Introduction to Microservices

Peter Dalbhanjan, Solutions Architect, AWS

10 min	Evolution from Monoliths to Microservices
10 min	Core Principle of Microservices Approaches to building microservices on AWS
15 min	Demo: Let's build a microservice!
15 min	Other Architectural Principles
5 min	Additional resources

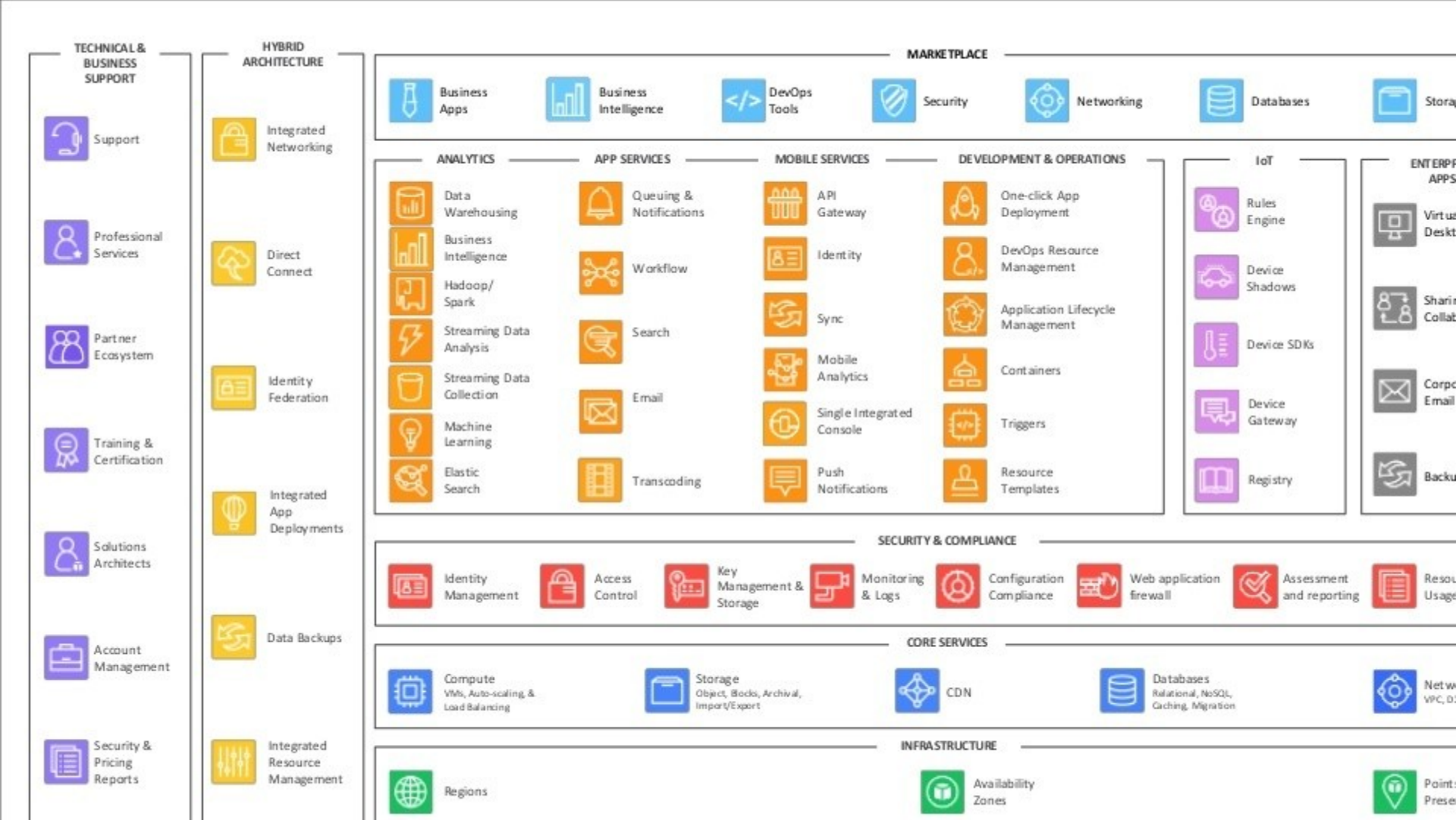
10 min	**Evolution from Monoliths to Microservices**
10 min	**Core Principle of Microservices** **Approaches to building microservices on AWS**
15 min	**Demo: Let's build a microservice!**
15 min	**Other Architectural Principles**
5 min	**Additional resources**

10 min	Evolution from Monoliths to Microservices
10 min	Core Principle of Microservices Approaches to building microservices on AWS
15 min	Demo: Let's build a microservice!
15 min	Other Architectural Principles
5 min	Additional resources

10 min	Evolution from Monoliths to Microservices
10 min	Core Principle of Microservices Approaches to building microservices on AWS
15 min	Demo: Let's build a microservice!
15 min	Other Architectural Principles
5 min	Additional resources

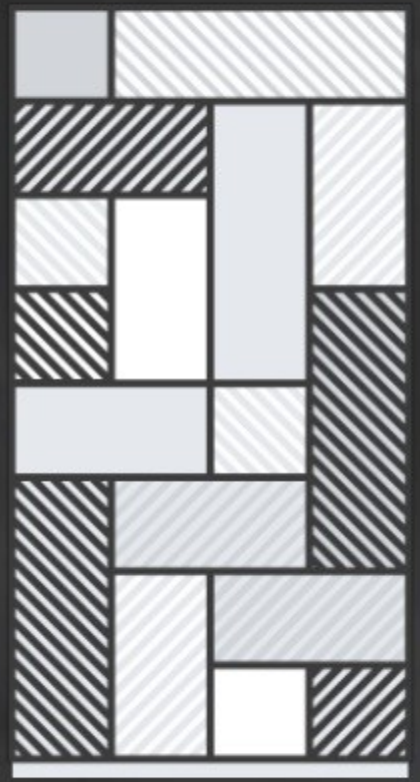
10 min	Evolution from Monoliths to Microservices
10 min	Core Principle of Microservices Approaches to building microservices on AWS
15 min	Demo: Let's build a microservice!
15 min	Other Architectural Principles
5 min	Additional resources

10 min	Evolution from Monoliths to Microservices
10 min	Core Principle of Microservices Approaches to building microservices on AWS
15 min	Demo: Let's build a microservice!
15 min	Other Architectural Principles
5 min	Additional resources


TECHNICAL & BUSINESS SUPPORT
Support
Professional Services
Partner Ecosystem
Training & Certification
Solutions Architects
Account Management
Security & Pricing Reports
HYBRID ARCHITECTURE
Integrated Networking
Direct Connect
Identity Federation
Integrated App Deployments
Data Backups
Integrated Resource Management
MARKETPLACE
Business Apps
Business Intelligence
DevOps Tools
Security
Networking
Databases
ANALYTICS
Data Warehousing
Business Intelligence
Hadoop/ Spark
Streaming Data Analysis
Streaming Data Collection
Machine Learning
Elastic Search
APP SERVICES
Queuing & Notifications
Workflow
Search
Email
Transcoding
MOBILE SERVICES
API Gateway
Identity
Sync
Mobile Analytics
Single Integrated Console
Push Notifications
DEVELOPMENT & OPERATIONS
One-click App Deployment
DevOps Resource Management
Application Lifecycle Management
Containers
Triggers
Resource Templates
IoT
Rules Engine
Device Shadows
Device SDKs
Device Gateway
Registry
SECURITY & COMPLIANCE
Identity Management
Access Control
Key Management & Storage
Monitoring & Logs
Configuration Compliance
Web application firewall
Assessment and reporting
CORE SERVICES
Compute
VMs, Auto-scaling & Load Balancing
Storage
Object, Blocks, Archival, Import/Export
CDN
Databases
Relational, NoSQL, Caching, Migration
INFRASTRUCTURE
Regions
Availability Zones


722
280
82
48
2009
2011
2013
2015

“The Monolith”



Challenges with monolithic software

Difficult to scale

Long Build/Test/Release Cycles (who broke the build?)

Operations is a nightmare (module X is failing, who's the owner?)

Architecture is hard to maintain and evolve

New releases take months

Long time to add new features

Lack of agility

Lack of innovation

Frustrated customers

Challenges with monolithic software

Difficult to scale

Architecture is hard to maintain and evolve

Lack of agility

Long Build/Test/Release Cycles (who broke the build?)

New releases take months

Lack of innovation

Operations is a nightmare (module X is failing, who's the owner?)

Long time to add new features

Frustrated customers

Challenges with monolithic software

Difficult to scale

Architecture is hard to maintain and evolve

Lack of agility

Long Build/Test/Release Cycles (who broke the build?)

New releases take months

Lack of innovation

Operations is a nightmare (module X is failing, who's the owner?)

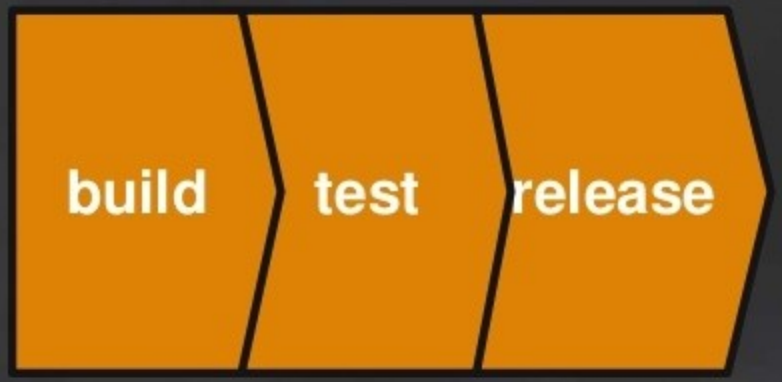
Long time to add new features

Frustrated customers

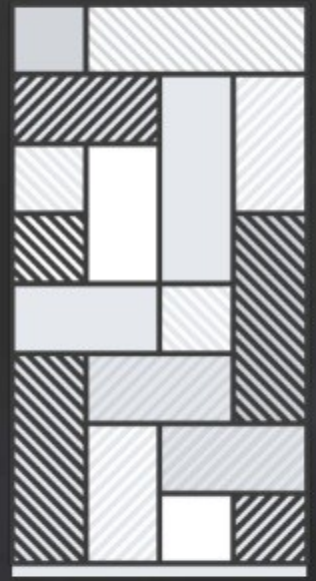
Monolith development lifecycle

developers

delivery pipeline

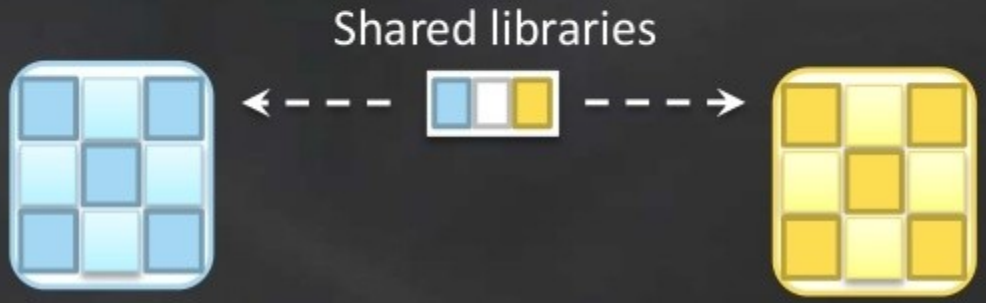


app
(aka the“monolith

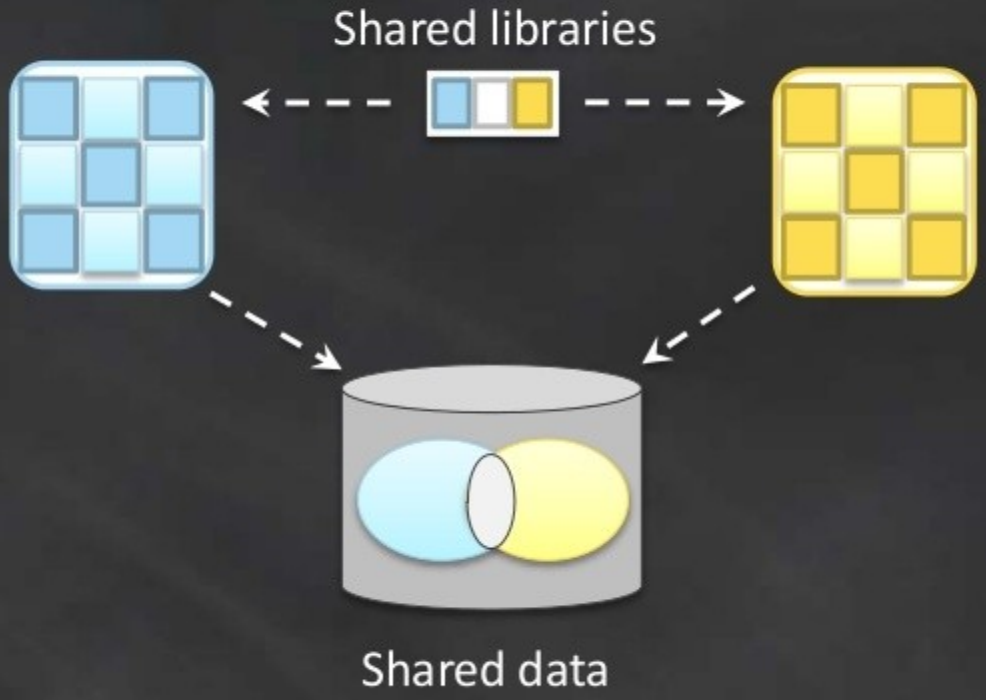


Too much software coupling

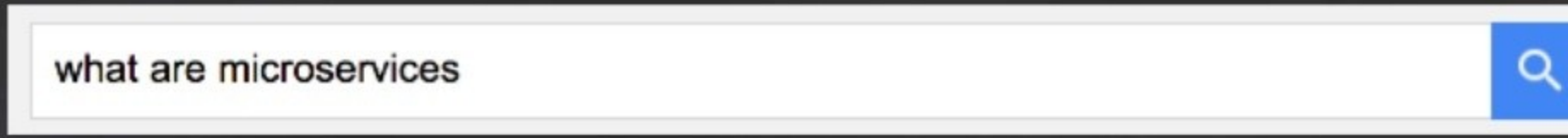
Too much software coupling



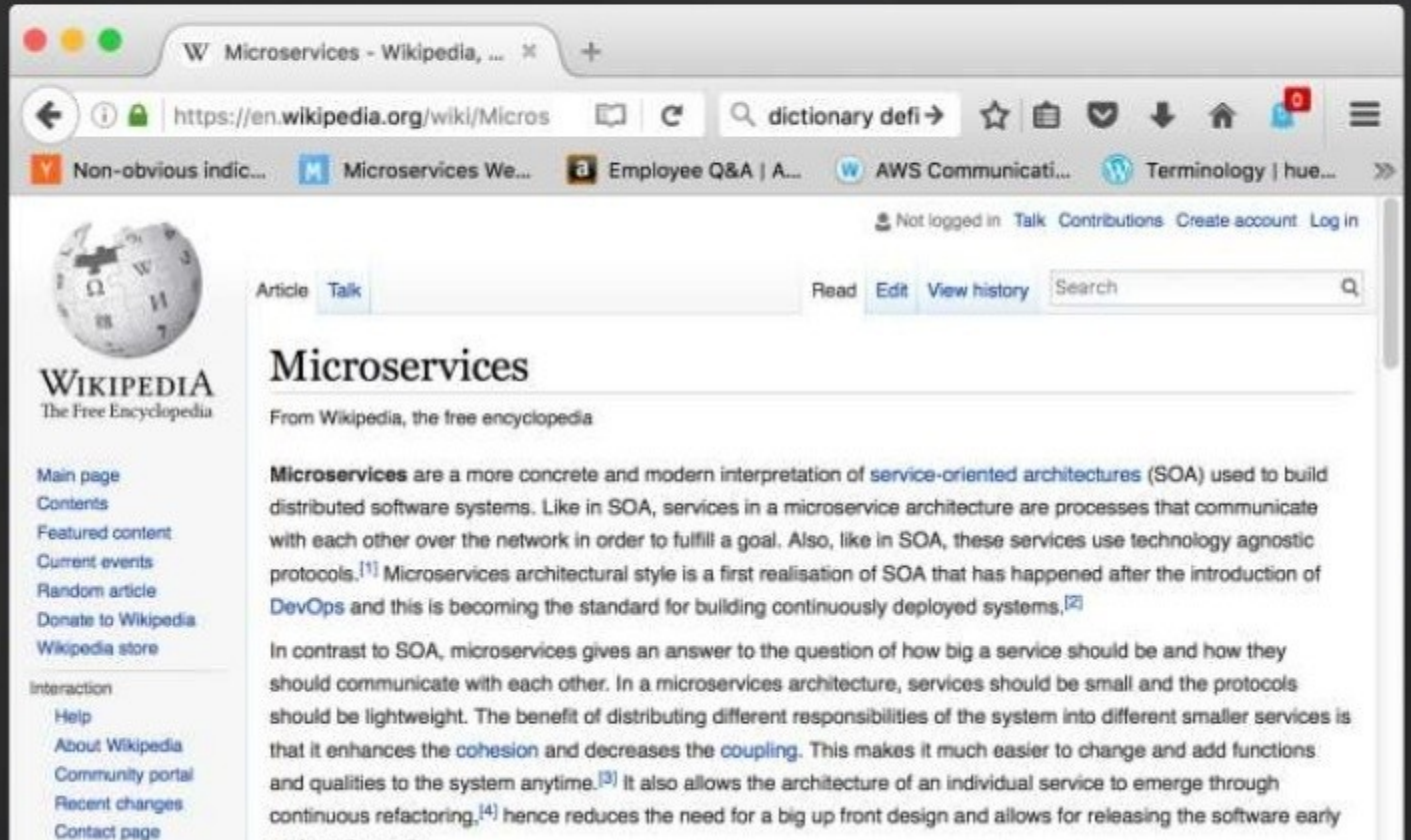
Too much software coupling




CHANGE
For The Better


what are microservices

what are microservices



"service-oriented architecture composed of loosely coupled elements that have bounded contexts"

Adrian Cockcroft (former Cloud Architect at Netflix, now Technology Fellow at Battery Ventures)

"service-oriented architecture composed of loosely coupled elements that have bounded contexts"

Services communicate wit
each other over the
network

Adrian Cockcroft (former Cloud Architect at Netflix, now Technology Fellow at Battery Ventures)

"service-oriented architecture composed of loosely coupled elements that have bounded contexts"

Adrian Cockcroft (former Cloud Architect at Netflix, now Technology Fellow at Battery Ventures)

You can update the servic
independently; updating
one service doesn't requir
changing any other servic

"service-oriented architecture composed of loosely coupled elements that have bounded contexts"

Adrian Cockcroft (former Cloud Architect at Netflix, now Technology Fellow at Battery Ventures)

Self-contained; you can update the code without knowing anything about t internals of other microservices

“Do one thing, and do it well”

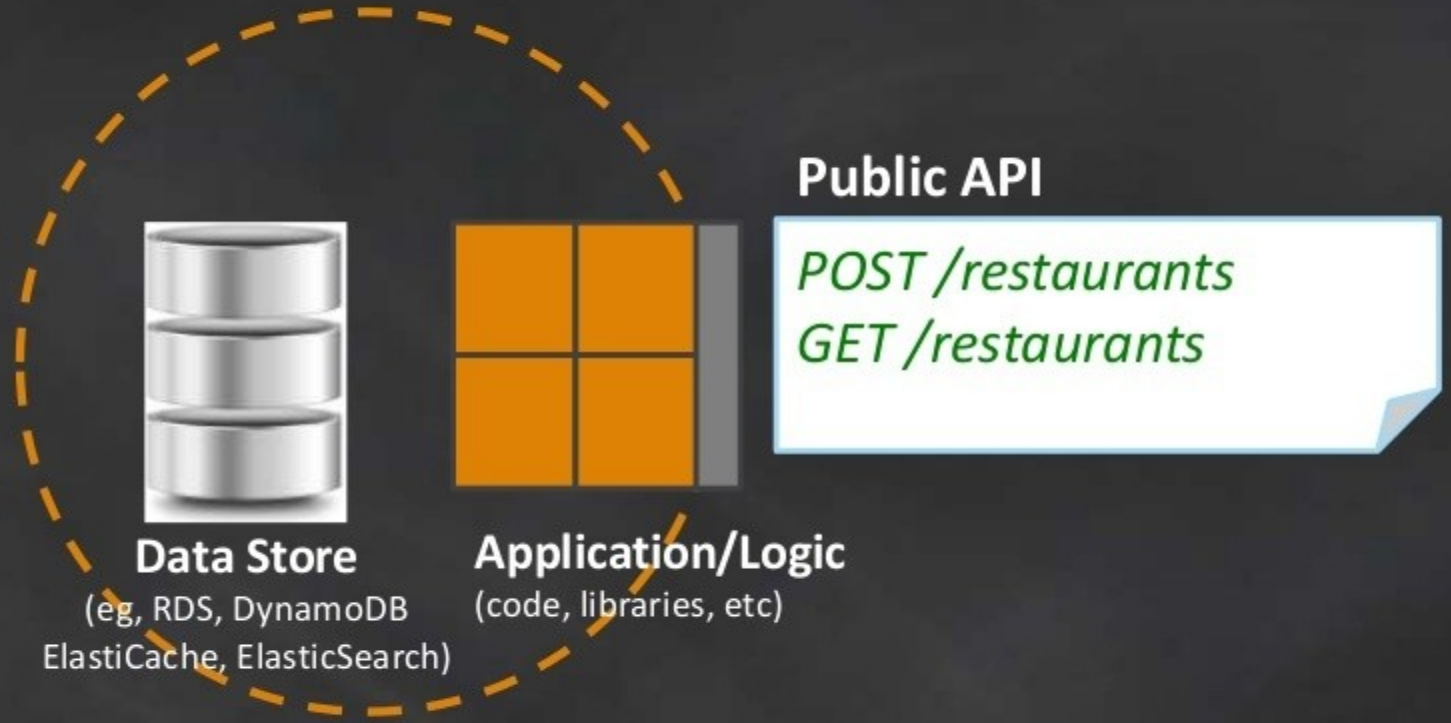
“Do one thing, and do it well”

