AWS SUMMIT

Microservices on AWS

AWS Summit Berlin 2016

Matthias Jung, Solutions Architect
Julien Simon, Evangelist

April, 12th, 2016

amazon web services

Agenda

What are Microservices?

Why Microservices?

Challenges of Microservices

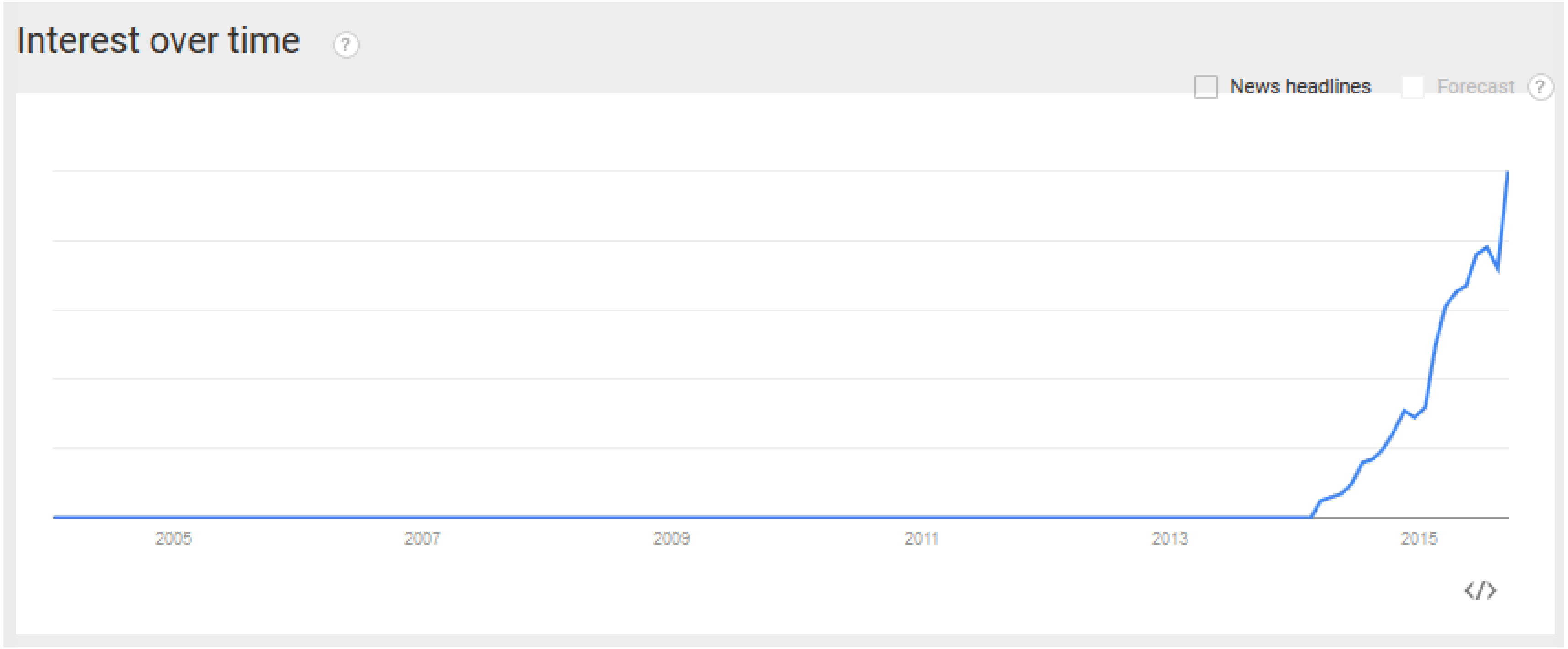
Microservices on AWS

Docker with ECR & ECS - Demo

What are Microservices?

What are Microservices?

Google Trends

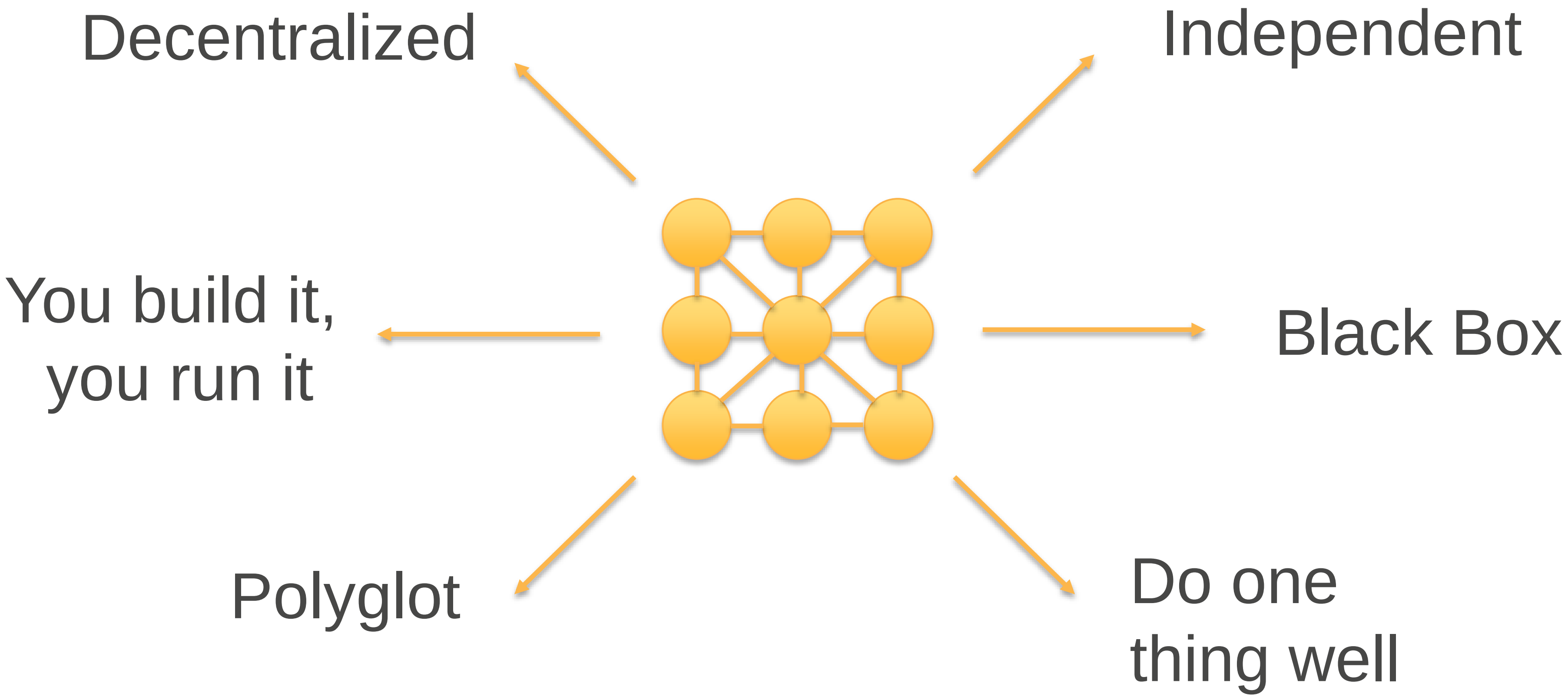


What are Microservices?

Related concepts

- Service Oriented Architectures
- API First
- Agile Software Development
- Continuous Delivery
- DevOps

Characteristics of Microservice Architectures



Why Microservices?

Why Microservices?

Gilt: “From Monolith Ruby App to Distributed Scala Micro-Services” (*NYC Tech Talks*) [Link]

Nike: “Nike’s Journey to Microservices” (*AWS Re:Invent 2014*) [Link]

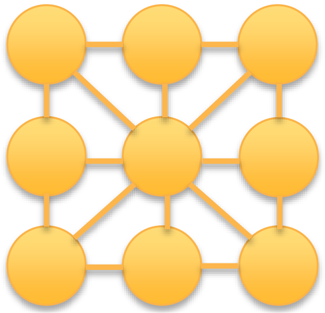
SoundCloud: ”Building Products at SoundCloud - Part III: Microservices in Scala and Finagle” [Link]

Capital One: “Lack Of Legacy Lets Capital One Build Nimble Infrastructure” [Link]

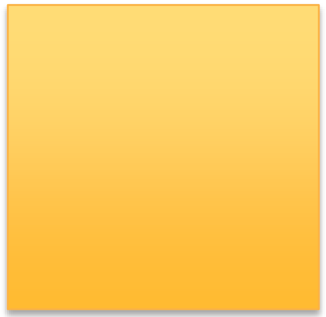
Hailo: “A Journey into Microservices” [Link]

Autoscout24: “Why Autoscout24 changes its technology” [Link]

Zalando: “From Monolith to Microservices” [Link]



vs



Microservices

Monolith

Problems of Monolithic Architectures

Code complexity and maintainability

Deployment becomes the bottleneck

Fear to change

Lack of ownership

Failure dependencies

One size doesn’t fit all (ex: relational DB)

