
AWS
re:Invent
NOV. 28 – DEC. 2, 2022 | LAS VEGAS, NV

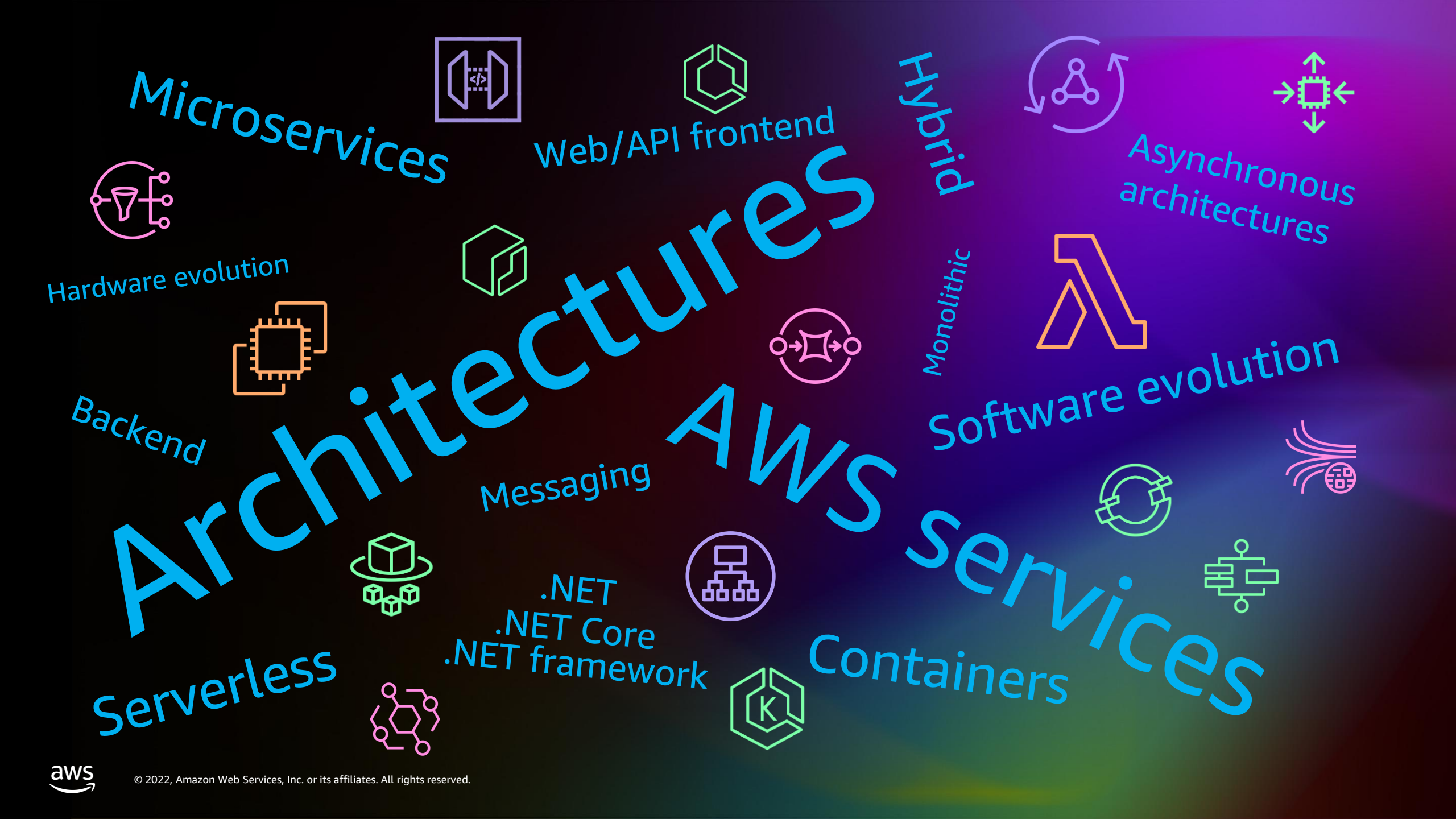
XNT405

Don’t overthink it: Identifying services for modernized .NET workloads

Rahul Chugh
Sr. Solutions Architect
AWS

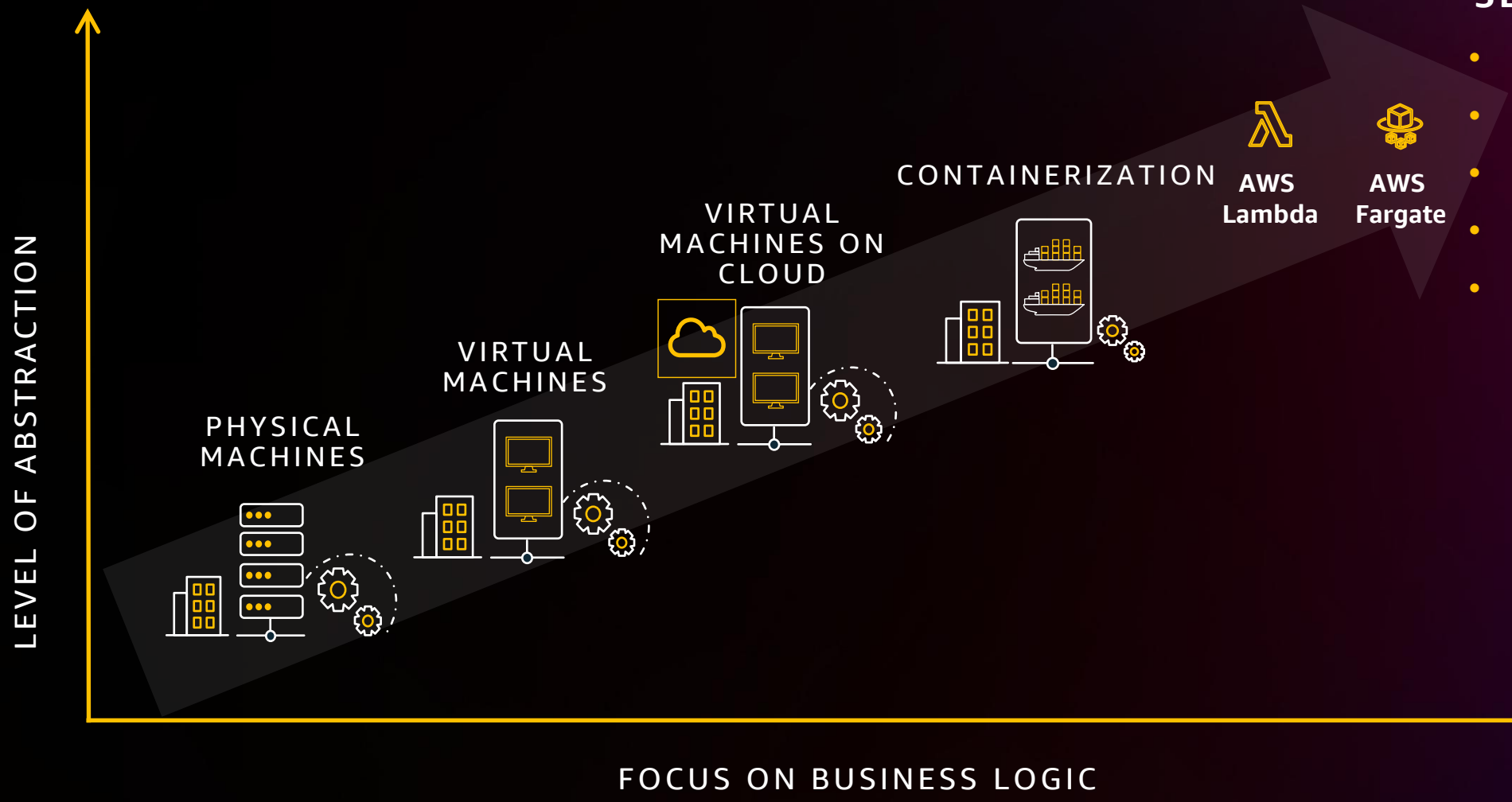
Saleha Haider
Sr. Solutions Architect
AWS

© 2022, Amazon Web Services, Inc. or its affiliates. All rights reserved.