Anatomy of a Micro-service

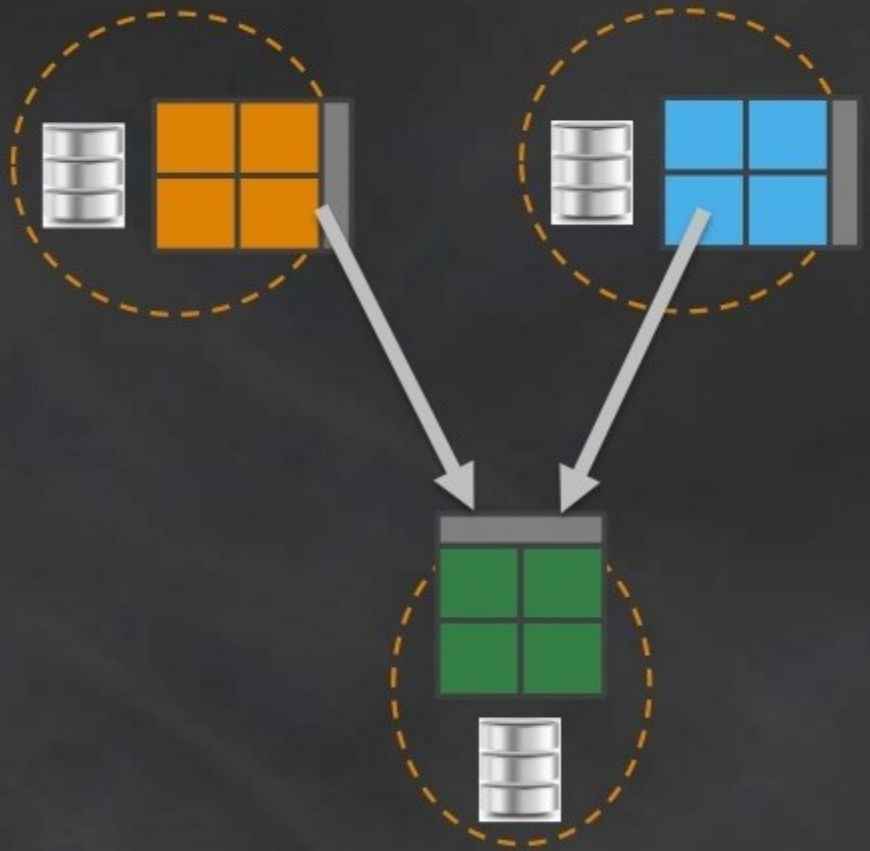
Anatomy of a Micro-service

Anatomy of a Micro-service

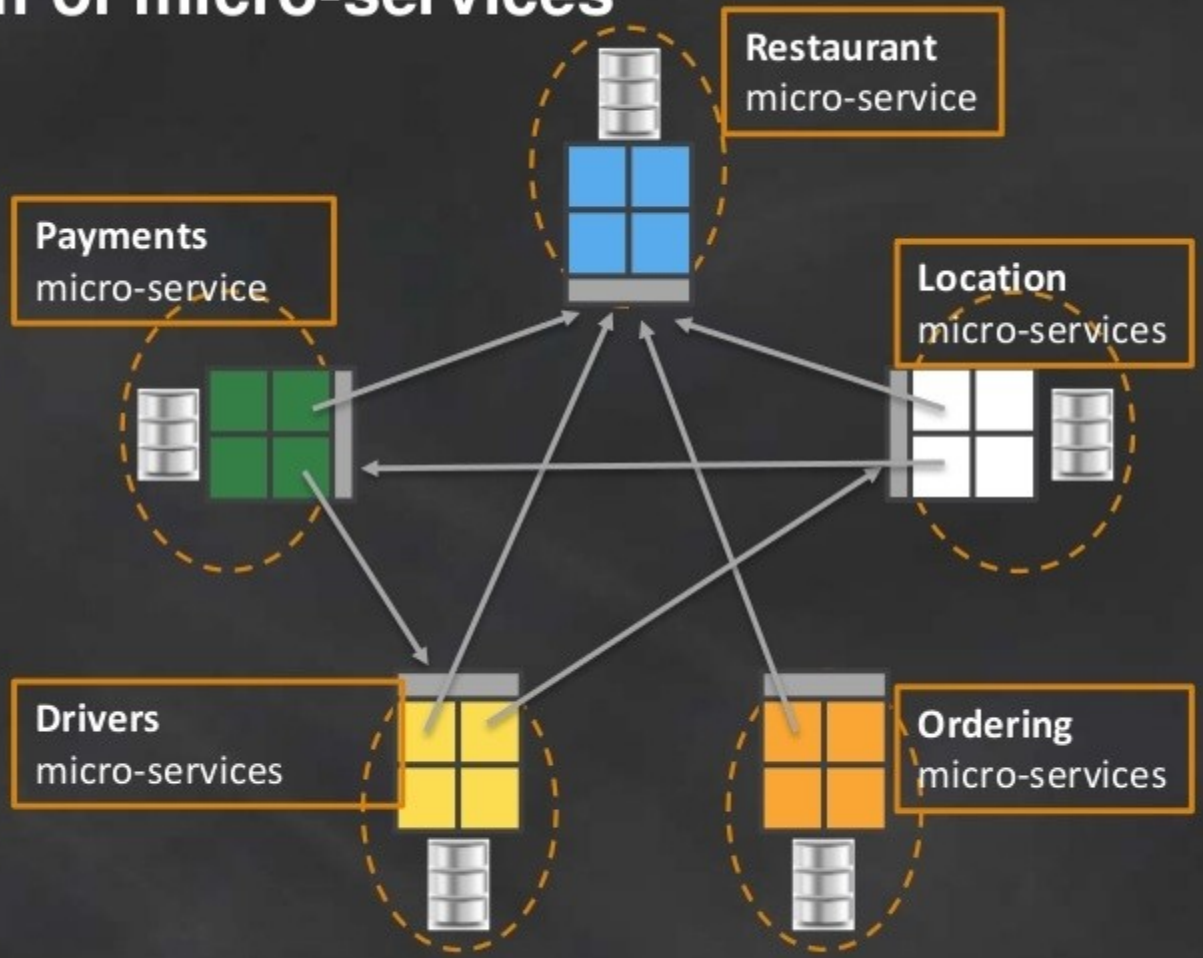
Anatomy of a Micro-service

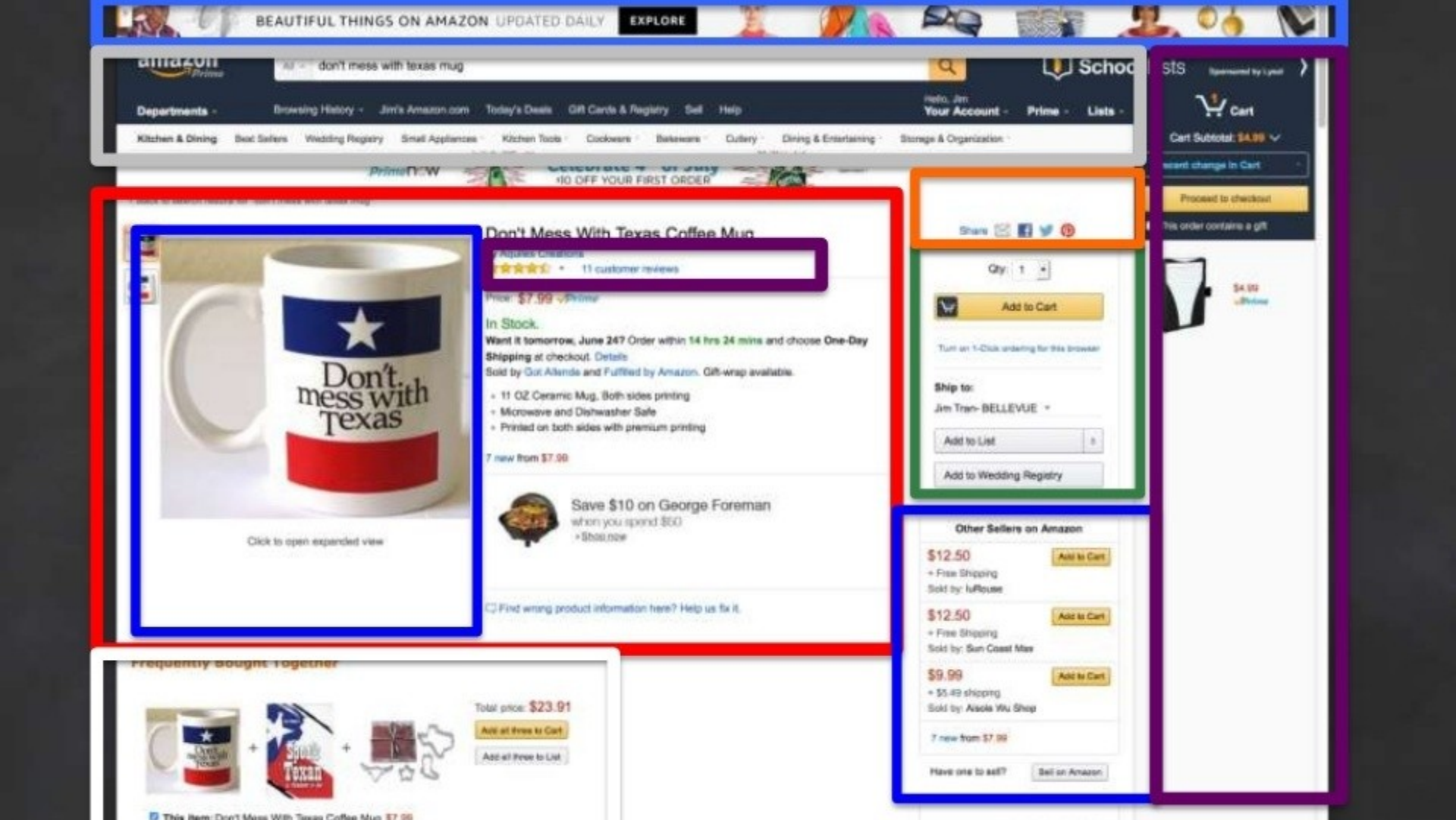


Avoid Software Coupling



Ecosystem of micro-services




BEAUTIFUL THINGS ON AMAZON UPDATED DAILY
EXPLORE
don't mess with texas mug
Departments
Browsing History
Jim's Amazon.com
Today's Deals
Gift Cards & Registry
Sell
Help
Your Account
Prime
Lists
Kitchen & Dining
Best Sellers
Wedding Registry
Small Appliances
Kitchen Tools
Cookware
Bakeware
Cutlery
Dining & Entertaining
Storage & Organization
Don't Mess With Texas Coffee Mug
11 customer reviews
Price: $7.99
In Stock.
Want it tomorrow, June 24? Order within 14 hrs 24 mins and choose One-Day Shipping at checkout. Details
Sold by Got Allanda and Fulfilled by Amazon. Gift-wrap available.
11 OZ Ceramic Mug, Both sides printing
Microwave and Dishwasher Safe
Printed on both sides with premium printing
7 new from $7.99
Don't mess with Texas
Click to open expanded view
Save $10 on George Foreman
when you spend $50
Shop now
Find wrong product information here? Help us fix it.
Add to Cart
Add to List
Add to Wedding Registry
Ship to:
Jim Tran- BELLEVUE
Other Sellers on Amazon
$12.50
+ Free Shipping
Sold by: lulRouse
$12.50
+ Free Shipping
Sold by: Sun Coast Max
$9.99
+ $5.49 shipping
Sold by: Alaola Wu Shop
7 new from $7.99
Have one to sell?
Sell on Amazon
Frequently Bought Together
Total price: $23.91
Add all three to Cart
Add all three to List
Cart
Cart Subtotal:
Proceed to checkout

‹ Back to search results for "don't mess with texas mug"

Click to open expanded view

Don't Mess With Texas Coffee Mug

by Aquiles Creations

★★★★½ • 11 customer reviews

Price: $7.99 ✓Prime

In Stock.

Want it tomorrow, June 24? Order within 14 hrs 24 mins and choose **One-Day Shipping** at checkout. Details

Sold by Gol Allende and Fulfilled by Amazon. Gift-wrap available.

- 11 OZ Ceramic Mug, Both sides printing
- Microwave and Dishwasher Safe
- Printed on both sides with premium printing

7 new from $7.99

Find wrong product information here? Help us fix it.