Hard to scale out

Problems of Monolithic Architectures

Code complexity and maintainability

Deployment becomes the bottleneck

Fear to change

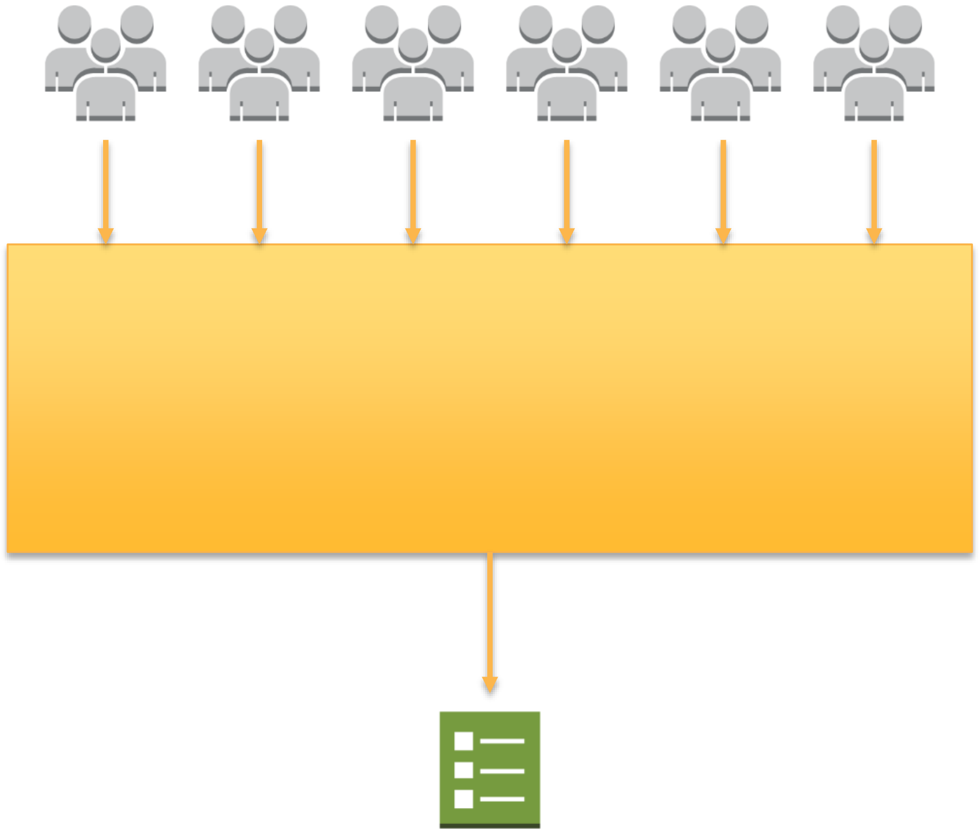
Lack of **ownership**

Failure dependencies

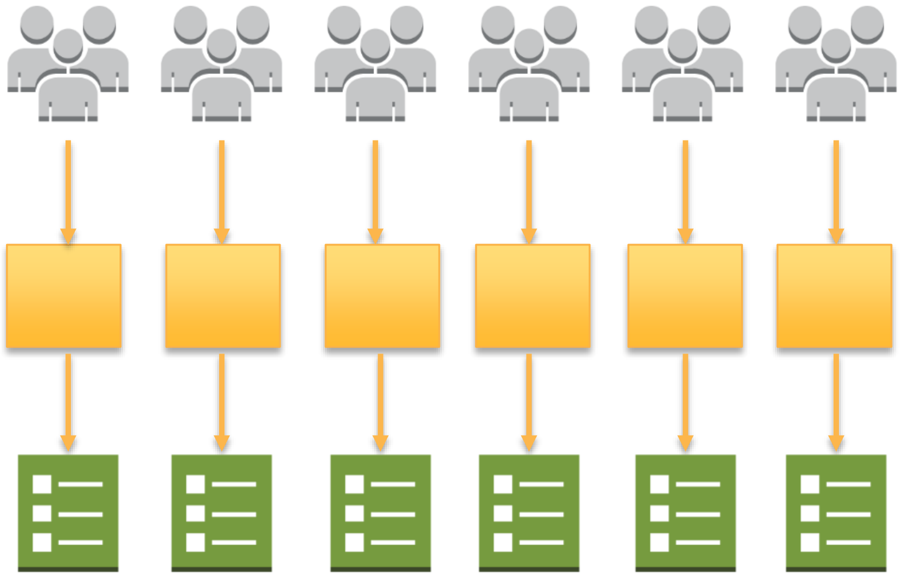
One size doesn’t fit all (ex: relational DB)

Hard to **scale** out

Problems of Monolithic Architectures



Development Life Cycle with Small Teams



Benefits of Microservices

Speed

- Faster development and deployment

Innovation

- Autonomy of teams, culture of change
- Ownership and DevOps culture

Quality

- Composability and reusability
- More maintainable code
- Better scaling and optimizations
- Failure Isolation and Resiliency

What Customers Say

“Avoid fear to change things”

“Applied SE best practices to operations”

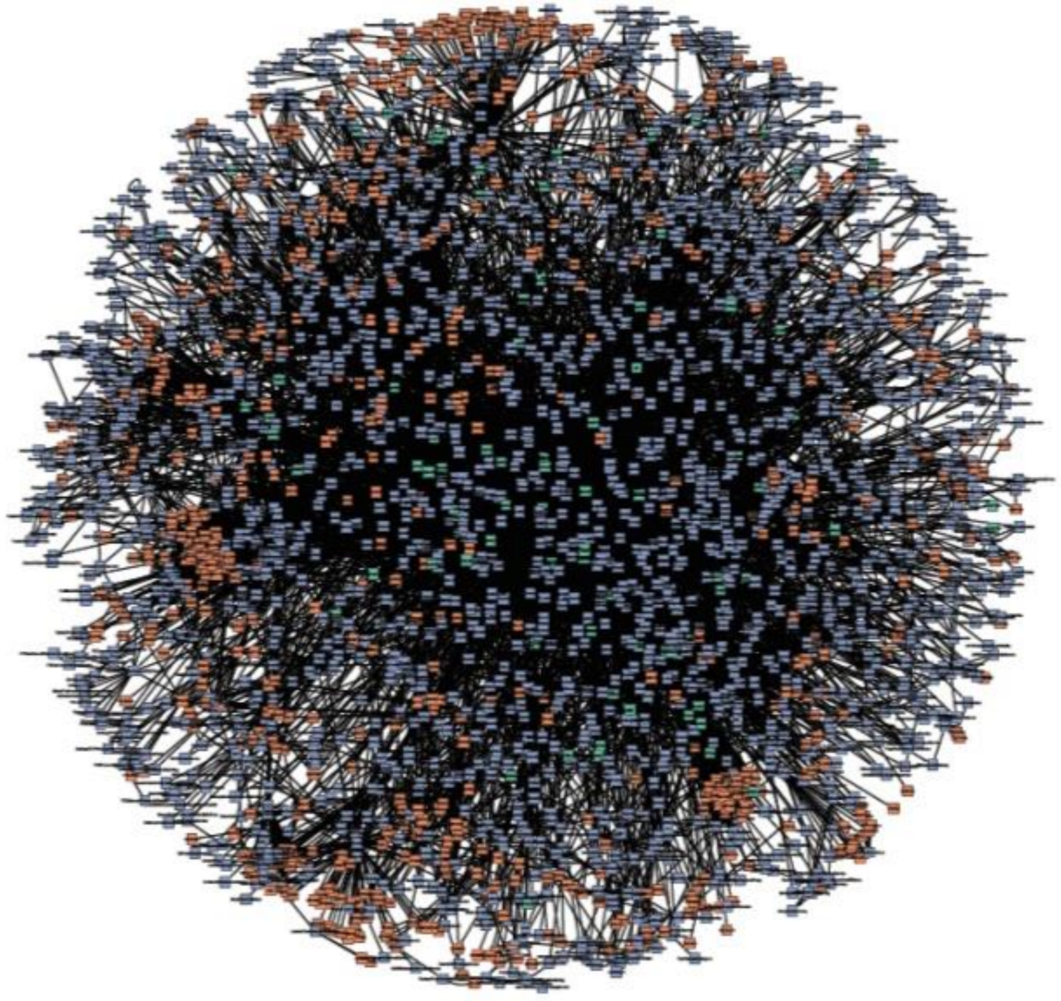
“Easily switch between synchronous and asynchronous communication”

“Easy to start new things from scratch”

“People take ownership”

“Deploy more – deploy faster – deploy better code”

The Amazon DevOps Story



Service-Oriented Architecture (SOA)

Everything gets a service interface

Primitives

“Microservices”

Decentralized

Two-pizza teams

Agility, autonomy, accountability, and ownership

“DevOps”

