AWS re:Inforce

JUNE 10 – 12, 2024 | PHILADELPHIA, PA

DAP301

Building resilient event-driven architectures, feat. United Airlines

Hemal Jani
(he/him)
Sr. Solutions Architect
AWS

Isael Pelletier
(he/him)
Pr. Solutions Architect
AWS

Vinay Patil
(he/him)
Principal Architect – PSS System
United Airlines

© 2024, Amazon Web Services, Inc. or its affiliates. All rights reserved.

Where are we going today?

1 Why event-driven architectures?

2 Best practices for governance & resiliency

3 Data protection mechanisms

4 Customer story – United Airlines mainframe modernization

© 2024, Amazon Web Services, Inc. or its affiliates. All rights reserved.

Application modernization

© 2024, Amazon Web Services, Inc. or its affiliates. All rights reserved.

Business requirements

Go faster

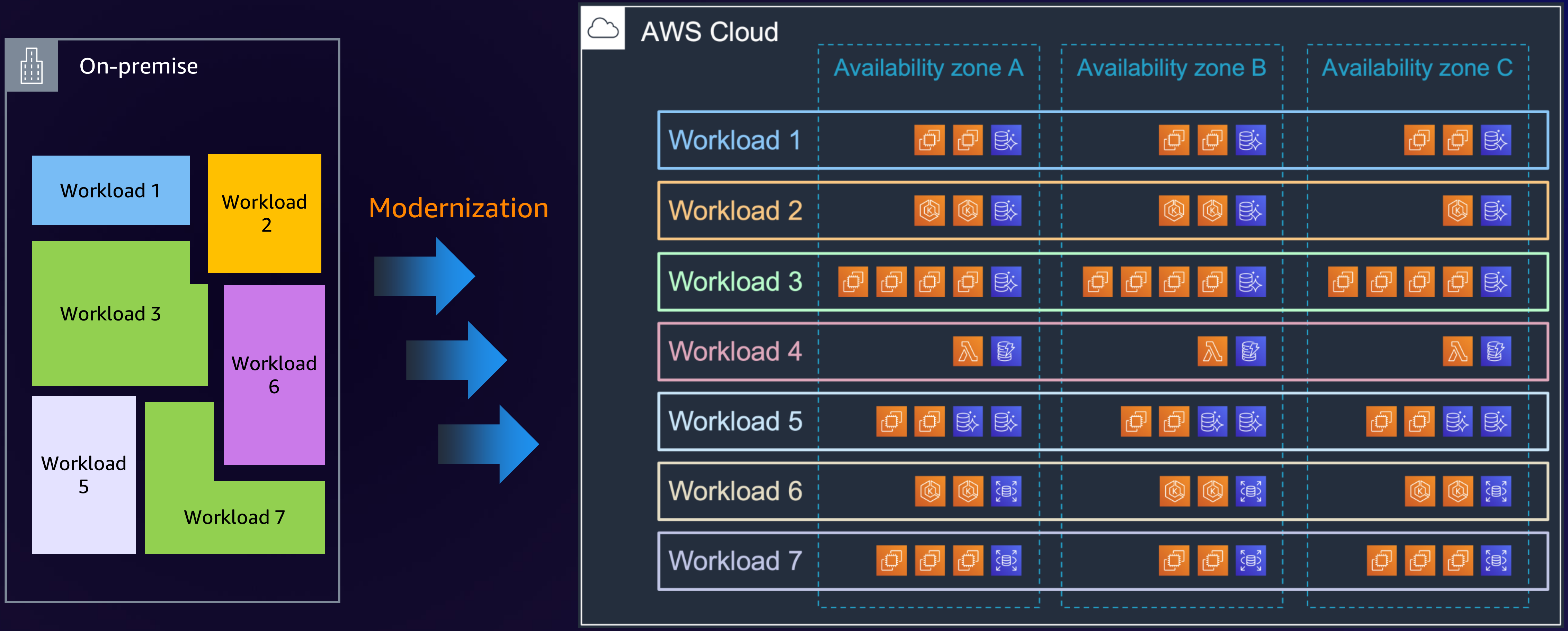
Be more stable

Increase quality

Be cheaper

© 2024, Amazon Web Services, Inc. or its affiliates. All rights reserved.

Strangling workload by workload



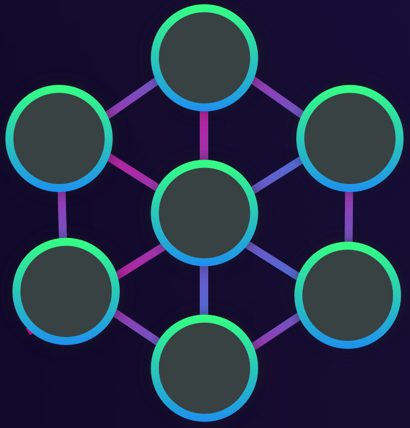
Guidance:
https://docs.aws.amazon.com/prescriptive-guidance/latest/modernization-decomposing-monoliths/welcome.html

aws

© 2024, Amazon Web Services, Inc. or its affiliates. All rights reserved.

Architecture review

Monolith
Does everything



Microservices
Does one thing

Scales to millions of users

Has global availability

Responds in milliseconds

Handles petabytes of data

Microservices approach advocates creating a system from a collection of small, isolated services, each of which owns their data and is independently isolated, scalable, and resilient to failure

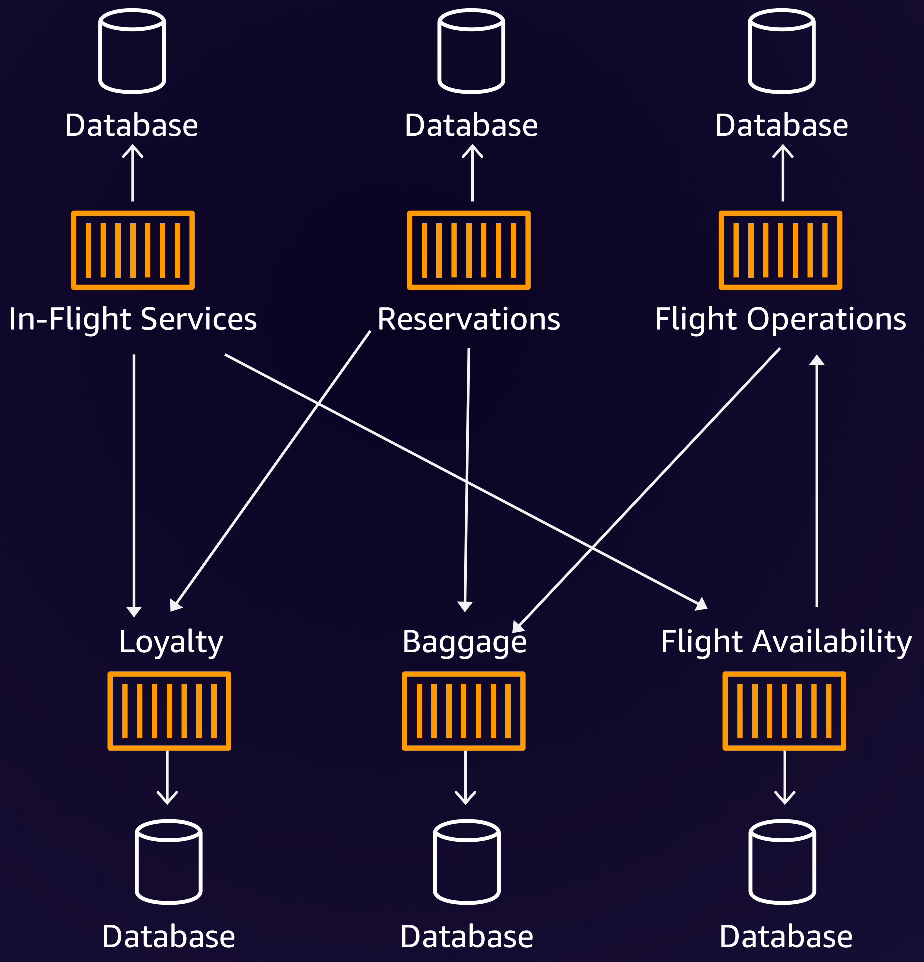
aws
© 2024, Amazon Web Services, Inc. or its affiliates. All rights reserved.

Why event-driven architecture

© 2024, Amazon Web Services, Inc. or its affiliates. All rights reserved.

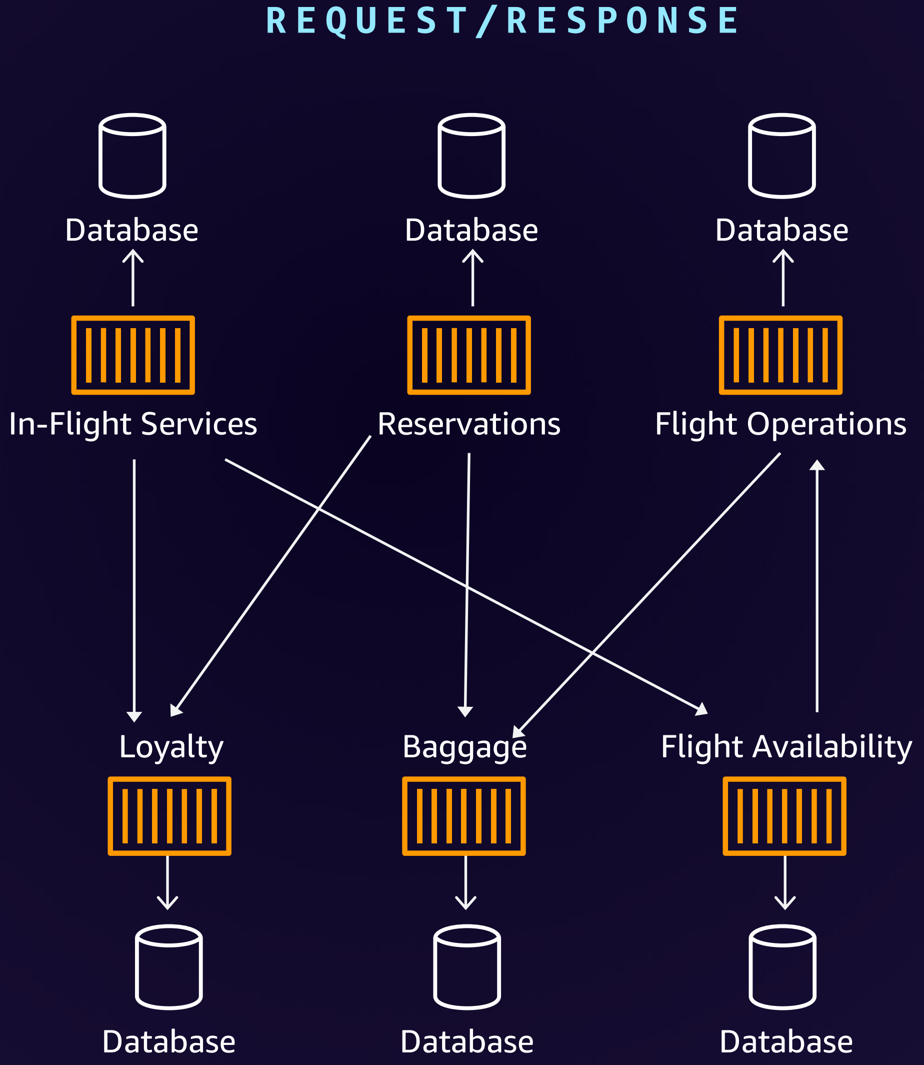
Loosely couple microservices

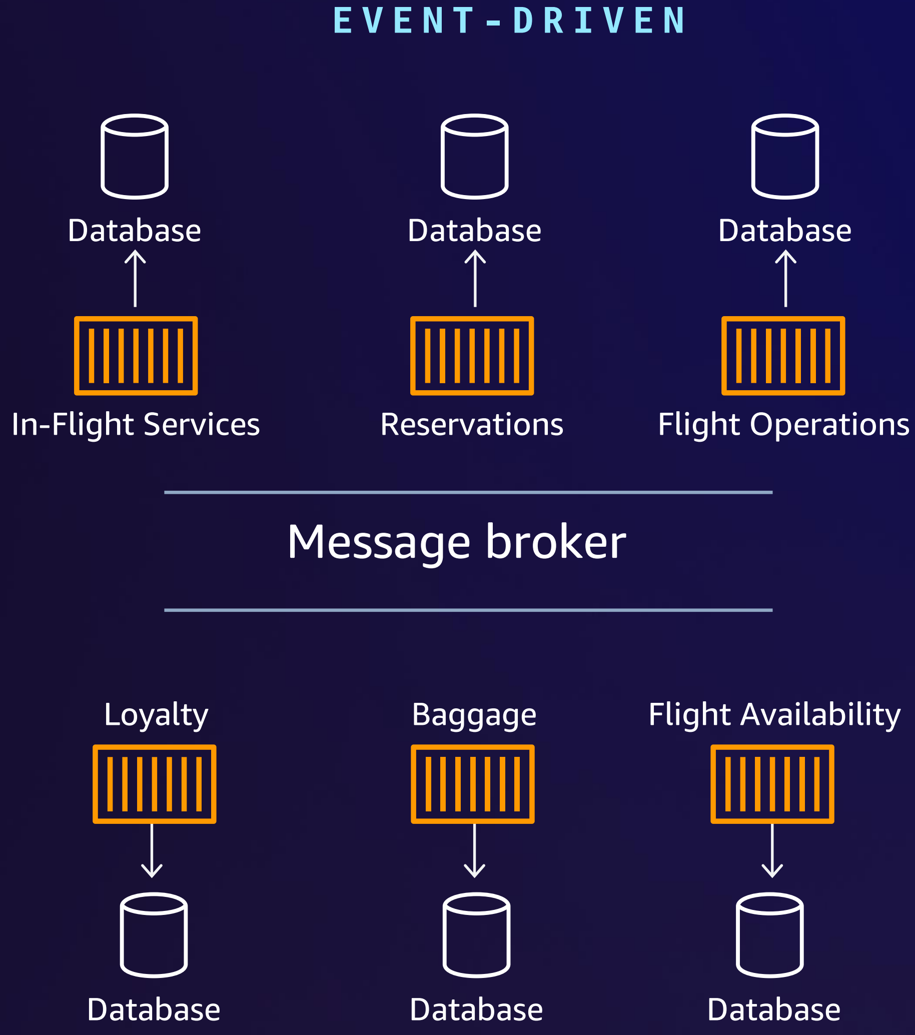
REQUEST/RESPONSE



© 2024, Amazon Web Services, Inc. or its affiliates. All rights reserved.

Loosely couple microservices





© 2024, Amazon Web Services, Inc. or its affiliates. All rights reserved.

Event-driven architectures

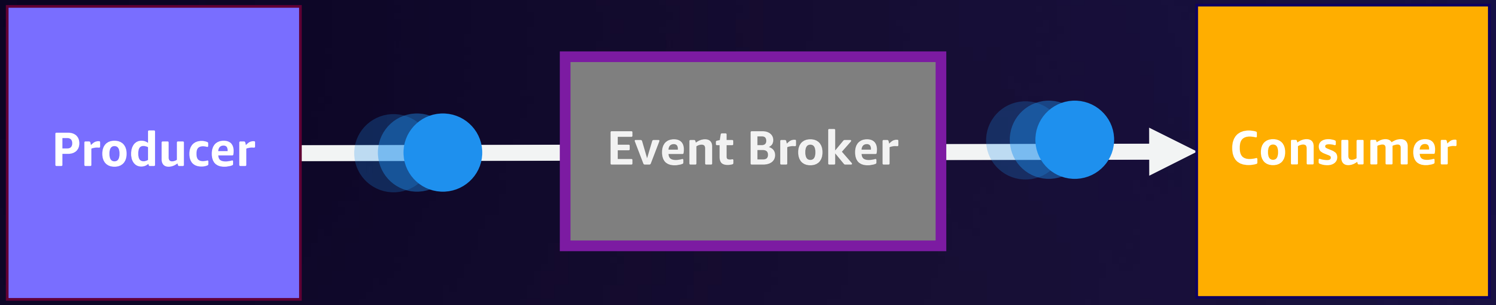
An architectural style of building loosely-coupled software systems that work together by emitting and responding to events

“If your application is cloud-native, or large-scale, or distributed, and doesn’t include a messaging component, that’s probably a bug.”

Source:
https://aws.amazon.com/blogs/compute/understanding-asynchronous-messaging-for-microservices/

© 2024, Amazon Web Services, Inc. or its affiliates. All rights reserved.

At the core of event-driven architecture



© 2024, Amazon Web Services, Inc. or its affiliates. All rights reserved.