Share

Qty: 1

Add to Cart

Turn on 1-Click ordering for this browser

Ship to:
Jim Tran- BELLEVUE

Add to List

Add to Wedding Registry

Other Sellers on Amazon

$12.50 Add to Cart
+ Free Shipping
Sold by: luRouse

$12.50 Add to Cart
+ Free Shipping
Sold by: Sun Coast Max

$9.99 Add to Cart
+ $5.49 shipping
Sold by: Alaola Wu Shop

7 new from $7.99

Have one to sell? Sell on Amazon

Frequently Bought Together

Total price: $23.91

Add all three to Cart

Add all three to List

This item: Don't Mess With Texas Coffee Mug $7.99

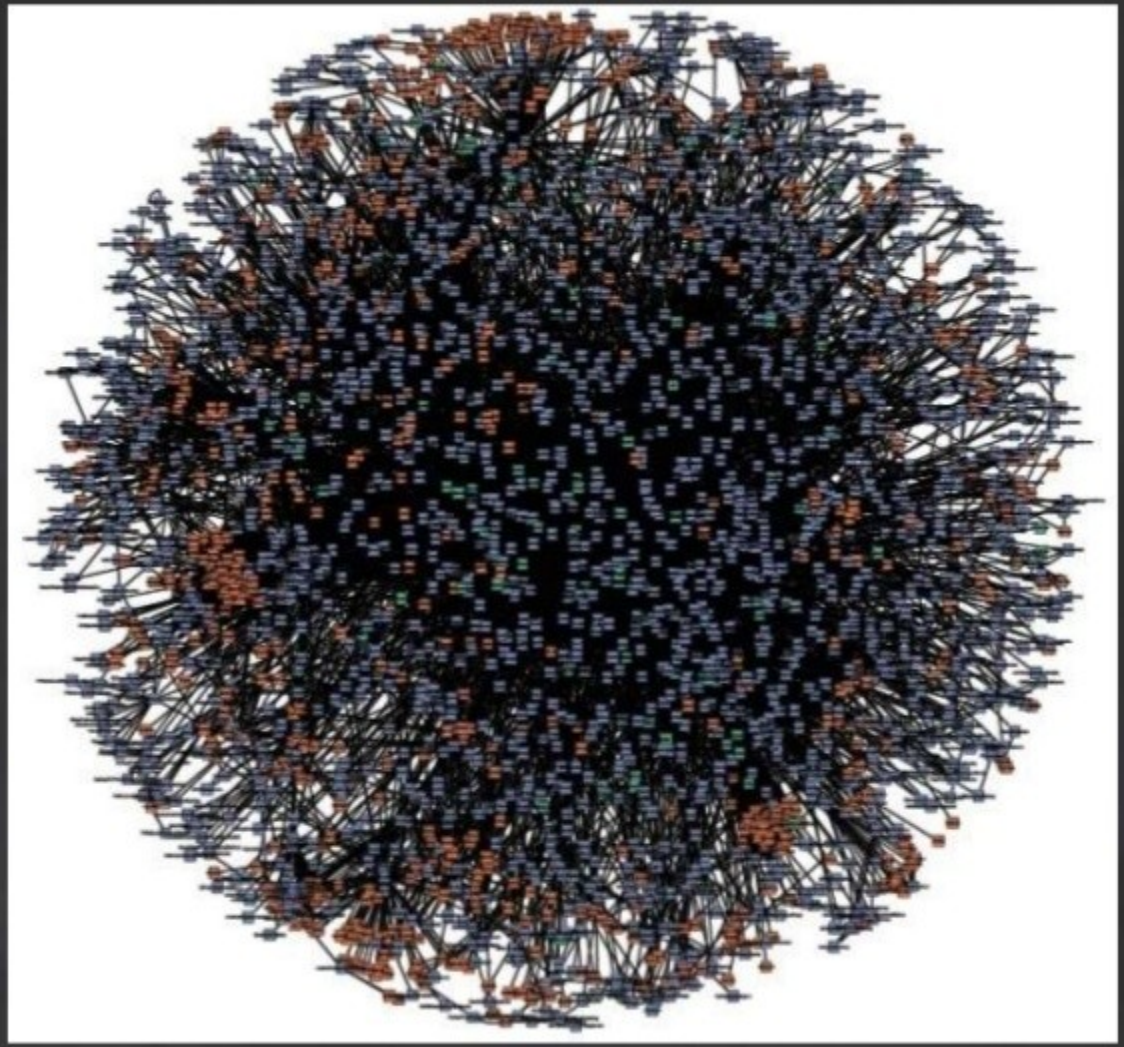
Cart Subtotal: $4.99

1 recent change in Cart

Proceed to checkout

This order contains a gift

$4.99 ✓Prime



Thousands of teams

× Microservice architecture

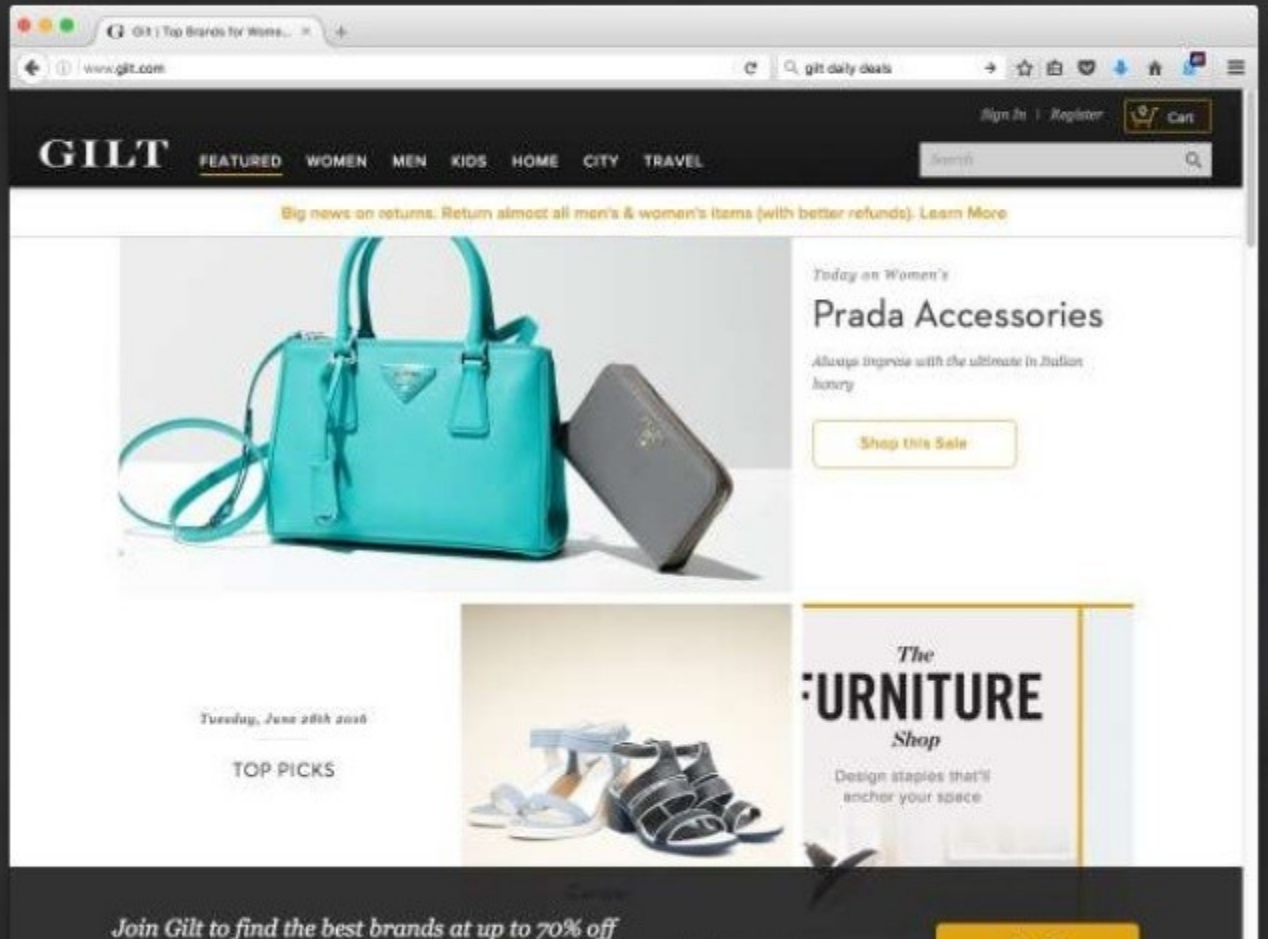
× Continuous delivery

× Multiple environments

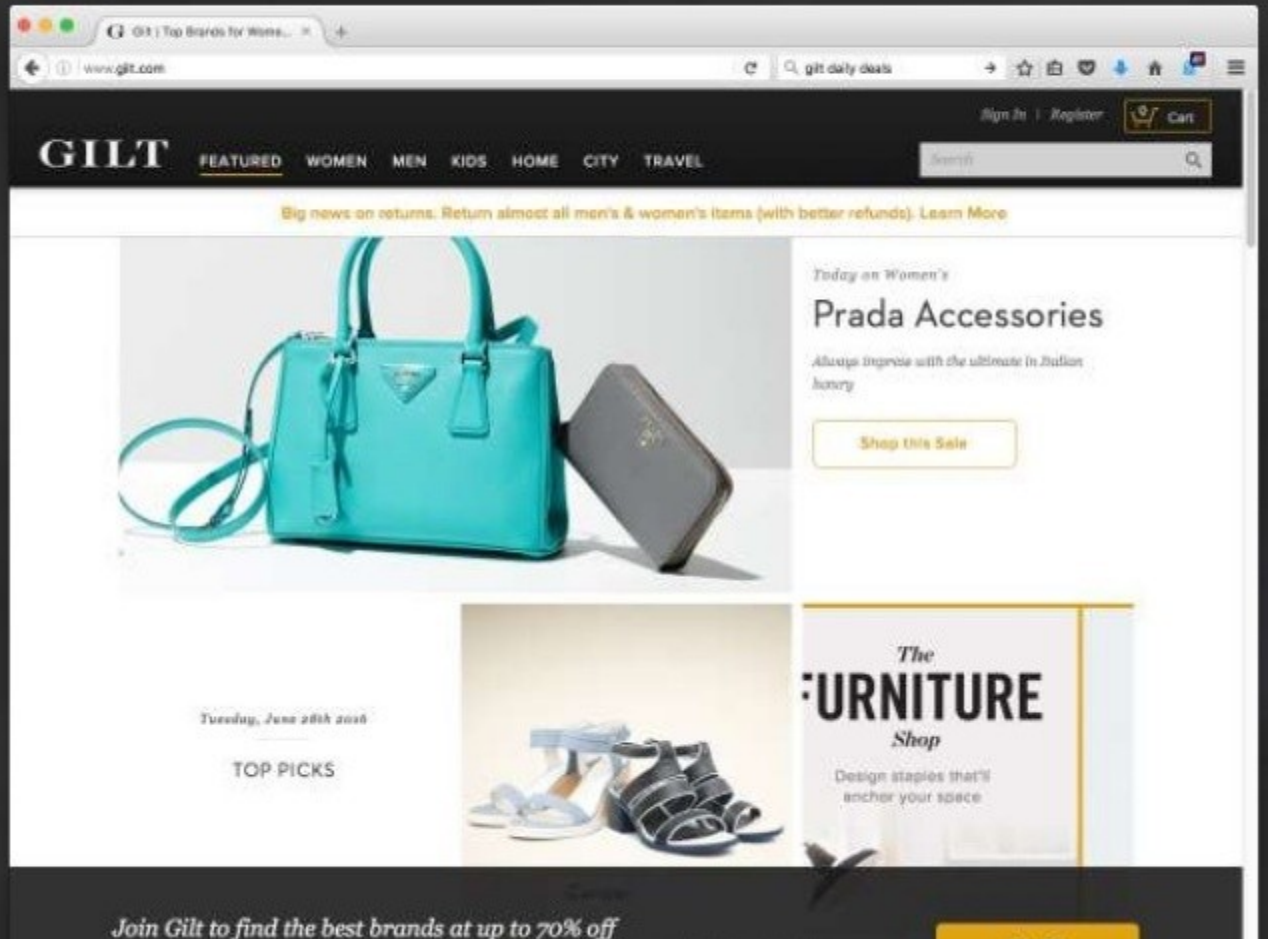
= 50 million deployments a year

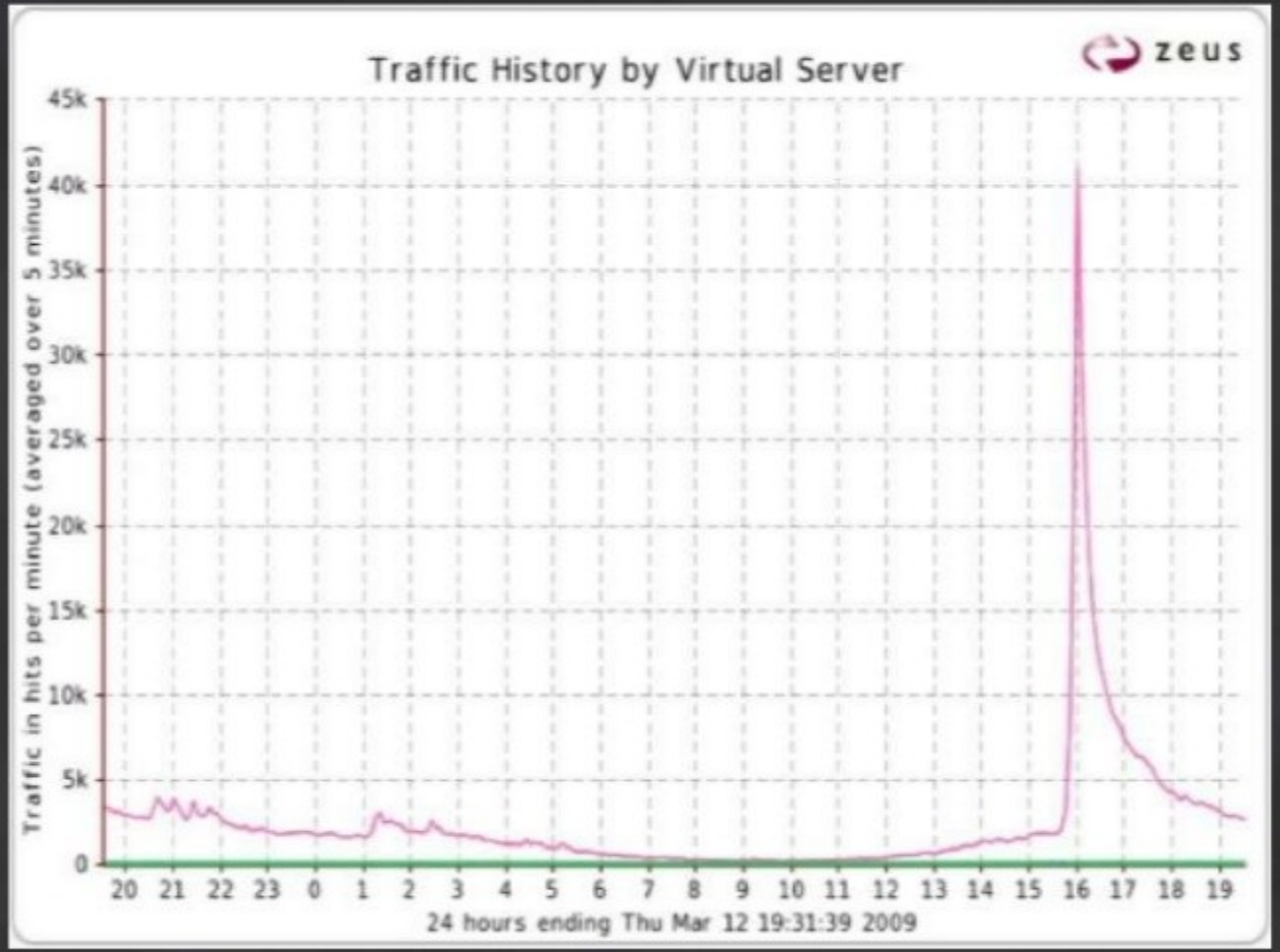
(5708 per hour, or every 0.63 second)

Gilt: Luxury designer brands at members-only prices



... Sale every day at noon EST




Traffic History by Virtual Server
zeus
Traffic in hits per minute (averaged over 5 minutes)
45k
40k
35k
30k
25k
20k
15k
10k
5k
0
20 21 22 23 0 1 2 3 4 5 6 7 8 9 10 11 12 13 14 15 16 17 18 19
24 hours ending Thu Mar 12 19:31:39 2009

AWS re:Invent

CONSUMER DIGITAL TECH

ARC 308

Nike's Journey to Microservices

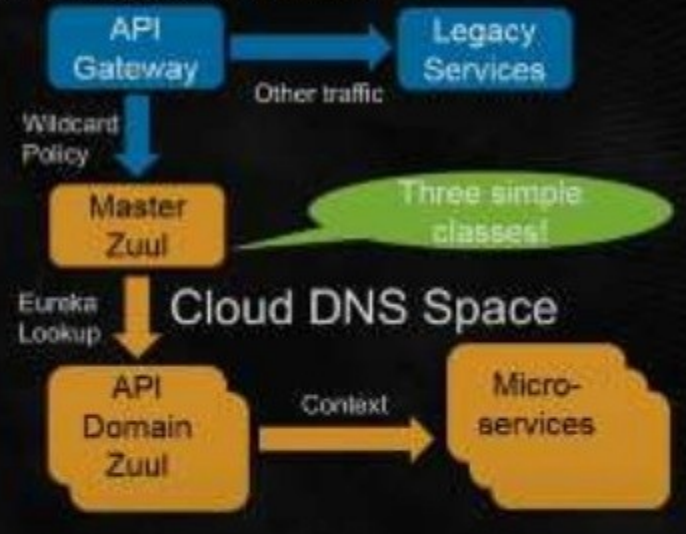
Jason Robey, Consumer Digital Technology (CDT), Nike, Inc.

November 12, 2014 | Las Vegas, NV

Getting to production fast

https://api.nike.com/plusv3/domain/context

- AMI
- AWS Console
- Conformity Monkey
- Asgard
- Ribbon
- Eureka
- Archaius
- Servo
- ASG

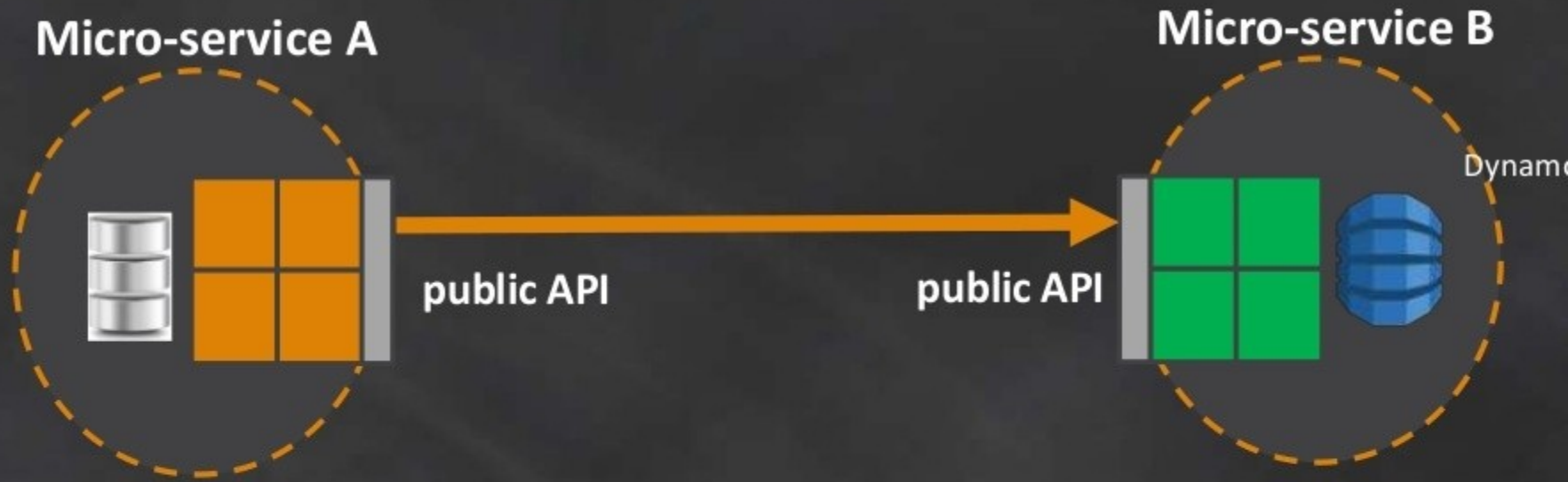


Principle 1

Micro-services only rely on each other's public AP

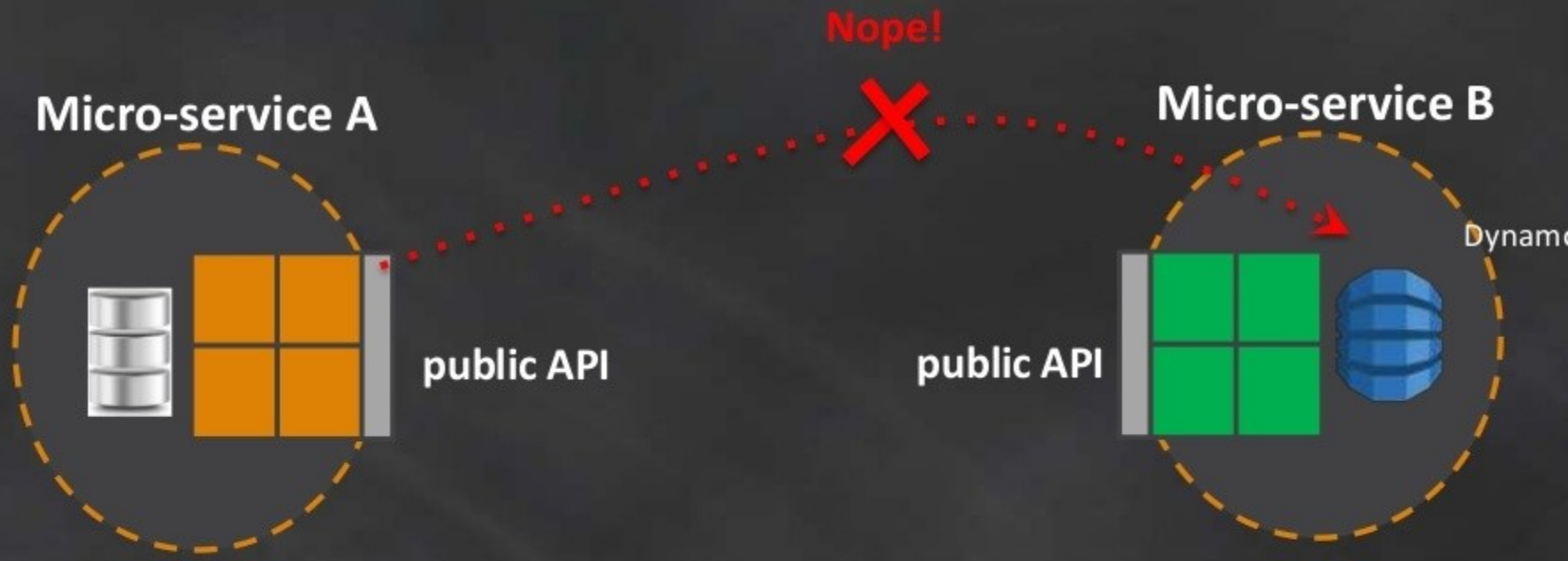
"Contracts" by NobMouse. No alterations other tha
https://www.flickr.com/photos/nobmouse/40

Principle 1: Microservices only rely on each other's public API

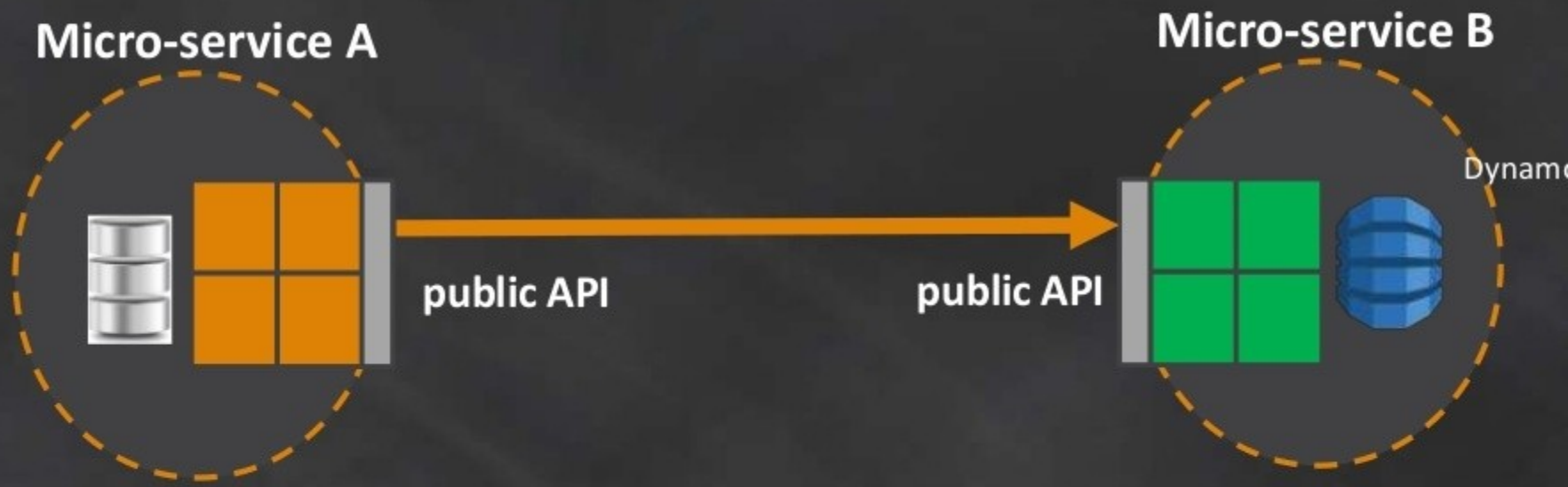


Principle 1: Microservices only rely on each other's public API
(Hide Your Data)

Principle 1: Microservices only rely on each other's public API
(Hide Your Data)



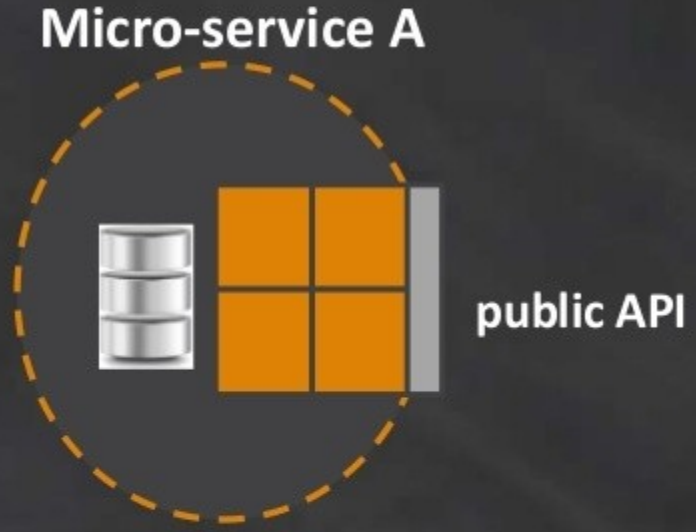
Principle 1: Microservices only rely on each other's public API
(Hide Your Data)



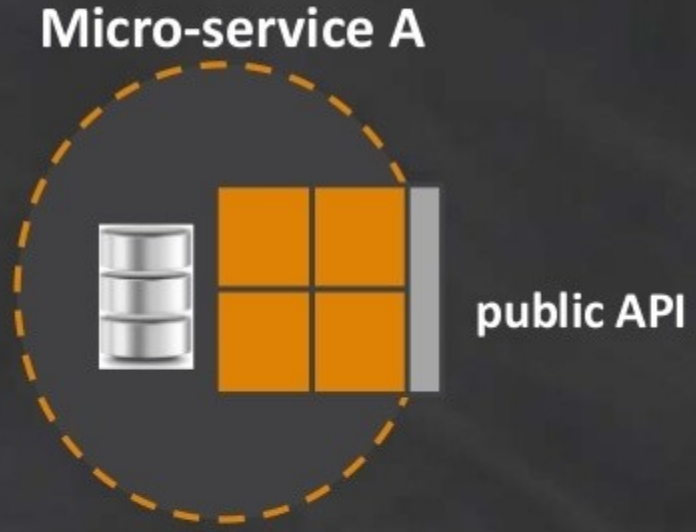
Principle 1: Microservices only rely on each other's public API (Evolve API in backward-compatible way…and document!)

Version 1.0.0

storeRestaurant (*id, name, cuisine*)



Principle 1: Microservices only rely on each other's public API (Evolve API in backward-compatible way…and document!)