Decentralized Ownership

Promote Best Practices

Support Agile
SW Dev Lifecycle

No gatekeepers

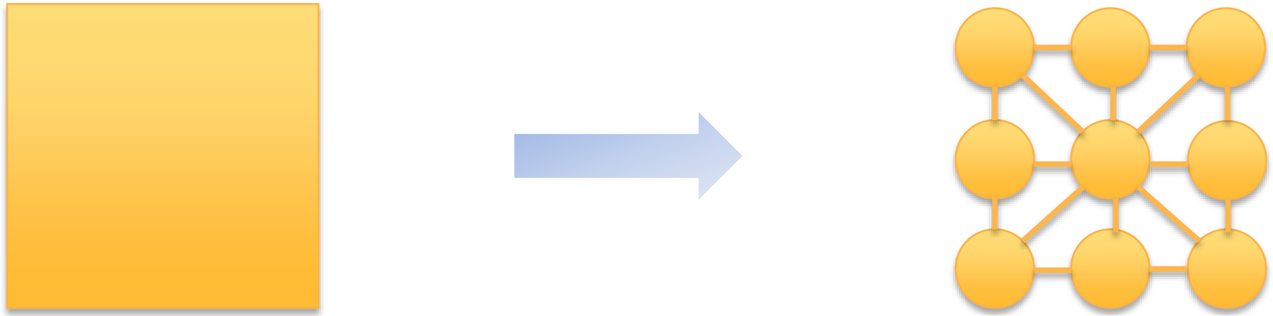
Technology Agnostic

DEPLOYMENTS AT AMAZON.COM

~11.6s	~1,079	~10,000	~30,000
Mean time between deployments (weekday)	Max number of deployments in a single hour	Mean number of hosts simultaneously receiving a deployment	Max number of hosts simultaneously receiving a deployment

Challenges of Microservices

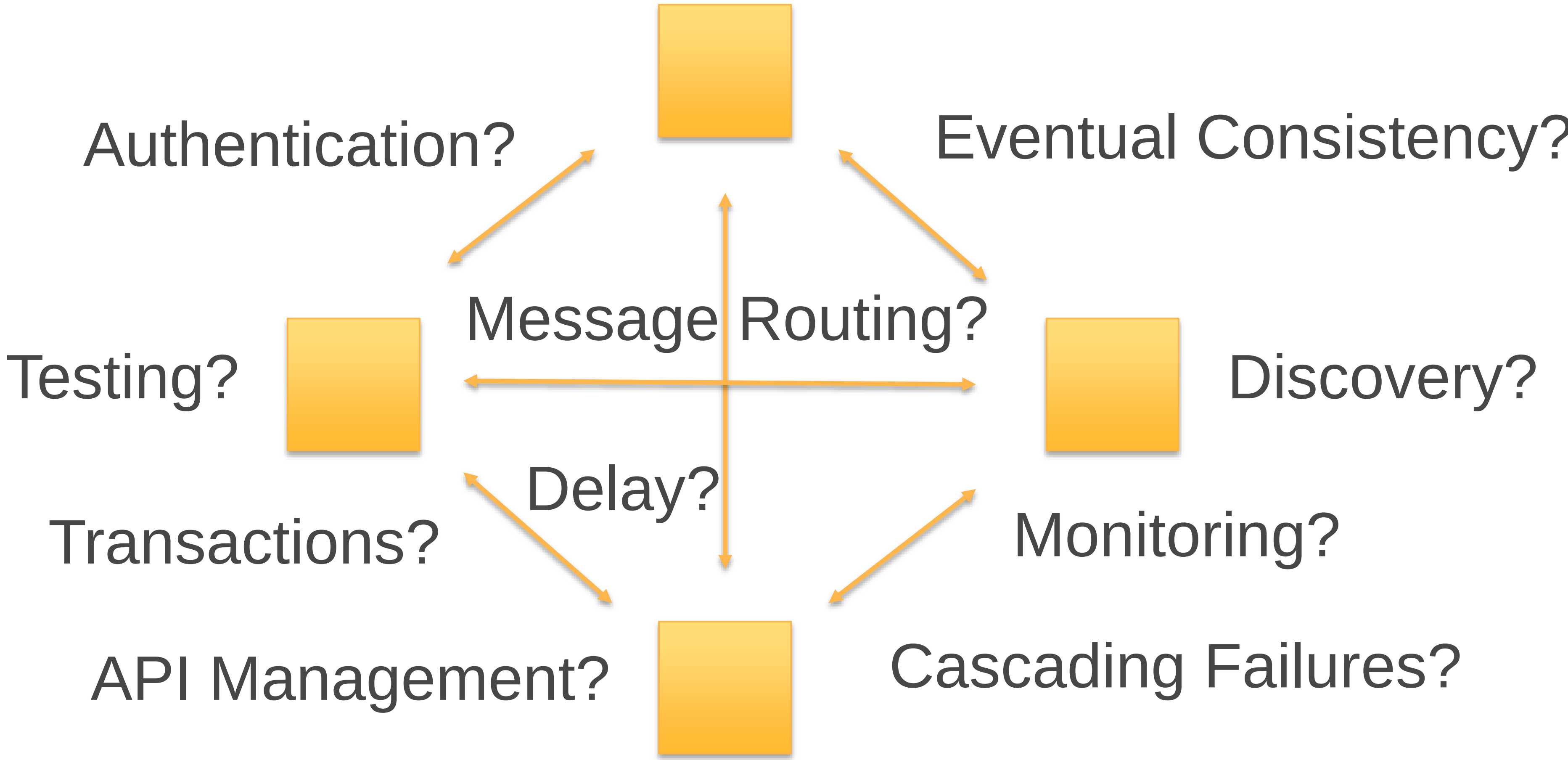
Challenges of Microservices



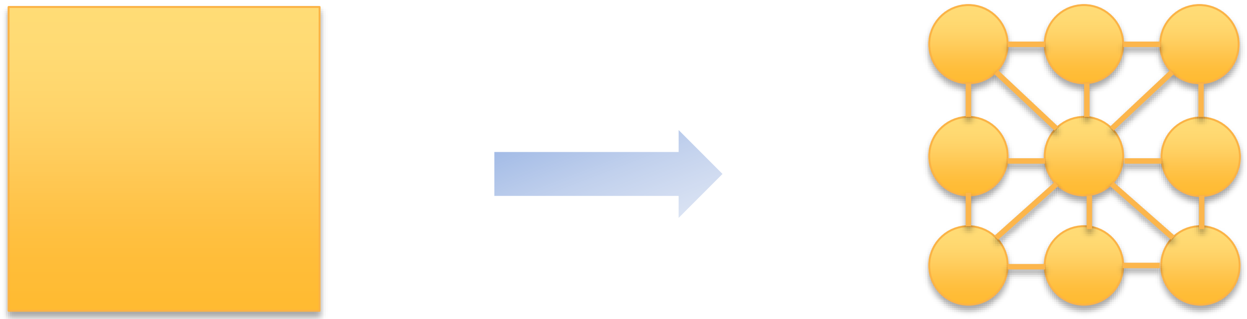
Complexity in Code Base

Complexity in Interactions

Complexity in Interactions



Challenges of Microservices



One size doesn't fit all

Heterogenity No Standards

Challenges

Organizational Cultural Challenges

- You built it, you run it

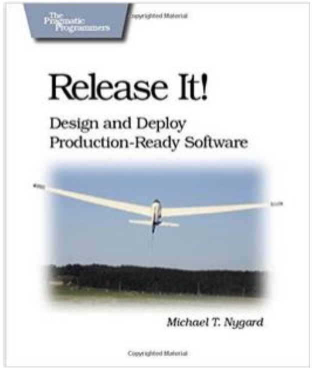
Architectural Challenges

- Dealing with asynchronicity
- Cascading failures
- Discovery and authentication of services
- Integration Tests

Operational Challenges

- Duplication of processes and tools
- Complexity moves from components to interactions
- Debugging across components
- Deployment

The Phoenix Project
REVISED WITH NEW RESOURCE GUIDE
A Novel About IT, DevOps, and Helping Your Business Win
Gene Kim, Kevin Behr, and George Spafford


The Pragmatic Programmers
Release It!
Design and Deploy Production-Ready Software
Michael T. Nygard