AWS messaging & event services

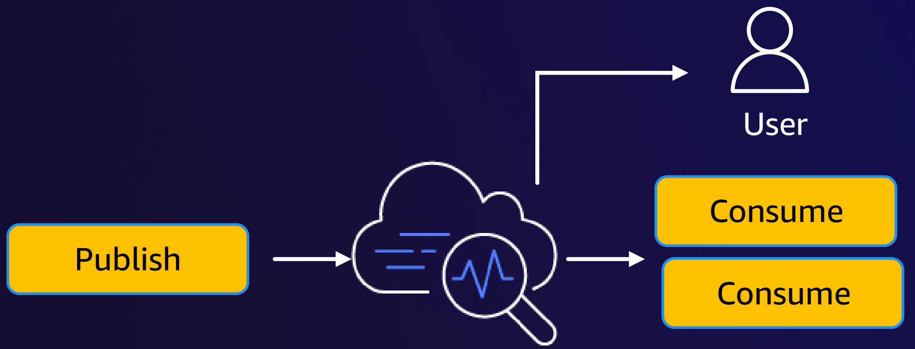
Born in the Cloud | Micro Services

Amazon SQS

Amazon SNS

Amazon EventBridge

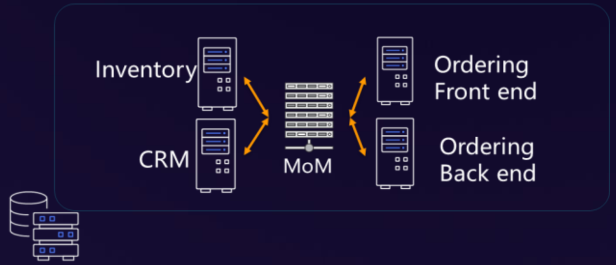
Event Bus & SaaS



Amazon EventBridge

Amazon Pinpoint

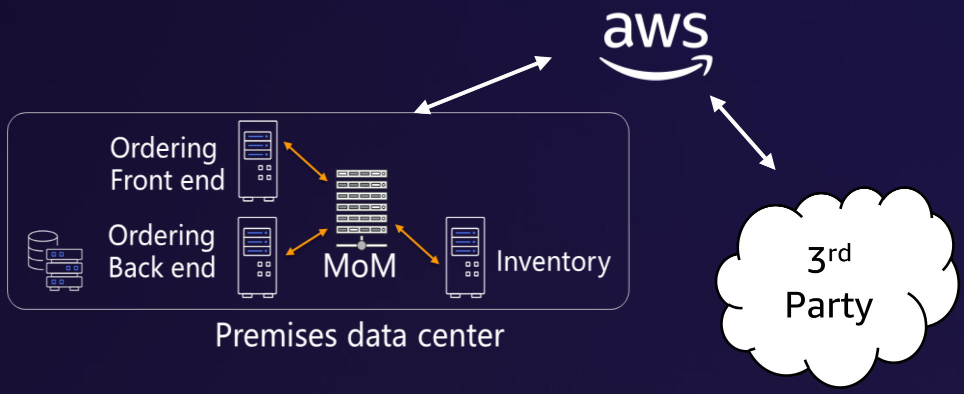
Cloud Migration & Modernization



Amazon MQ

Amazon MSK

Hybrid & Multicloud



Amazon MQ

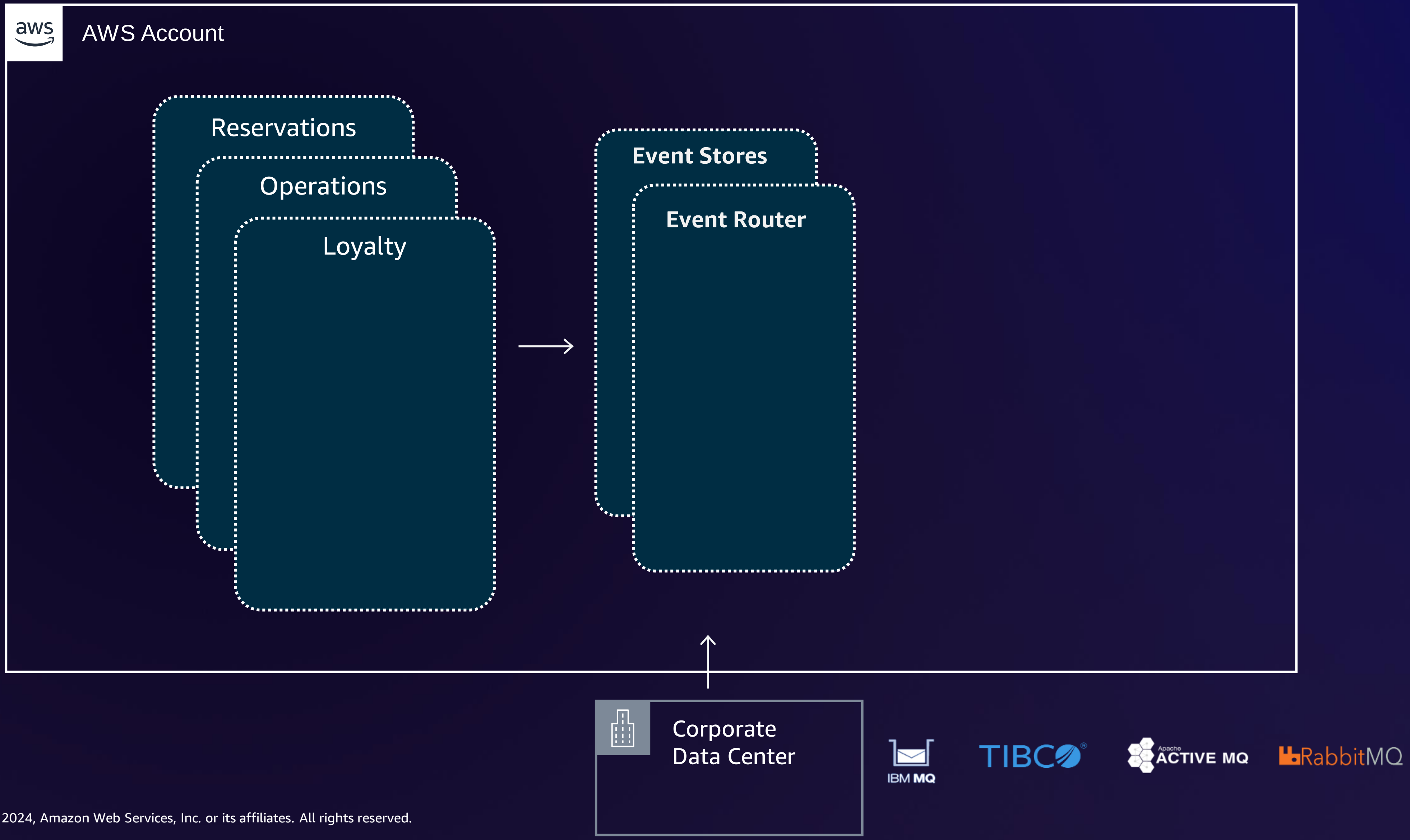
Amazon MSK

© 2024, Amazon Web Services, Inc. or its affiliates. All rights reserved.

Governance architecture

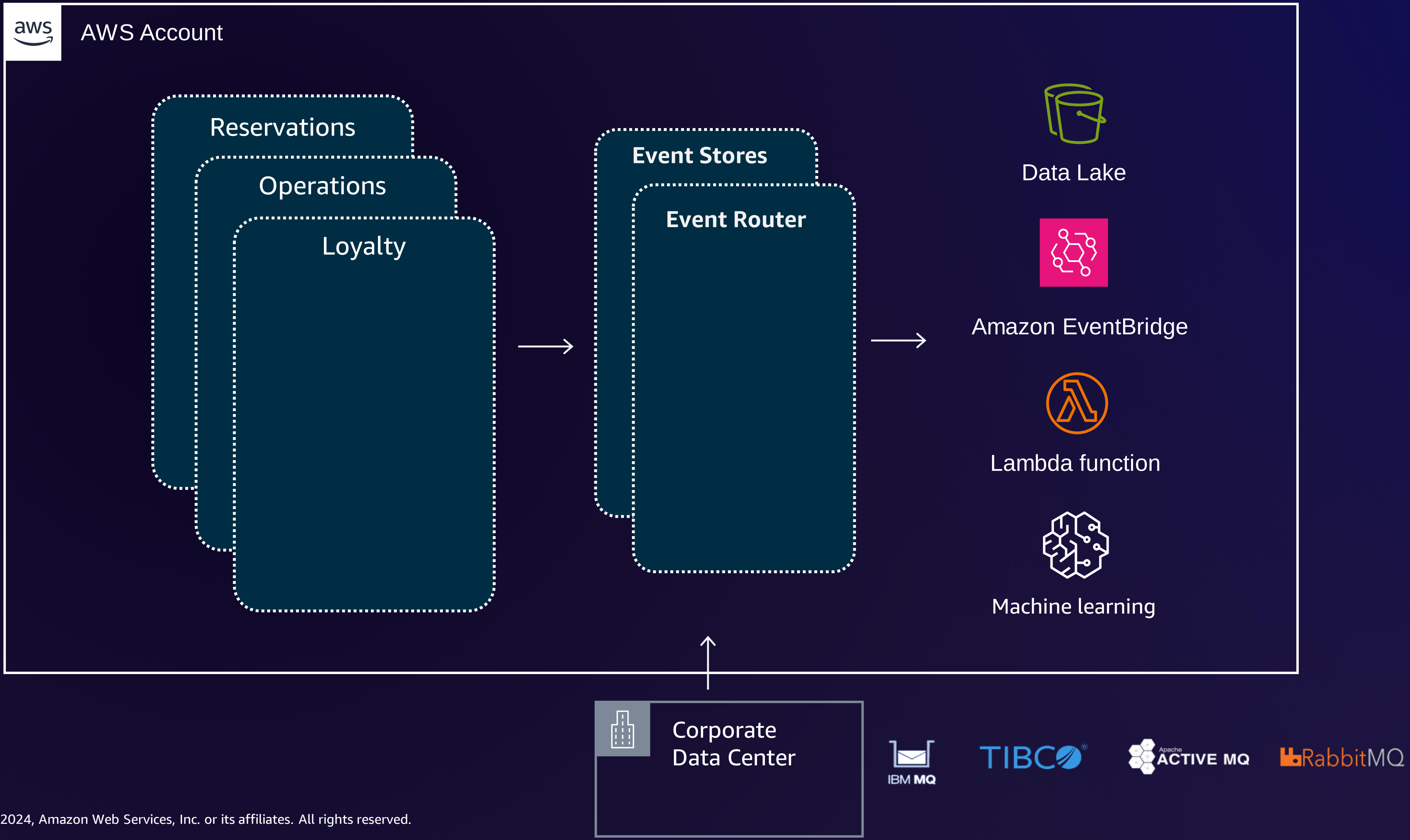
© 2024, Amazon Web Services, Inc. or its affiliates. All rights reserved.

Event-driven architecture implementation



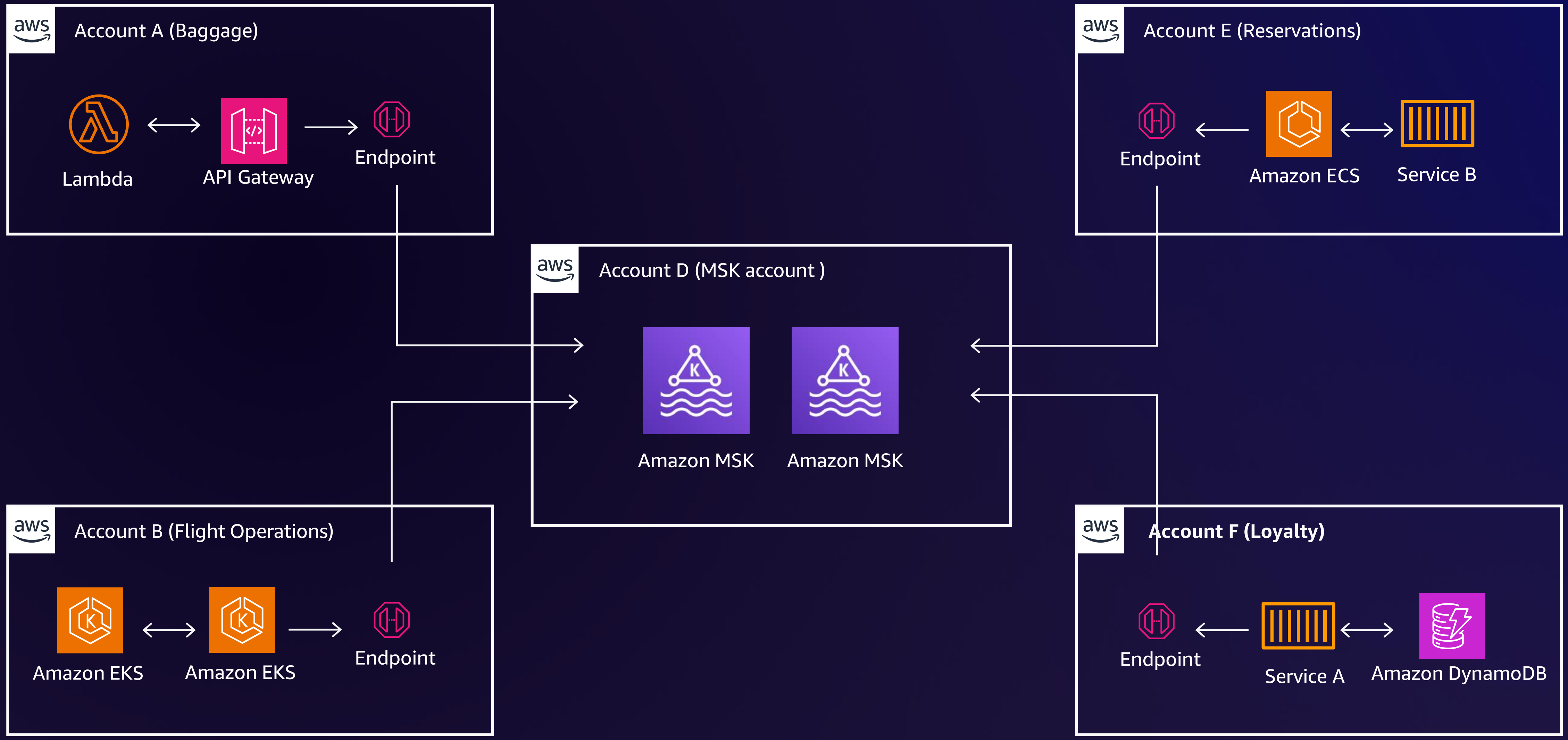
© 2024, Amazon Web Services, Inc. or its affiliates. All rights reserved.

Event-driven architecture implementation



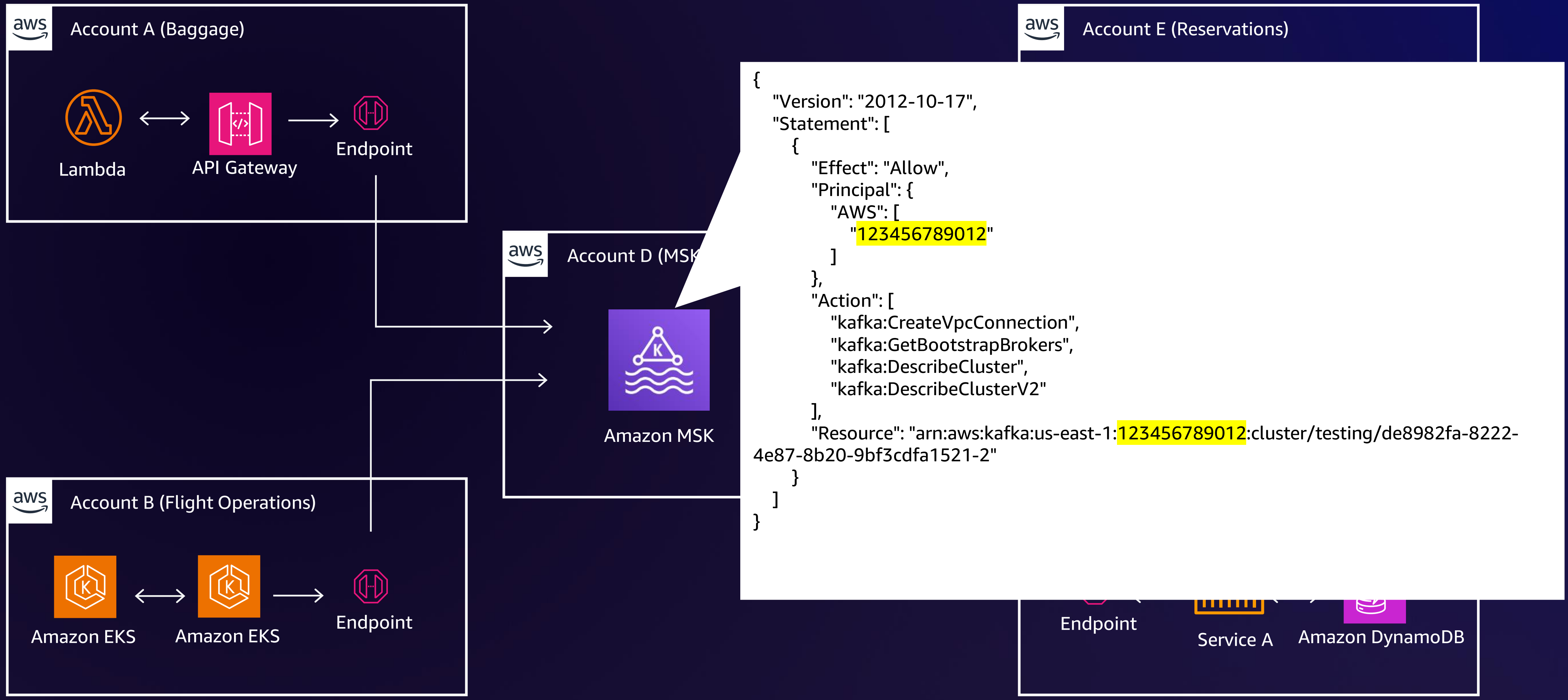
© 2024, Amazon Web Services, Inc. or its affiliates. All rights reserved.

Micro account architectures



© 2024, Amazon Web Services, Inc. or its affiliates. All rights reserved.

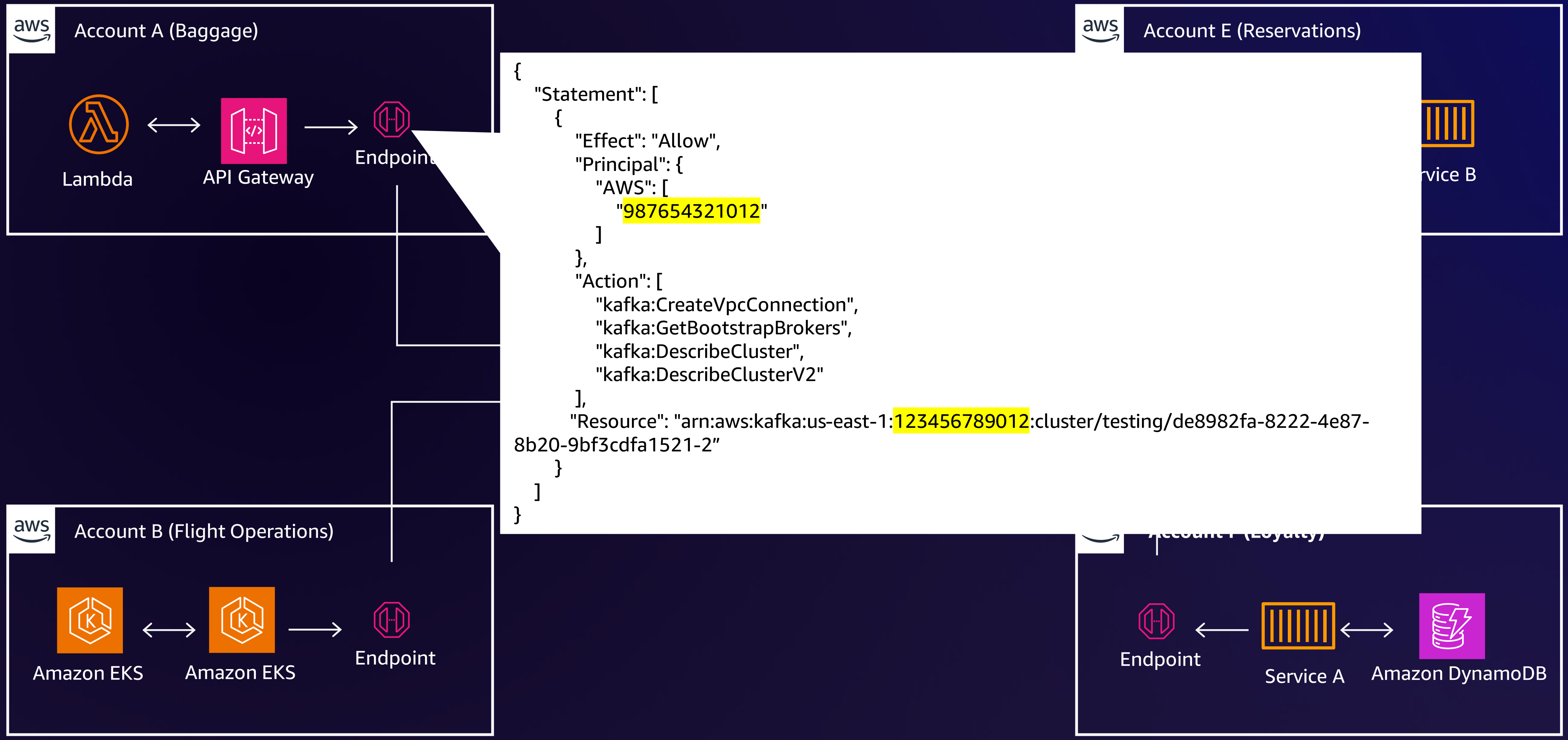
Micro account architectures



```
{
   "Version": "2012-10-17",
   "Statement": [
      {
         "Effect": "Allow",
         "Principal": {
            "AWS": [
               "123456789012"
            ]
         },
         "Action": [
            "kafka:CreateVpcConnection",
            "kafka:GetBootstrapBrokers",
            "kafka:DescribeCluster",
            "kafka:DescribeClusterV2"
         ],
         "Resource": "arn:aws:kafka:us-east-1:123456789012:cluster/testing/de8982fa-8222-
4e87-8b20-9bf3cdfa1521-2"
      }
   ]
}
```

© 2024, Amazon Web Services, Inc. or its affiliates. All rights reserved.

