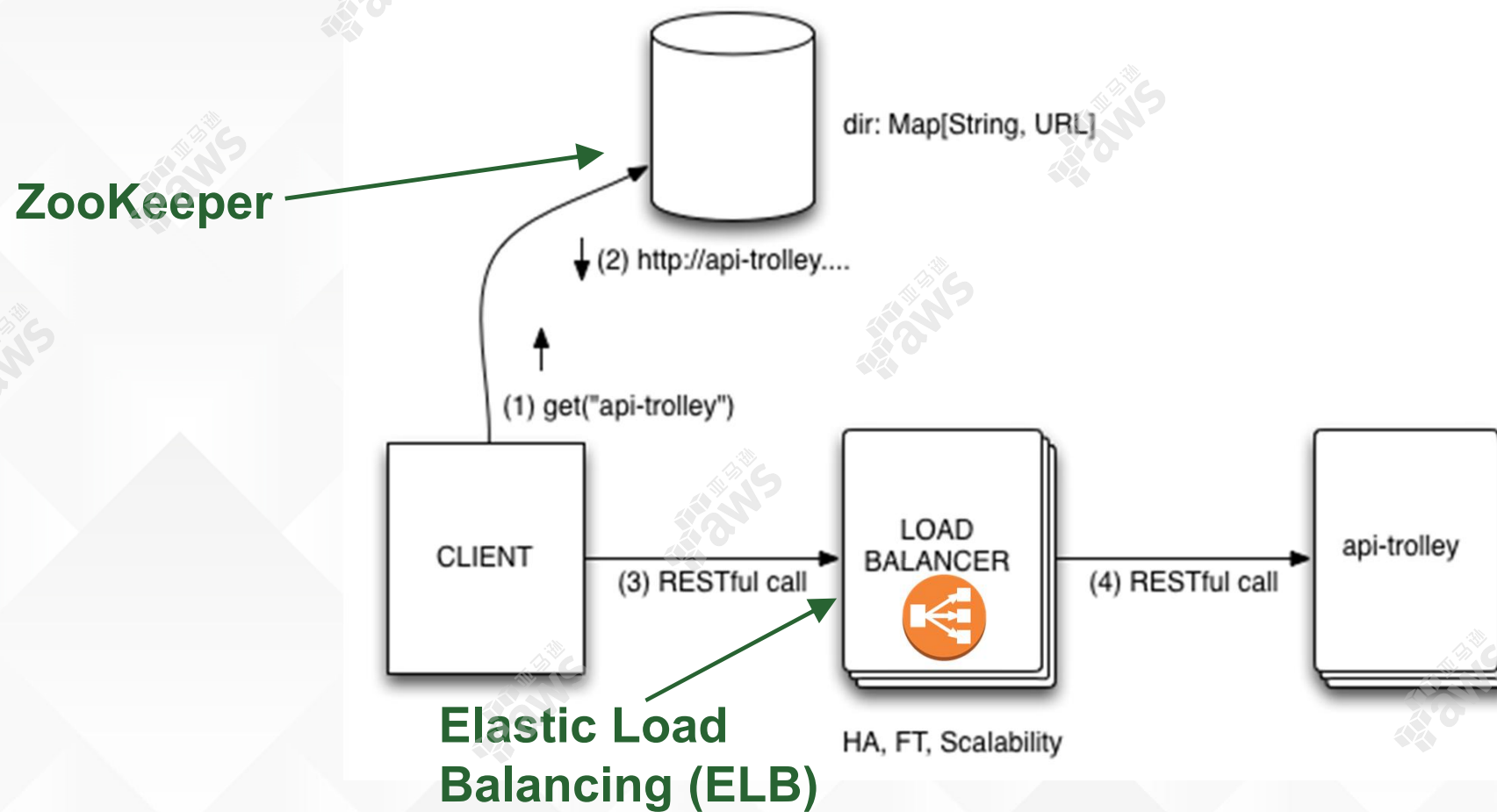
## AWS Instance Size



采用的EC2实例类型

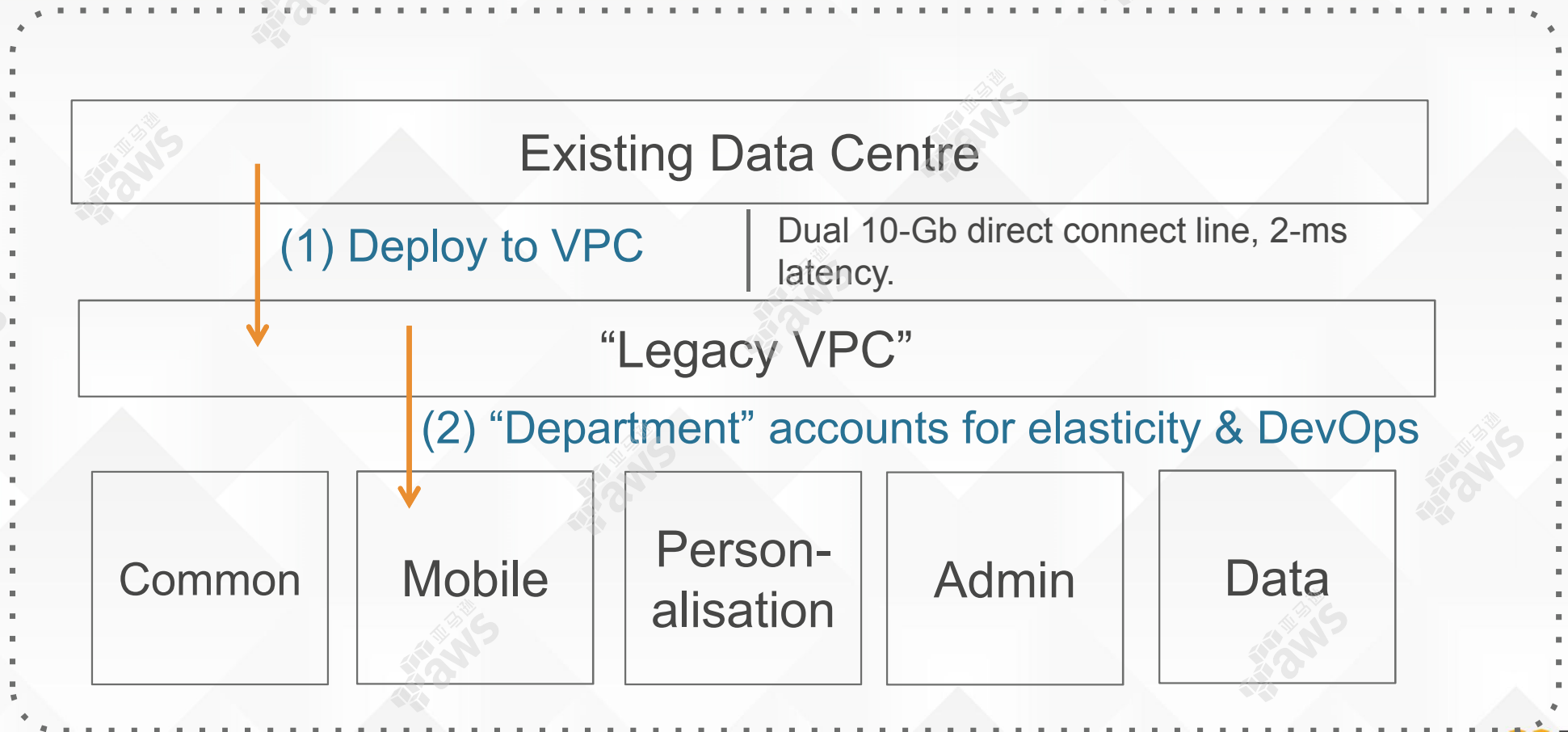
# 服务发现

Gilt案例



# 在AWS之上的部署

Gilt案例





# 微服务+云计算

## 微服务的好处

- 减少团队之间的依赖关系：更快的从代码到产品
- 大量的并行活动
- 每个微服务选择自己的技术、语言、框架
- 优雅地降级服务
- 可自由使用地代码：容易创新、容易失败、继续前进

## 云计算的好处

- 自动化DevOps
- 降低使用新技术的门槛（Amazon DynamoDB、Amazon Kinesis、...）
- 隔离
- 费用可见
- 安全工具（IAM）
- 良好的文档
- 方便的弹性
- 方便的混合
- 良好的性能



Thank You