powered by amazon web services

Microservices on AWS

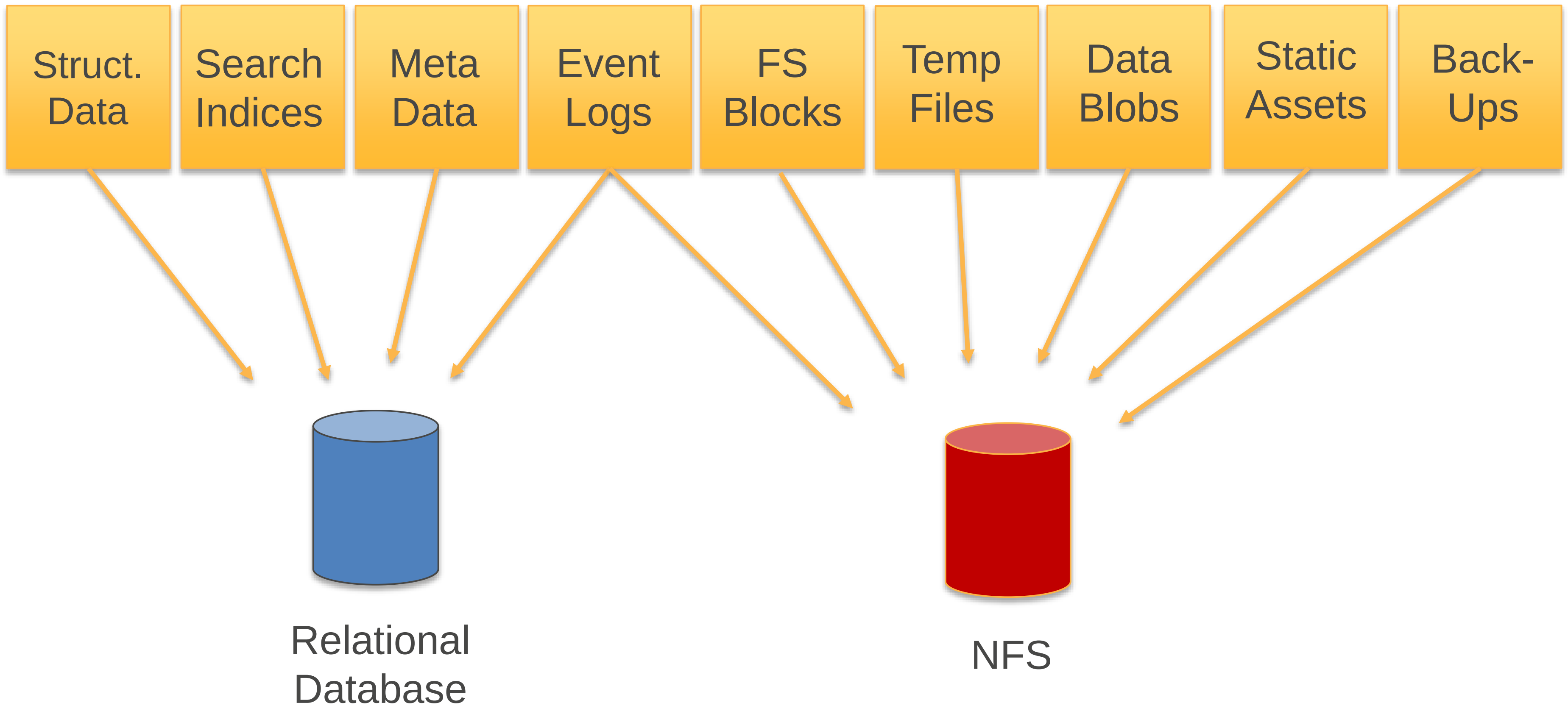
How Can AWS Help with Operational Complexity?

- **On Demand Resources**
    - no capacity guessing
    - resources in any size
    - parallel environments

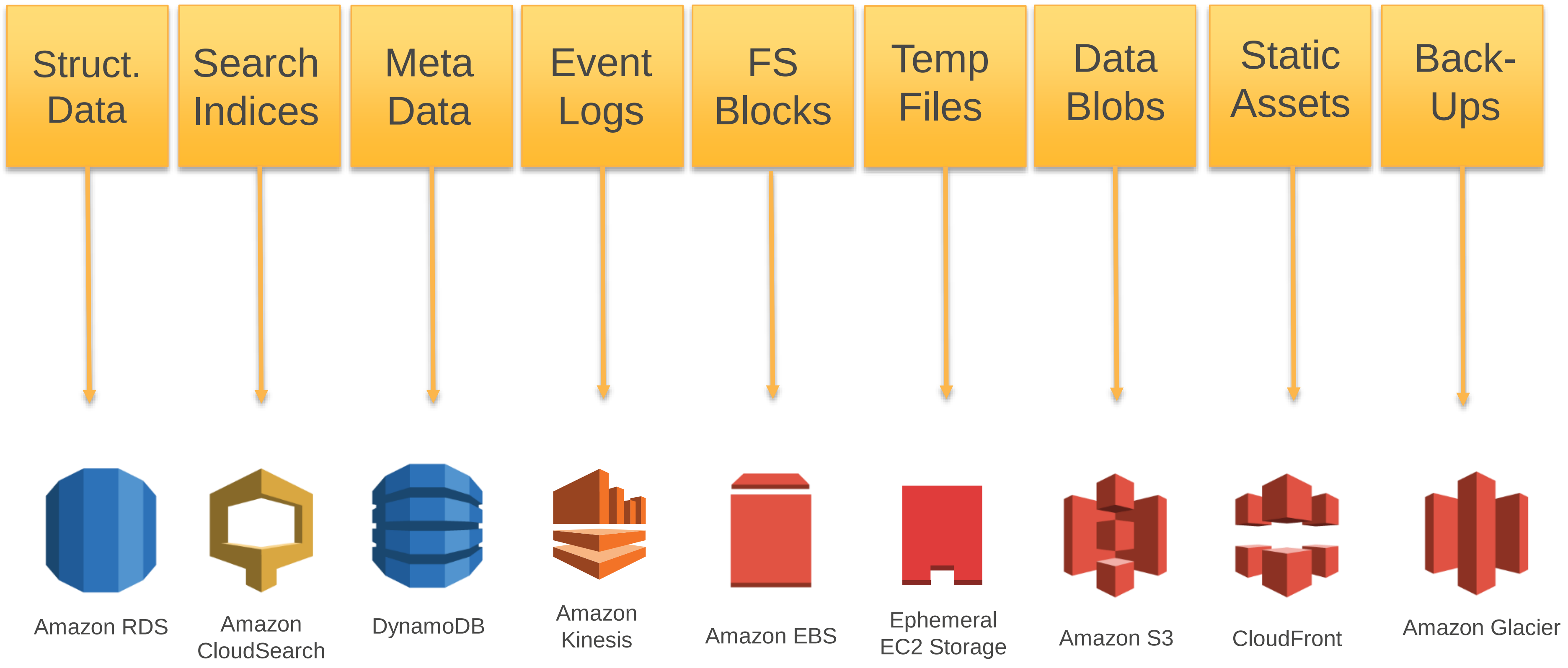
How Can AWS Help with Operational Complexity?

- On Demand Resources
- **Managed Services**

Storage Options in the Traditional World



Storage Options in the Cloud



Don’t Reinvent the Wheel

If you find yourself writing your own…

Notification system
E-Mail component
Search engine
Workflow engine
Queue
Transcoding system
Monitoring system