Micro account architectures

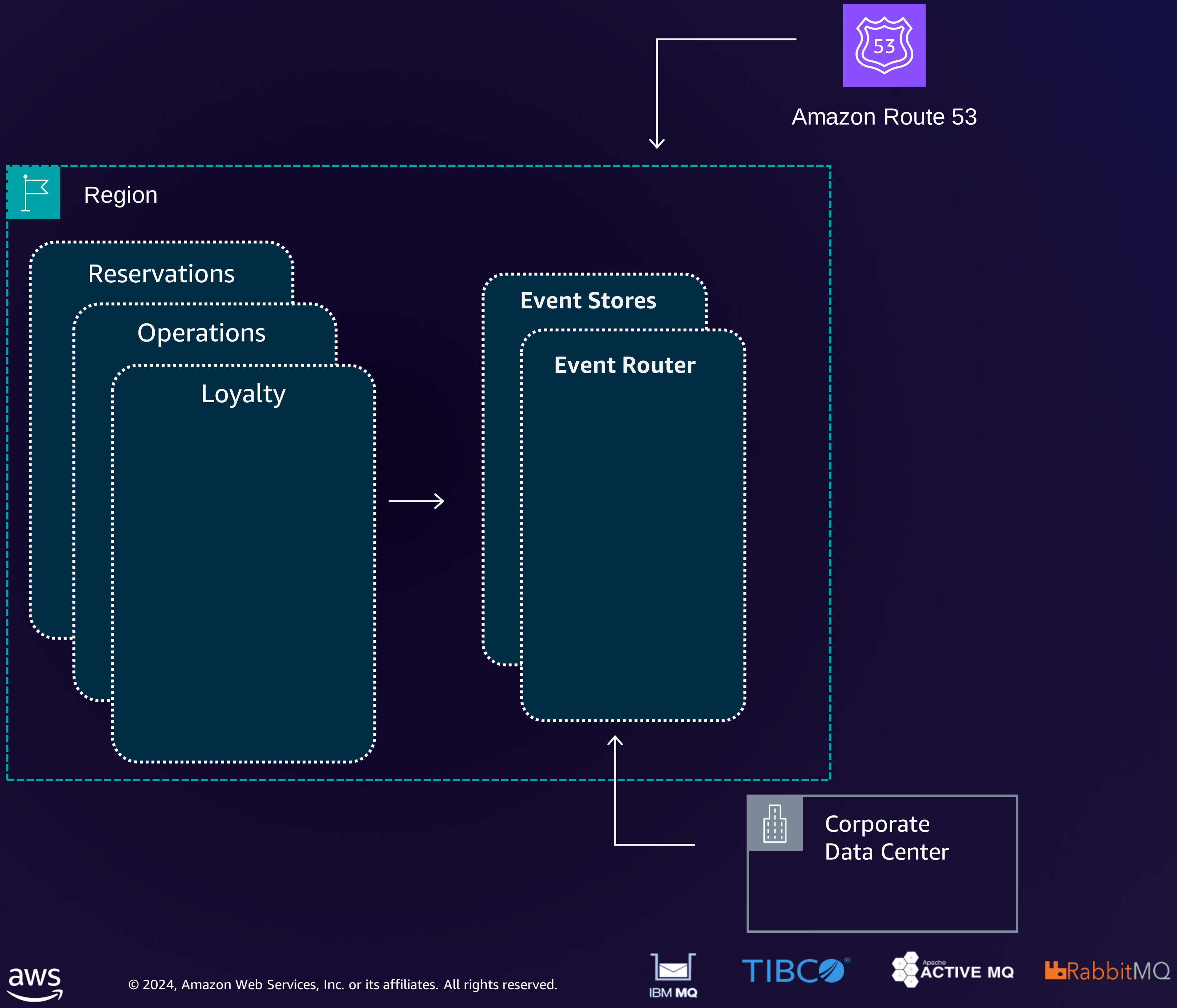


```
{
   "Statement": [
      {
         "Effect": "Allow",
         "Principal": {
            "AWS": [
               "987654321012"
            ]
         },
         "Action": [
            "kafka:CreateVpcConnection",
            "kafka:GetBootstrapBrokers",
            "kafka:DescribeCluster",
            "kafka:DescribeClusterV2"
         ],
        "Resource": "arn:aws:kafka:us-east-1:123456789012:cluster/testing/de8982fa-8222-4e87-
8b20-9bf3cdfa1521-2”
      }
   ]
}
```

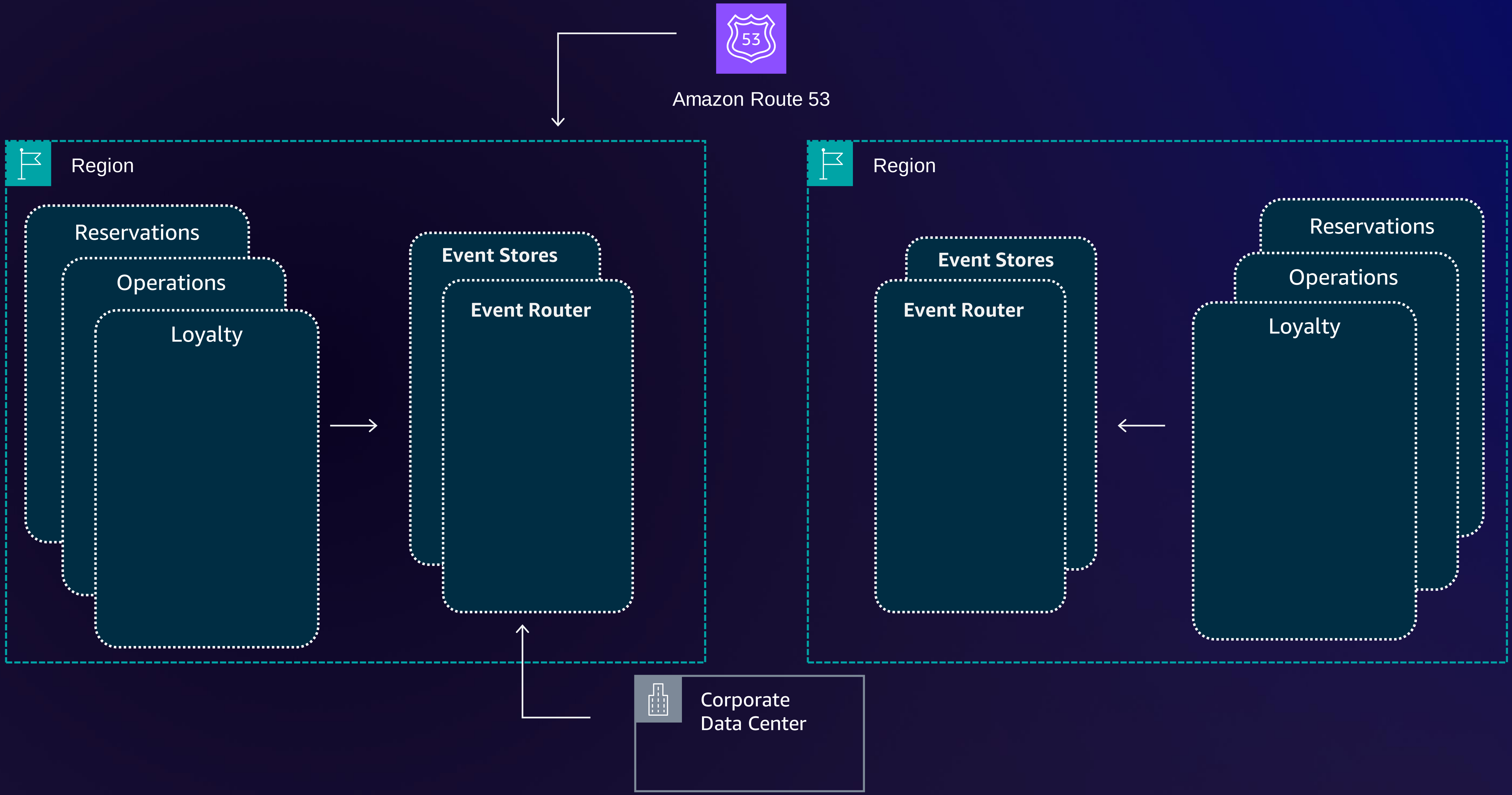
© 2024, Amazon Web Services, Inc. or its affiliates. All rights reserved.

Cross-Region architecture

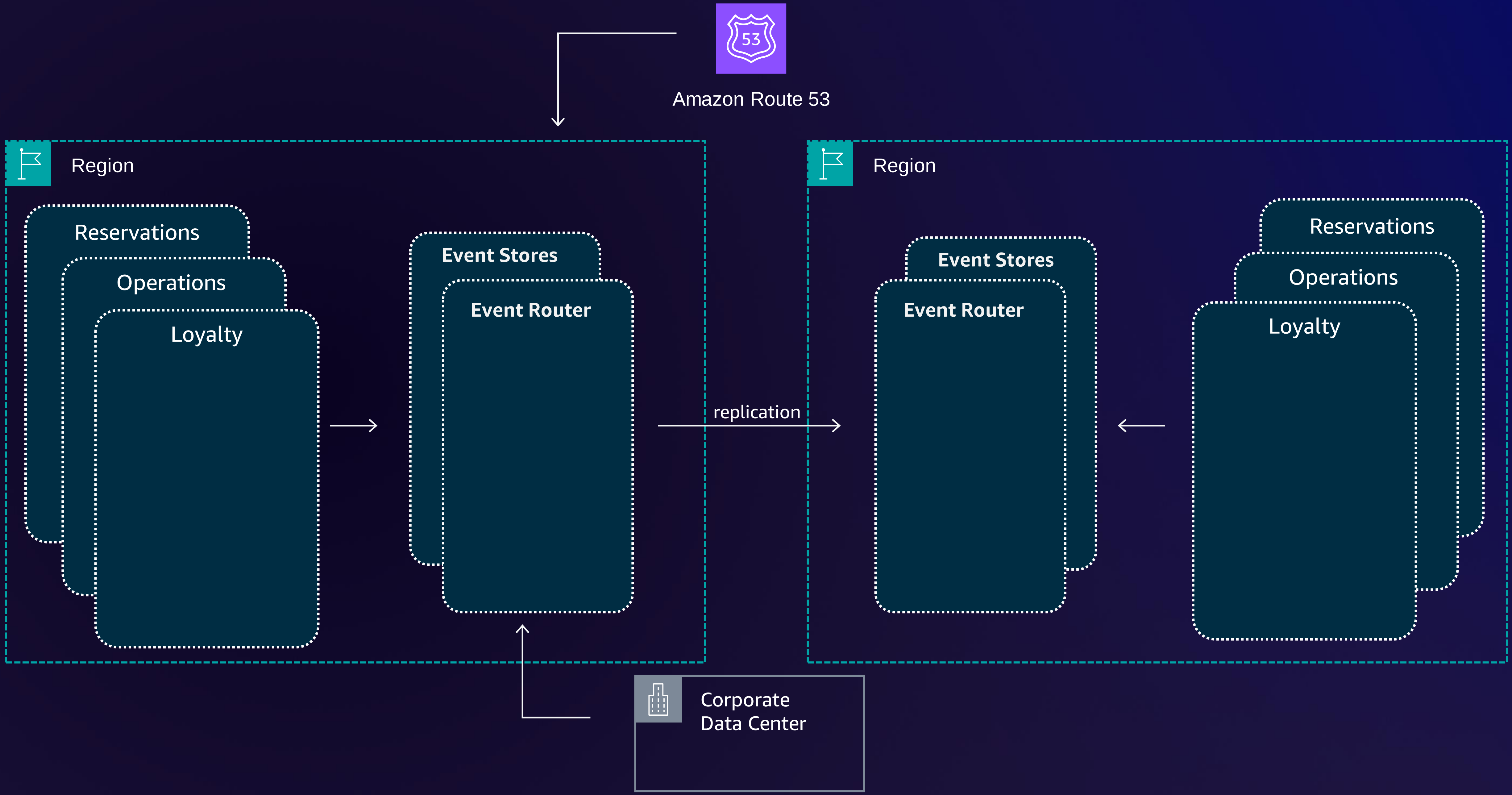
© 2024, Amazon Web Services, Inc. or its affiliates. All rights reserved.



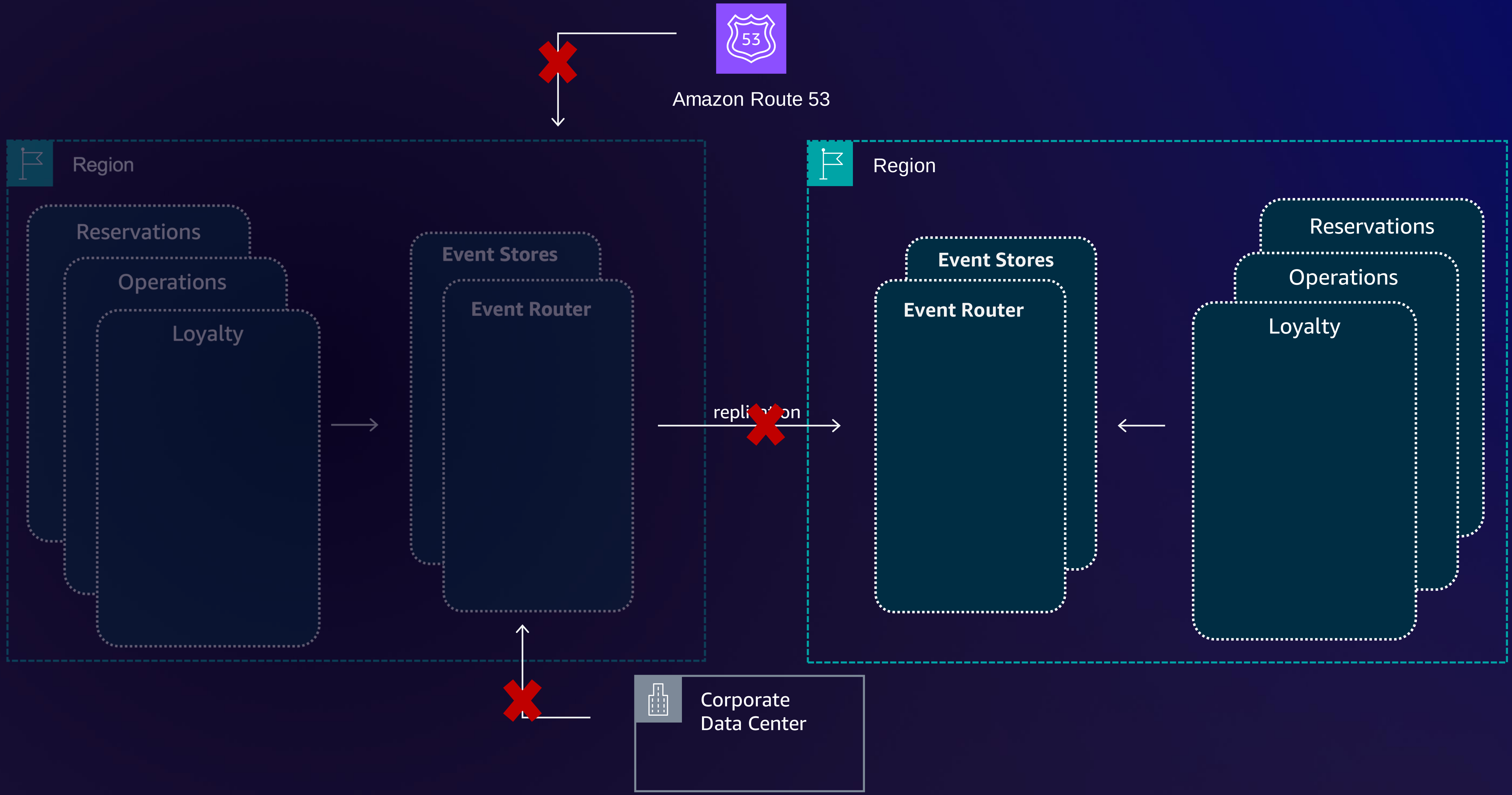
aws
© 2024, Amazon Web Services, Inc. or its affiliates. All rights reserved.