© 2022, Amazon Web Services, Inc. or its affiliates. All rights reserved.

Evolution of computing



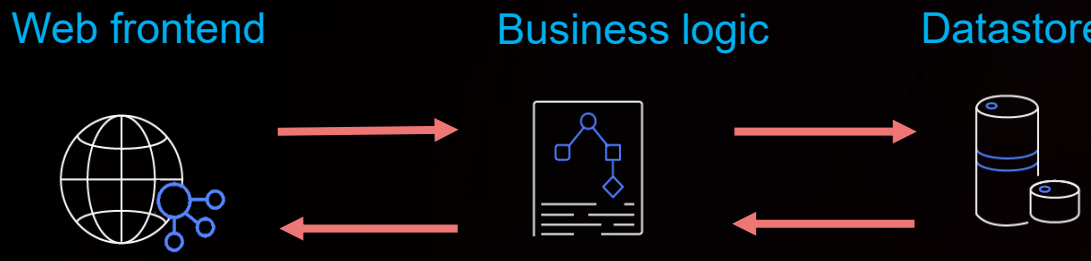
SERVERLESS

- Continuous scaling
- Fault tolerance built in
- Pay for value
- Zero maintenance
- Focus on business value

© 2022, Amazon Web Services, Inc. or its affiliates. All rights reserved.