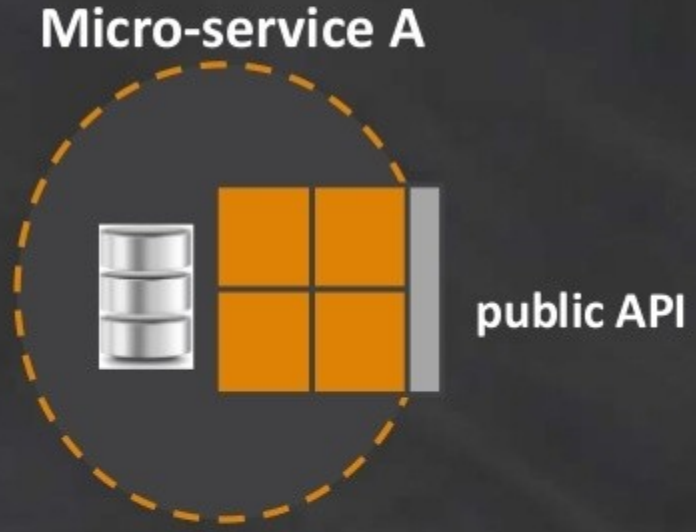
Version 1.0.0

storeRestaurant (*id, name, cuisine*)

Version 1.1.0

storeRestaurant (id, name, cuisine)
storeRestaurant (id, name, arbitrary_metadata)
addReview (restaurantId, rating, comments)

Principle 1: Microservices only rely on each other's public API (Evolve API in backward-compatible way…and document!)



Version 1.0.0

storeRestaurant (*id, name, cuisine*)

Version 1.1.0

storeRestaurant (id, name, cuisine)
storeRestaurant (id, name, arbitrary_metadata)
addReview (restaurantId, rating, comments)

Version 2.0.0

storeRestaurant (id, name, arbitrary_metadata)
addReview (restaurantId, rating, comments)

Principle 2

Use the right tool for the job

Principle 2: Use the right tool for the job
(Embrace polyglot persistence)

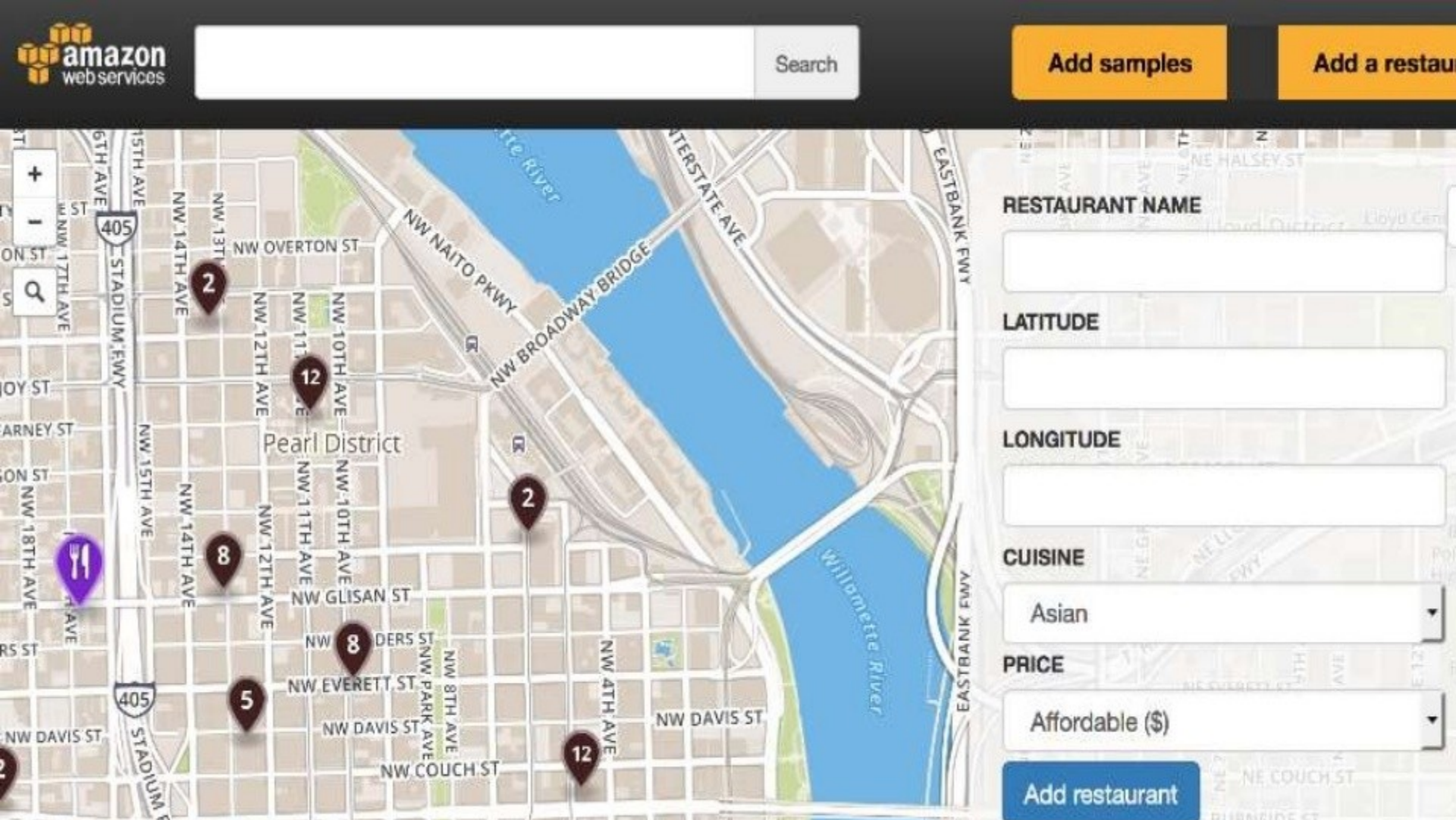
Principle 2: Use the right tool for the job
(Embrace polyglot persistence)

Principle 2: Use the right tool for the job
(Embrace polyglot persistence)

Principle 2: Use the right tool for the job
(Embrace polyglot programming frameworks)

Principle 2: Use the right tool for the job
(Embrace polyglot programming frameworks)

Let's build
a microservice!


amazon web services
Search
Add samples
Add a restau
RESTAURANT NAME
LATITUDE
LONGITUDE
CUISINE
Asian
PRICE
Affordable ($)
Add restaurant
Pearl District
NW NAITO PKWY
NW BROADWAY BRIDGE
INTERSTATE AVE
EASTBANK FWY
Willamette River
NW OVERTON ST
NW GLISAN ST
NW EVERETT ST
NW DAVIS ST
NW COUCH ST
NW 14TH AVE
NW 12TH AVE
NW 11TH AVE
NW 10TH AVE
NW 15TH AVE
NW 4TH AVE
NW PARK AVE
NW 8TH AVE
STADIUM FWY
405

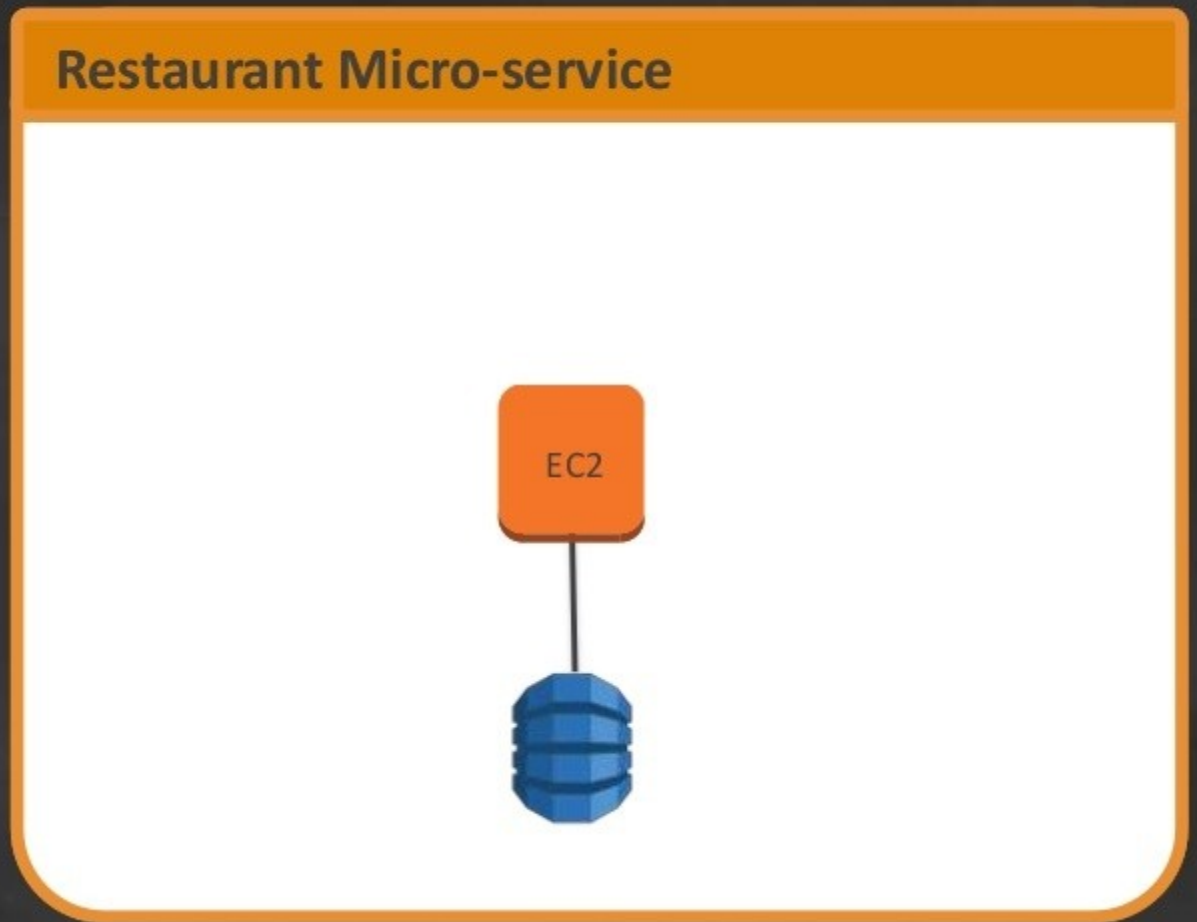

Restaurant Micro-service
POST /restaurants
GET /restaurants

Restaurant Micro-service

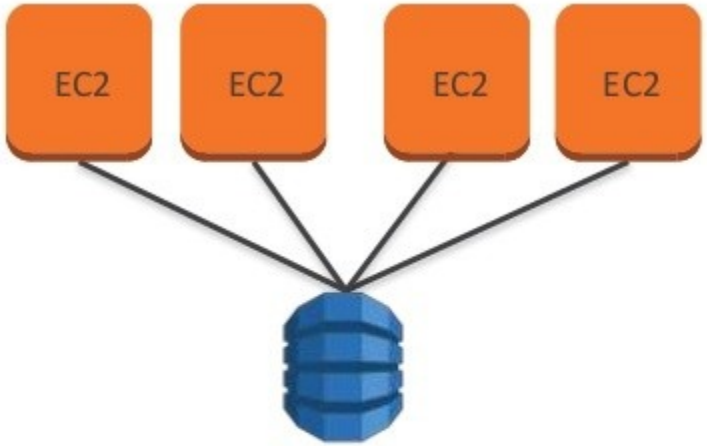
Approach #1

EC2

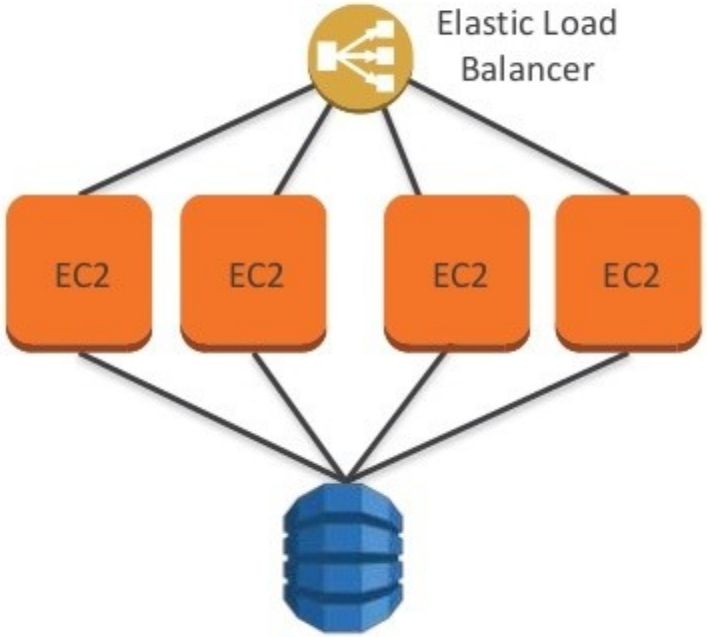

Restaurant Micro-service
EC2


Restaurant Micro-service
EC2

Restaurant Micro-service



Restaurant Micro-service

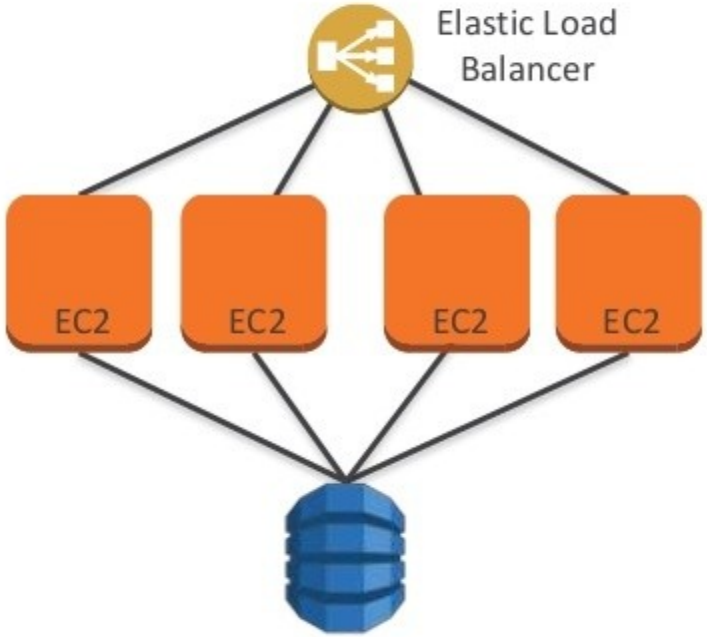


Restaurant Micro-service

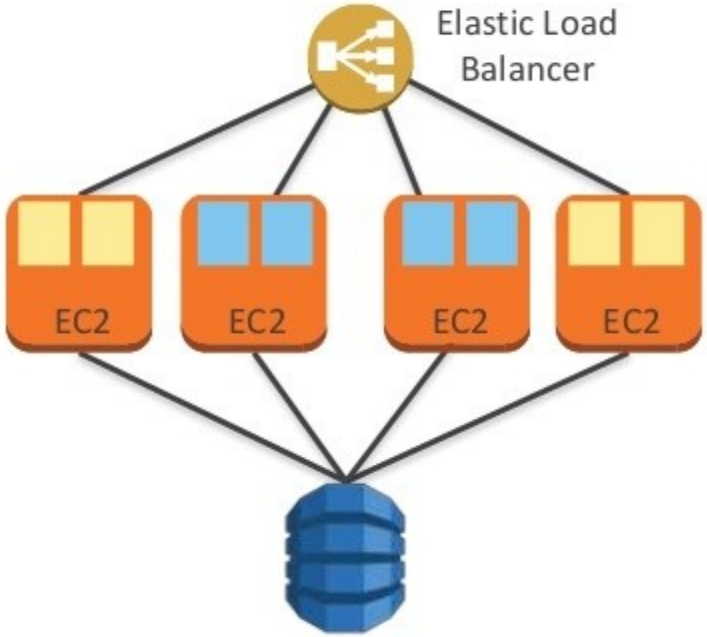
Approach #2

Containers Using ECS

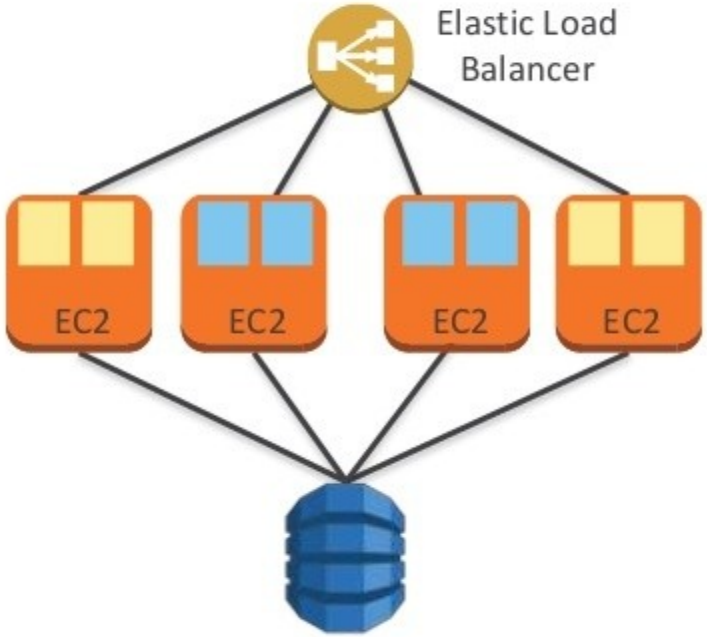
Restaurant Micro-service



Restaurant Micro-service



Restaurant Micro-service



Amazon EC2 Container Service (ECS) to manage containers

Coursera

13 million users from 190 countries
1,000 courses from 119 institutions

- Prototype in less than 2 months
- Deployment time: hours → minu
- Each team can now develop its respective applications independently

Restaurant Micro-service

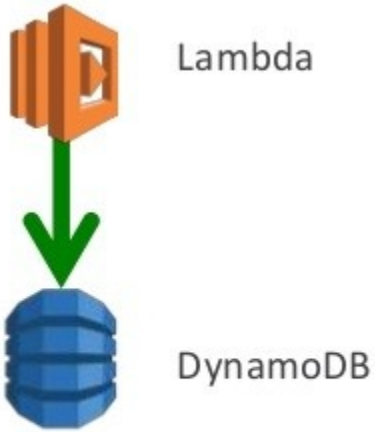
Approach #3

API Gateway + Lambda

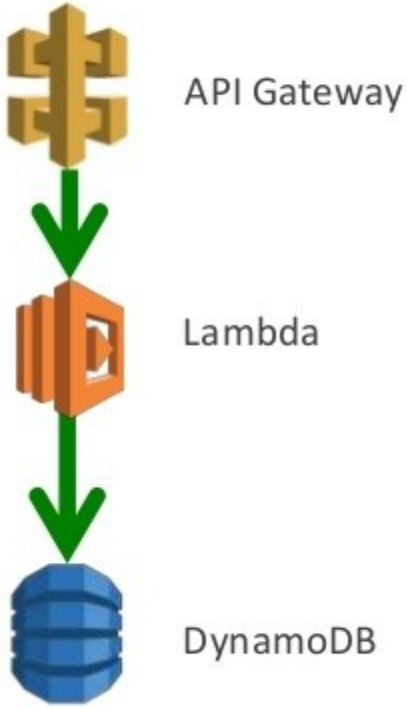
Restaurant Micro-service

DynamoDB

Restaurant Micro-service



Restaurant Micro-service

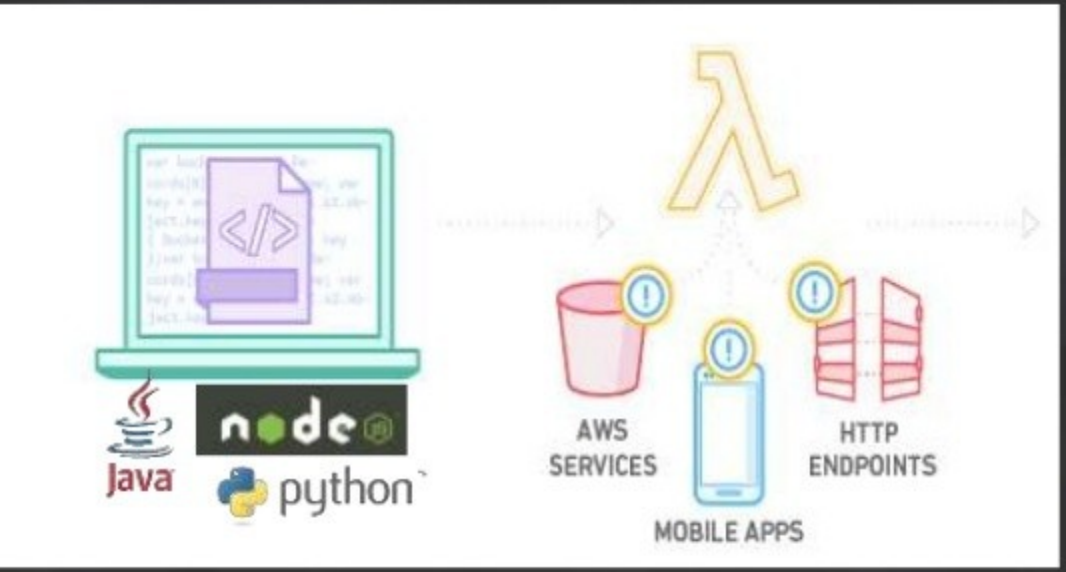


AWS Lambda

lets you run code

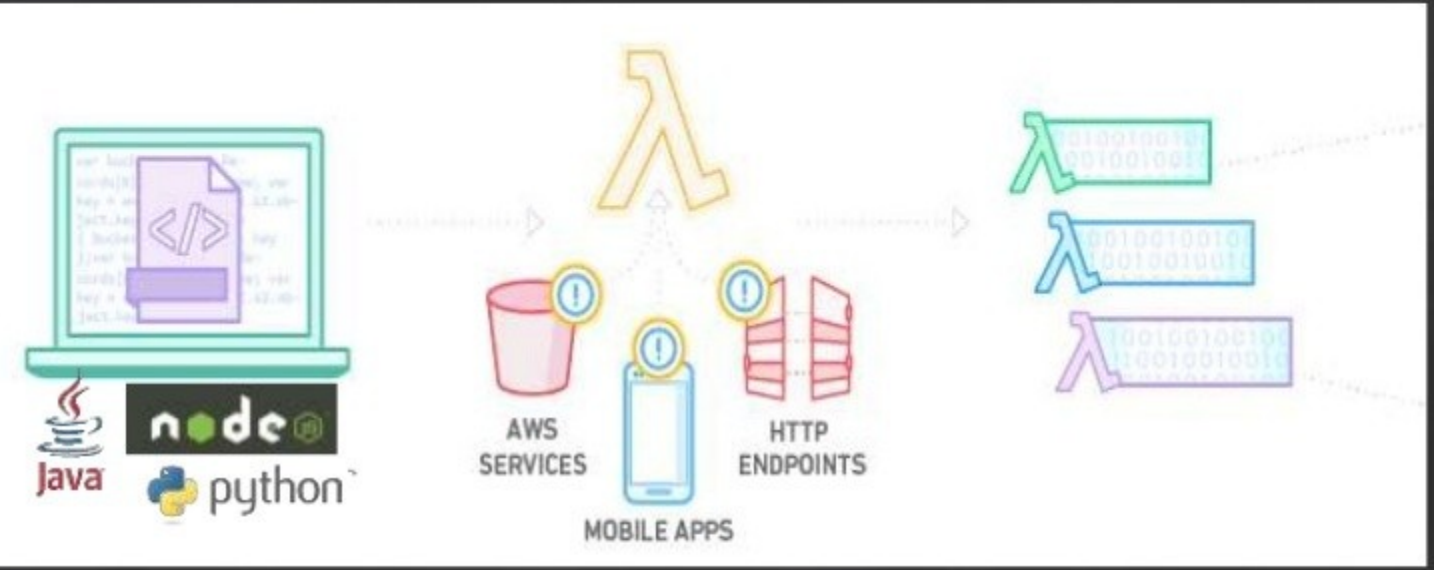
without managing servers

Upload your code (Java, JavaScript, Python)