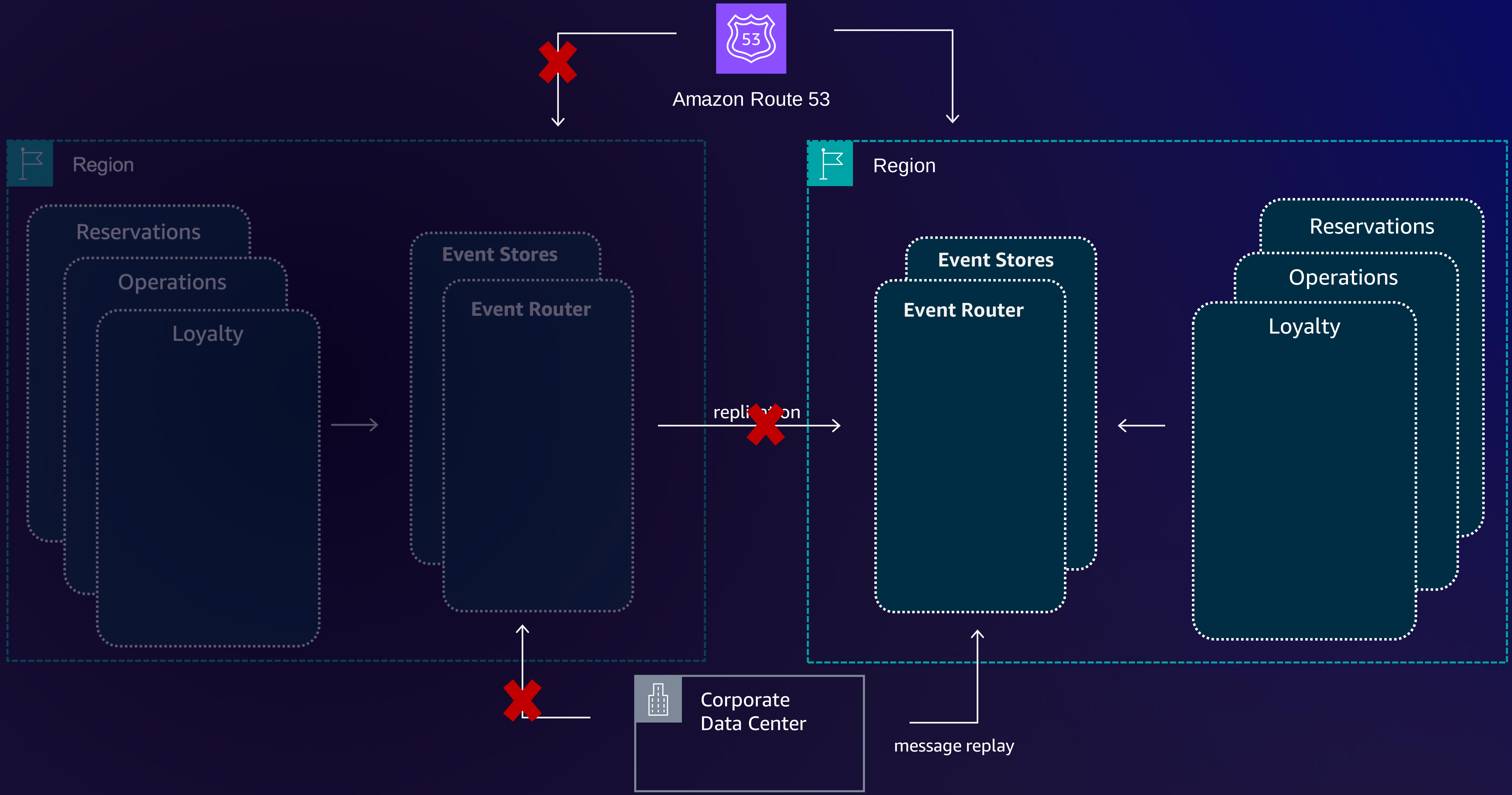
© 2024, Amazon Web Services, Inc. or its affiliates. All rights reserved.



© 2024, Amazon Web Services, Inc. or its affiliates. All rights reserved.

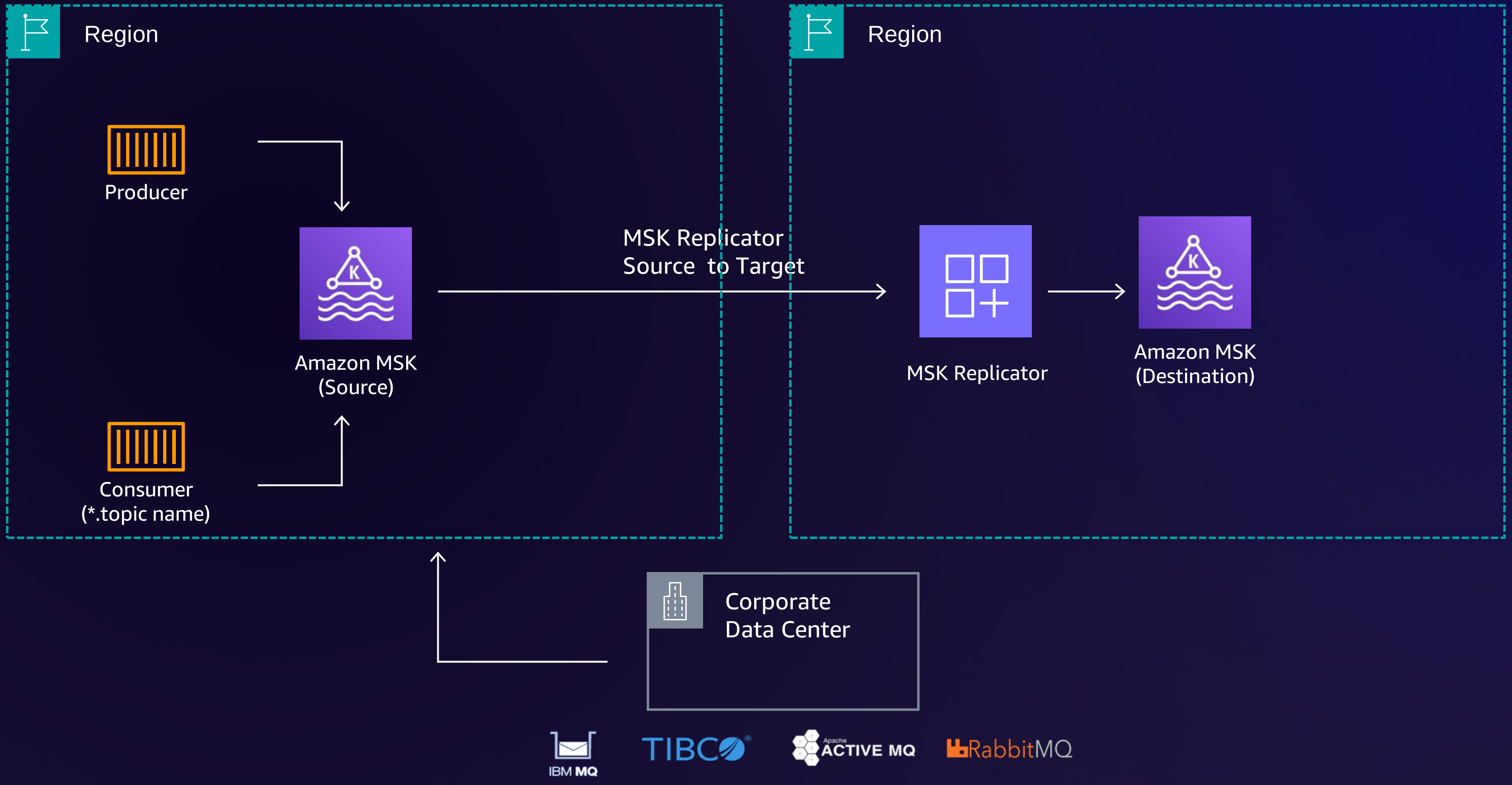


© 2024, Amazon Web Services, Inc. or its affiliates. All rights reserved.



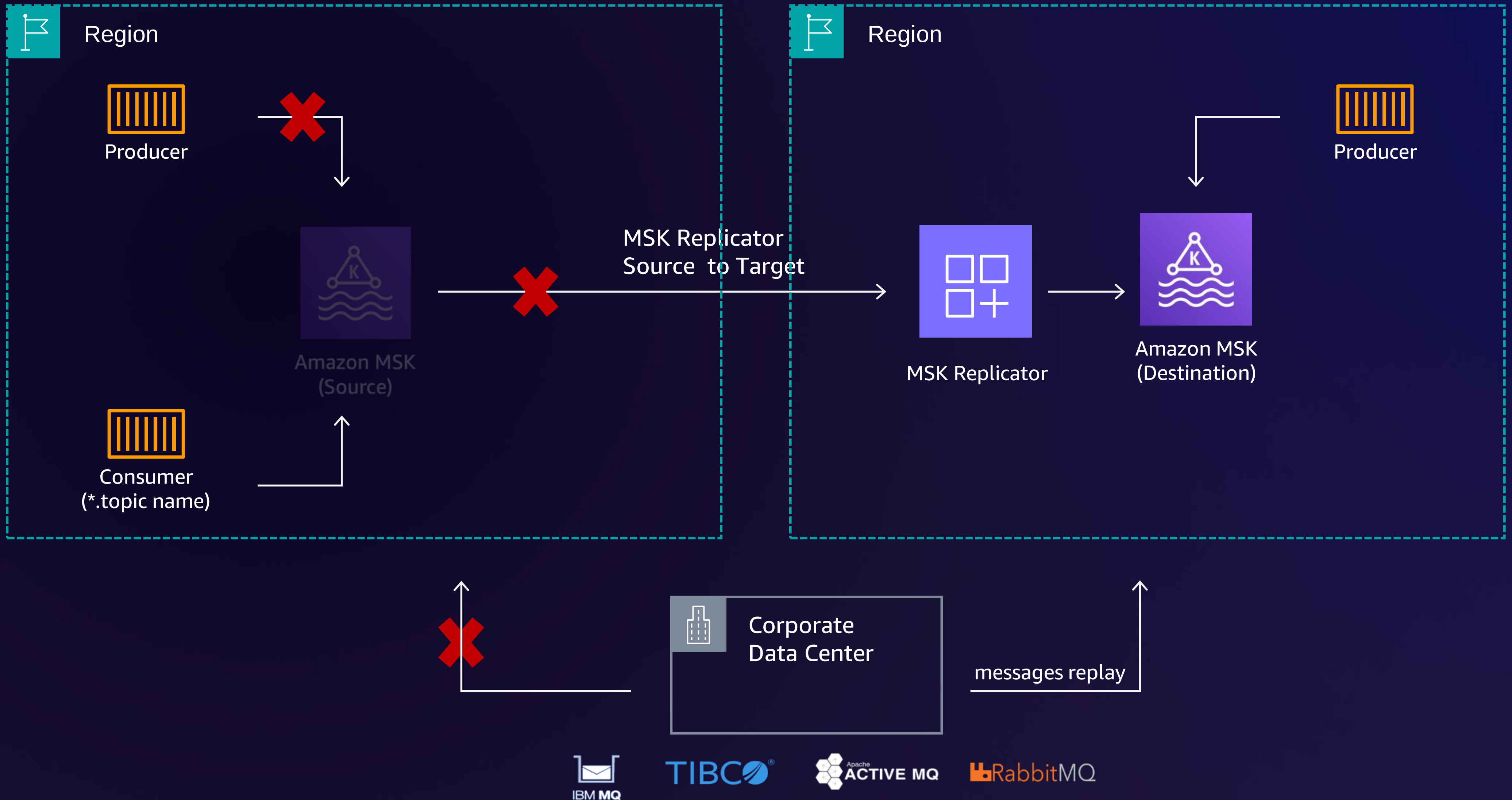
© 2024, Amazon Web Services, Inc. or its affiliates. All rights reserved.

Amazon MSK Replicator



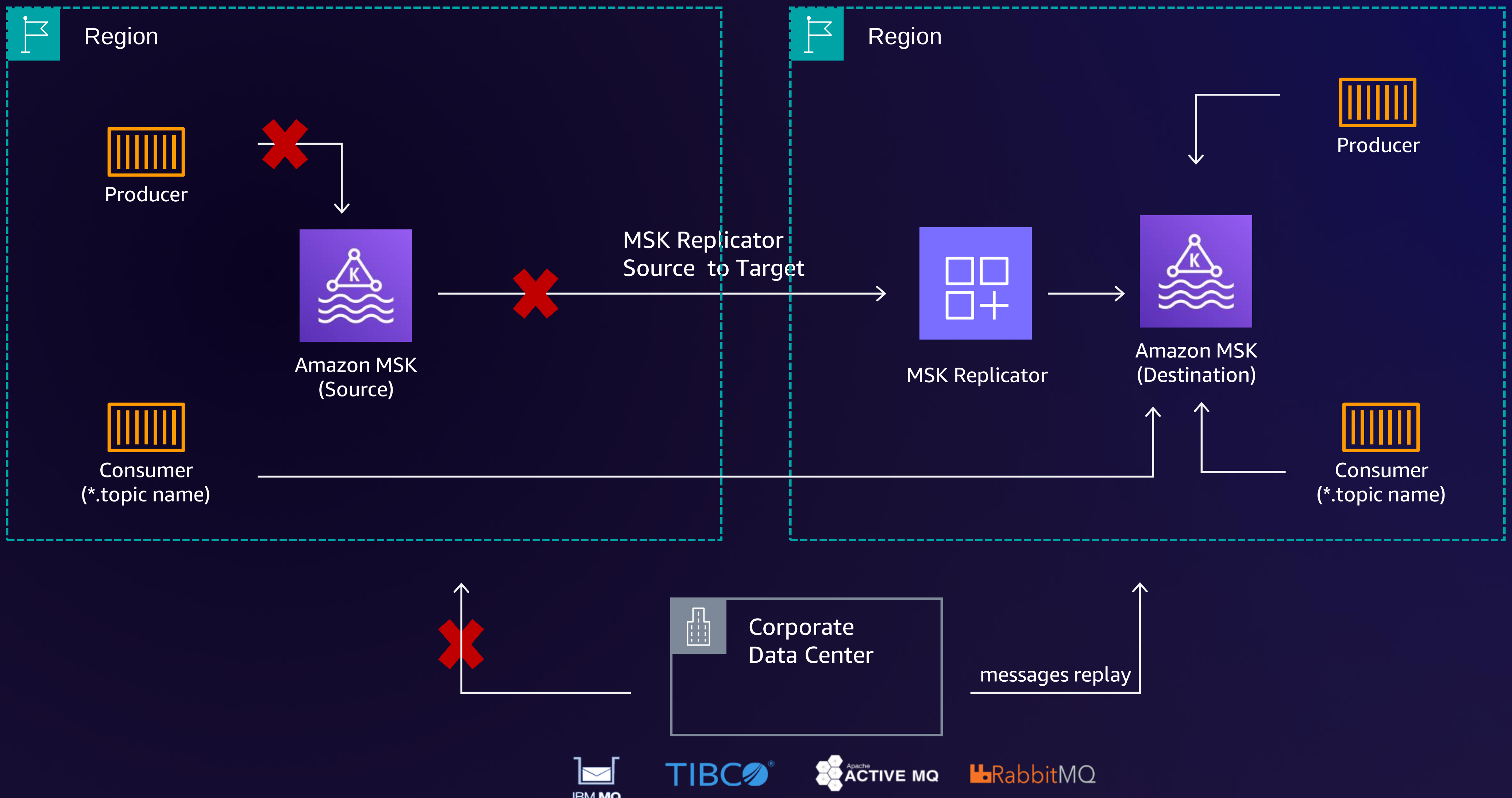
© 2024, Amazon Web Services, Inc. or its affiliates. All rights reserved.

Amazon MSK Replicator



© 2024, Amazon Web Services, Inc. or its affiliates. All rights reserved.

Amazon MSK Replicator



© 2024, Amazon Web Services, Inc. or its affiliates. All rights reserved.

Other considerations

1. Failover process automation
2. Asynchronous replication
3. Strong observability practice & health checks
4. Latency across Regions
5. Fine-grained access to secrets

© 2024, Amazon Web Services, Inc. or its affiliates. All rights reserved.

Data protection for event-driven architecture

© 2024, Amazon Web Services, Inc. or its affiliates. All rights reserved.

Characteristics of EDA

In event-driven architecture (EDA) focuses on the production, detection, and reaction to events

Events can **propagate across a network**, updating data stores as they go, and can be initiated anywhere

It consists of **event emitters** (or agents), **event consumers** (or sinks), and **event channels**, with a focus on **decoupling producers and consumers** of events to allow for independent operation and scalability

© 2024, Amazon Web Services, Inc. or its affiliates. All rights reserved.