Amazon SNS

Amazon CloudSearch

Amazon SQS

Amazon SES

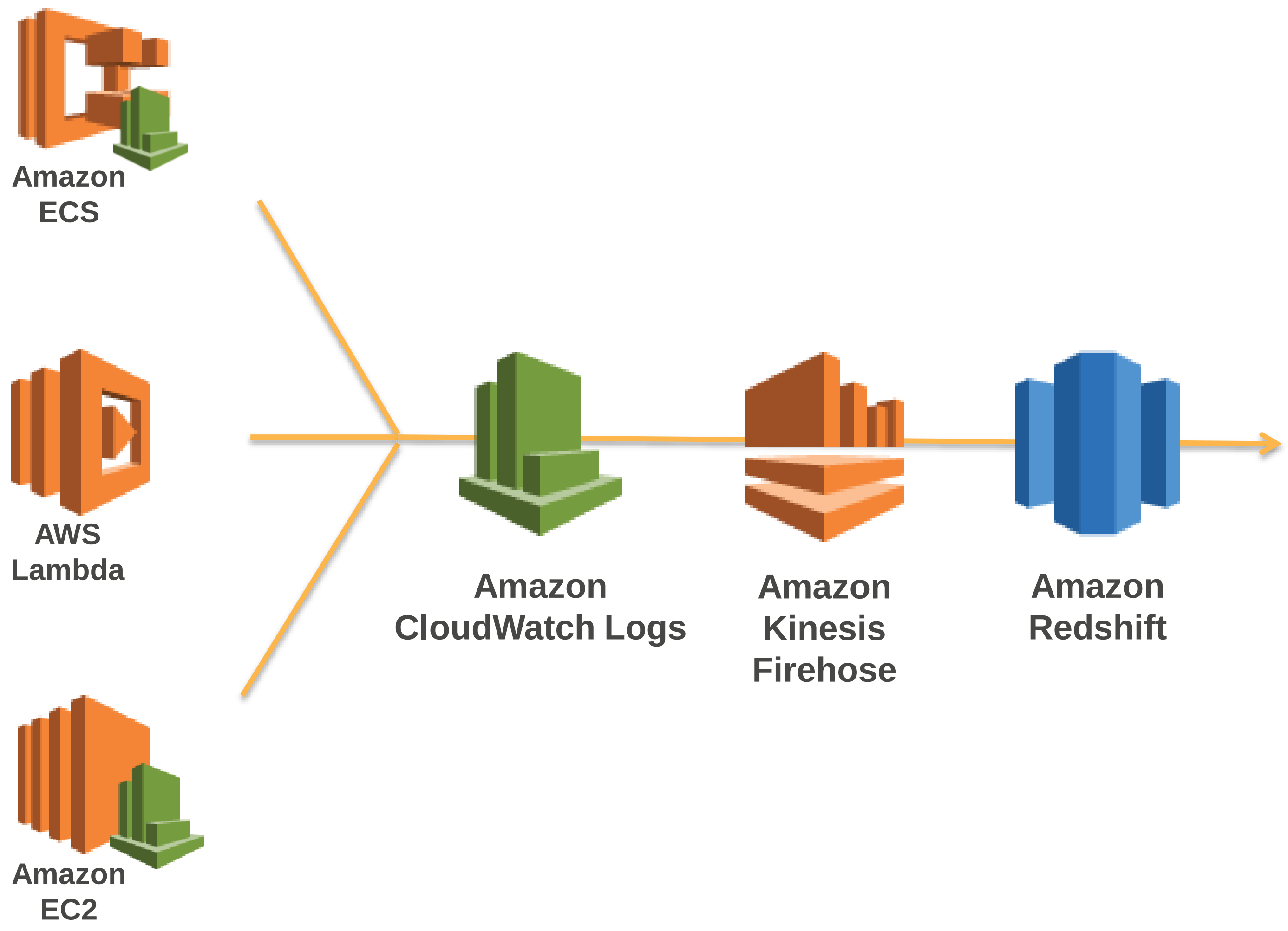
Amazon SWF

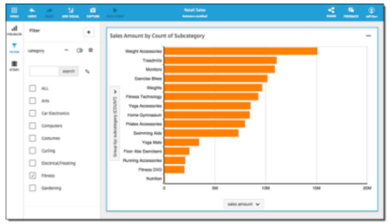
Amazon Elastic Transcoder

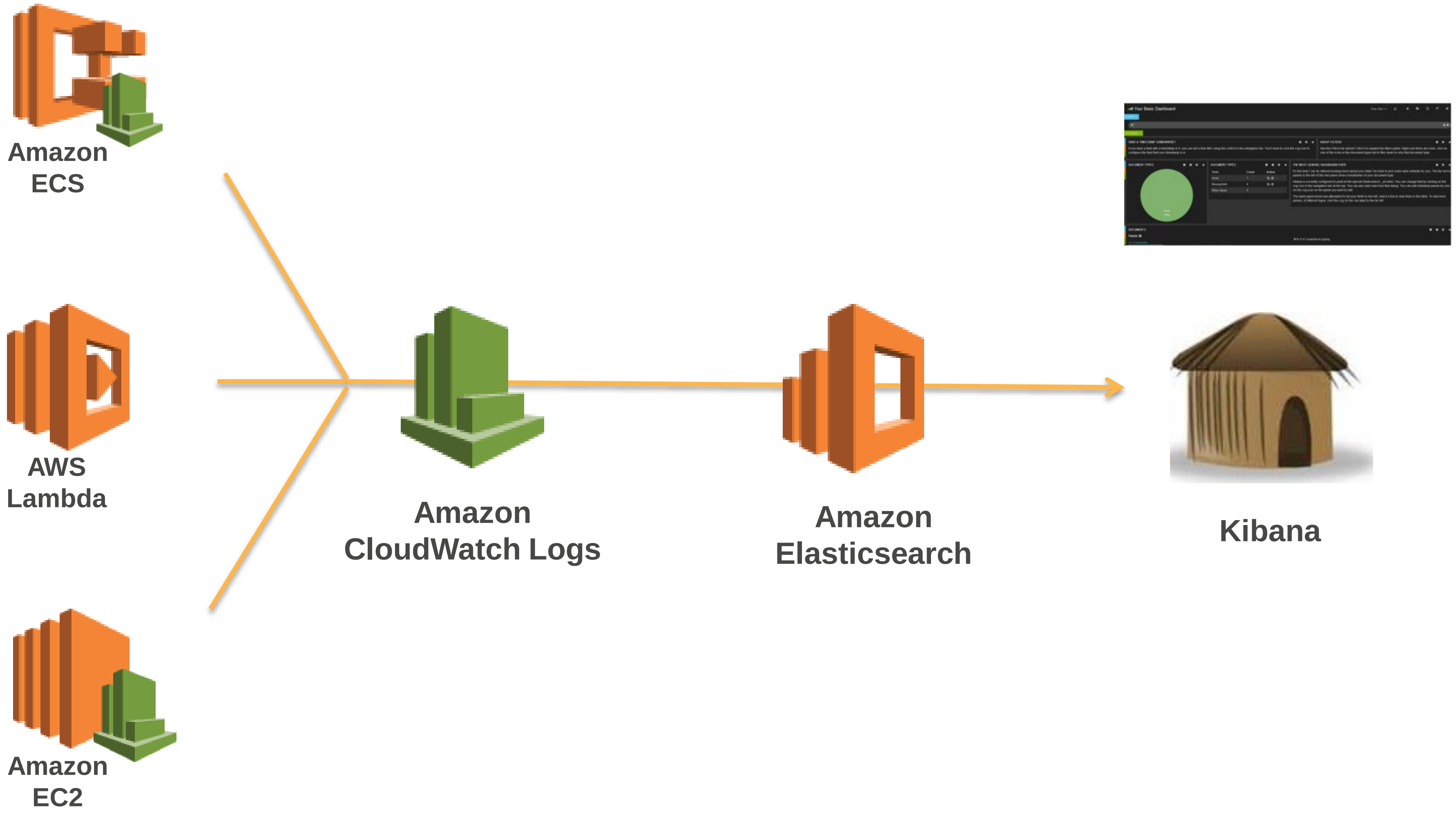
…take a deep breath and stop it now!

How Can AWS Help with Operational Complexity?

- On Demand Resources
- Managed Services
- **Built-in features**
  - Monitoring via CloudWatch
  - Security: IAM, CloudTrail, KMS, …
  - Logging: CloudWatch Logs
  - Scalability: Auto-Scaling, ELB, S3, …
  - Availability: multiple Availability Zones


Amazon ECS
AWS Lambda
Amazon EC2
Amazon CloudWatch Logs
Amazon Kinesis Firehose
Amazon Redshift
Amazon QuickSight


MENU
UNDO
REDO
ADD VISUAL
CAPTURE
PLAY STORY
Retail Sales
SHARE
FEEDBACK
VISUALIZE
FILTER
STORY
Filter
category
search
ALL
Arts
Car Electronics
Computers
Costumes
Cycling
Electrical/Heating
Fitness
Gardening
Sales Amount by Count of Subcategory
Weight Accessories
Treadmills
Monitors
Exercise Bikes
Weights
Fitness Technology
Yoga Accessories
Home Gymnasium
Pilates Accessories
Swimming Aids
Yoga Mats
Floor Abs Exercisers
Running Accessories
Fitness DVD
Nutrition
Group by: subcategory (COUNT)
0
5M
10M
15M
20M
sales amount


Amazon ECS
AWS Lambda
Amazon EC2
Amazon CloudWatch Logs
Amazon Elasticsearch
Kibana
Your Basic Dashboard

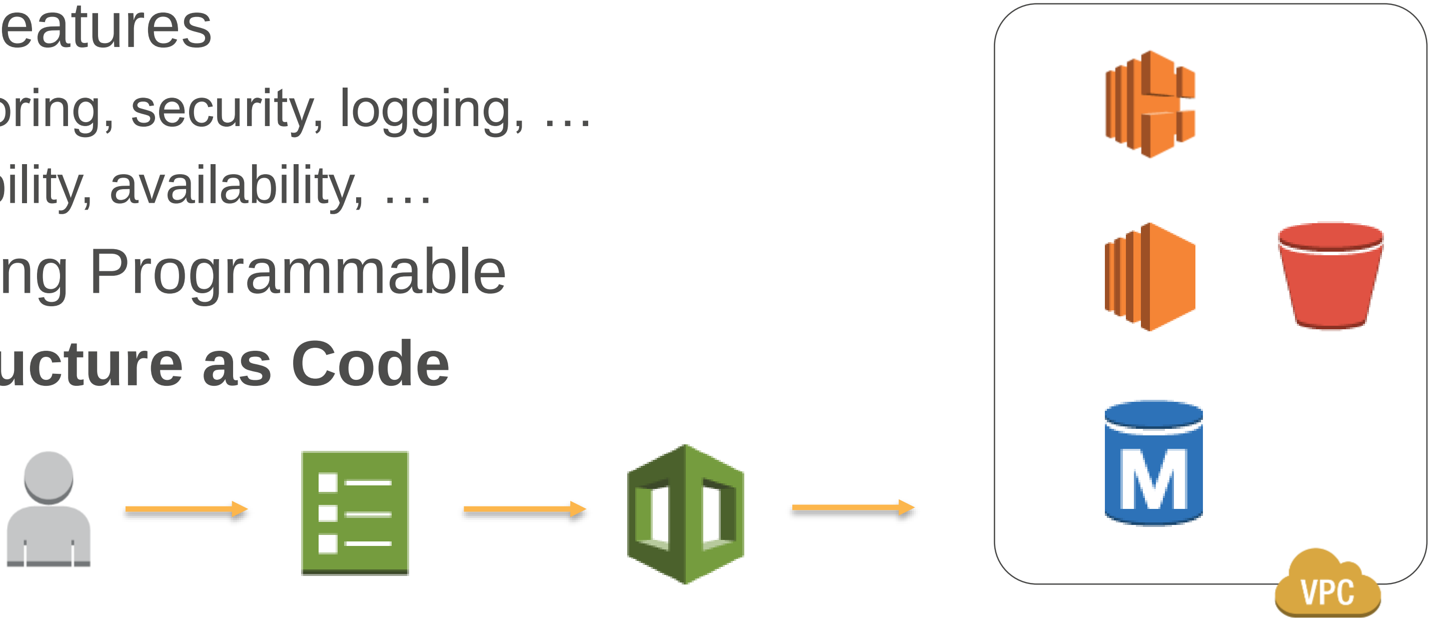
How Can AWS Help with Operational Complexity?

- On Demand Resources
- Managed Services
- Built-in features
  - monitoring, security, logging, …
  - scalability, availability, …
- **Everything Programmable**

AWS CLI & SDKs

How Can AWS Help with Operational Complexity?