Upload your code (Java, JavaScript, Python)

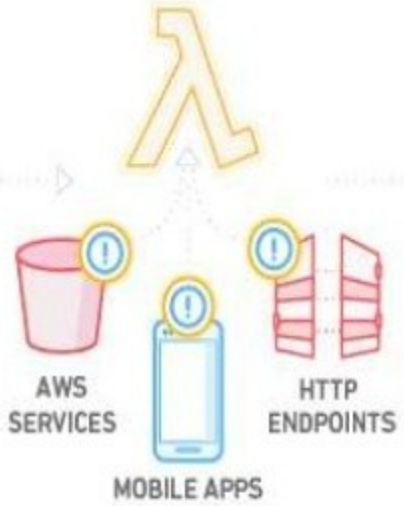
Set up your code to trigger from other AWS services, webservice calls, or app activity

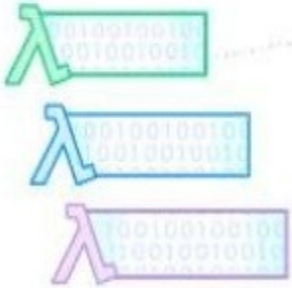


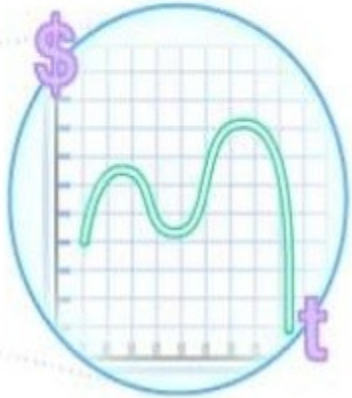
Upload your code (Java, JavaScript, Python)

Set up your code to trigger from other AWS services, webservice calls, or app activity

Lambda automatically scales







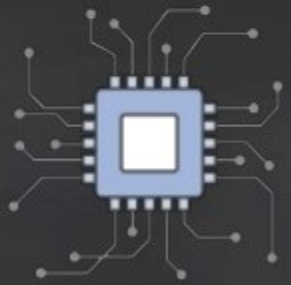
Upload your code (Java, JavaScript, Python)

Set up your code to trigger from other AWS services, webservice calls, or app activity

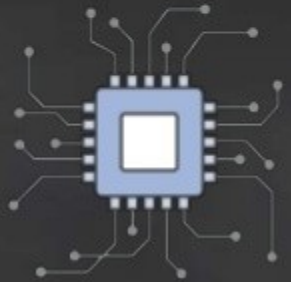
Lambda automatically scales

Pay for only th compute time you use (sub-second metering)

AWS API Gateway
is the easiest way to
deploy micro-services

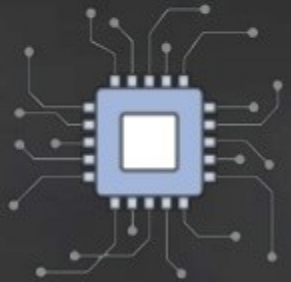


Create a unified API frontend for multiple micro-services



Create a unified API frontend for multiple micro-services

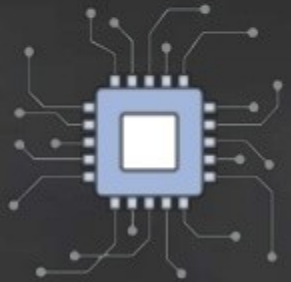
Authenticate and authorize requests



Create a unified API frontend for multiple micro-services

Authenticate and authorize requests

Handles DDoS protection and API throttling

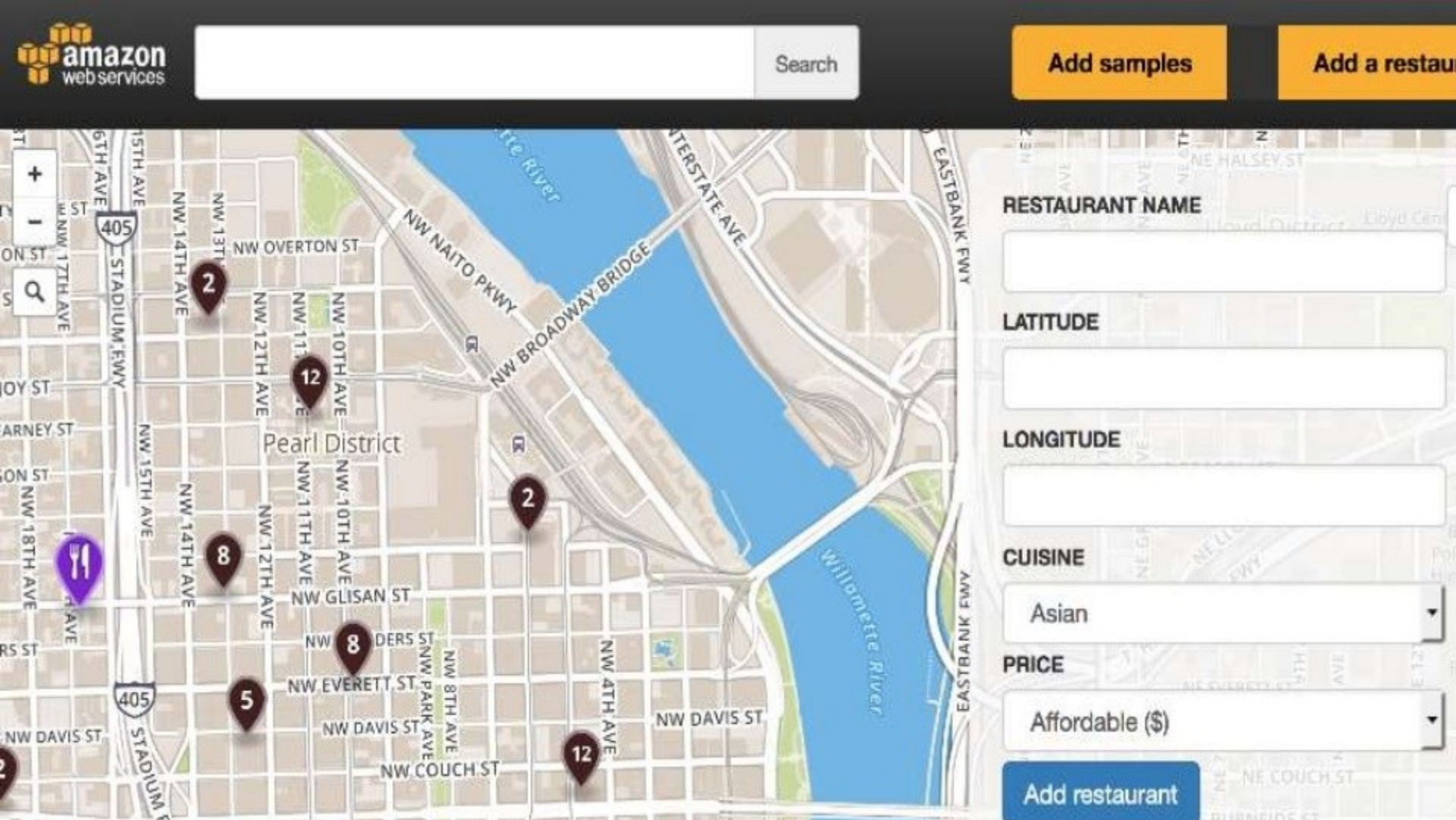


Create a unified API frontend for multiple micro-services

Authenticate and authorize requests

Handles DDoS protection and API throttling

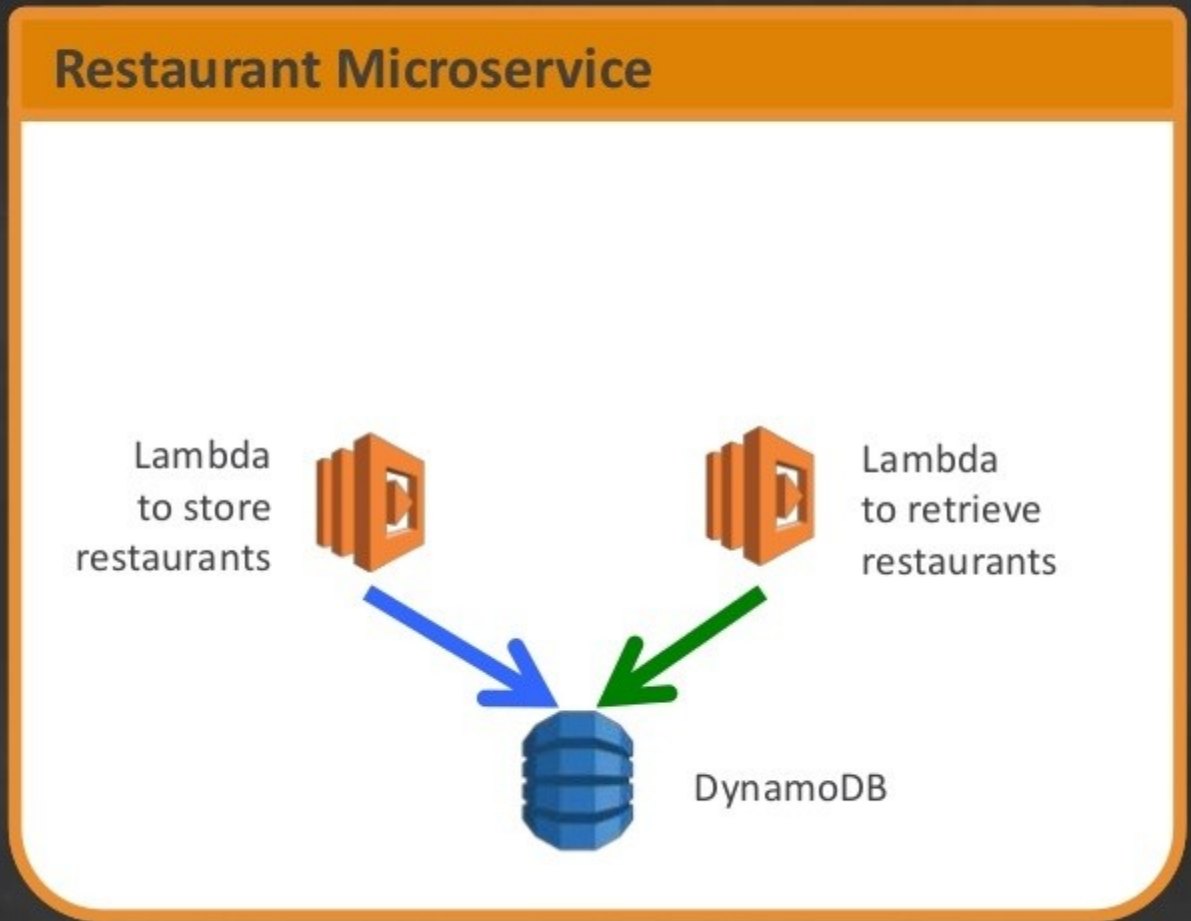
…as well as monitoring, logging, rollbacks, client SDK generation…

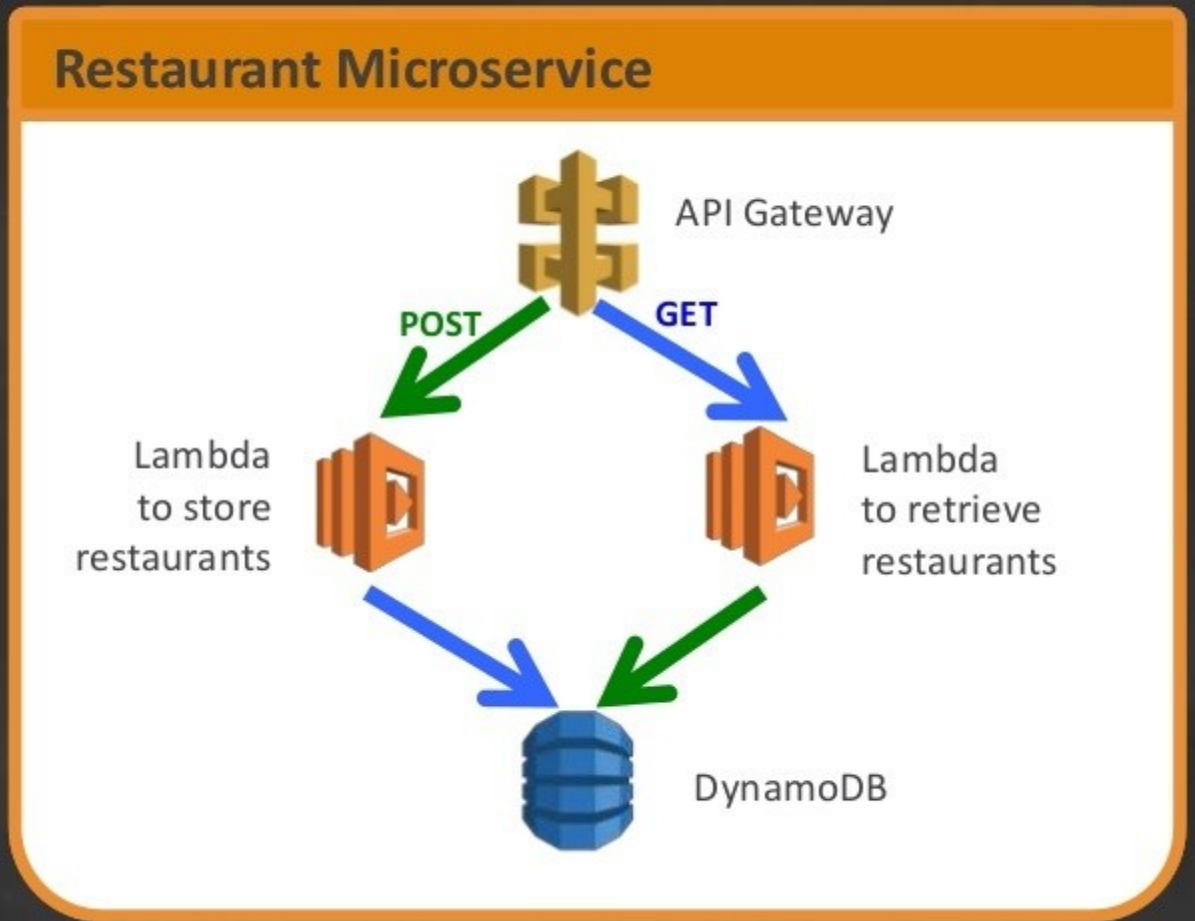

amazon web services
Search
Add samples
Add a restau
RESTAURANT NAME
LATITUDE
LONGITUDE
CUISINE
Asian
PRICE
Affordable ($)
Add restaurant
Pearl District
NW NAITO PKWY
NW BROADWAY BRIDGE
INTERSTATE AVE
EASTBANK FWY
Willamette River
NW OVERTON ST
NW GLISAN ST
NW EVERETT ST
NW DAVIS ST
NW COUCH ST
NW 14TH AVE
NW 12TH AVE
NW 11TH AVE
NW 10TH AVE
NW 15TH AVE
NW 4TH AVE
NW PARK AVE
NW 8TH AVE
STADIUM FWY
405