Software architectures

MONOLITHIC ARCHITECTURE



DRAWBACKS



Hard to iterate fast



Hard to scale efficiently

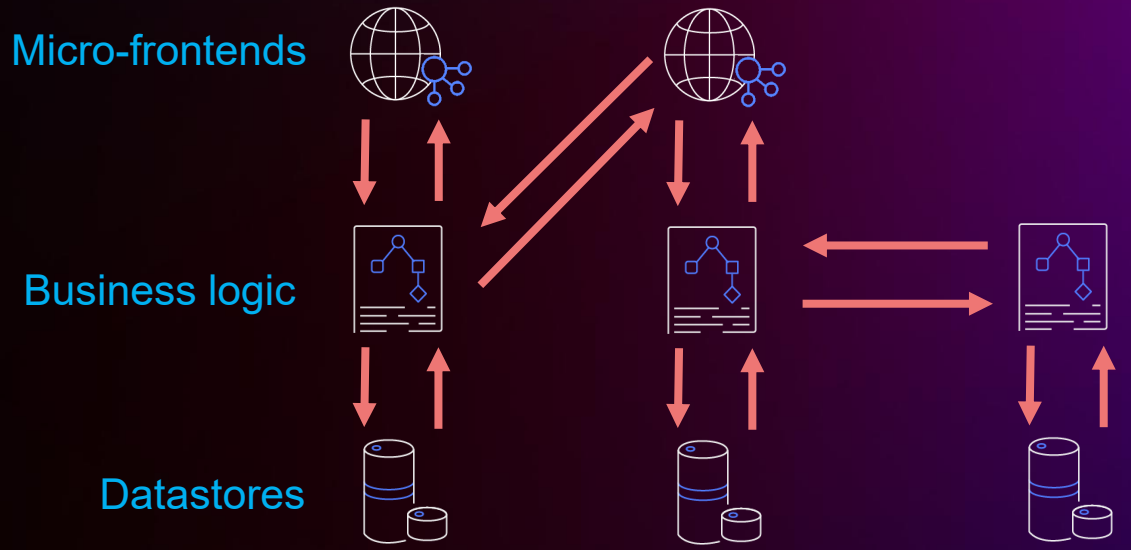


Reliability challenges



CI/CD time consuming

MICROSERVICES ARCHITECTURE



BENEFITS



Increased agility



Easy to scale



Improved innovation



Reduced human errors

aws

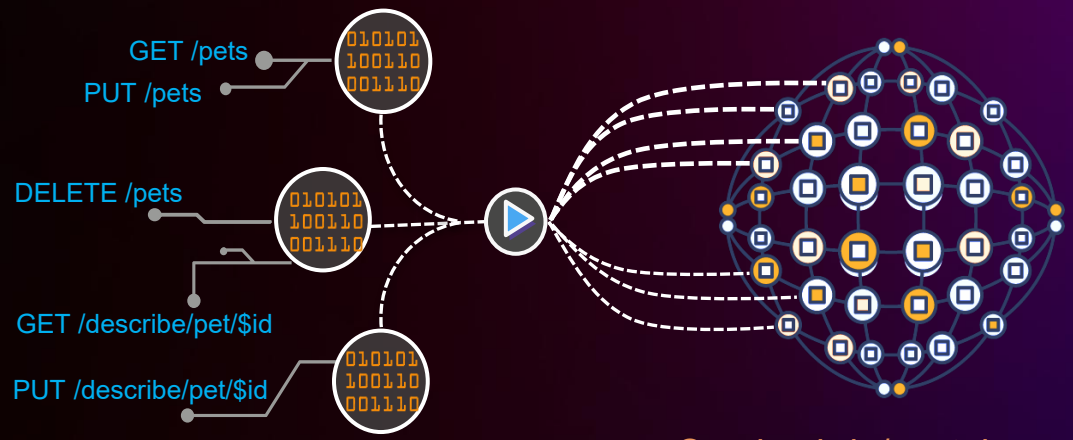
© 2022, Amazon Web Services, Inc. or its affiliates. All rights reserved.

Combining H/W and S/W architectures

MONOLITHIC SERVERLESS ARCHITECTURE

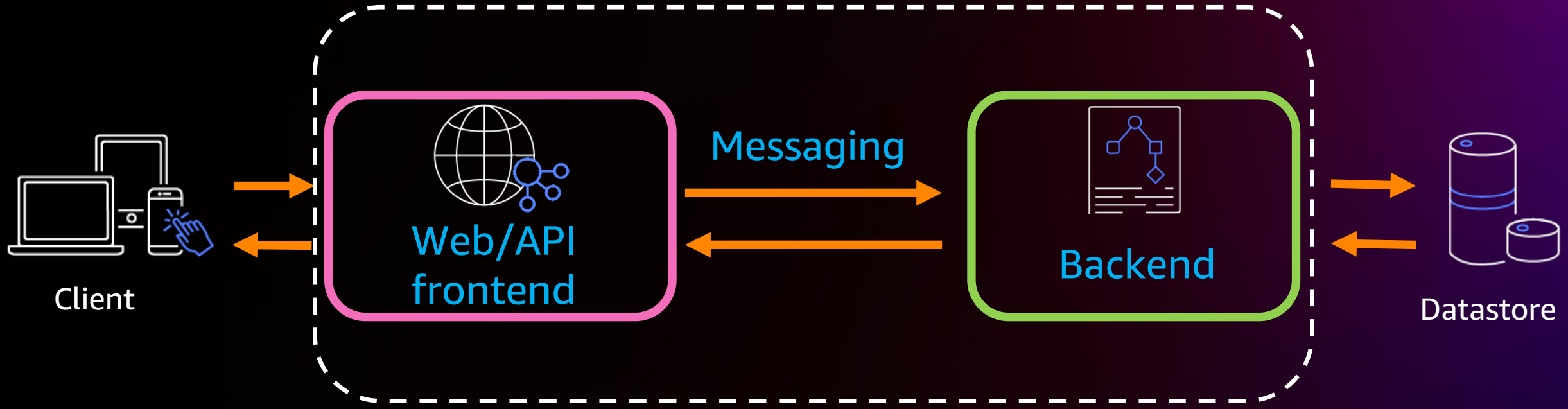


MICROSERVICES SERVERLESS ARCHITECTURE



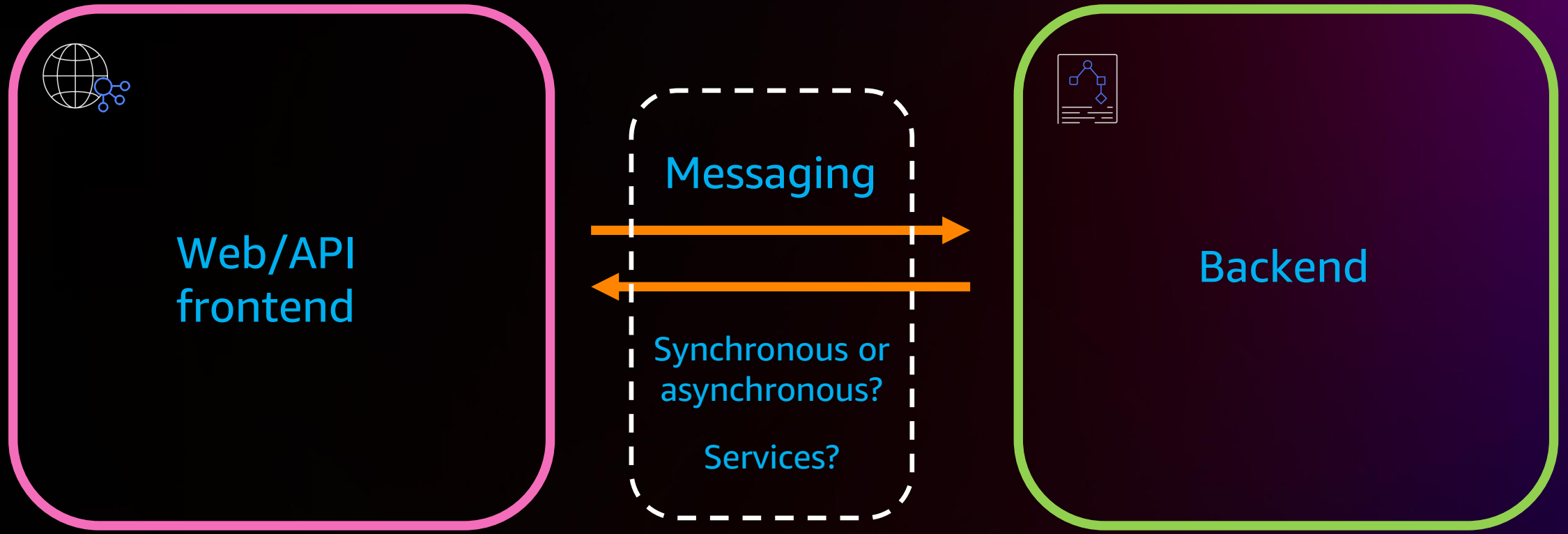
© 2022, Amazon Web Services, Inc. or its affiliates. All rights reserved.