Characteristics of EDA

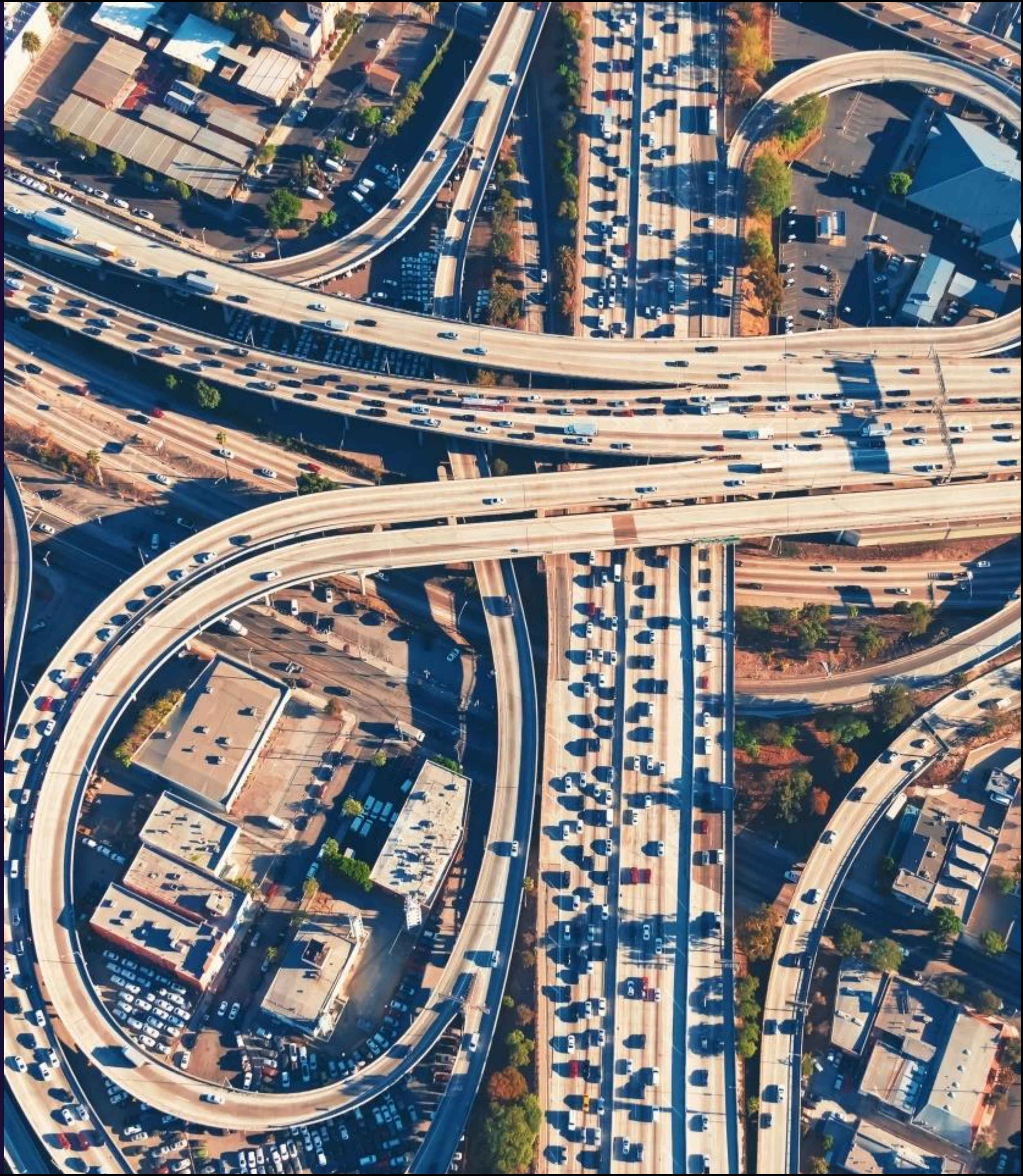
Data in movement

Across locations
on-premises <-> AWS <-> Regions & AZs

Across system components
producers, channels, consumers & storages

On top of a network layer

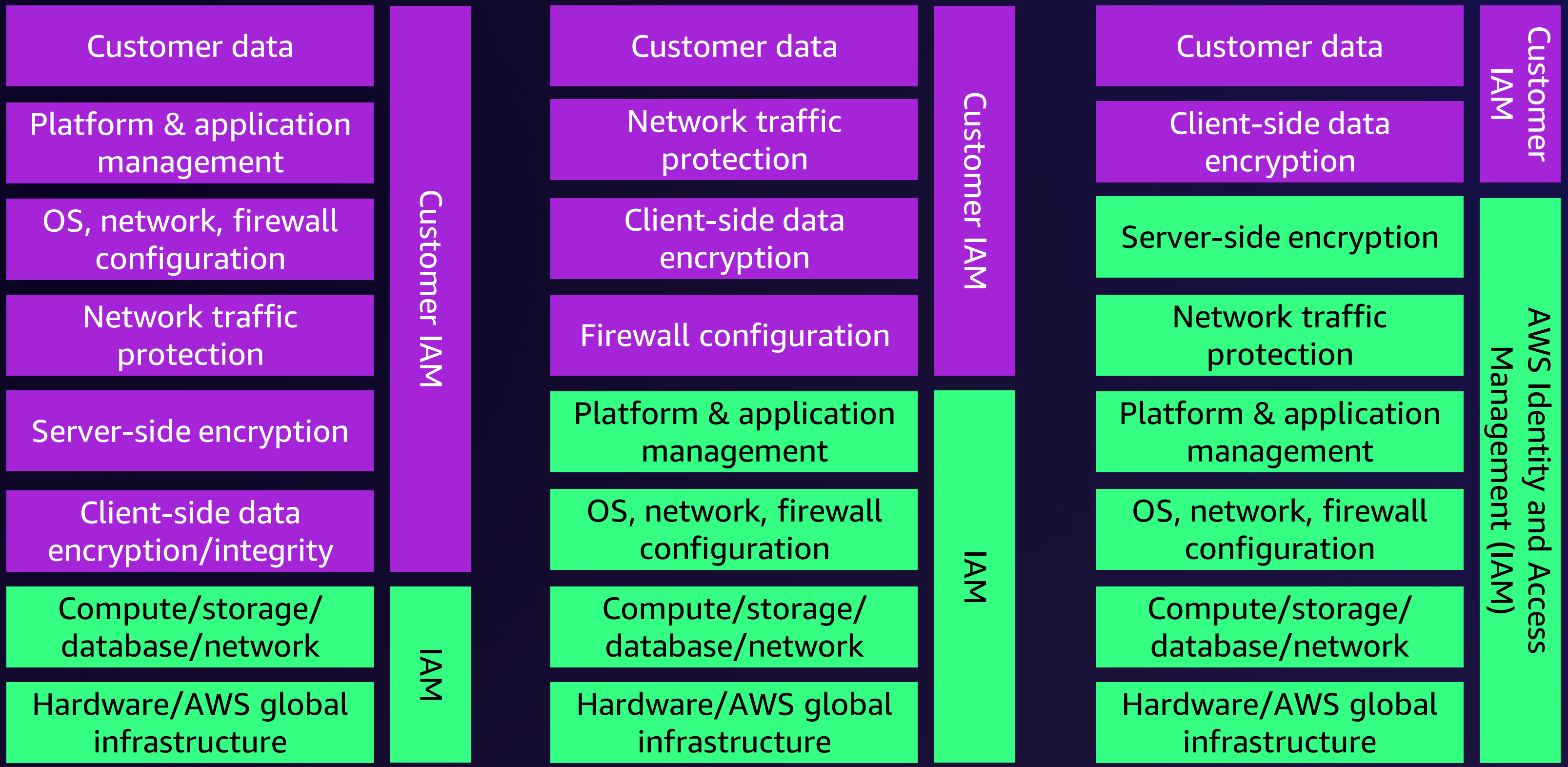
aws © 2024, Amazon Web Services, Inc. or its affiliates. All rights reserved.



Shared responsibility

Amazon EC2 | AWS Fargate | Amazon RDS | Amazon S3 | AWS KMS | Amazon DynamoDB

More customizable
+
More customer
Responsibility


Customer data
Platform & application management
OS, network, firewall configuration
Network traffic protection
Server-side encryption
Client-side data encryption/integrity
Customer IAM
Compute/storage/ database/network
Hardware/AWS global infrastructure
IAM
Customer data
Network traffic protection
Client-side data encryption
Firewall configuration
Customer IAM
Platform & application management
OS, network, firewall configuration
Compute/storage/ database/network
Hardware/AWS global infrastructure
IAM
Customer data
Client-side data encryption
Customer IAM
Server-side encryption
Network traffic protection
Platform & application management
OS, network, firewall configuration
Compute/storage/ database/network
Hardware/AWS global infrastructure
AWS Identity and Access Management (IAM)

Less customizable
+
Less customer
responsibility
+
More best practices
built-in

Infrastructure services | **Container services** | **Abstracted services**

© 2024, Amazon Web Services, Inc. or its affiliates. All rights reserved.

EDA data protection best practices

- Move only needed data, attributes, or metadata (avoid verbose events/messages)
- Filter and remove unneeded sensitive data (at source or at the edge)
- Always use encryption

© 2024, Amazon Web Services, Inc. or its affiliates. All rights reserved.

Encryption

In Transit

Protects data being transferred (network) using a secure protocols

At Rest

Protects data stored (includes short-term storage like cached data)

Client-Side

Protects data at the client or event producer, before it is transmitted

© 2024, Amazon Web Services, Inc. or its affiliates. All rights reserved.

AWS KMS 101

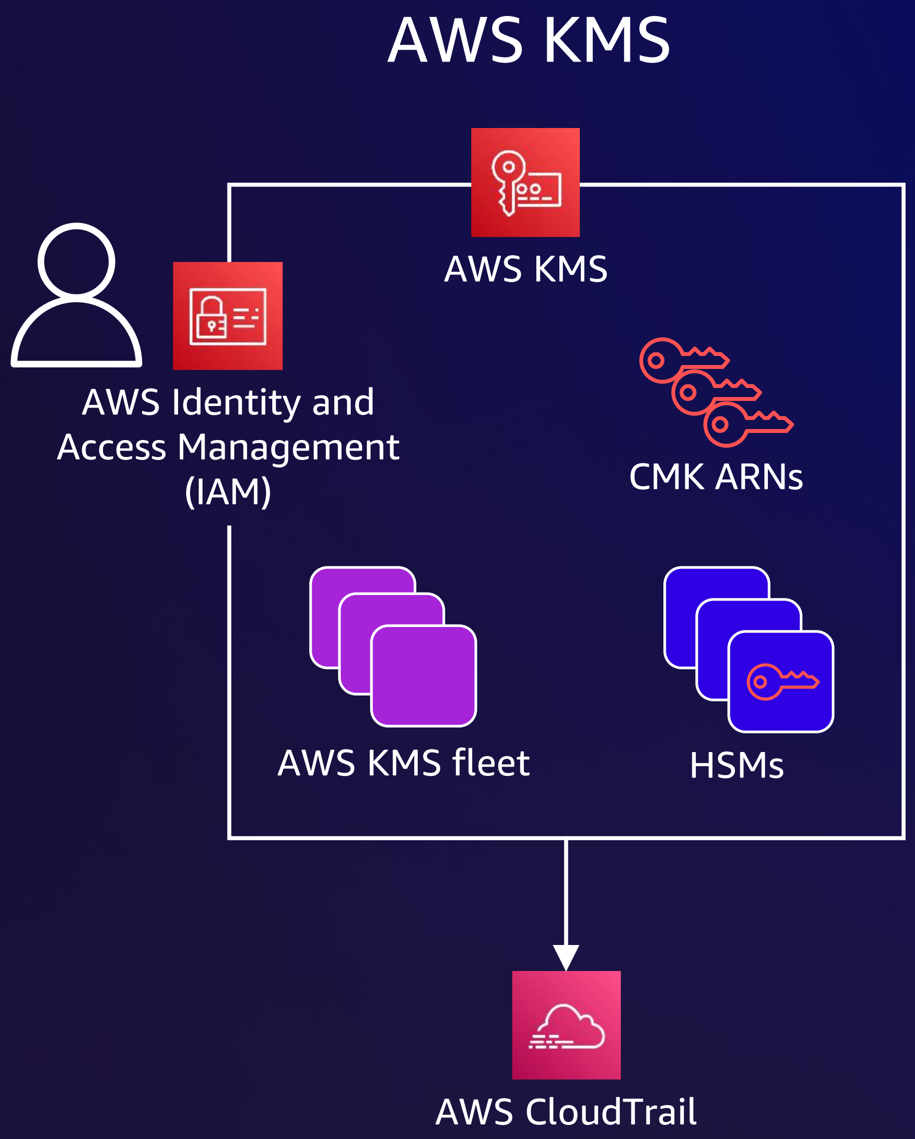
AWS managed **web service** with console, REST API, and CLI access

Database of encrypted keys leveraging role- and attribute-based access controls

Backed by a fleet of **hardware security modules (HSMs)**

All requests authorized with **IAM**

All activity tracked with **AWS CloudTrail**



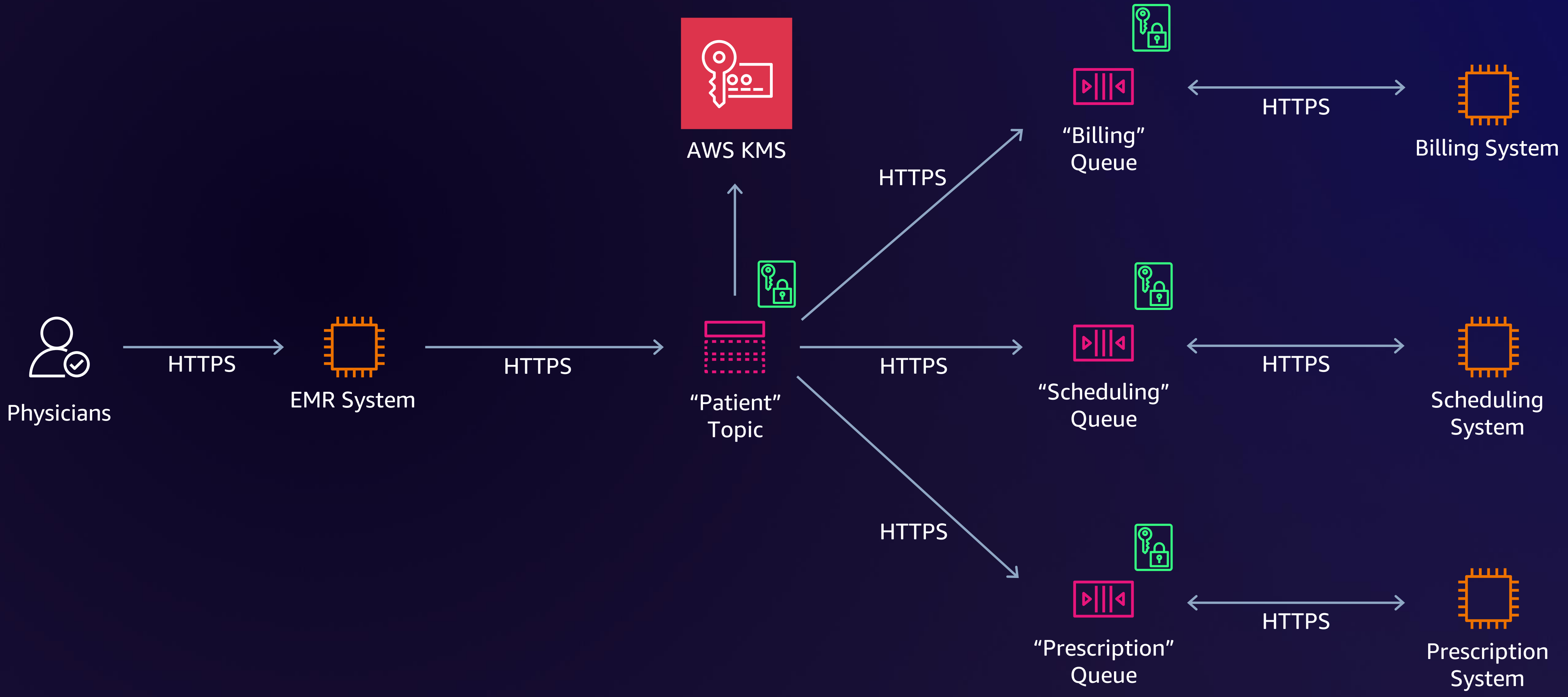
© 2024, Amazon Web Services, Inc. or its affiliates. All rights reserved.

EDA services encryption

	AWS Native				Managed Open Source	
	Queue	Stream	Topic	Bus	Stream	Broker
	Amazon SQS	Amazon Kinesis Data Streams	Amazon SNS	Amazon EventBridge	Amazon MSK	Amazon MQ *(ActiveMQ/RabbitMQ)*
In Transit	TLS 1.2+	TLS 1.2+	TLS 1.2+ (optional but recommended for subscription)	TLS 1.2+	TLS 1.2 (default between brokers of a cluster)	TLS 1.2+ *see RabbitMQ inter-node encryption
At Rest SSE server-side encryption	Supported SSE-SQS or KMS	Supported KMS	Supported KMS	Default | AWS owned key KMS	Always encrypted KMS	Always encrypted KMS

© 2024, Amazon Web Services, Inc. or its affiliates. All rights reserved.

Electronic medical record (EMR) example



© 2024, Amazon Web Services, Inc. or its affiliates. All rights reserved.

Client-side encryption

Plaintext:

```
{
  “title” : { S : “Encryption Is Fun” },
  “year” : { N : “2023” },
  “keywords” : { SS : [ “B”, “A”, “C” ] }
}
```

Encrypted form:

```
{
 “title” : { B : “FkdT1jwRcRb0MvONY=” },
 “year” : { B : “Jc9o1w==” },
 “keywords” : { B : “FYAHd8jenWb8Zf1a” }
}
```

© 2024, Amazon Web Services, Inc. or its affiliates. All rights reserved.

Client-side encryption and event-based systems

- Consider payload-level VS field-level encryption (overhead VS accessible fields)
- Your bus/broker/topic may need plaintext access to specific fields (filter rules, routing)
- Avoid adding sensitive information in metadata, context, or bus/topic names

© 2024, Amazon Web Services, Inc. or its affiliates. All rights reserved.

Selective field-level client-side encryption

```
Plaintext:
{
  "name" : { S : "My Name" },
  "SSN" : { N : "111-11-1111" },
  "keywords" : { SS : [ "B", "A", "C" ] }
}
```

```
Encrypted selective form:
{
 "name" : { B : "FkdT1jwRcRb0MvONY=" },
 "SSN" : { B : "Jc9o1w==" },
 "keywords" : { SS : [ "B", "A", "C" ] }
}
```

© 2024, Amazon Web Services, Inc. or its affiliates. All rights reserved.