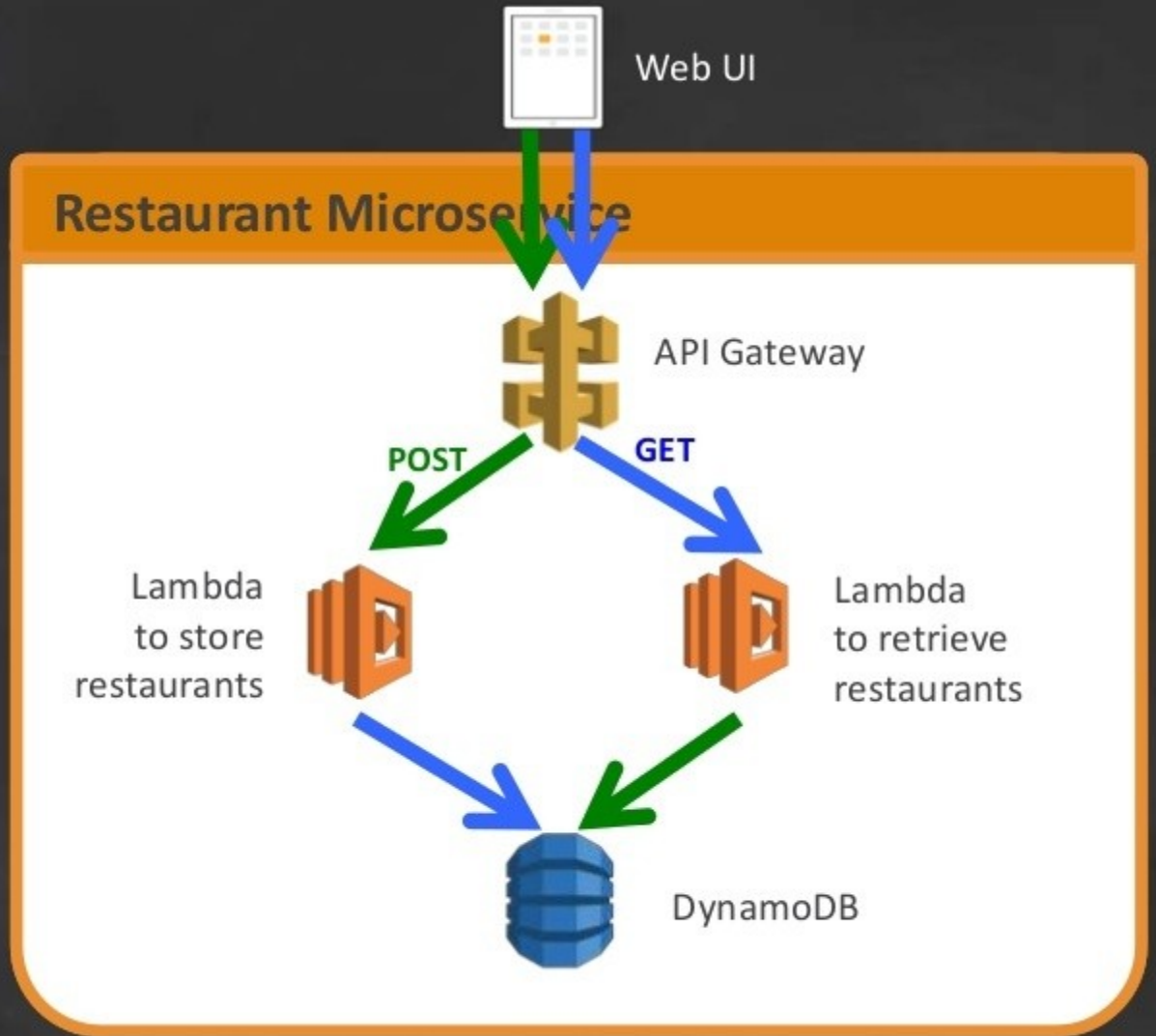

Restaurant Microservice

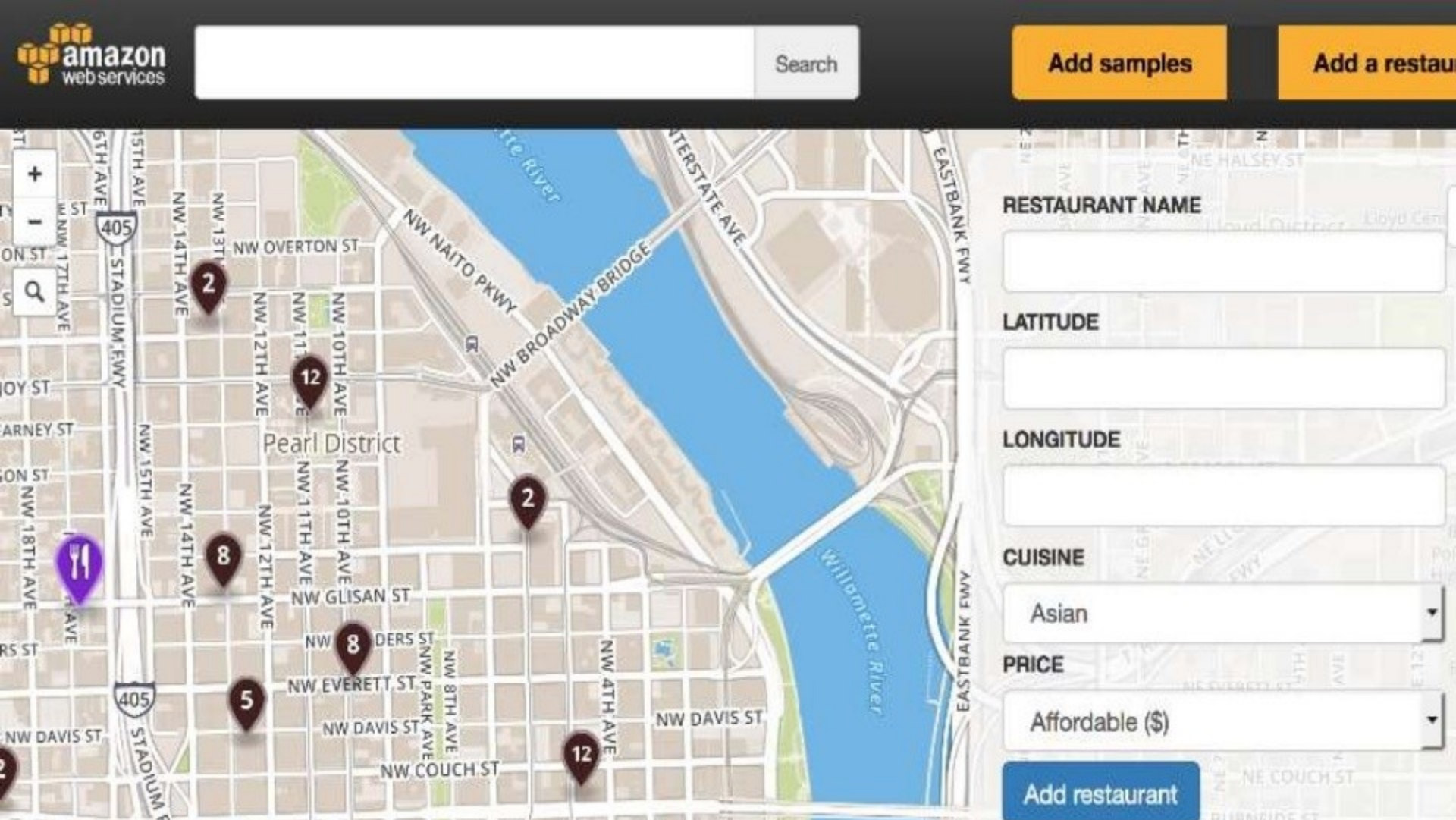
Restaurant Microservice

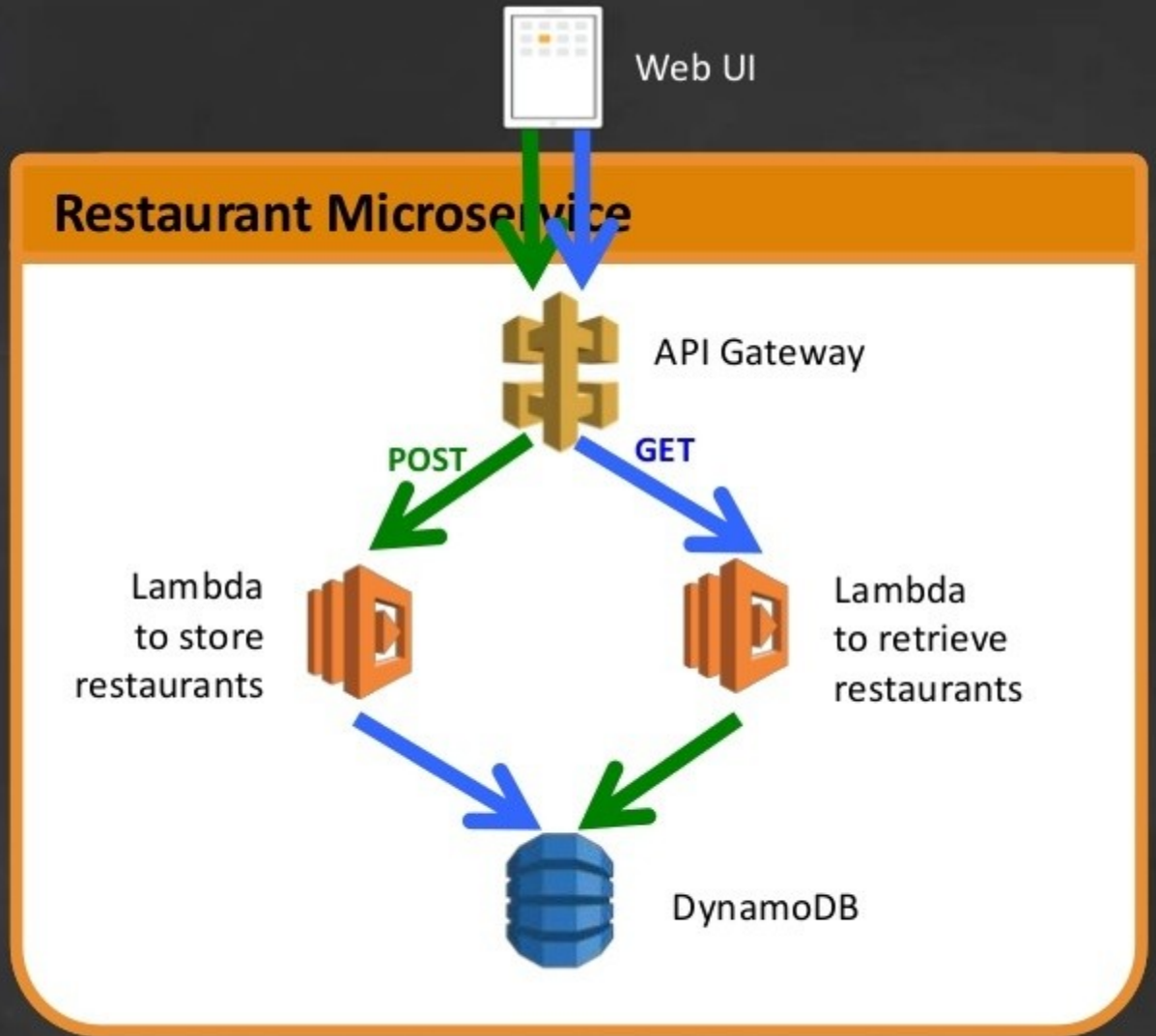
DynamoDB

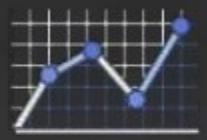

Restaurant Microservice
Lambda to store restaurants
Lambda to retrieve restaurants
DynamoDB


Restaurant Microservice
API Gateway
POST
GET
Lambda
to store
restaurants
Lambda
to retrieve
restaurants
DynamoDB


Web UI
Restaurant Microservice
API Gateway
POST
GET
Lambda
to store
restaurants
Lambda
to retrieve
restaurants
DynamoDB


amazon web services
Search
Add samples
Add a restau
RESTAURANT NAME
LATITUDE
LONGITUDE
CUISINE
Asian
PRICE
Affordable ($)
Add restaurant
Pearl District
NW NAITO PKWY
NW BROADWAY BRIDGE
INTERSTATE AVE
EASTBANK FWY
Willamette River
NW OVERTON ST
NW GLISAN ST
NW EVERETT ST
NW DAVIS ST
NW COUCH ST
NW 14TH AVE
NW 12TH AVE
NW 11TH AVE
NW 10TH AVE
NW 15TH AVE
NW 4TH AVE
NW PARK AVE
NW 8TH AVE
STADIUM FWY
405


Web UI
Restaurant Microservice
API Gateway
POST
GET
Lambda
to store
restaurants
Lambda
to retrieve
restaurants
DynamoDB



Highly Scalable

- Inherently scalable

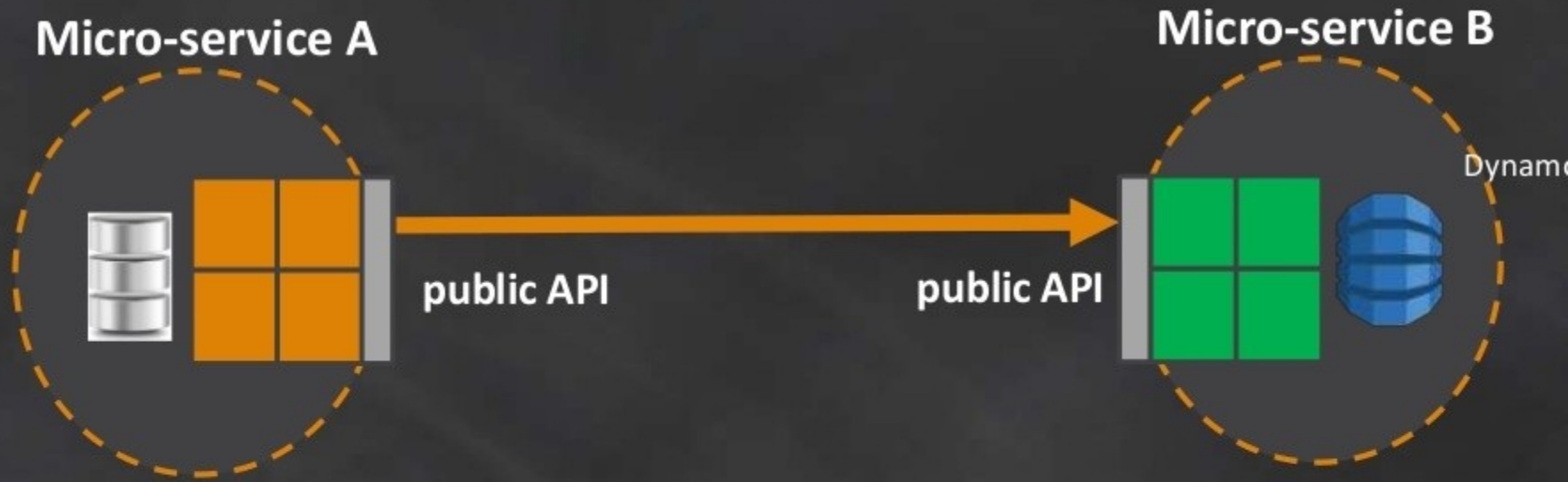
Secure

- API Gateway acts as “front door”
- Can add authN/authZ; or throttle API if needed
- S3 bucket policies
- IAM Roles for Lambda invocations

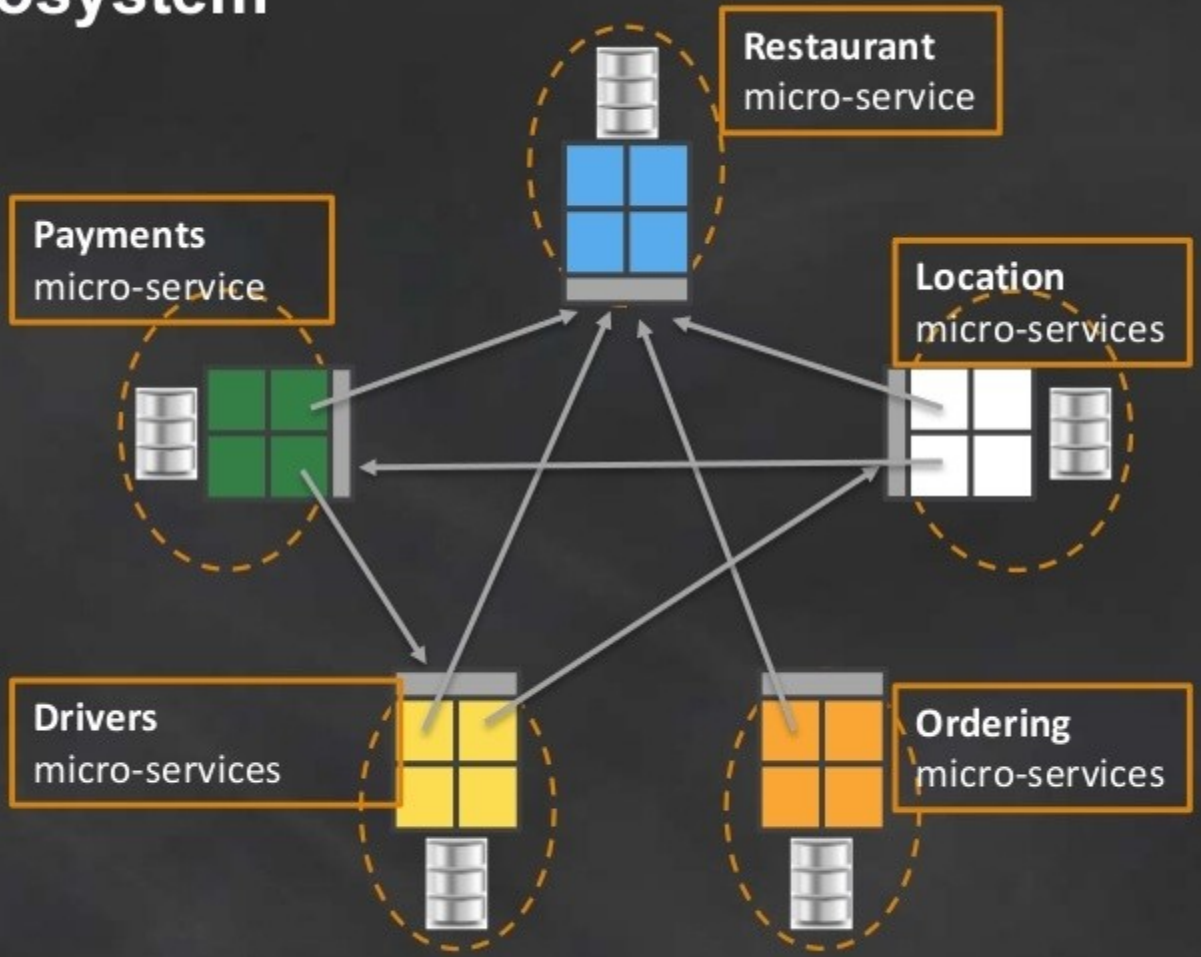
Cost-efficient

- Only pay for actual microservice usage

From a few…



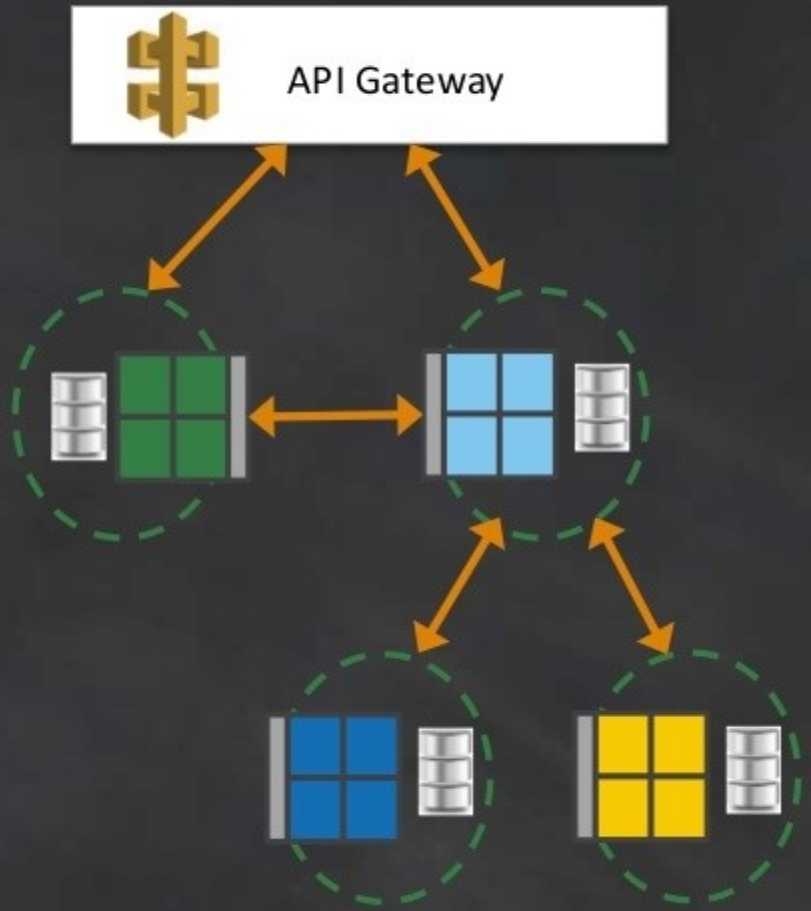
…to an ecosystem



Principle 3

Secure Your Services

Principle 3: Secure Your Services

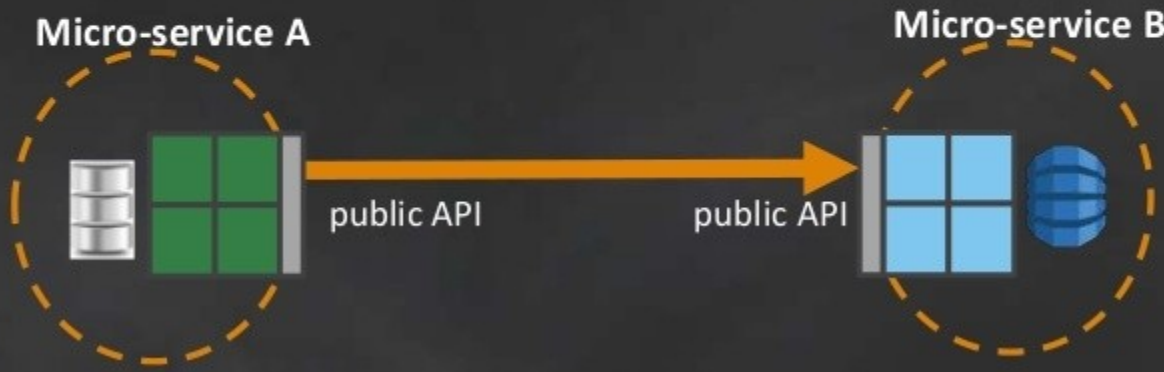


- **Defense-in-depth**
  - Network level (e.g. VPC, Security Groups, TLS)
  - Server/container-level
  - App-level
  - IAM policies
- **Gateway** ("Front door")
- **API Throttling**
- **Authentication & Authorization**
  - Client-to-service, as well as service-to-service
  - API Gateway: custom Lambda authorizers
  - IAM-based Authentication
  - Token-based auth (JWT tokens, OAuth 2.0)
- **Secrets management**
  - S3 bucket policies + KMS + IAM
  - Open-source tools (e.g. Vault, Keywhiz)

Principle 4

Be a good citizen within the ecosystem

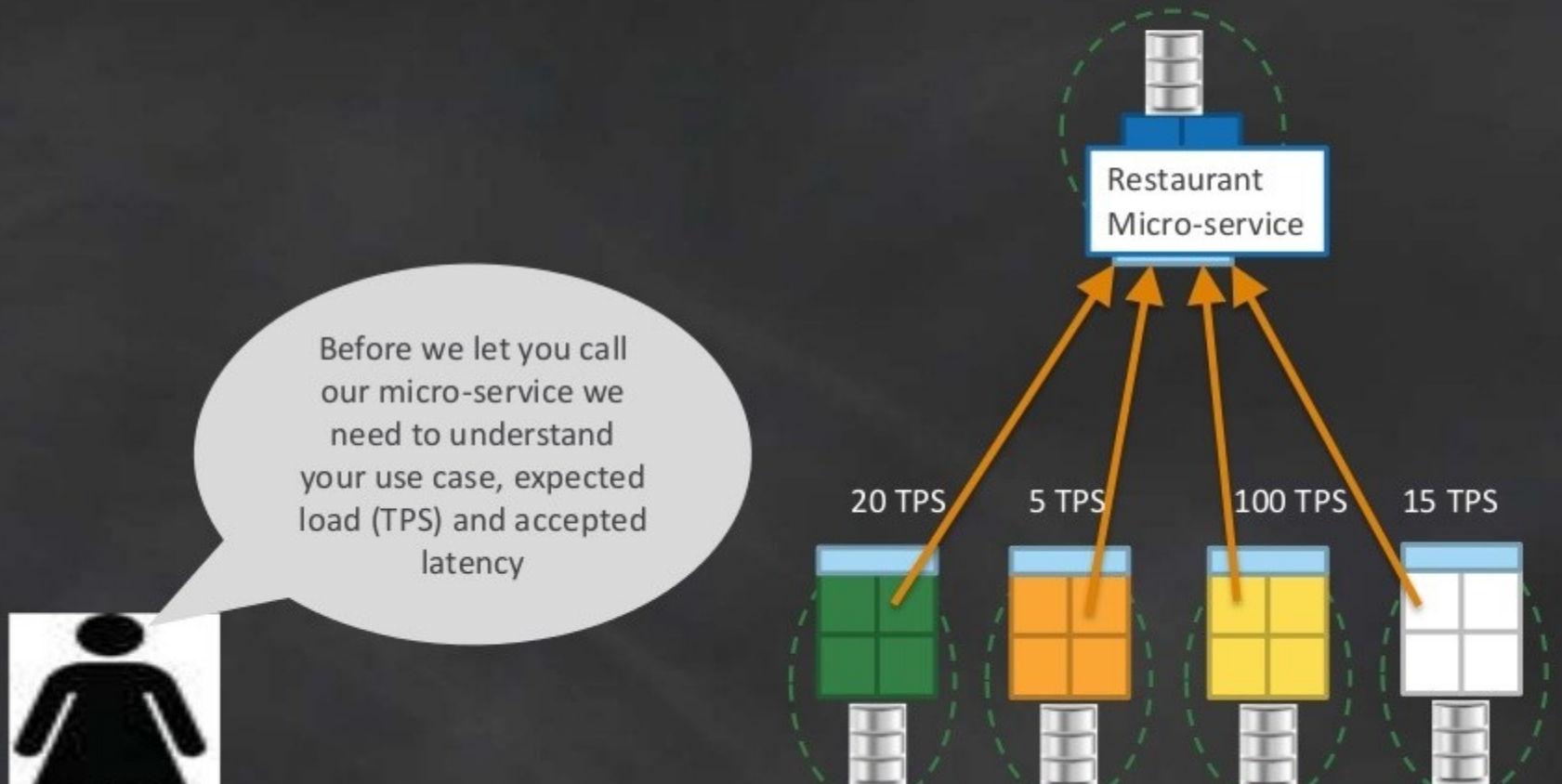
Principle 4: Be a good citizen within the ecosystem



Hey Sally, we need to call your micro-service to fetch restaurants details.