A standard architecture



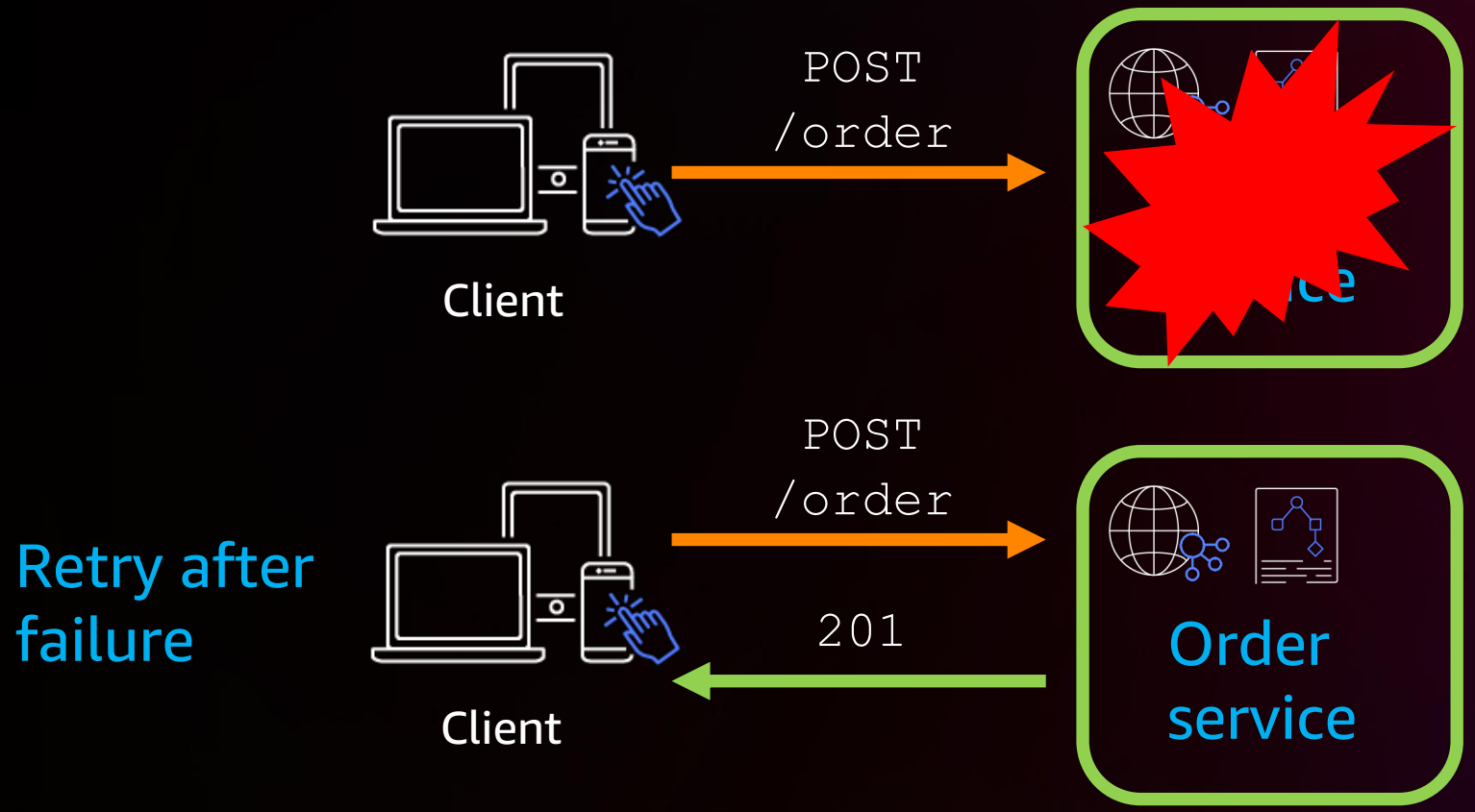
aws
© 2022, Amazon Web Services, Inc. or its affiliates. All rights reserved.

Messaging



© 2022, Amazon Web Services, Inc. or its affiliates. All rights reserved.

Synchronous APIs – Scenario 1



© 2022, Amazon Web Services, Inc. or its affiliates. All rights reserved.