Client-side encryption – The easy way

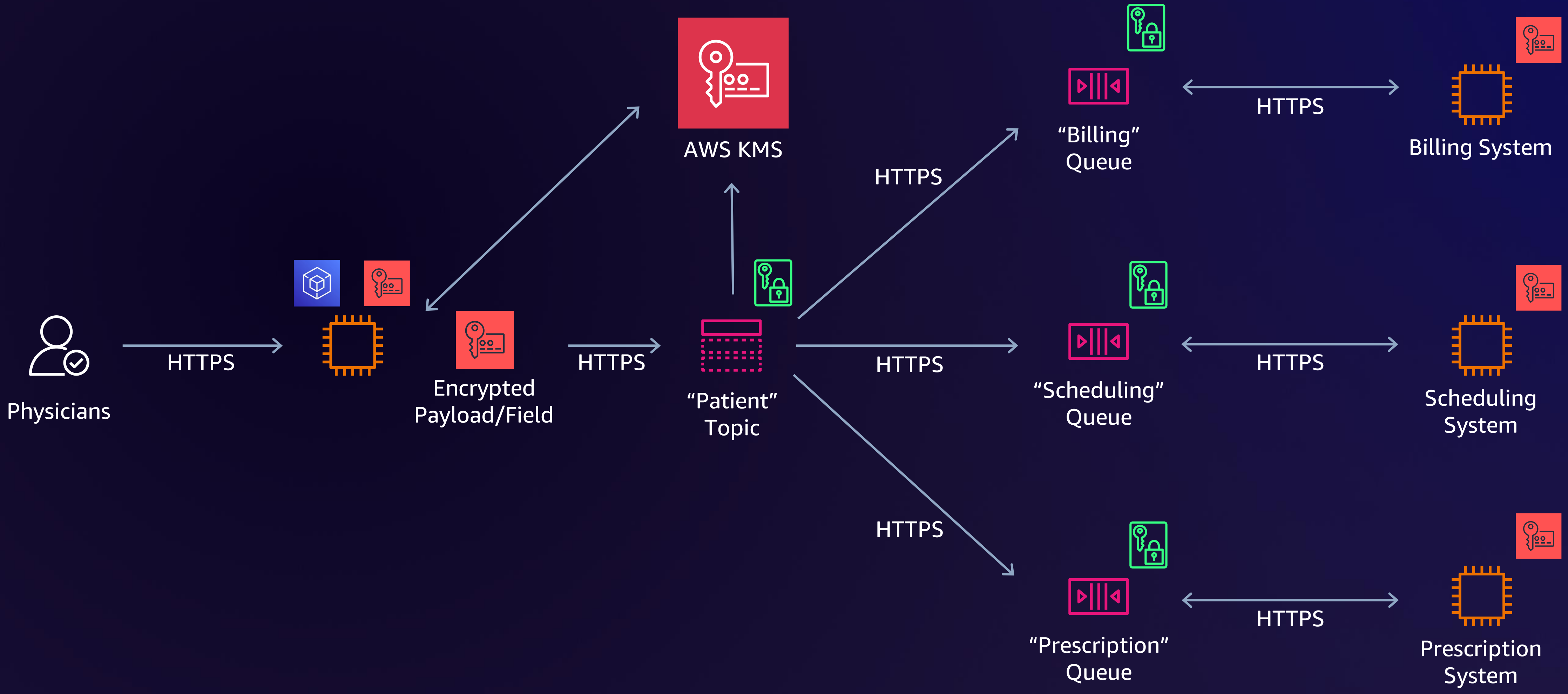
AWS Encryption SDK

Making client-side encryption safer and easier to use

- Open source
- C/.NET/Java/JavaScript/Python/CLI
- Supports Keyrings and multi-Region KMS keys
- Simplifies the development process
- Envelope encryption
- **Data protection** (encryption) and **data integrity** (signature) in a single tool

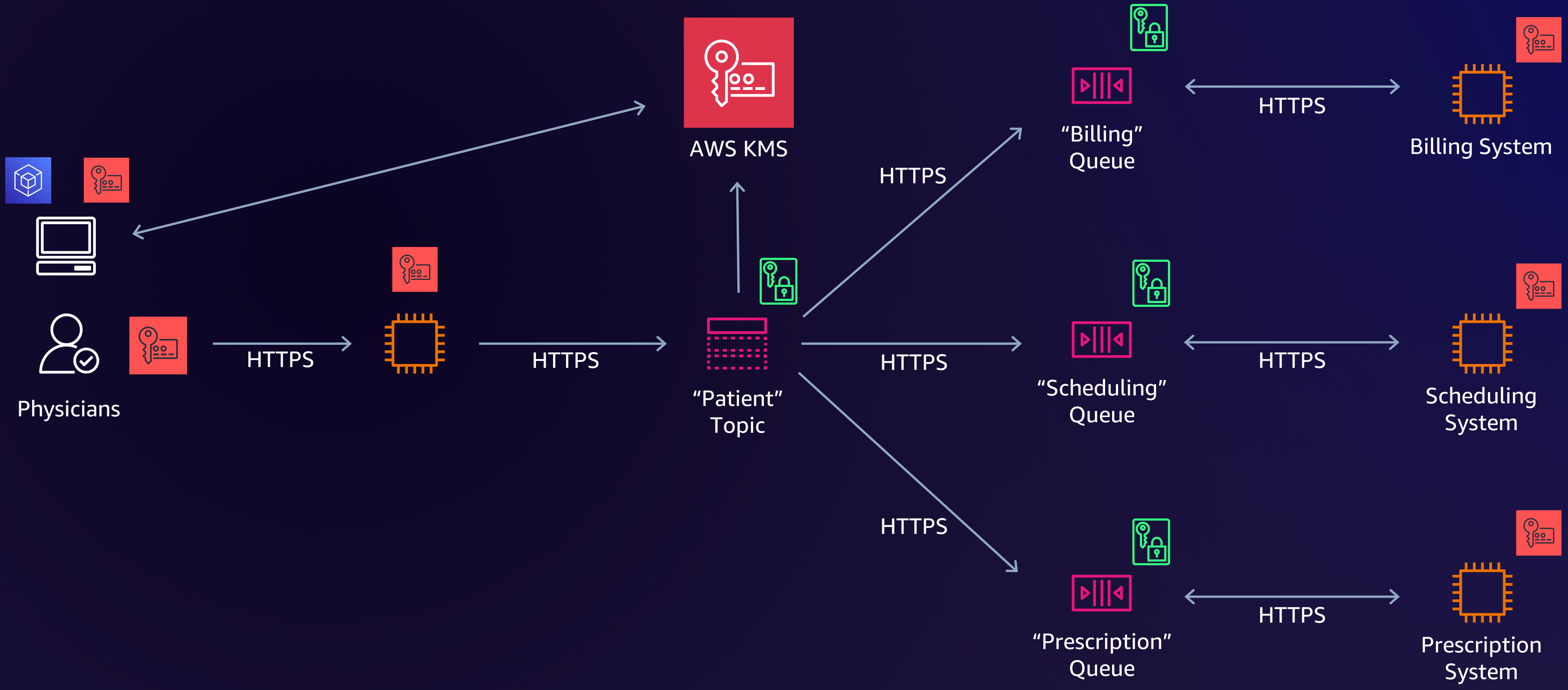
© 2024, Amazon Web Services, Inc. or its affiliates. All rights reserved.

Electronic medical record (EMR) example 2



© 2024, Amazon Web Services, Inc. or its affiliates. All rights reserved.

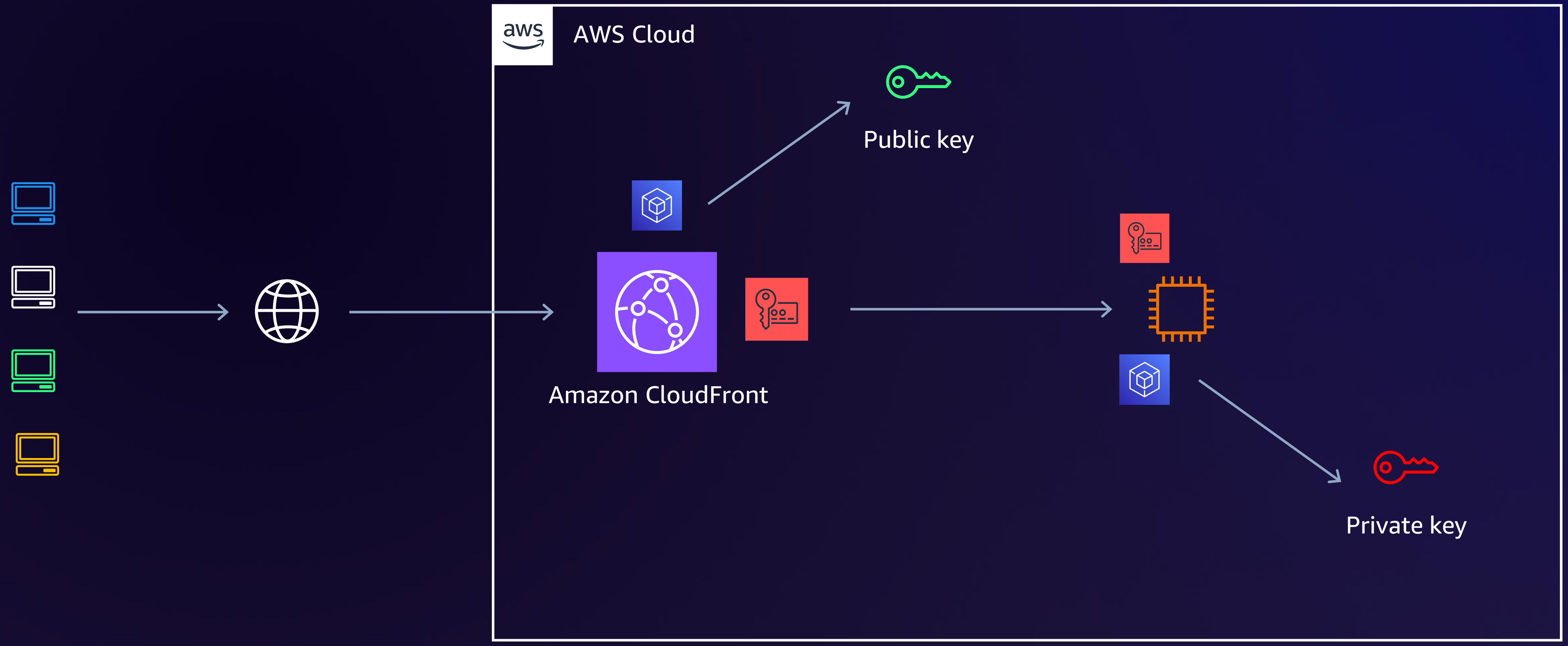
Electronic medical record (EMR) example 3



© 2024, Amazon Web Services, Inc. or its affiliates. All rights reserved.

What if I don't control external clients?

Field-level encryption at the edge with Amazon CloudFront



© 2024, Amazon Web Services, Inc. or its affiliates. All rights reserved.

UNITED

United Airlines

© 2024, Amazon Web Services, Inc. or its affiliates. All rights reserved.

Agenda

Passenger **S**ervice **S**ystem

Transformation experience using event-driven architecture

Setting up for success

© 2024, Amazon Web Services, Inc. or its affiliates. All rights reserved.

UNITED NEXT

700+

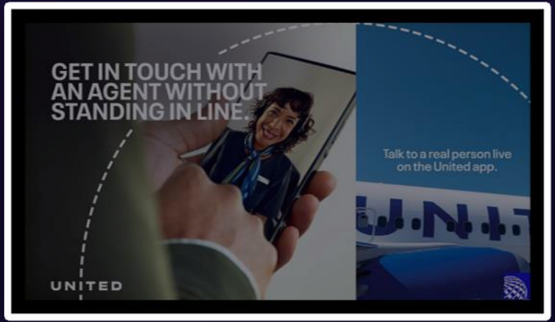
New Aircrafts

50,000

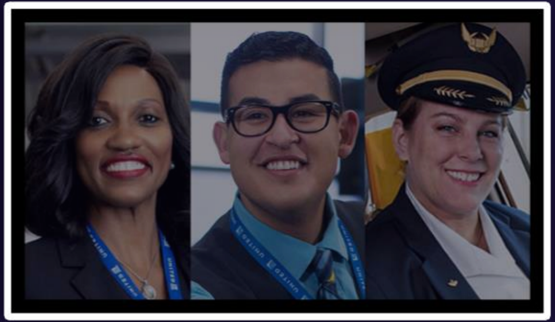
Jobs

LARGEST WIDEBODY ORDER BY A U.S. CARRIER IN COMMERCIAL AVIATION HISTORY.
Up to 200 new 787 Dreamliners.
UNITED
DREAMLINER
© 2022 UNITED AIRLINES, INC. ALL RIGHTS RESERVED.

© 2024, Amazon Web Services, Inc. or its affiliates. All rights reserved.



Success so far built upon decades old technology

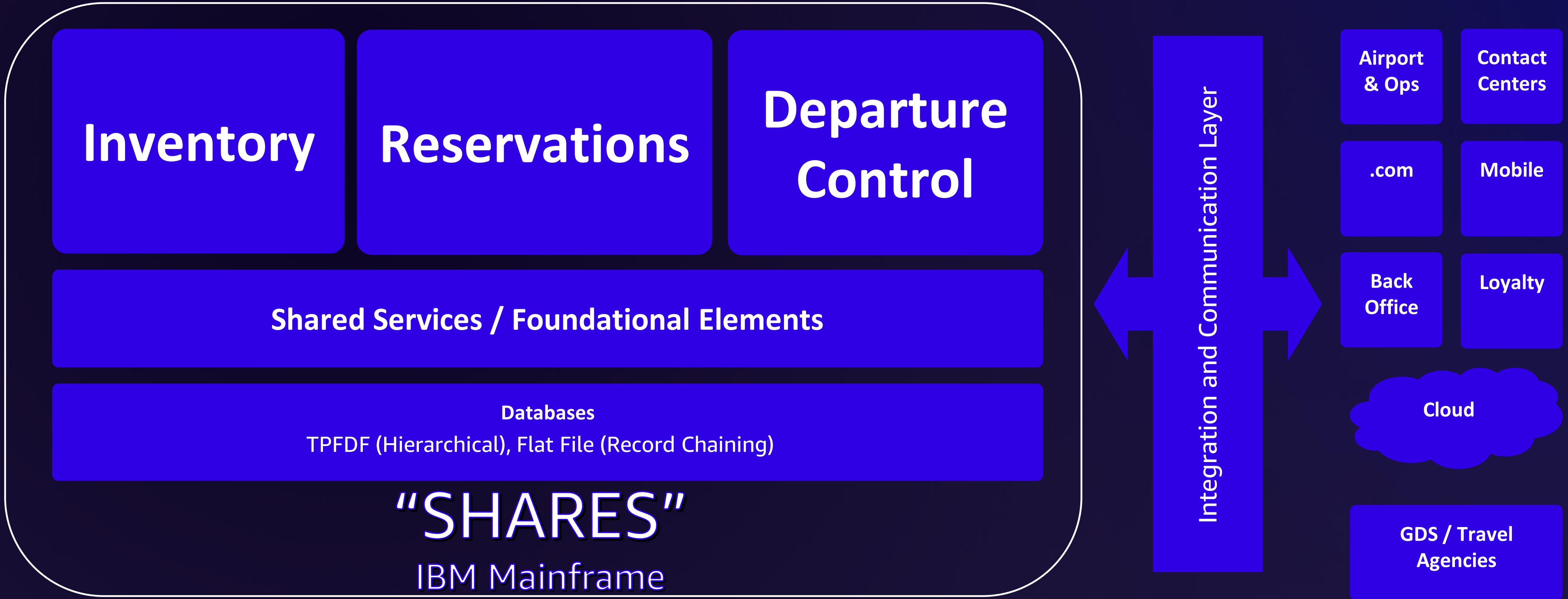


Bag drop shortcut
Already checked in and paid for your bags?
Use these kiosks to skip the lines.
UNITED

© 2024, Amazon Web Services, Inc. or its affiliates. All rights reserved.

What is a Passenger Service System (PSS)?

A passenger service system (PSS) is a series of critical systems used by airlines. The PSS usually comprises an airline reservations system, an airline inventory system and a departure control system (DCS).



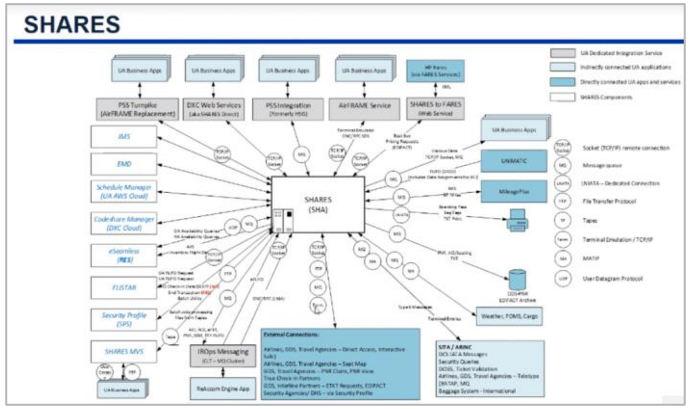
© 2024, Amazon Web Services, Inc. or its affiliates. All rights reserved.

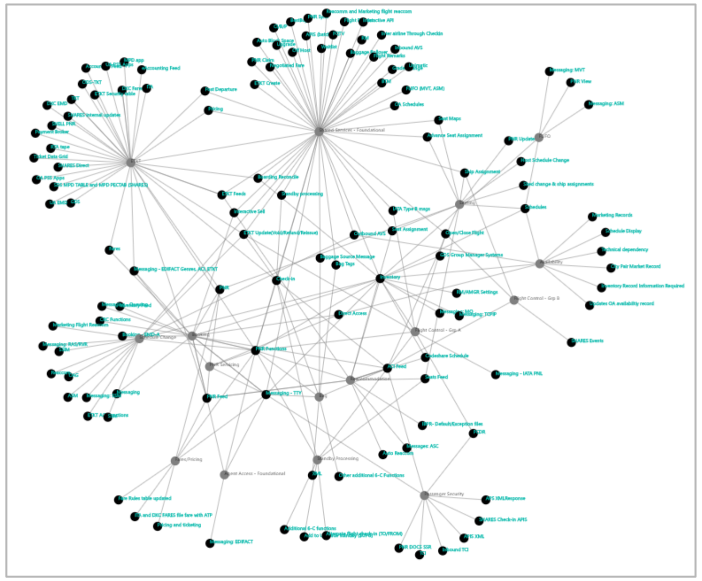
Transformation requires careful examination of dependencies and customer priorities

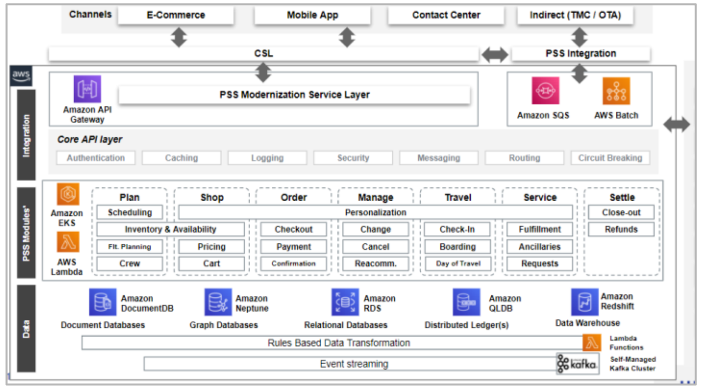
1 Document Current-State PSS Components and Dependencies

2 Examine **Component Dependencies**

3 Implement Future-State PSS with Mitigation of Dependencies







© 2024, Amazon Web Services, Inc. or its affiliates. All rights reserved.

Pivot PSS Design to Order Management System

- Customer centricity
- Alignment to domain-driven architecture*
- Lead industry transformation
- Technology transformation

*Redefine and remain compliant

Source: https://twitter.com/united/status/542688661083807746

© 2024, Amazon Web Services, Inc. or its affiliates. All rights reserved.