- On Demand Resources
- Managed Services
- Built-in features
    - monitoring, security, logging, …
    - scalability, availability, …
- Everything Programmable
- **Infrastructure as Code**



How Can AWS Help with Operational Complexity?

- On Demand Resources
- Managed Services
- Built-in features
  - monitoring, security, logging, …
  - scalability, availability, …
- Everything Programmable
- Infrastructure as Code
- **No Servers**

AWS Lambda

How Can AWS Help with Operational Complexity?

- Run code without infrastructure
- Backend at any scale
- No administration
- JavaScript, Java, and Python

AWS Lambda

How Can AWS Help with Managing APIs?

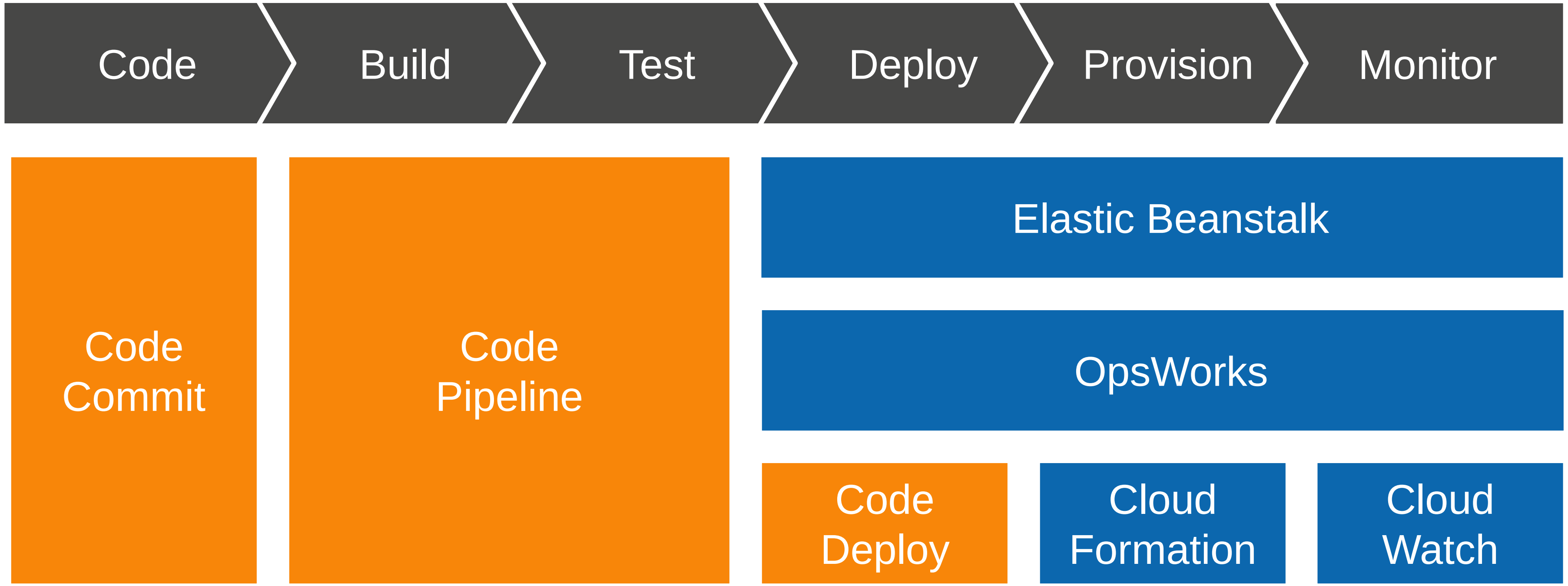
- Managing multiple versions and stages?
- Monitoring 3rd party developer access?
- Access authorization?
- Traffic spikes ?
- Caching ?

How Can AWS Help with Managing APIs?

- Managing multiple versions and stages
- Monitoring 3rd party developer access
- Access authorization
- Traffic spikes
- Caching
- Swagger Support
- Request/Response Transformation
- API Mocking

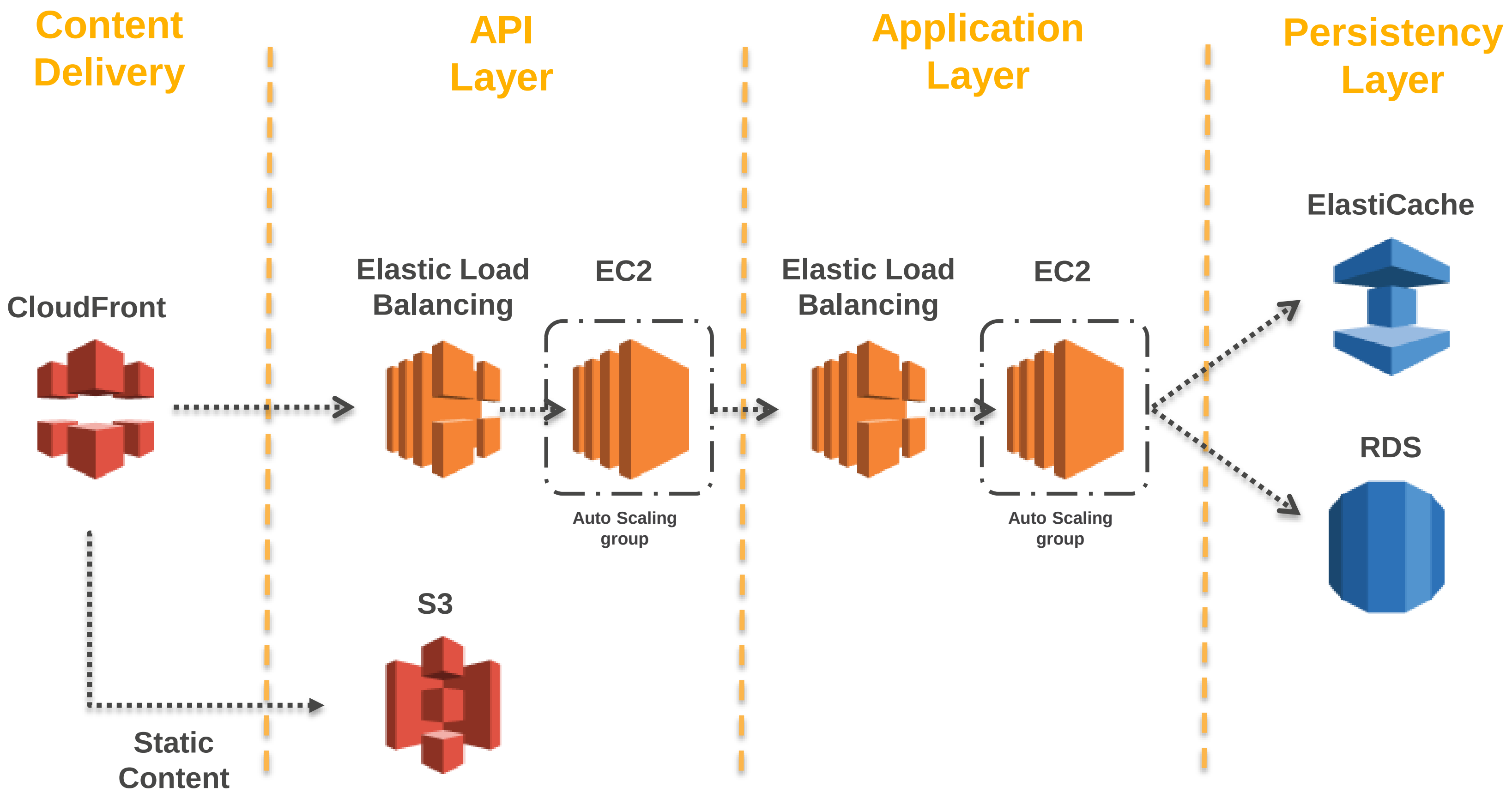
API Gateway

How Can AWS Help with Scaling Deployments?