Synchronous APIs – Scenario 2

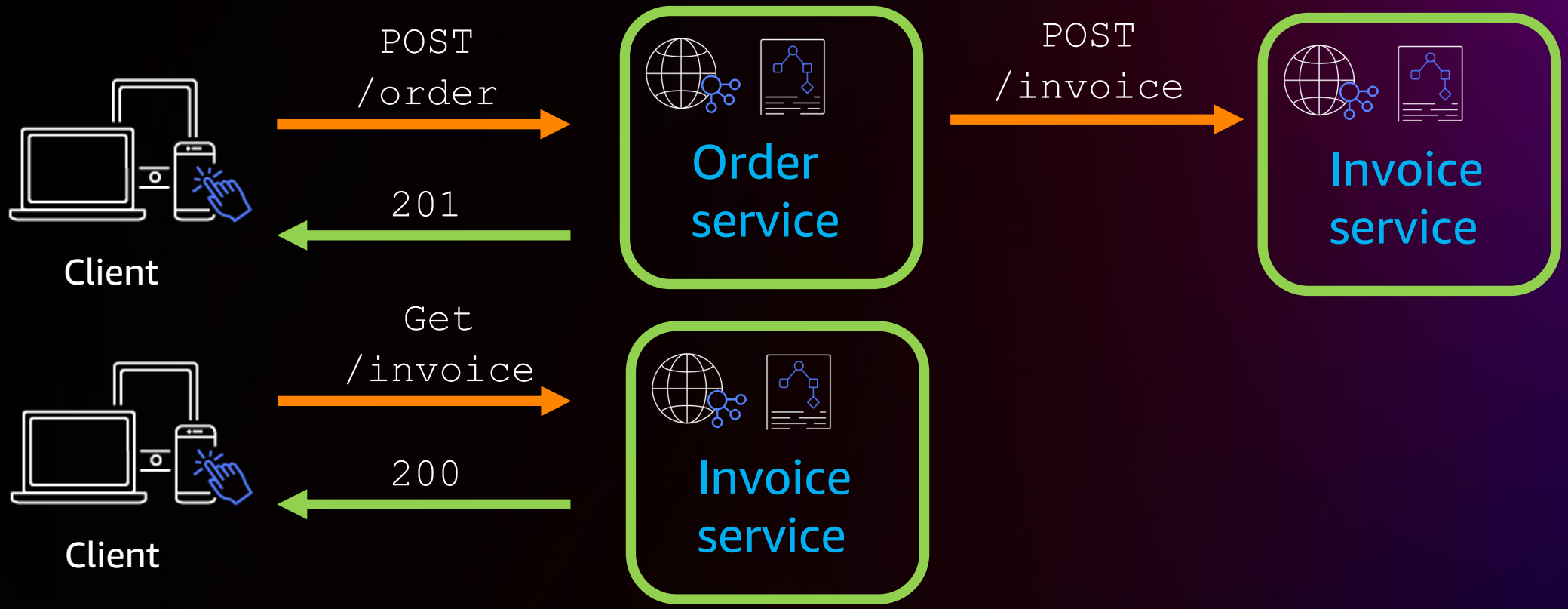


Who owns the retry? For how long? Does the client ever know? Etc..

This effectively creates a “tight coupling,” where failures become harder to recover from

aws
© 2022, Amazon Web Services, Inc. or its affiliates. All rights reserved.

Asynchronous APIs



© 2022, Amazon Web Services, Inc. or its affiliates. All rights reserved.

“The time spent to try making a process async will pay for itself in you gaining on performance, utilization, and having a deeper understanding of what is really happening with your data as it flows through the application.”

Me

Right now, 2022

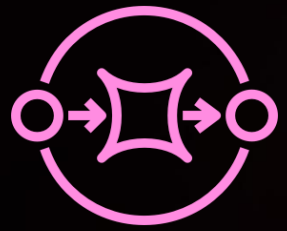
© 2022, Amazon Web Services, Inc. or its affiliates. All rights reserved.

Messaging services

MESSAGING SERVICES



Amazon Simple Notification Service (Amazon SNS)



Amazon Simple Queue Service (Amazon SQS)



Amazon EventBridge

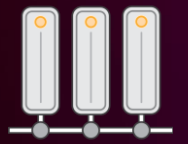


Amazon Kinesis Data Streams

WAYS TO COMPARE



Scale/concurrency controls



Durability



Persistence



Consumption models



Retries



Pricing

aws

© 2022, Amazon Web Services, Inc. or its affiliates. All rights reserved.

Async messaging service decider cheat sheet

Massive throughput/ordering/multiple consumers/replay?

- Amazon Kinesis Data Streams

Time critical updates, push mechanism, one to many fanout, direct to Lambda/SQS/other AWS services?

- Amazon SNS

Buffer requests until they can be consumed, whether ordered or not?

- Amazon SQS

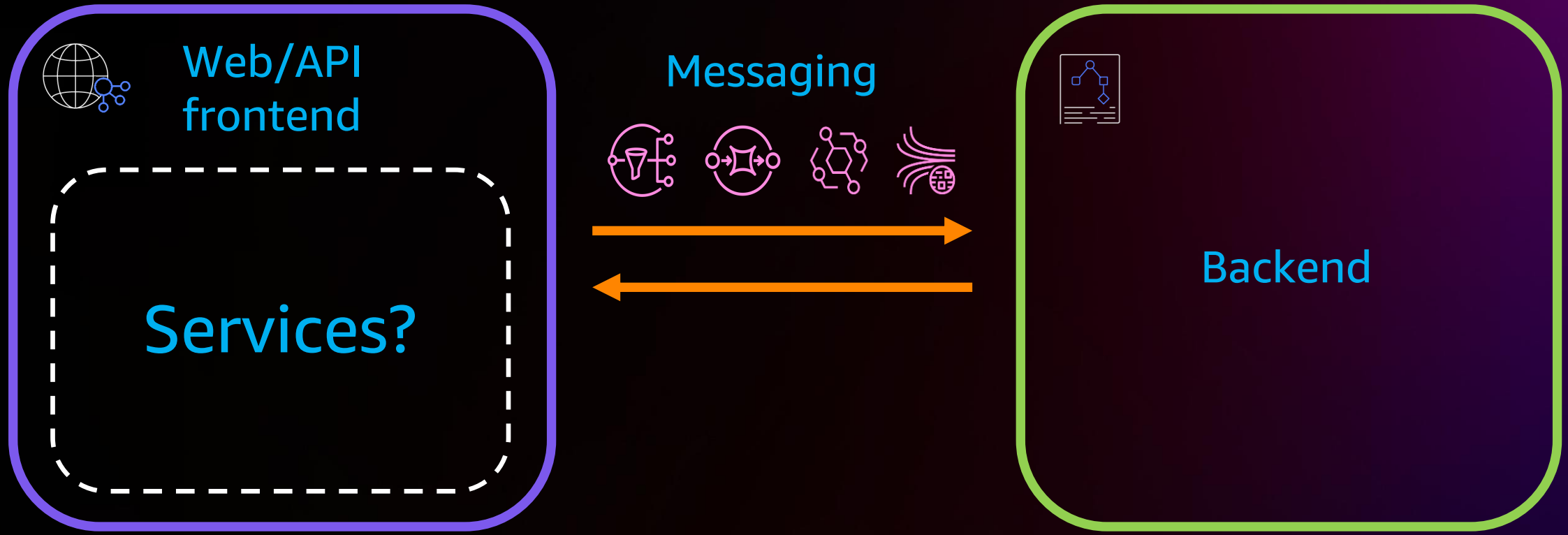
One to many fanout, lots of different consumer targets, schema matching, granular target rules?