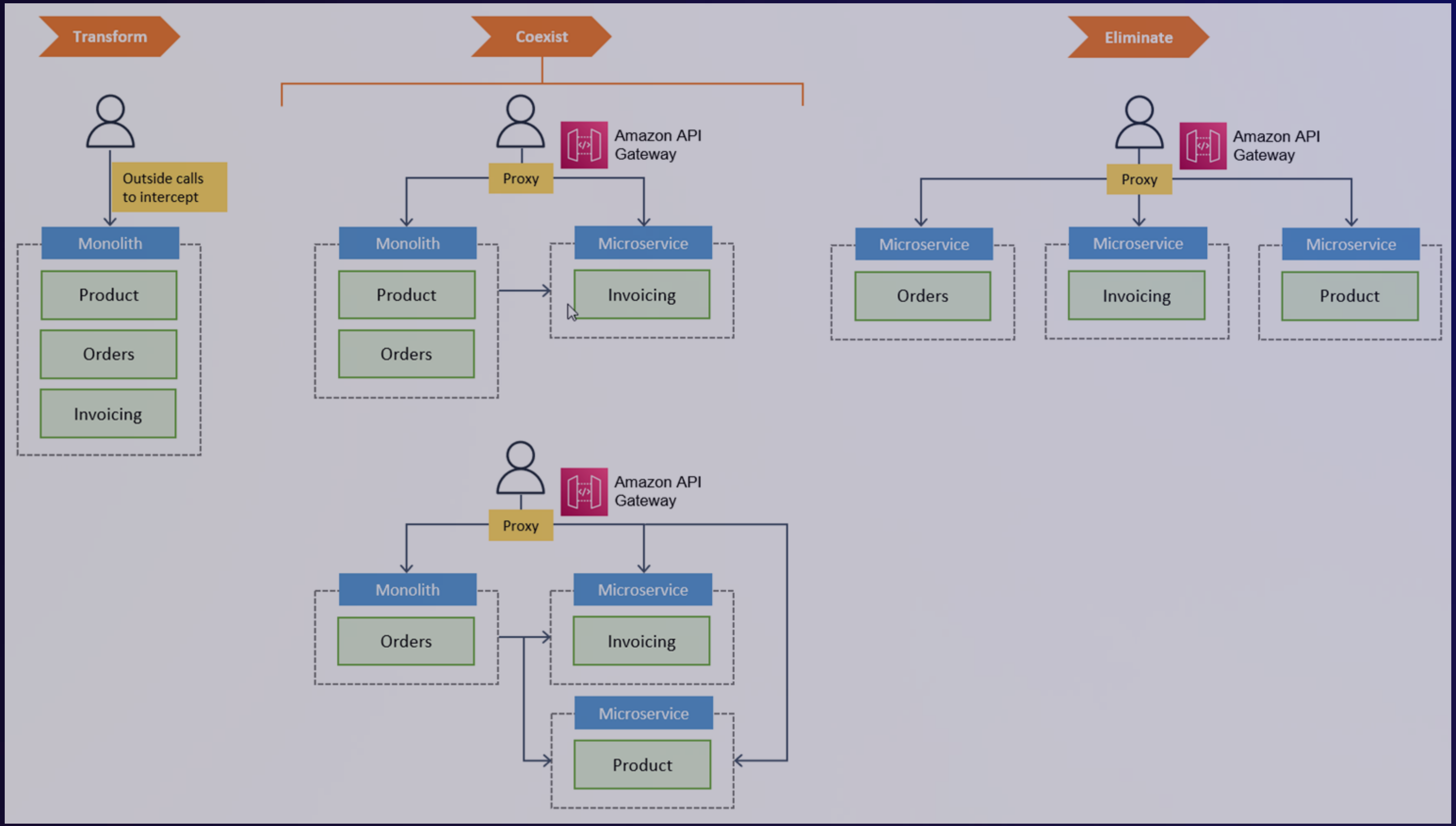
Transformation stages of strangler fig pattern

Complexity of **mainframe**

Coexist: Dependencies on other airlines and external systems

Eliminate: Green screen commands

Natural progression to event-driven architecture



Source: Strangler fig pattern - AWS Prescriptive Guidance (amazon.com)

© 2024, Amazon Web Services, Inc. or its affiliates. All rights reserved.

Key business requirements

15M+ daily transactions

1.7M+ flights

8.5M+ daily seat map view

500k+ daily passengers

Resilience & High Availability

Ensuring recovery from failures or disruptions and minimize downtime; ability to handle increased demand

Monitoring & Observability

End-to-end transaction tracing and proactive issue detection

Data Management

Maintain data integrity between legacy and modern system as well data storage, retrieval, processing, and analysis

Cost-Effectiveness

Variablize cost-structure, serverless where applicable

Interoperability

Interoperability between on premises and cloud; ability to support current business needs while modernizing for future

Agility & DevOps

Iterative approach to deliver business benefits; enable faster time to market, frequent updates, and improved quality

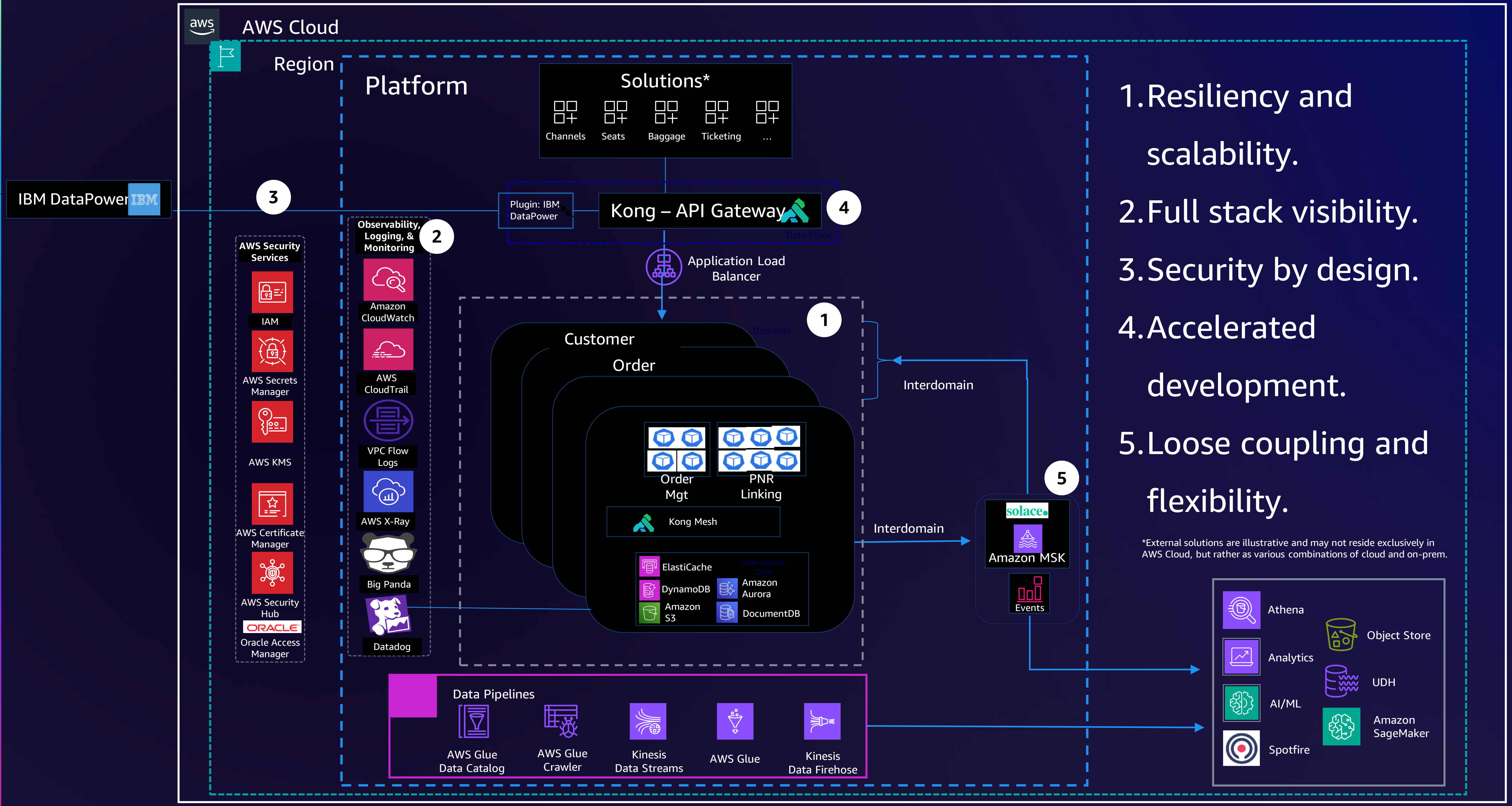
Security

Incorporate robust security measures, such as encryption, authentication, authorization, and audit logging

© 2024, Amazon Web Services, Inc. or its affiliates. All rights reserved.

Reference architecture

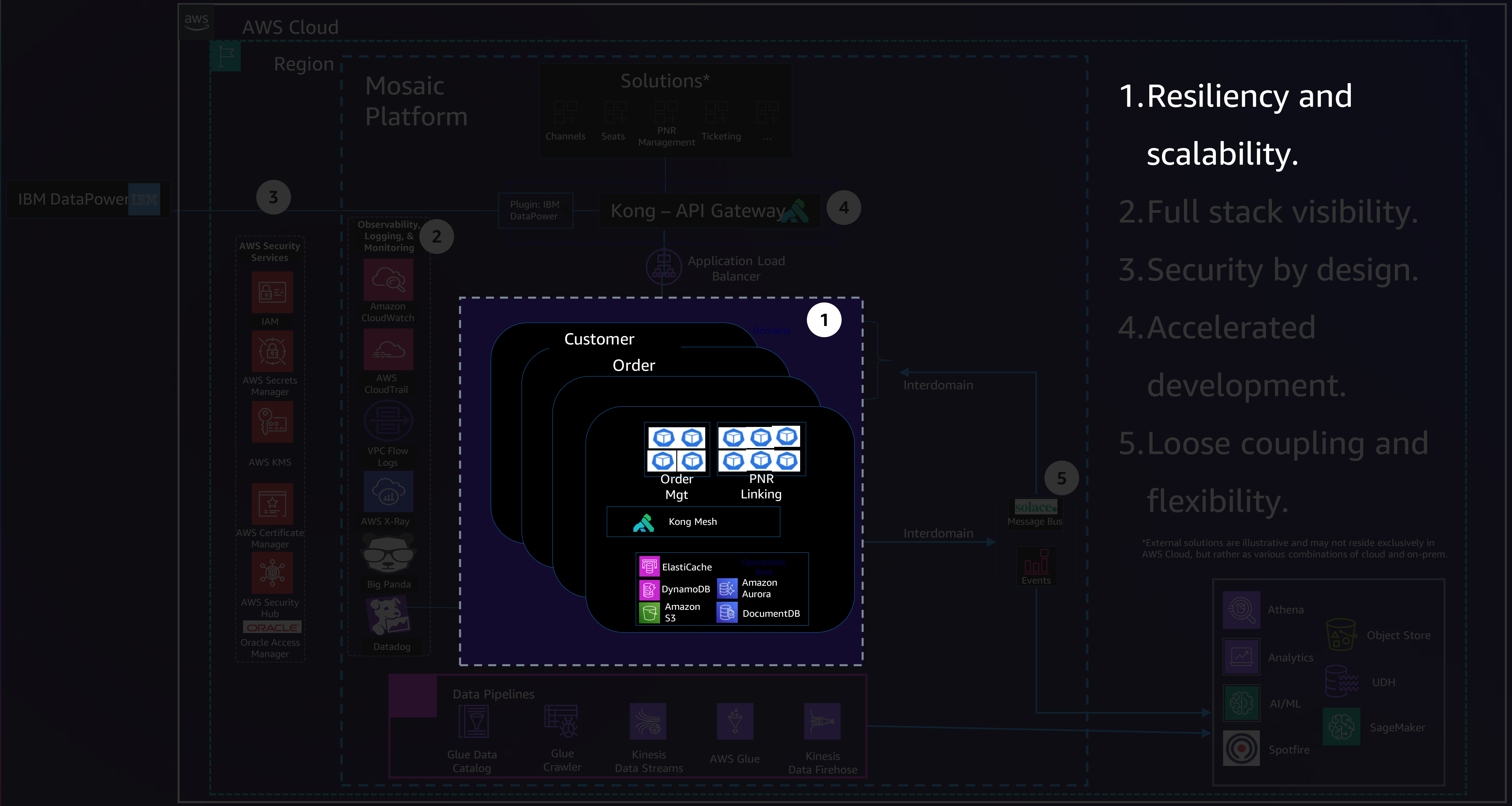
© 2024, Amazon Web Services, Inc. or its affiliates. All rights reserved.



1. Resiliency and scalability.
2. Full stack visibility.
3. Security by design.
4. Accelerated development.
5. Loose coupling and flexibility.

*External solutions are illustrative and may not reside exclusively in AWS Cloud, but rather as various combinations of cloud and on-prem.

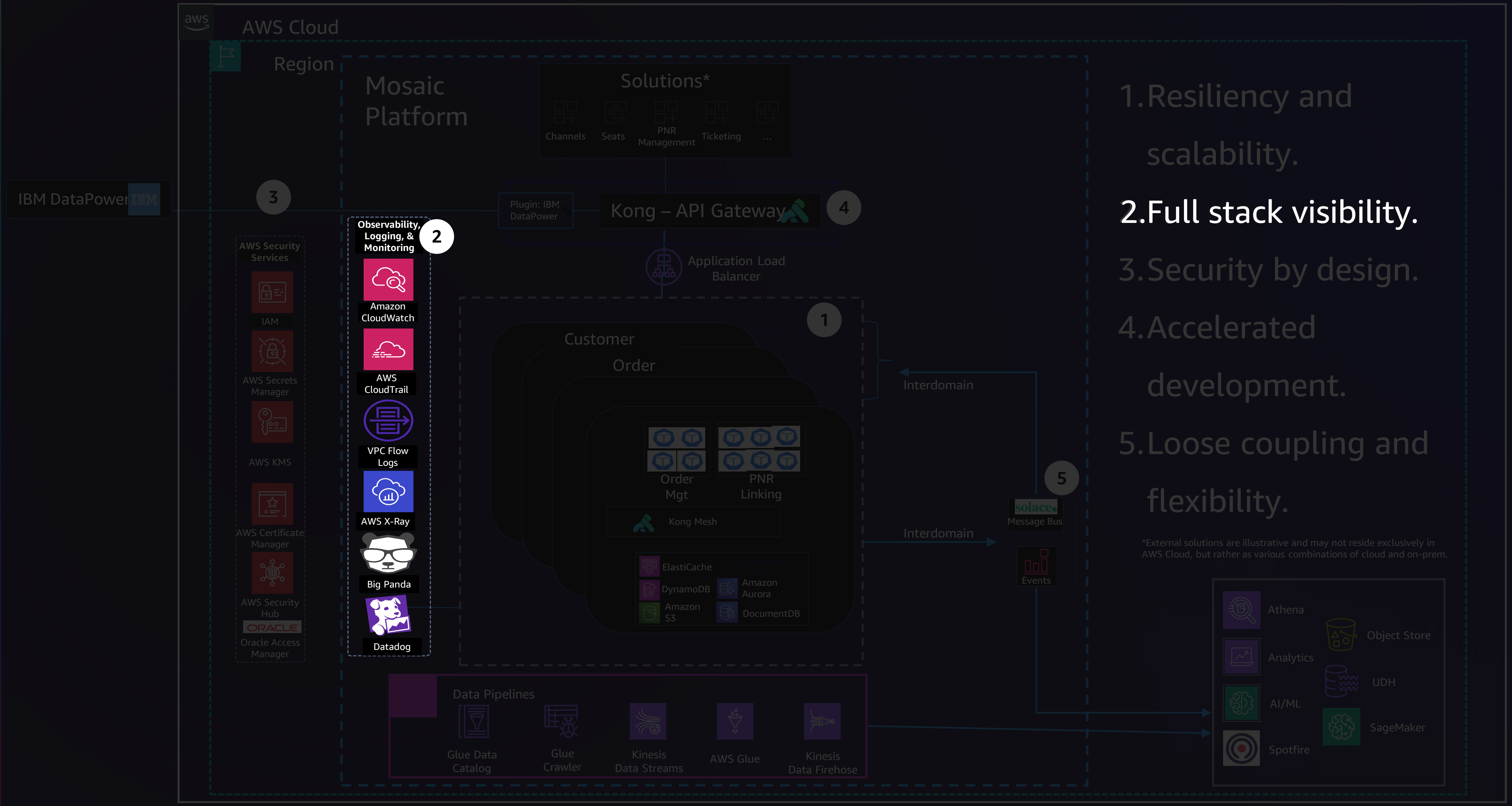
© 2024, Amazon Web Services, Inc. or its affiliates. All rights reserved.



1. Resiliency and scalability.
2. Full stack visibility.
3. Security by design.
4. Accelerated development.
5. Loose coupling and flexibility.

*External solutions are illustrative and may not reside exclusively in AWS Cloud, but rather as various combinations of cloud and on-prem.

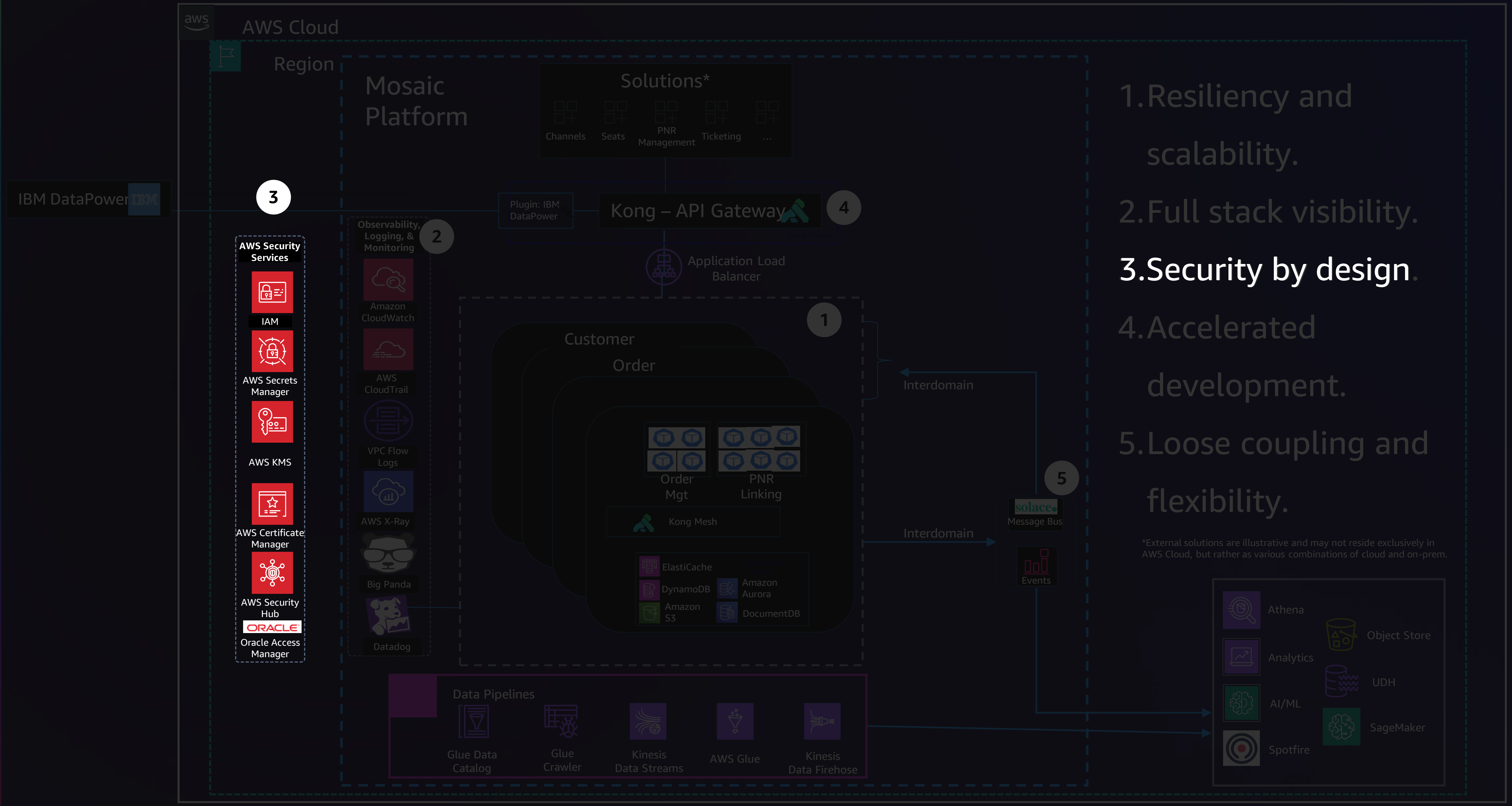
© 2024, Amazon Web Services, Inc. or its affiliates. All rights reserved.