Sure Paul. Which APIs you need to call? Once I know better your use cases I'll give you permission to register your service as a client on our service's directory entry.

Principle 4: Be a good citizen within the ecosystem (Have clear SLAs)



Principle 4: Be a good citizen within the ecosystem (Distributed monitoring, logging and tracing)

Distributed monitoring and tracing

- "Is the service meeting its SLA?"
- "Which services were involved in a request?"
- "How did downstream dependencies perform?"

Shared metrics

- e.g. request time, time to first byte

Distributed tracing

- e.g. Zipkin, OpenTracing

User-experience metrics

Principle 5

More than just technology transformatio

"rowing on the river in Bedford" by Matthew Hunt. No alterations other tha
https://www.flickr.com/photos/mattphotos

Conway's Law

"Any organization that designs a system will inevitably produce a design whose structure is a copy of the organization's communication structure."

Melvin E. Conway, 1967

Decentralize governance and data management

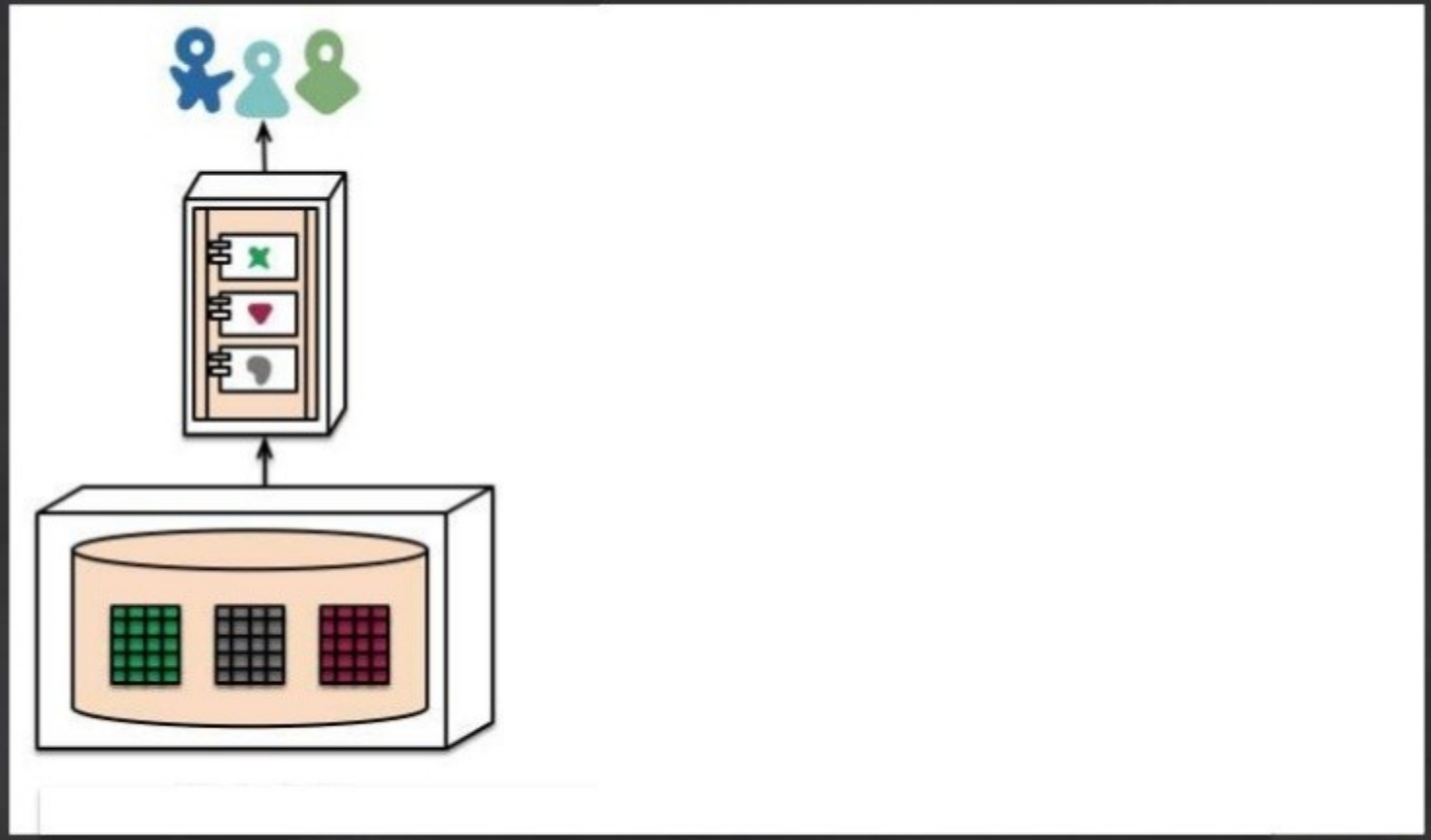


Image from Martin Fowler's article on micro
http://martinfowler.com/articles/microse

Decentralize governance and data management

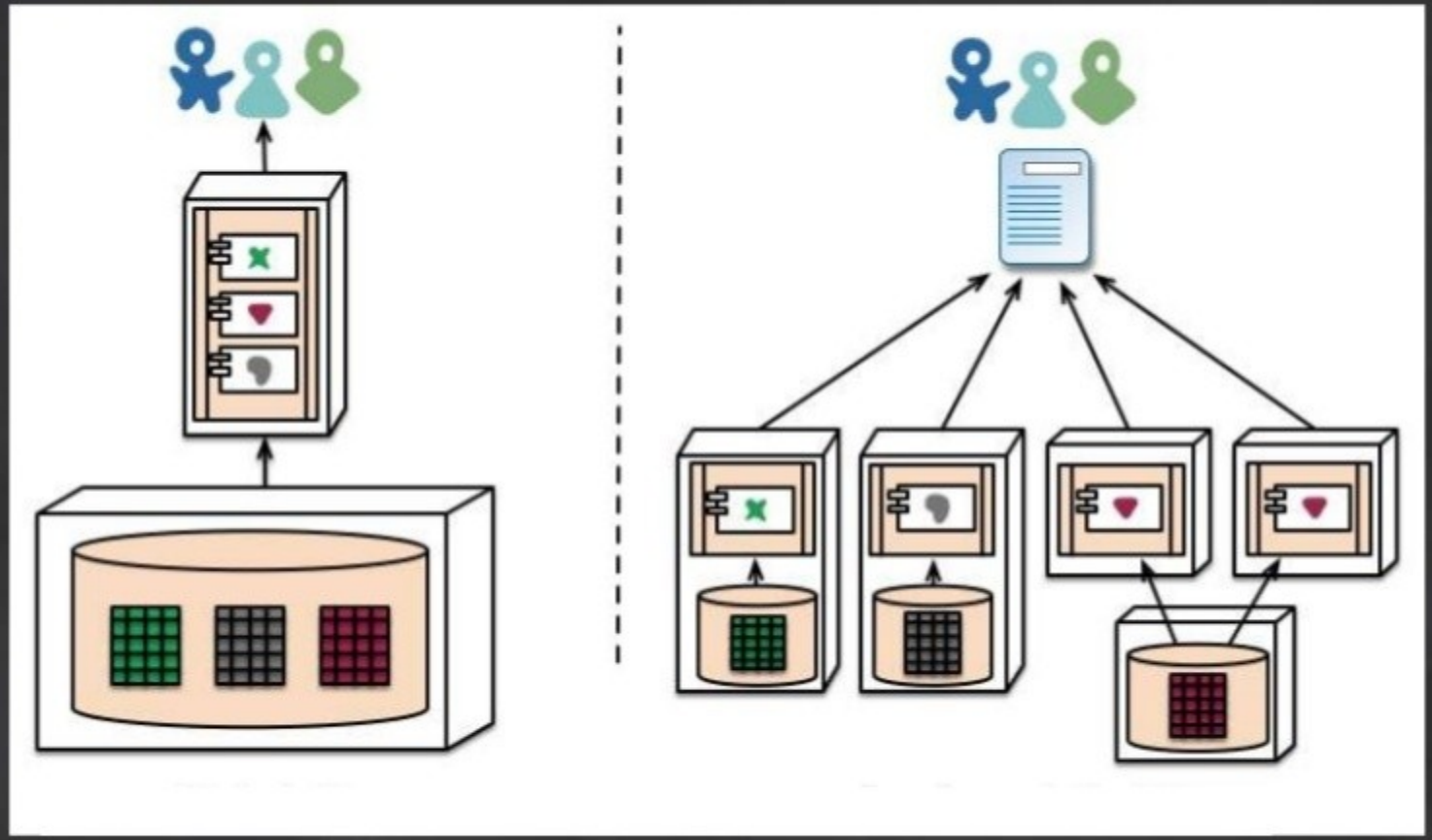
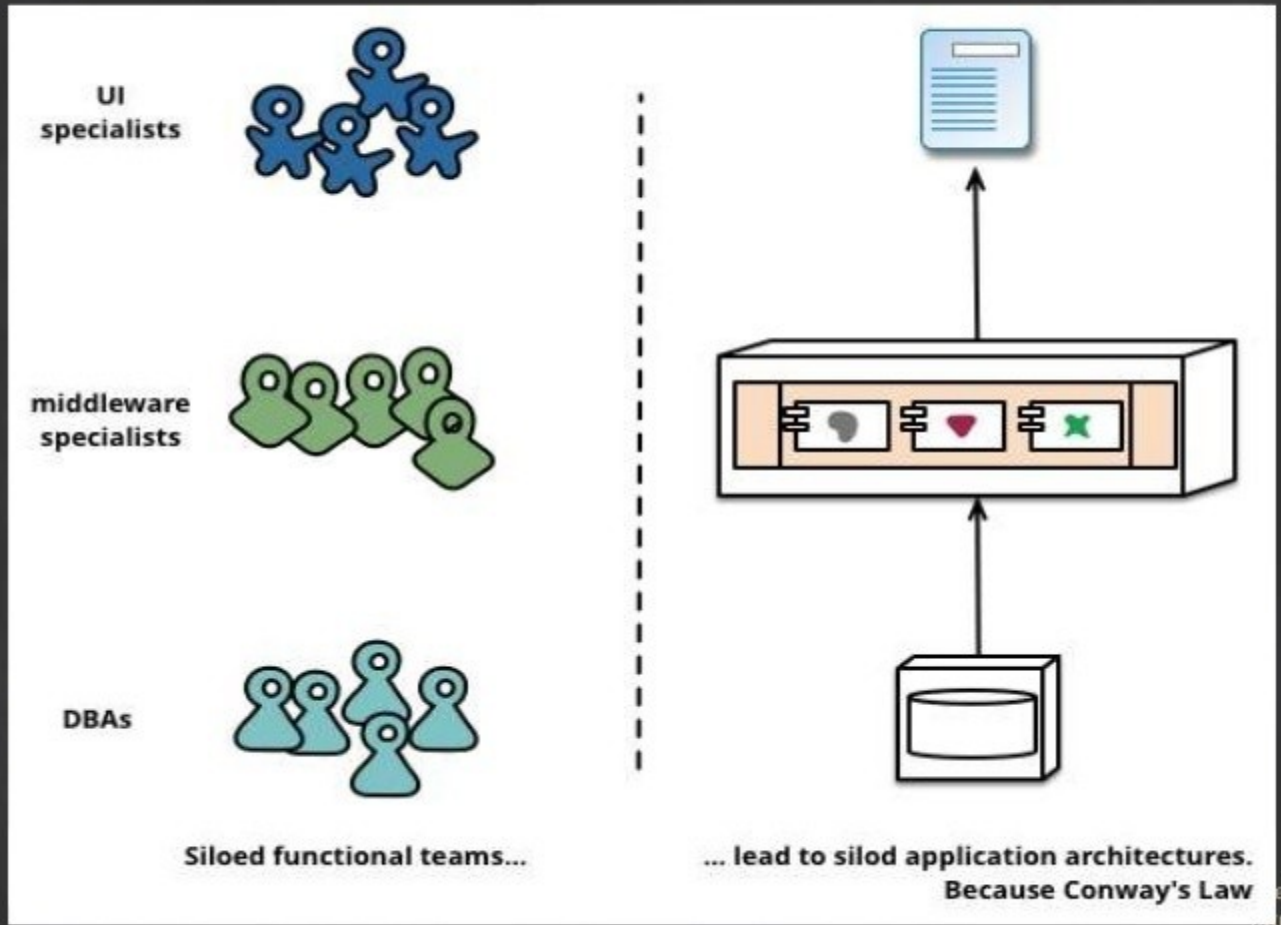


Image from Martin Fowler's article on micro
http://martinfowler.com/articles/microse

Silo'd functional teams → silo'd application architectures



e from Martin Fowler's article on micro
p://martinfowler.com/articles/microse

Cross functional teams → self-contained services

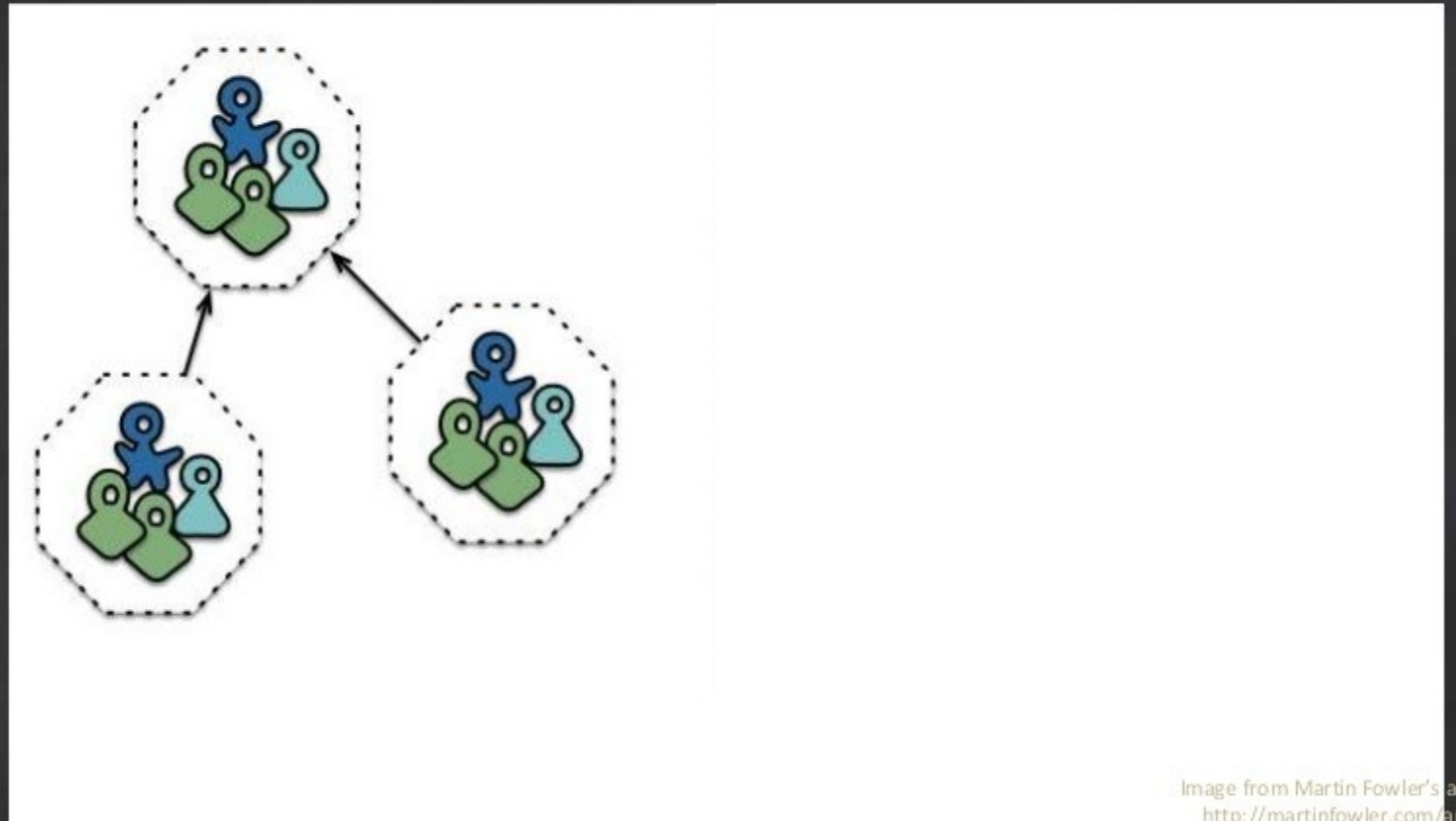


Image from Martin Fowler's article on micro
http://martinfowler.com/articles/microse