- Amazon EventBridge

aws

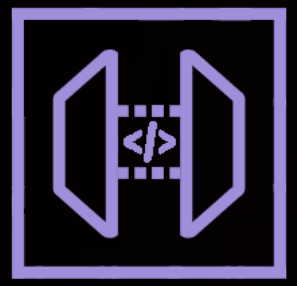
© 2022, Amazon Web Services, Inc. or its affiliates. All rights reserved.

Frontend services

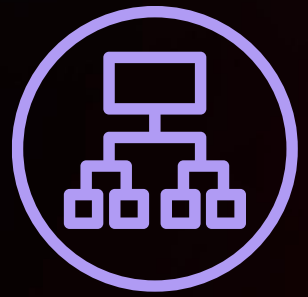


© 2022, Amazon Web Services, Inc. or its affiliates. All rights reserved.

Accessing the Frontend



Amazon API Gateway



Application Load Balancer



AWS AppSync

© 2022, Amazon Web Services, Inc. or its affiliates. All rights reserved.

Picking an API fronting service cheat sheet

Complex API with multiple data sources or very unique queries against data or offline sync?

- AWS AppSync

WebSockets?

- Amazon API Gateway

Need transforms, throttling, usage tiers, flexible auth?

- Amazon API Gateway

Potentially high requests per month, no need for added transform capabilities?

- Application Load Balancer

Else: Normal API, with < tens of millions of requests per month?

- Amazon API Gateway

© 2022, Amazon Web Services, Inc. or its affiliates. All rights reserved.

Where to handle request routing?

AWS services vs. language framework

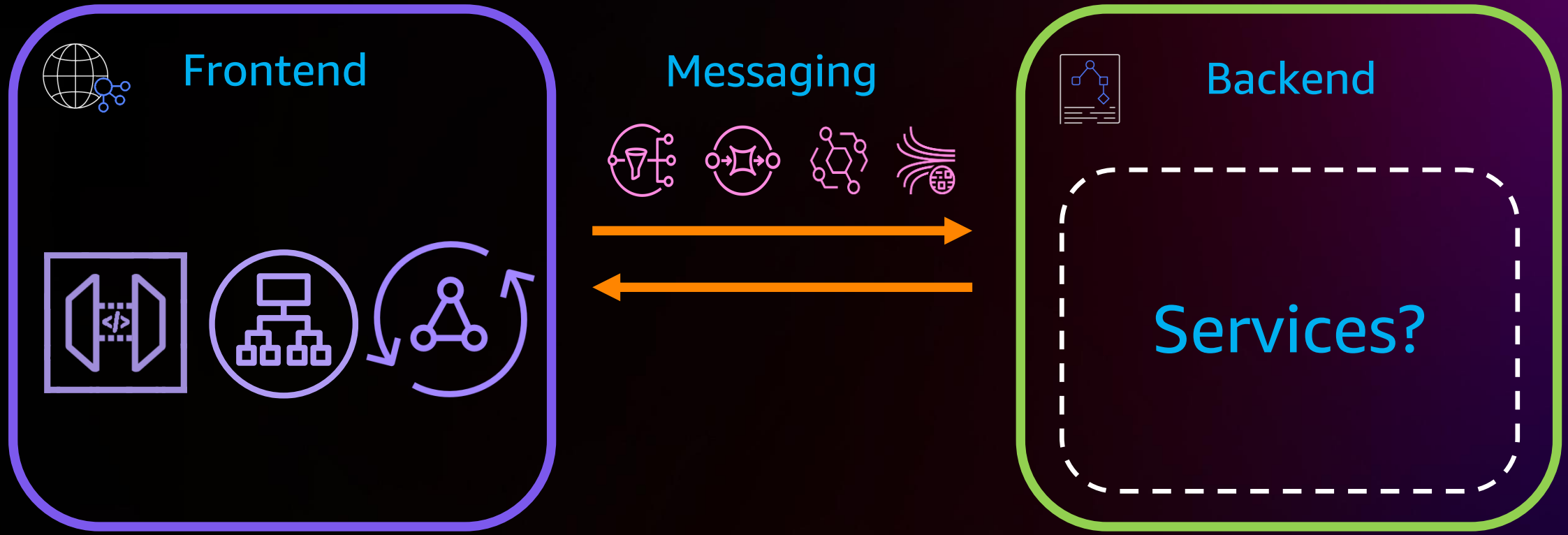
Customer feedback seems split

If you love your framework, stick with it and its capabilities

If you want to do less and write less code, go with the benefits of the managed services