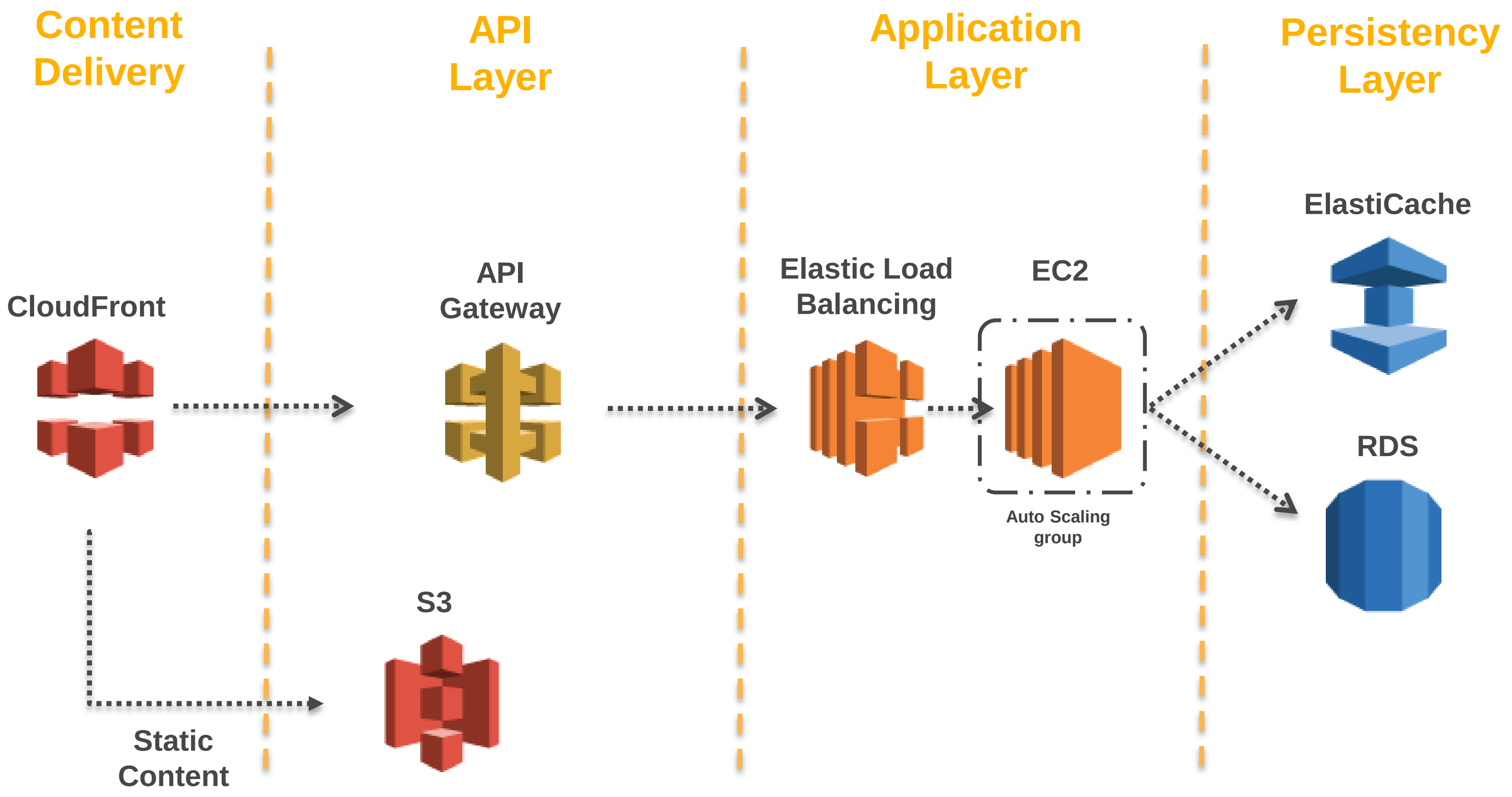


Microservice Architectures

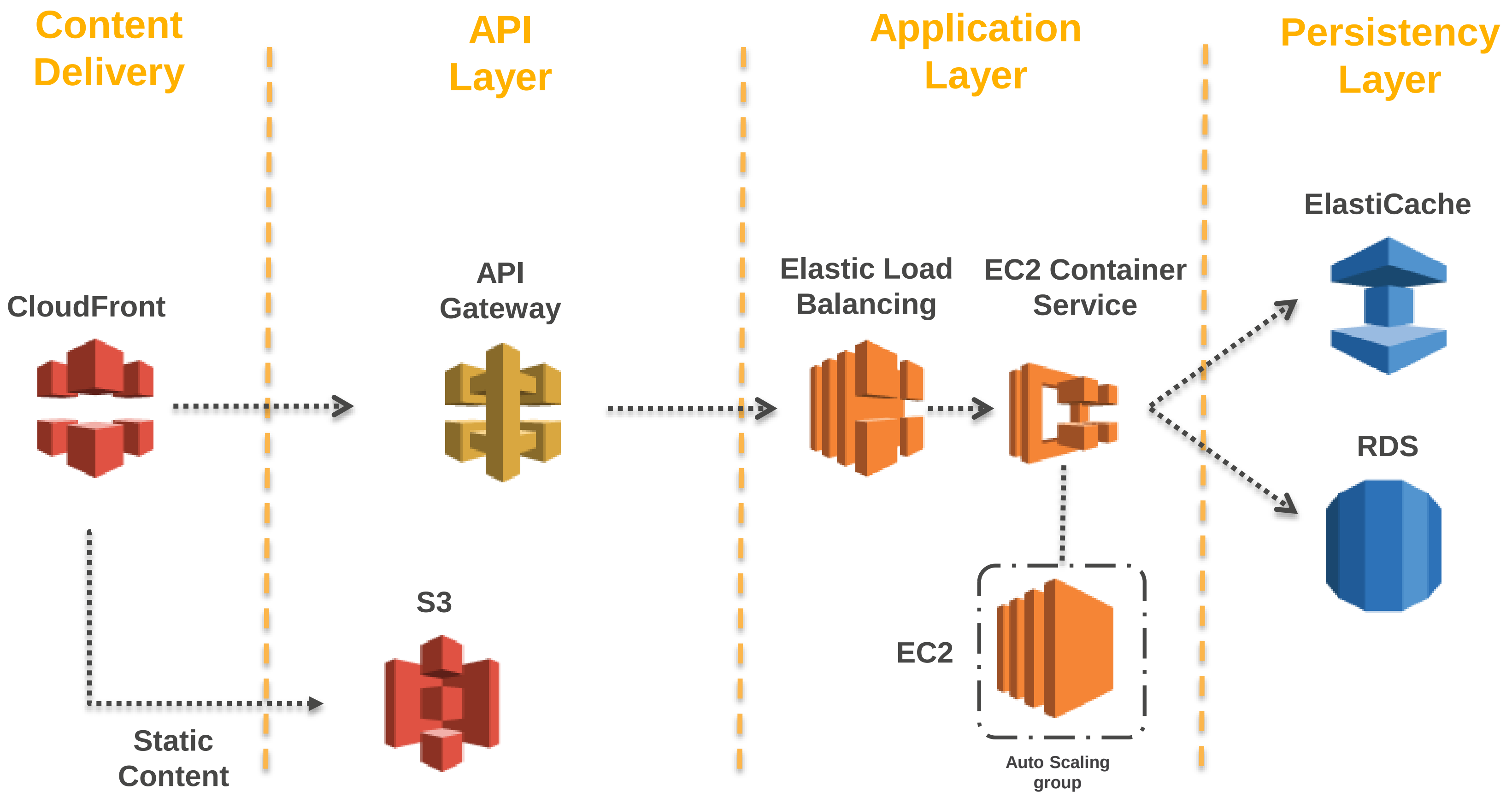
A Typical Microservice Architecture on AWS



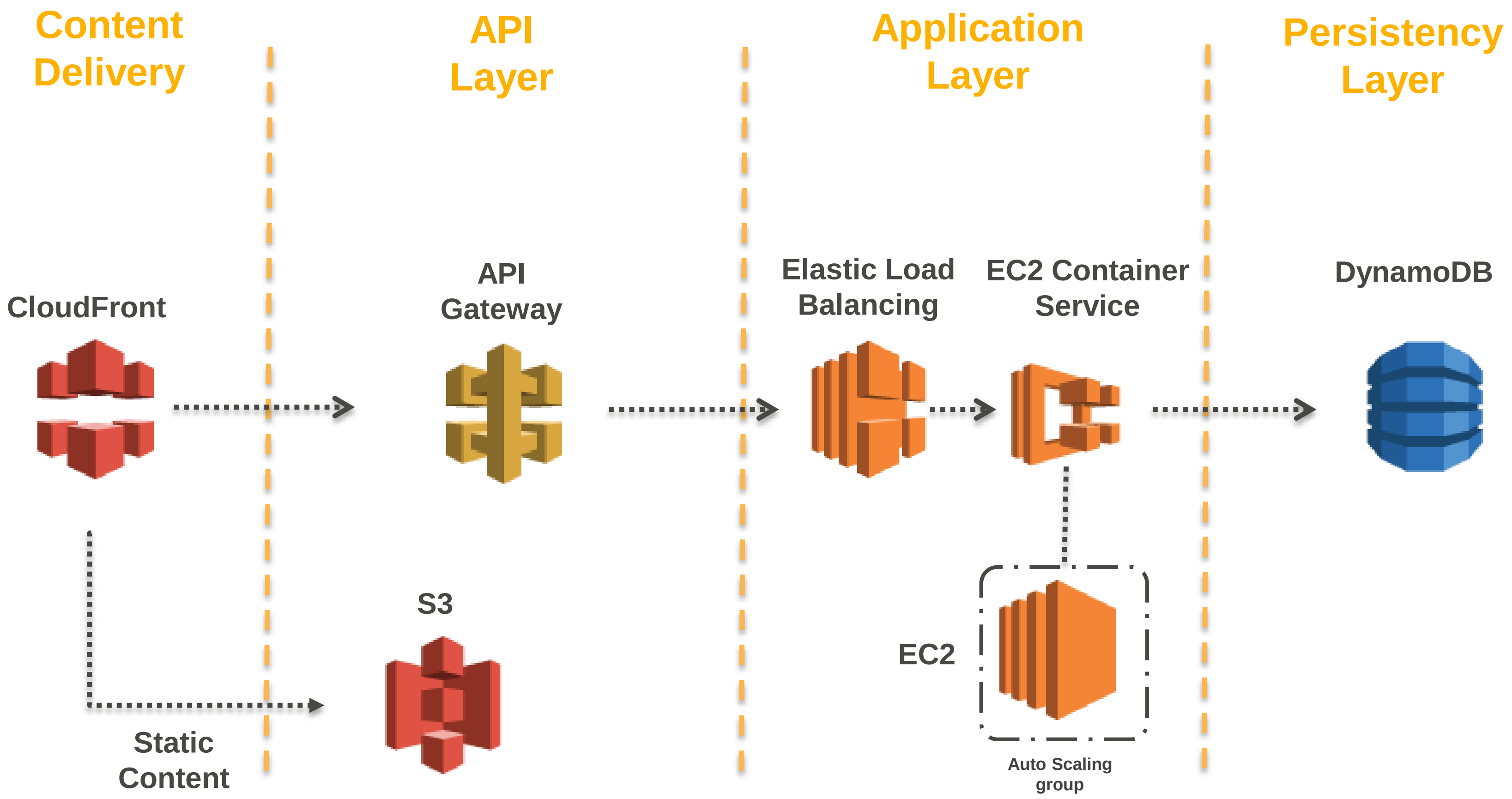
A Typical Microservice Architecture on AWS