Cross functional teams → self-contained services

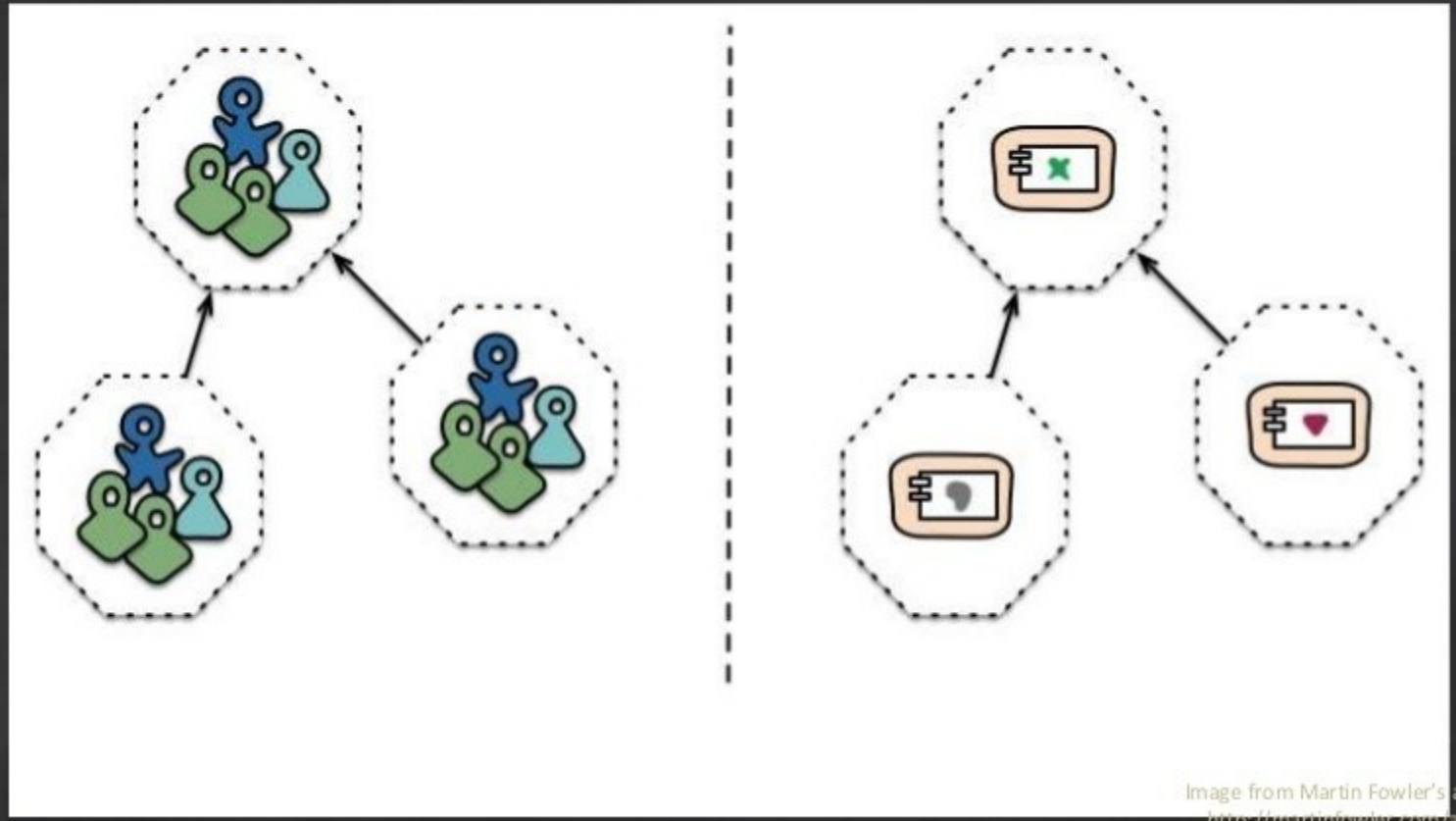
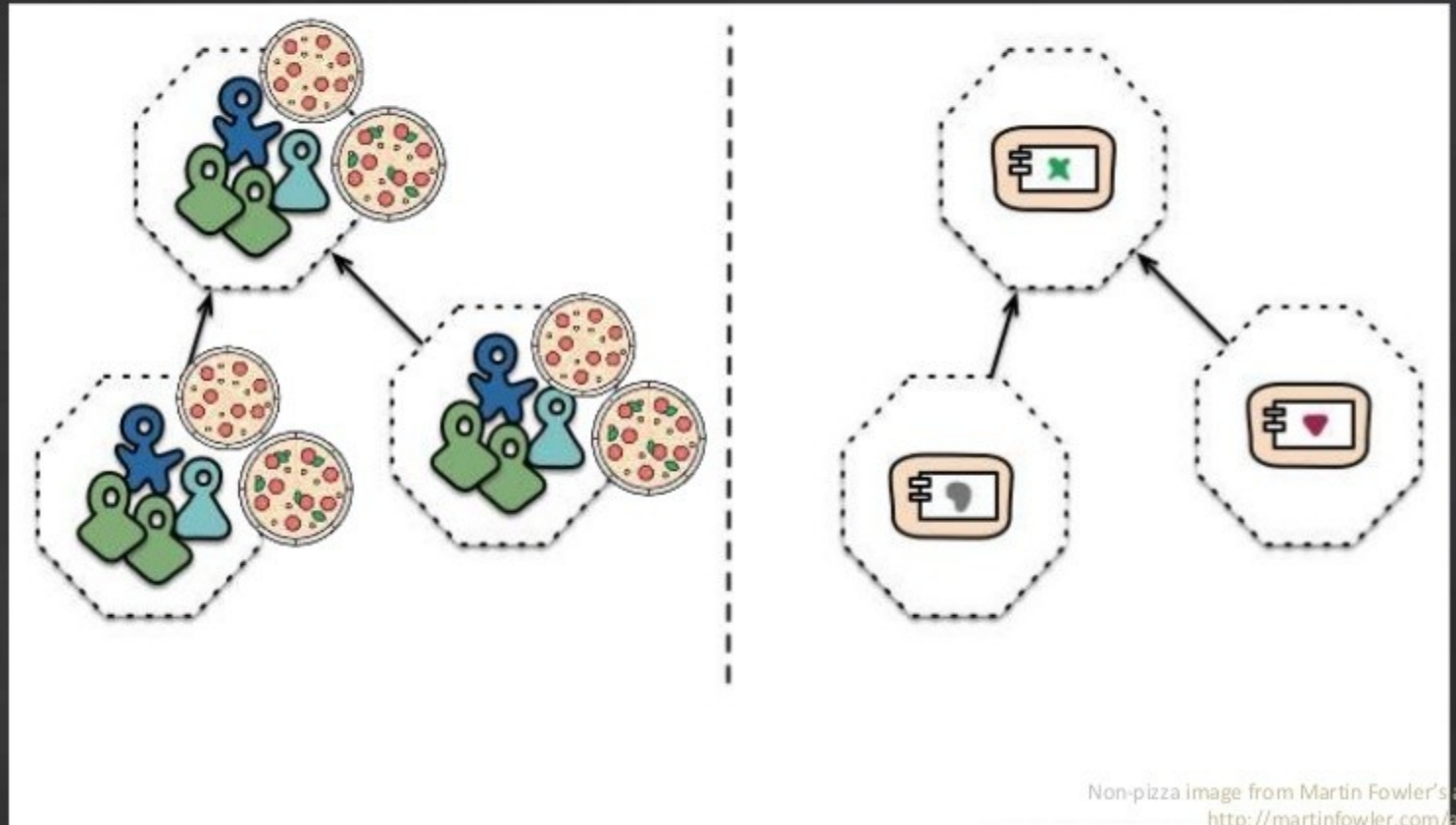


Image from Martin Fowler's article on micro

Cross functional teams → self-contained services

("Two-pizza teams" at Amazon)



Non-pizza image from Martin Fowler's article on micro
http://martinfowler.com/articles/microse

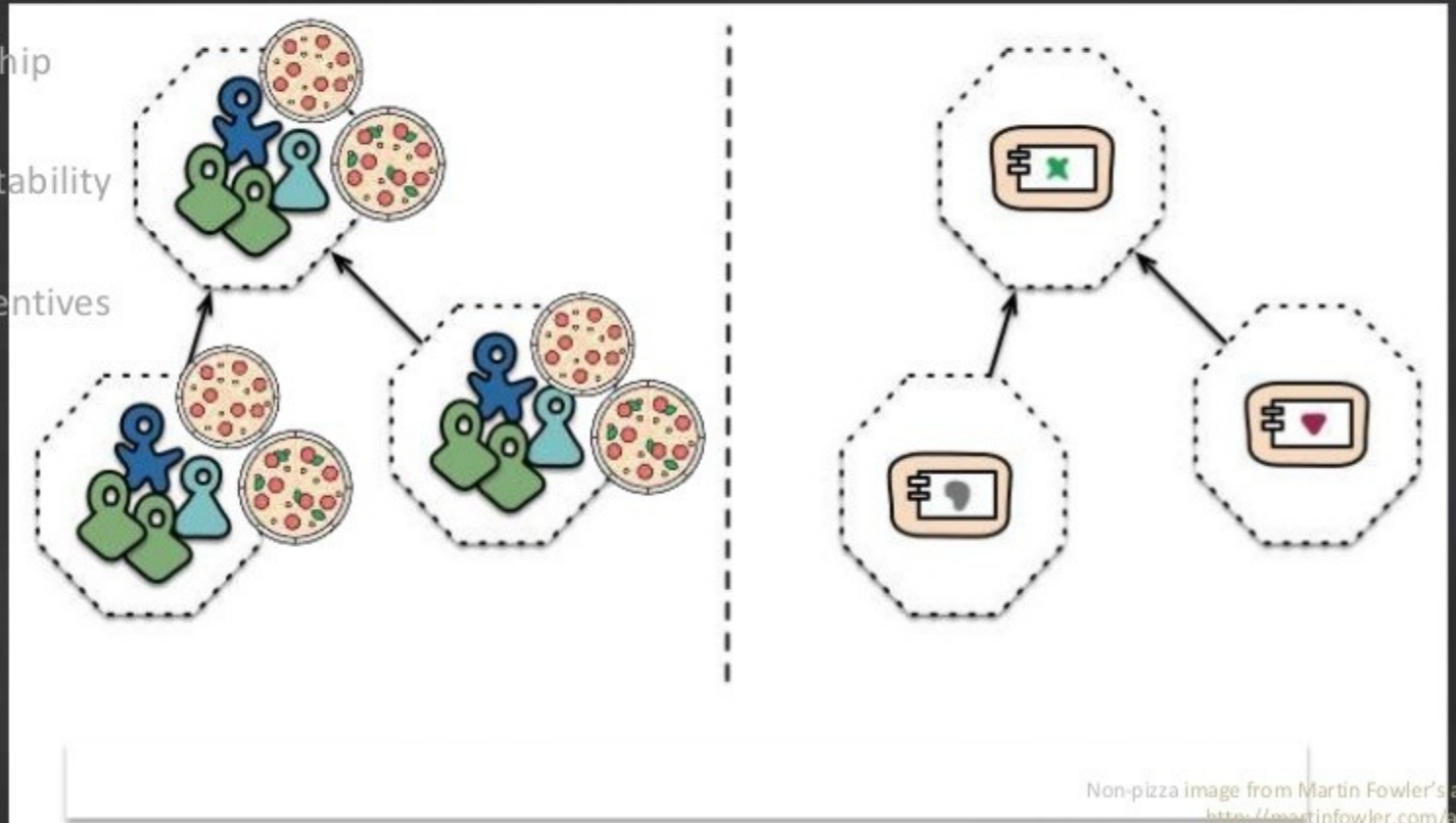
Cross functional teams → self-contained services ("Two-pizza teams" at Amazon)

Full ownership

Full accountability

Aligned incentives

"DevOps"



Non-pizza image from Martin Fowler's article on micro

Principle 6

Automate Everything

"Robot" by Robin Zebrowski. No alterations other tha
https://www.flickr.com/photos/firepile/4

Focused agile teams

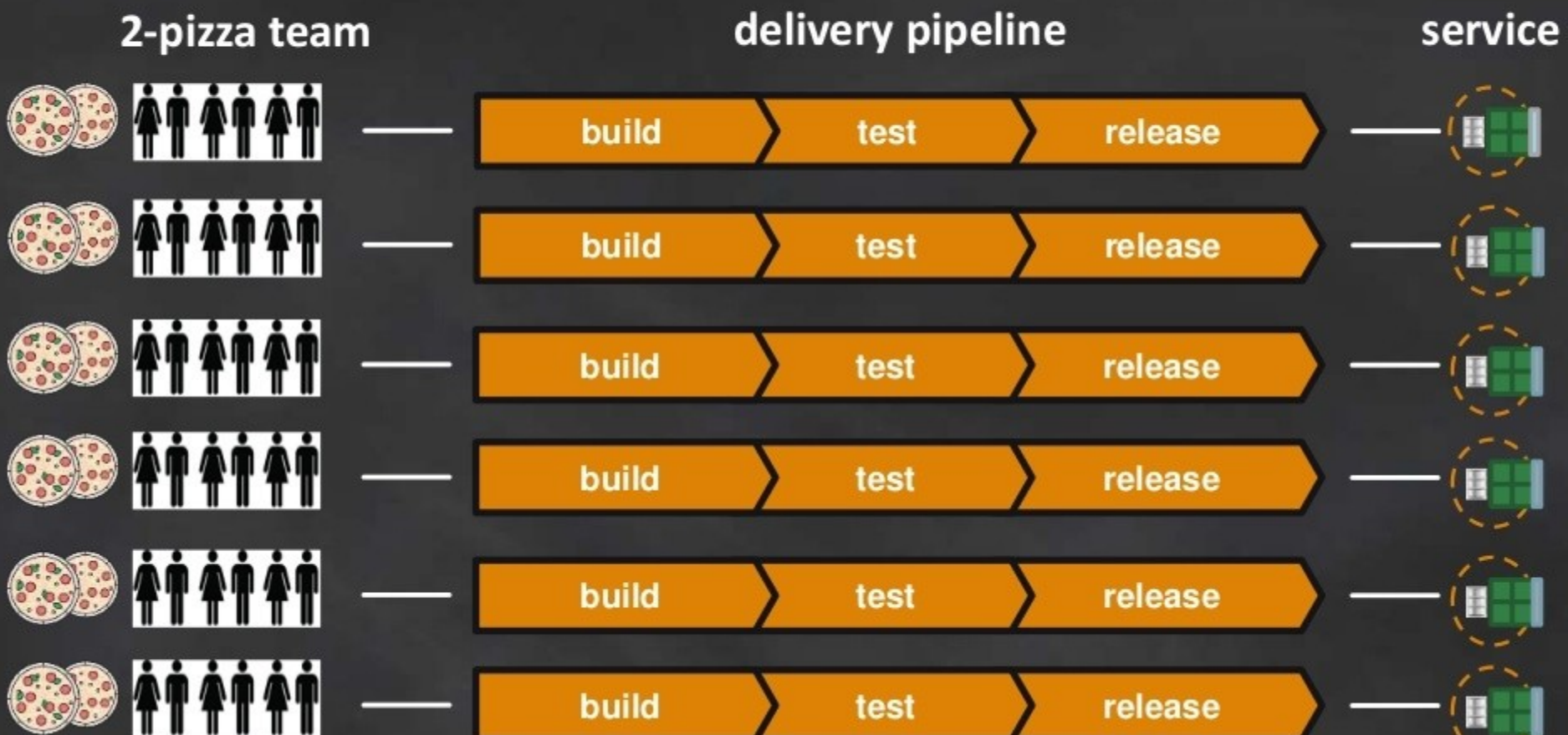
Focused agile teams

Focused agile teams

Focused agile teams

Focused agile teams

Focused agile teams



Principle 6: Automate everything

AWS
CodeCommit

AWS
CodePipeline

AWS
CodeDeploy

RDS

DynamoDB

ElastiCache

CloudWatch

Cloud Trail

API Gateway

SQS

SWF

EC2

ECS

Lambda

Auto
Scaling

ELB

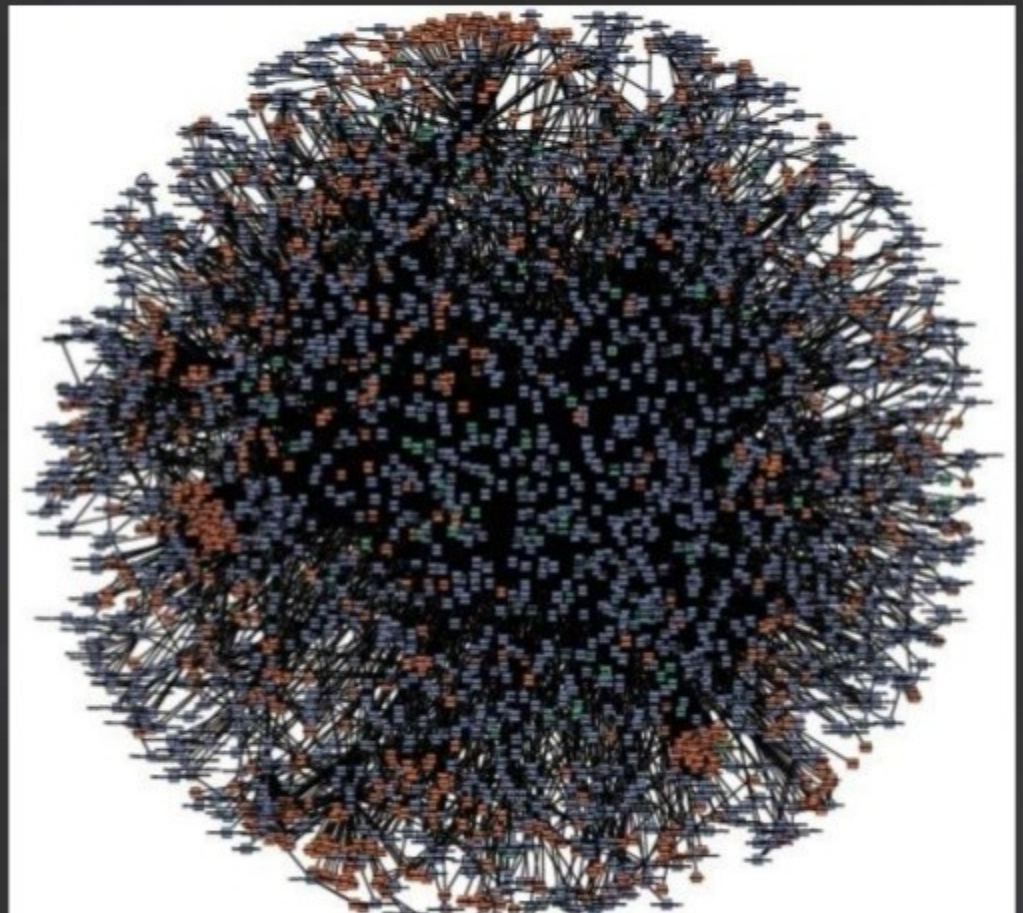
Elastic
Beanstalk

SES

SNS

Kinesis

It's a journey…



Expect challenges along the way…

- Understanding of business domains
- Coordinating txns across multiple servic
- Eventual Consistency
- Service discovery
- Lots of moving parts requires increased coordination
- Complexity of testing / deploying / operating a distributed system
- Cultural transformation

Principles of Microservices

1. Rely only on the public API
   - Hide your data
   - Document your APIs
   - Define a versioning strategy

2. Use the right tool for the job
   - Polygot persistence (data layer)
   - Polyglot frameworks (app layer)

3. Secure your services
   - Defense-in-depth
   - Authentication/authorization

4. Be a good citizen within the ecosystem
   - Have SLAs
   - Distributed monitoring, logging, tracin

5. More than just technology transformation
   - Embrace organizational change
   - Favor small focused dev teams

6. Automate everything
   - Adopt DevOps

Benefits of microservices

Easier to scale each individual micro-service

Rapid Build/Test/Release Cycles

Clear ownership and accountability

Benefits of microservices

Easier to scale each individual micro-service

Easier to maintain and evolve system

Rapid Build/Test/Release Cycles

New releases take minutes

Clear ownership and accountability

Short time to add new features

Benefits of microservices

Easier to scale each individual micro-service

Easier to maintain and evolve system

Increased agility

Rapid Build/Test/Release Cycles

New releases take minutes

Faster innovation

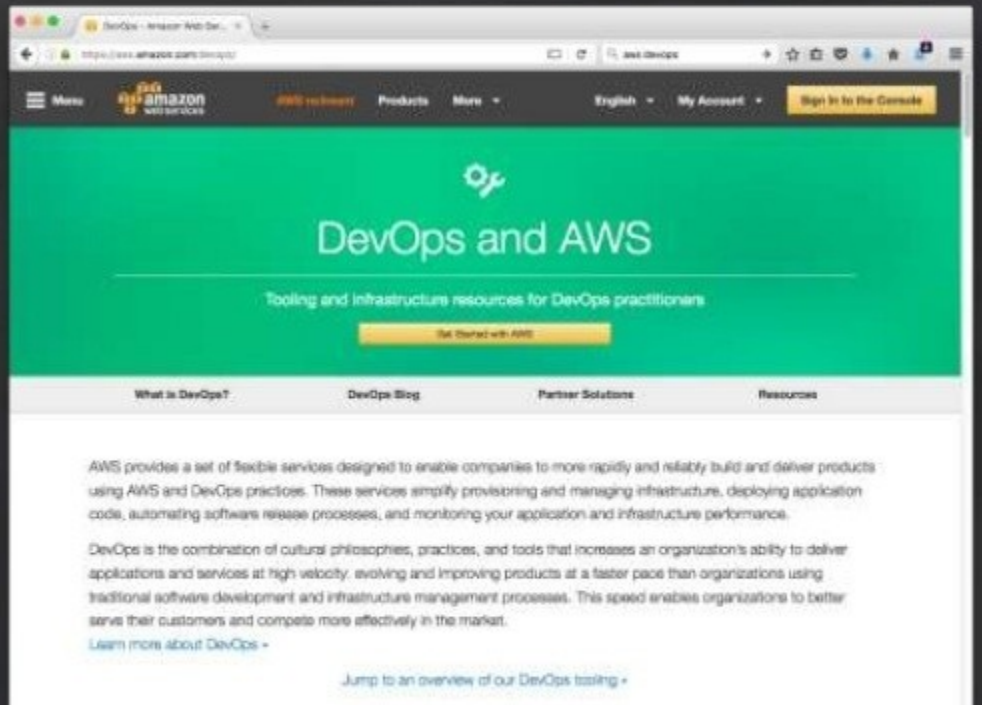
Clear ownership and accountability

Short time to add new features

Delighted customers

Additional resources

https://aws.amazon.com/devops/



Additional AWS resources:

- **Zombie Microservices Workshop:** https://github.com/awslabs/aws-lambda-zombie-wor
- **Serverless Webapp - Reference Architecture:** https://github.com/awslabs/lambda-refarch-webapp
- **Microservices with ECS:** https://aws.amazon.com/blogs/compute/using-amaz api-gateway-with-microservices-deployed-on-amazon
- **Serverless Service Discovery:** https://aws.amazon.com/blogs/developer/ serverless-service-discovery-part-1-get-started/
- **ECS Service Discovery:** https://aws.amazon.com/blogs/compute/ service-discovery-an-amazon-ecs-reference-architectu
- **Microservices without the Servers** https://aws.amazon.com/blogs/compute/ microservices-without-the-servers

Popular open-source tools:

- **Serverless** – http://serverless.com
- **Apex** - http://apex.run/

Thank you!

Peter Dalbhanjan
dalbhanj@amazon.com