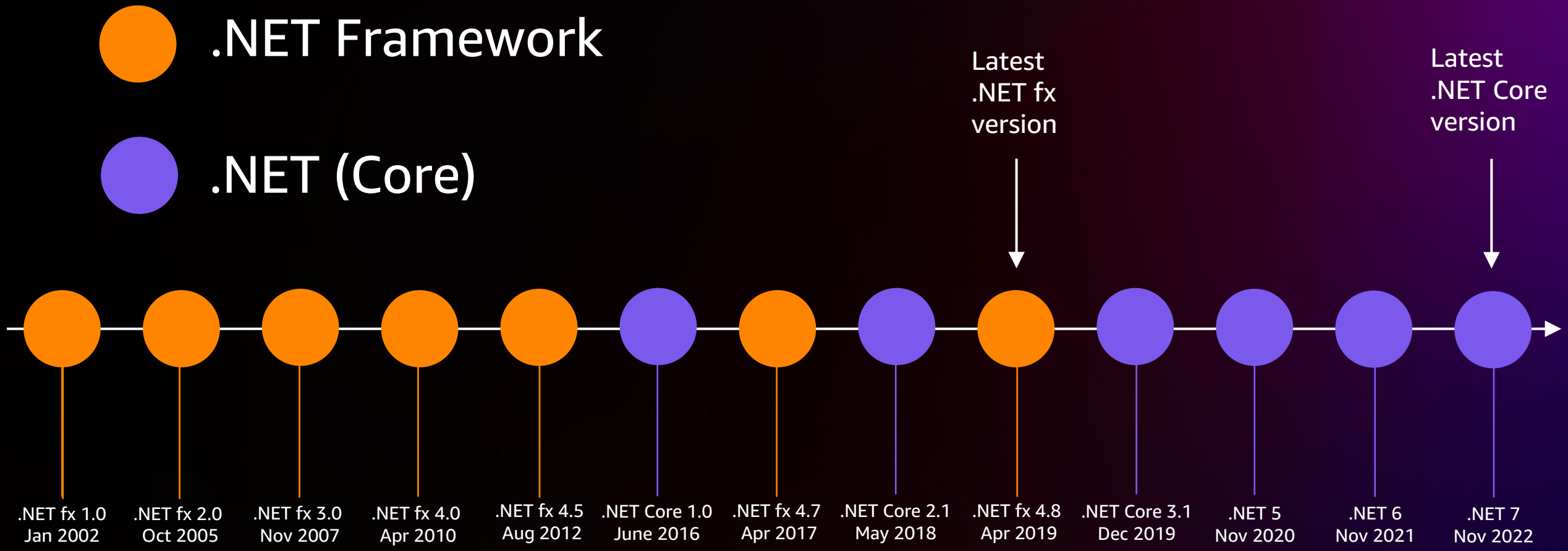
© 2022, Amazon Web Services, Inc. or its affiliates. All rights reserved.

Backend services



aws
© 2022, Amazon Web Services, Inc. or its affiliates. All rights reserved.

.NET versions



© 2022, Amazon Web Services, Inc. or its affiliates. All rights reserved.

Compute

	Benefits	.NET (Core)	.NET Framework	Management
Amazon EC2	• AWS Graviton • Amazon EC2 Spot Instances • AWS Savings Plans	• Windows/Linux • Gain in performance for Graviton instances when using .NET 5+	• Windows only • Significantly costlier as Windows license adds to the running cost	Auto Scaling group
AWS Lambda	• Serverless func • Good for one-off tasks • Event driven, pay per use	• .NET Core can run on lambdas • Utilize Native AOT for .NET 7	• Cannot run .NET framework applications on lambdas	AWS Step Functions Amazon Managed Workflows for Apache Airflow

Containers	• Package dependencies
	• Isolate software
	• Higher utilization
		• Container OS
		  • Linux
		  • Windows Nano Server
		  • Windows Server Core
		  • Windows Server (full)
		• Lightweight images: Linux & Nano Server
			• Container OS
			  • Windows Server Core
			  • Windows Server (full)
			• Heavy images and slow to start
				Container orchestration services

aws
© 2022, Amazon Web Services, Inc. or its affiliates. All rights reserved.

Picking a backend service cheat sheet

Older monolithic applications, has OS dependencies (like Windows Registry), have third-party software dependencies that can only be installed on a VM

- Amazon EC2

Serverless, functions, intermittent short tasks, quick event-driven actions originating from other AWS services

- AWS Lambda

Requires complex orchestration, no hard dependencies on OS services that are not available to containers, want granular control on compute scaling, long-running tasks

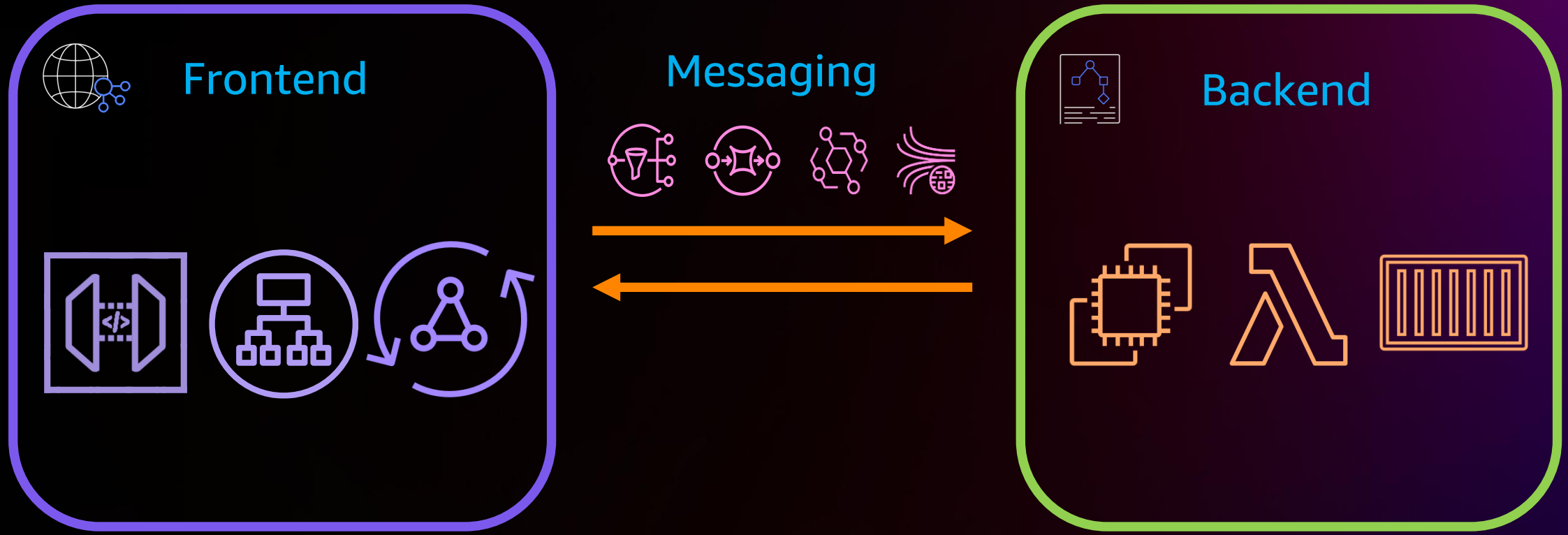
- Containers

For managing events between tiers/microservices, use event choreography with the help of one of the messaging services we discussed earlier; for managing business logic within a tier, use event orchestration with the help of orchestration or workflow services

aws

© 2022, Amazon Web Services, Inc. or its affiliates. All rights reserved.