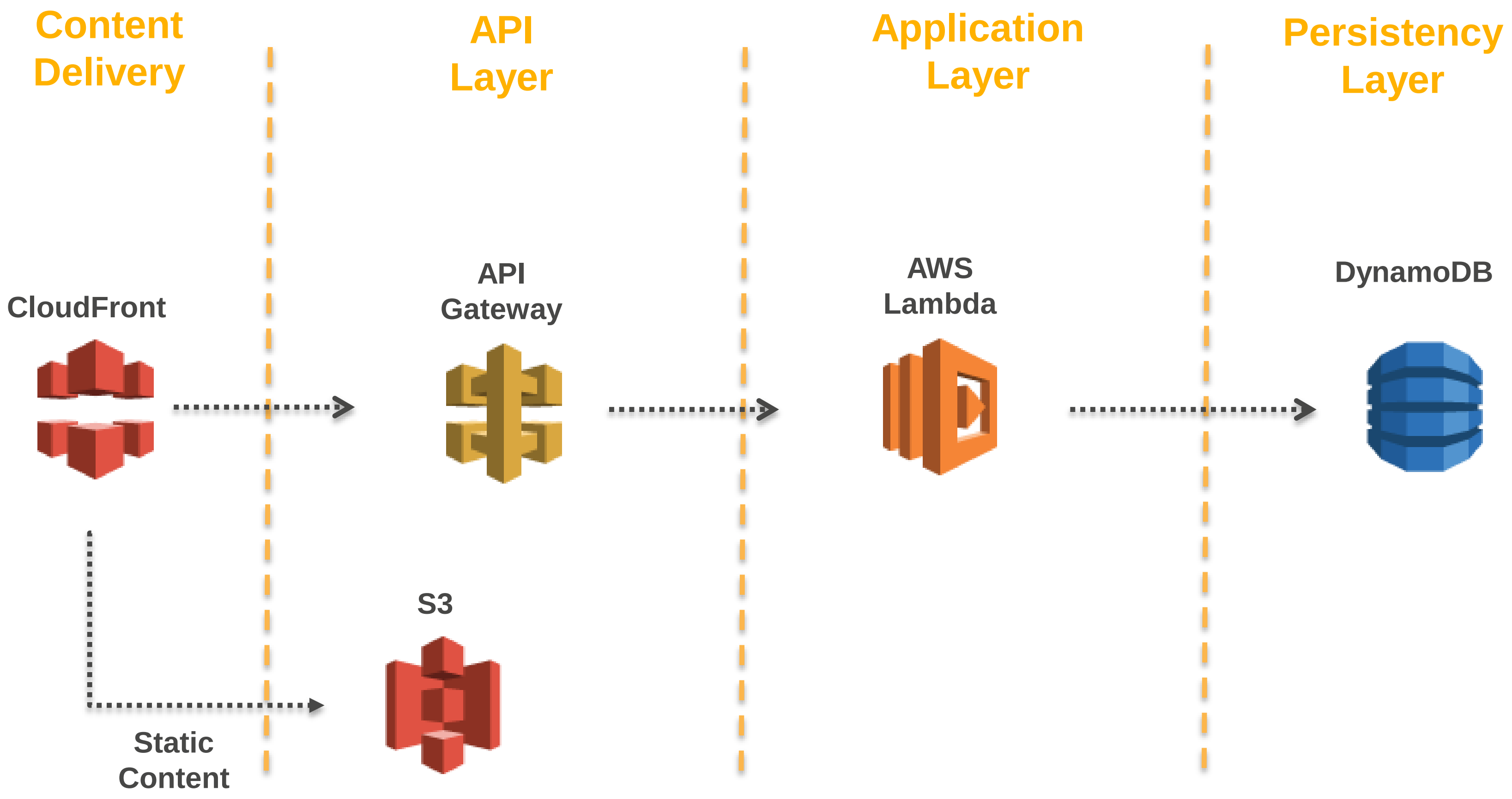
A Typical Microservice Architecture on AWS



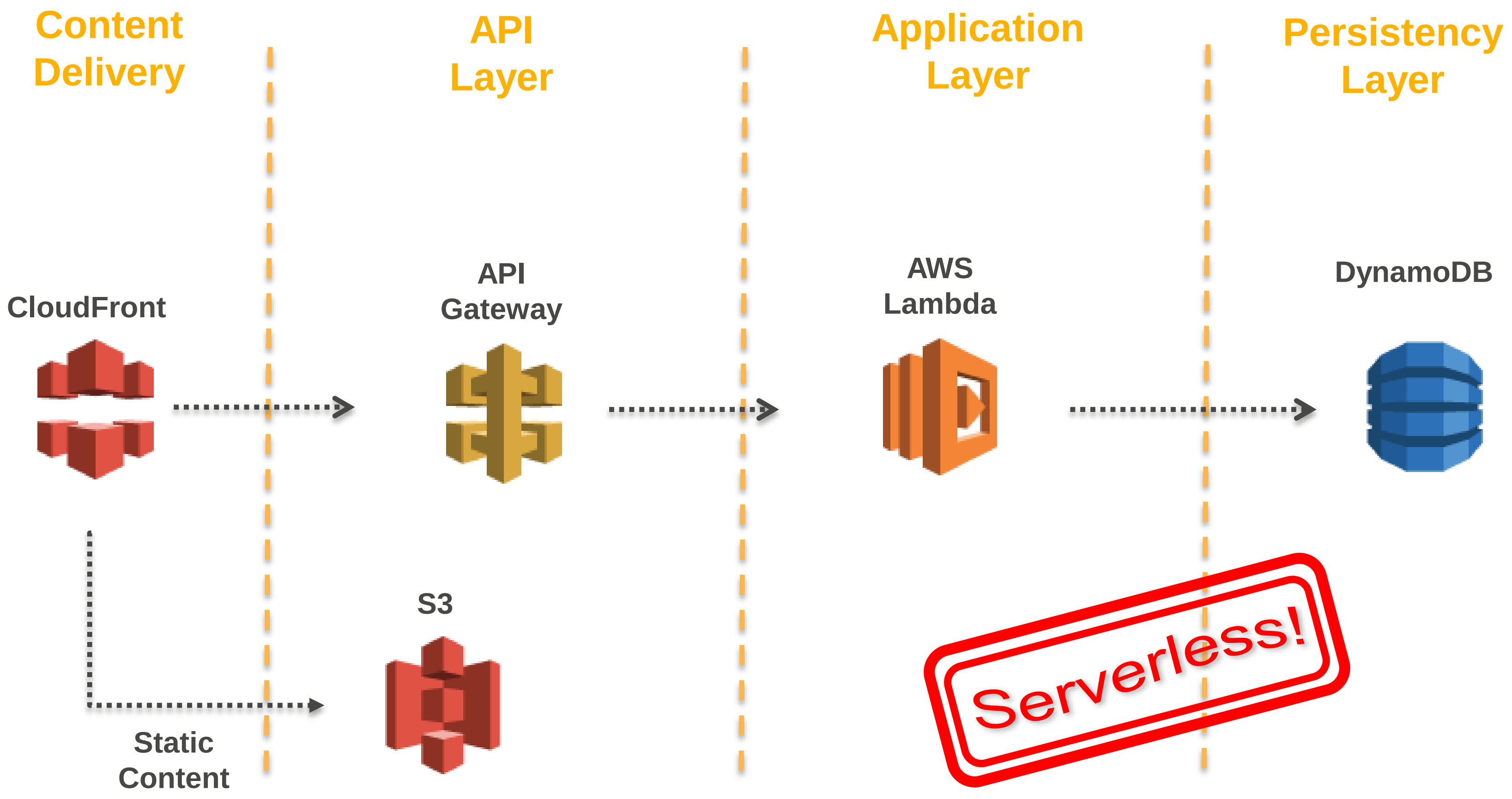
A Typical Microservice Architecture on AWS



A Typical Microservice Architecture on AWS



A Typical Microservice Architecture on AWS



Docker with ECR & ECS - Demo

AWS

SUMMIT

amazon web services