A standard architecture



© 2022, Amazon Web Services, Inc. or its affiliates. All rights reserved.

Container landscape on AWS

IMAGE REGISTRY

- Amazon Elastic Container Registry (Amazon ECR)
- Amazon ECR Public Gallery

ORCHESTRATION

- Amazon Elastic Container Service (Amazon ECS)
- Amazon Elastic Kubernetes Service (Amazon EKS)
- AWS App Runner
- Red Hat OpenShift on AWS (ROSA)

COMPUTE

- Amazon Elastic Compute Cloud (Amazon EC2)
- AWS Fargate
- Containers on Lambda

aws

© 2022, Amazon Web Services, Inc. or its affiliates. All rights reserved.

Picking a container service cheat sheet

Completely serverless (both control and data plane) containerization option

- AWS AppRunner

Want to use serverless functions in your workflows and use the logic hosted on containers

- Containers on AWS Lambda

Powerful simplicity, run containers at scale

- Amazon ECS

Managed Kubernetes service

- Amazon EKS

Turnkey platform to use OpenShift

- Red Hat OpenShift Service on AWS (ROSA)

With Amazon ECS and Amazon EKS, if you are looking for serverless compute engine

- Use AWS Fargate with them, else use Amazon EC2 which gives you more control

© 2022, Amazon Web Services, Inc. or its affiliates. All rights reserved.

AWS for the Edge



AWS Outposts



AWS Wavelength



AWS Local Zones

© 2022, Amazon Web Services, Inc. or its affiliates. All rights reserved.

Running containers on premises



Amazon ECS Anywhere



Amazon EKS Anywhere

aws

© 2022, Amazon Web Services, Inc. or its affiliates. All rights reserved.

Hybrid Service decider cheat sheet

Run near end users (accessing over 5G) requiring ultra low latency

- AWS Wavelength

Run near end users requiring low latency

- AWS Local Zones

Run near or inside customer's data center for low latency data access or compliance

- AWS Outposts

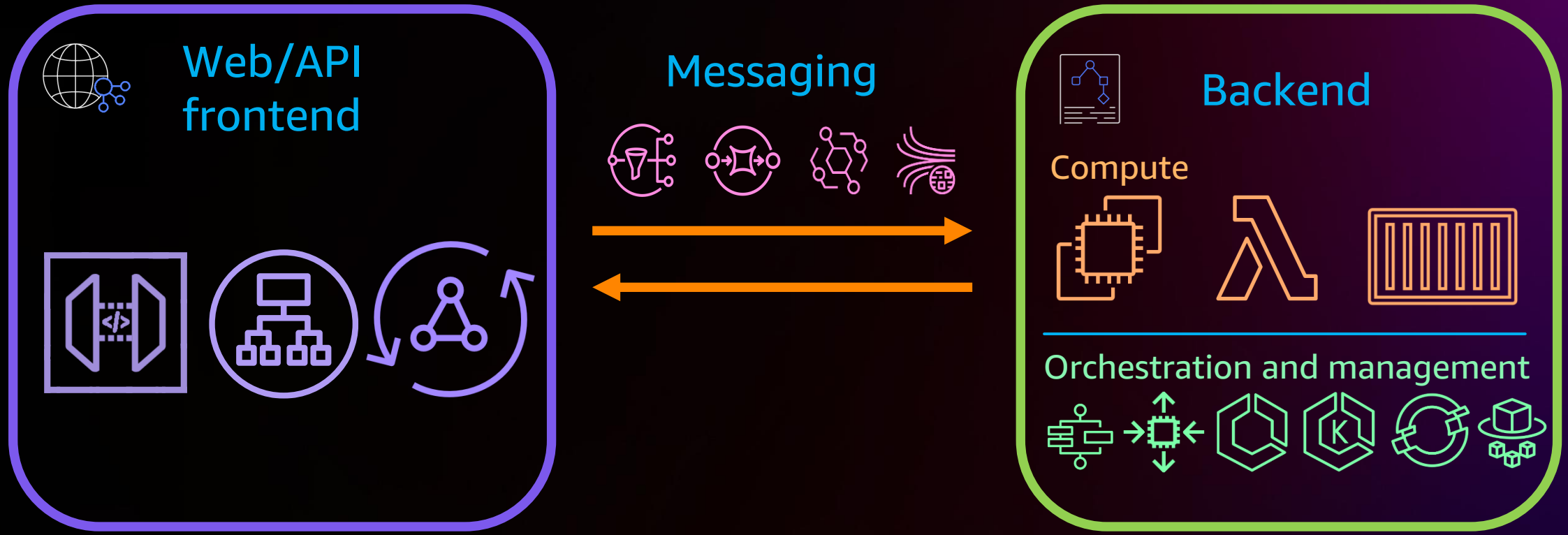
For container options on edge services

- With control plane in AWS Region
  - Either Amazon ECS or Amazon EKS with any of the edge services
- With control plane on the Edge
  - AWS Outposts with EKS control plane

aws

© 2022, Amazon Web Services, Inc. or its affiliates. All rights reserved.

A standard architecture – Complete view



© 2022, Amazon Web Services, Inc. or its affiliates. All rights reserved.

Thank you!

Rahul Chugh
imRahulChugh
/in/rchugh/

Saleha Haider
iSalehaHaider
/in/saleha-haider/



Please complete the session survey in the **mobile app**

© 2022, Amazon Web Services, Inc. or its affiliates. All rights reserved.