1. Resiliency and scalability.
2. Full stack visibility.
3. Security by design.
4. Accelerated development.
5. Loose coupling and flexibility.

*External solutions are illustrative and may not reside exclusively in AWS Cloud, but rather as various combinations of cloud and on-prem.

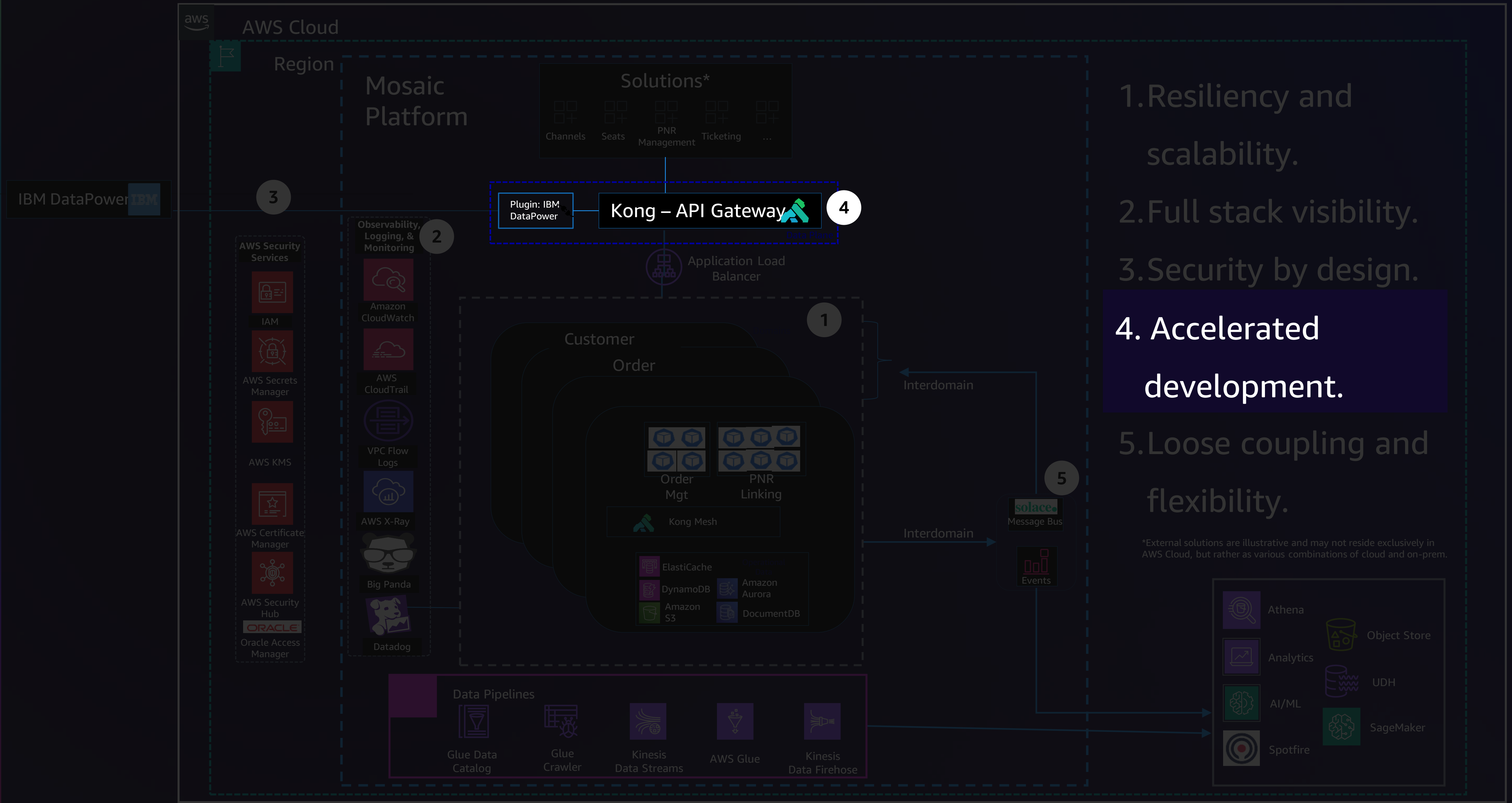
© 2024, Amazon Web Services, Inc. or its affiliates. All rights reserved.



1. Resiliency and scalability.
2. Full stack visibility.
3. Security by design.
4. Accelerated development.
5. Loose coupling and flexibility.

*External solutions are illustrative and may not reside exclusively in AWS Cloud, but rather as various combinations of cloud and on-prem.

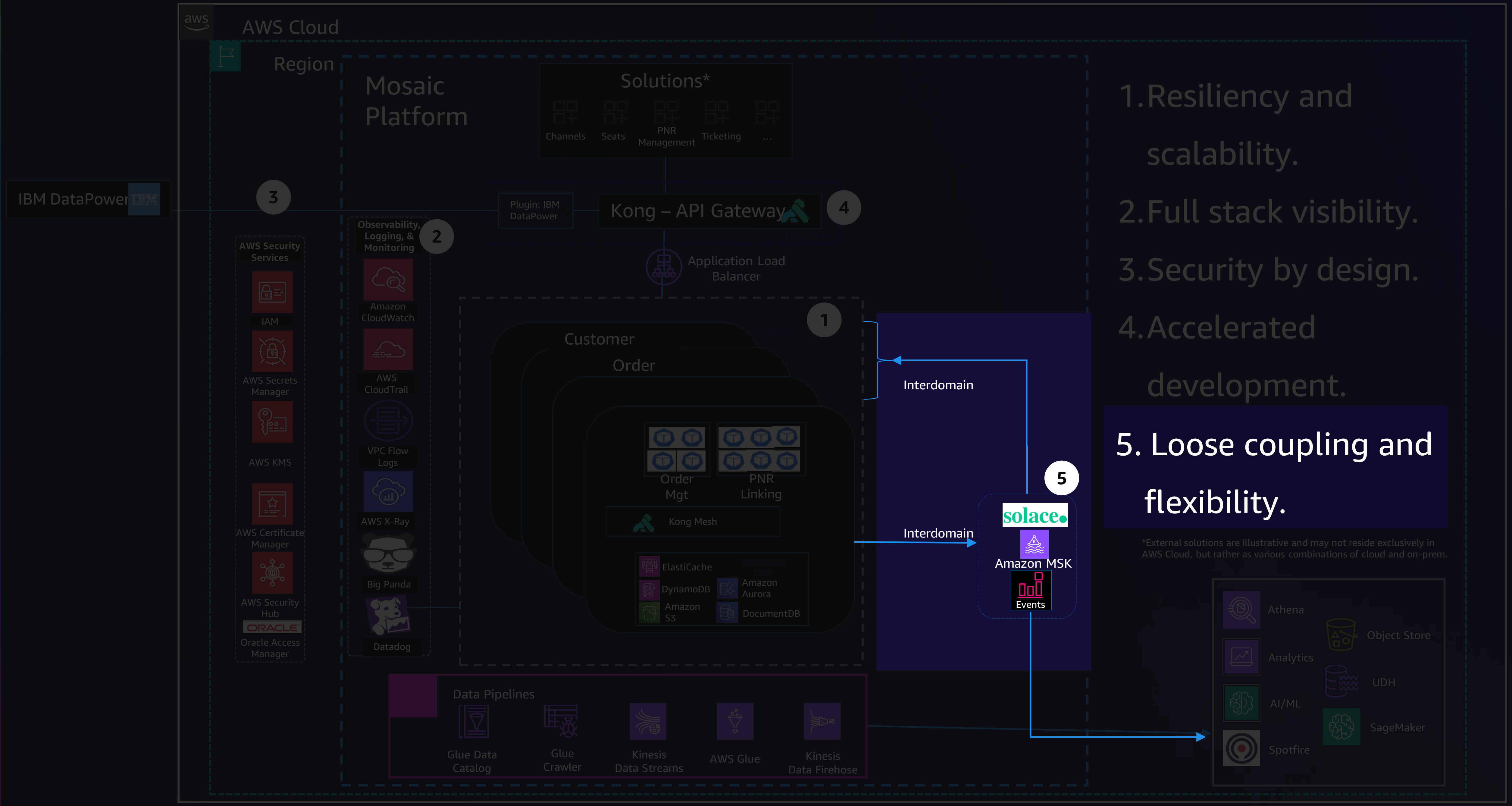
aws © 2024, Amazon Web Services, Inc. or its affiliates. All rights reserved.



1. Resiliency and scalability.
2. Full stack visibility.
3. Security by design.
4. Accelerated development.
5. Loose coupling and flexibility.

*External solutions are illustrative and may not reside exclusively in AWS Cloud, but rather as various combinations of cloud and on-prem.

© 2024, Amazon Web Services, Inc. or its affiliates. All rights reserved.



1. Resiliency and scalability.
2. Full stack visibility.
3. Security by design.
4. Accelerated development.
5. Loose coupling and flexibility.

*External solutions are illustrative and may not reside exclusively in AWS Cloud, but rather as various combinations of cloud and on-prem.

© 2024, Amazon Web Services, Inc. or its affiliates. All rights reserved.

EDA patterns

© 2024, Amazon Web Services, Inc. or its affiliates. All rights reserved.

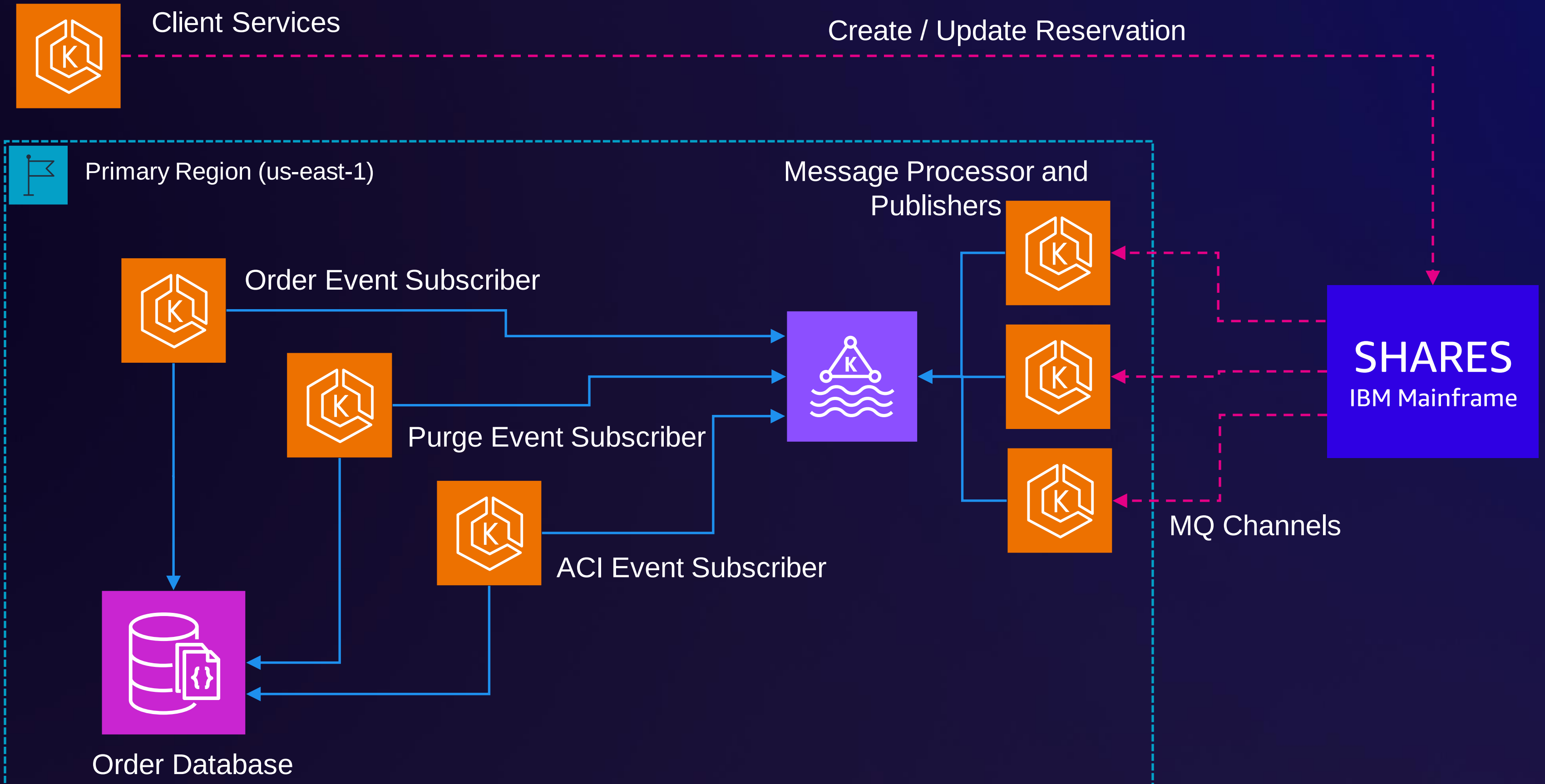
Pattern 1 – Coexist with mainframe

Transformed application in passive state

Order Event

Purge Lifecycle Event

Automated Check In Event



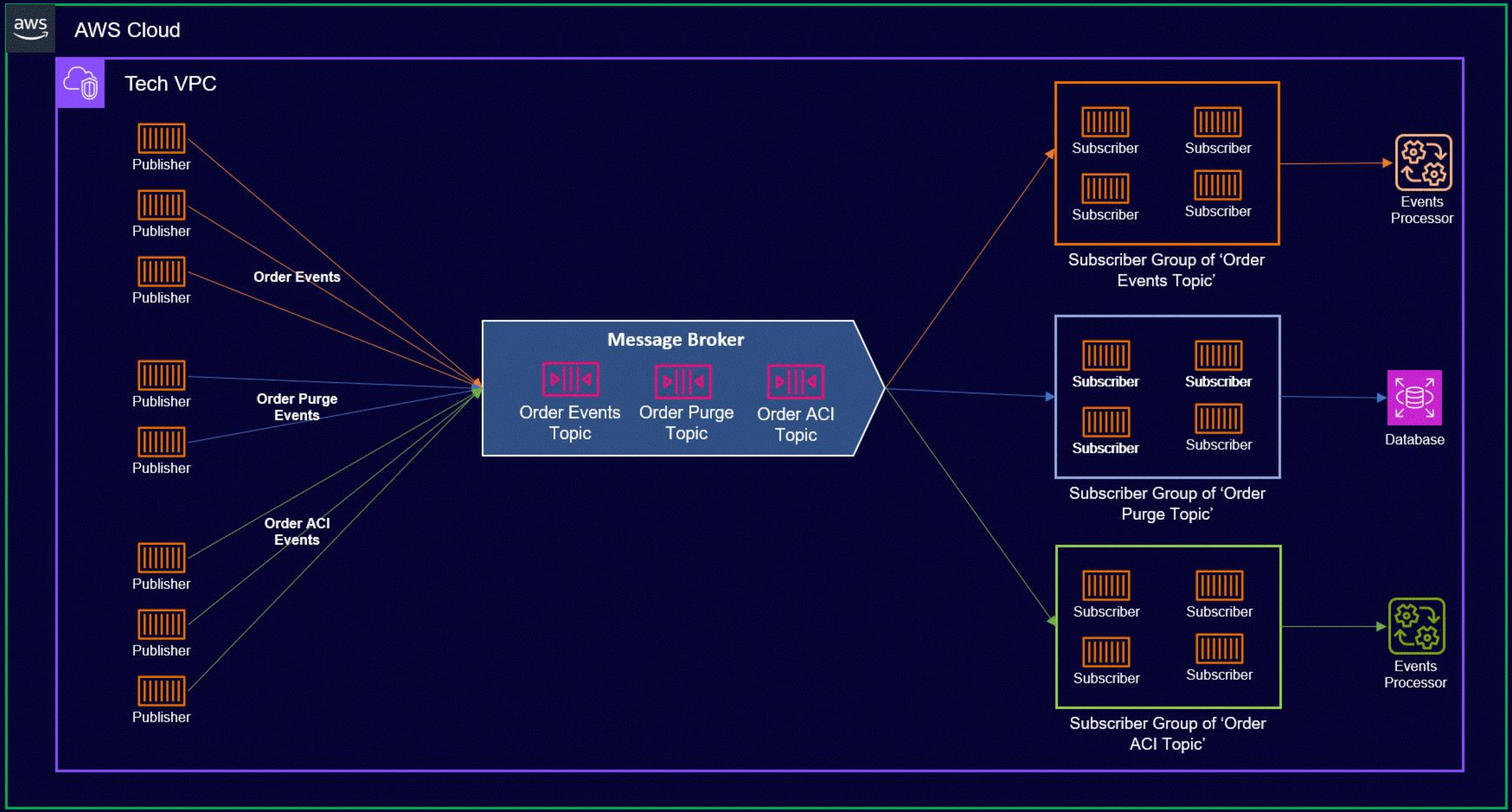
© 2024, Amazon Web Services, Inc. or its affiliates. All rights reserved.

Pattern 2 - Event streaming

Order events for Baggage

Order events to serve customers

Guaranteed sequence for events during IRROPS



© 2024, Amazon Web Services, Inc. or its affiliates. All rights reserved.

Pattern 3 - Data integrity check

Multiple System Of Record

Maintain Seat Assignment Sequence

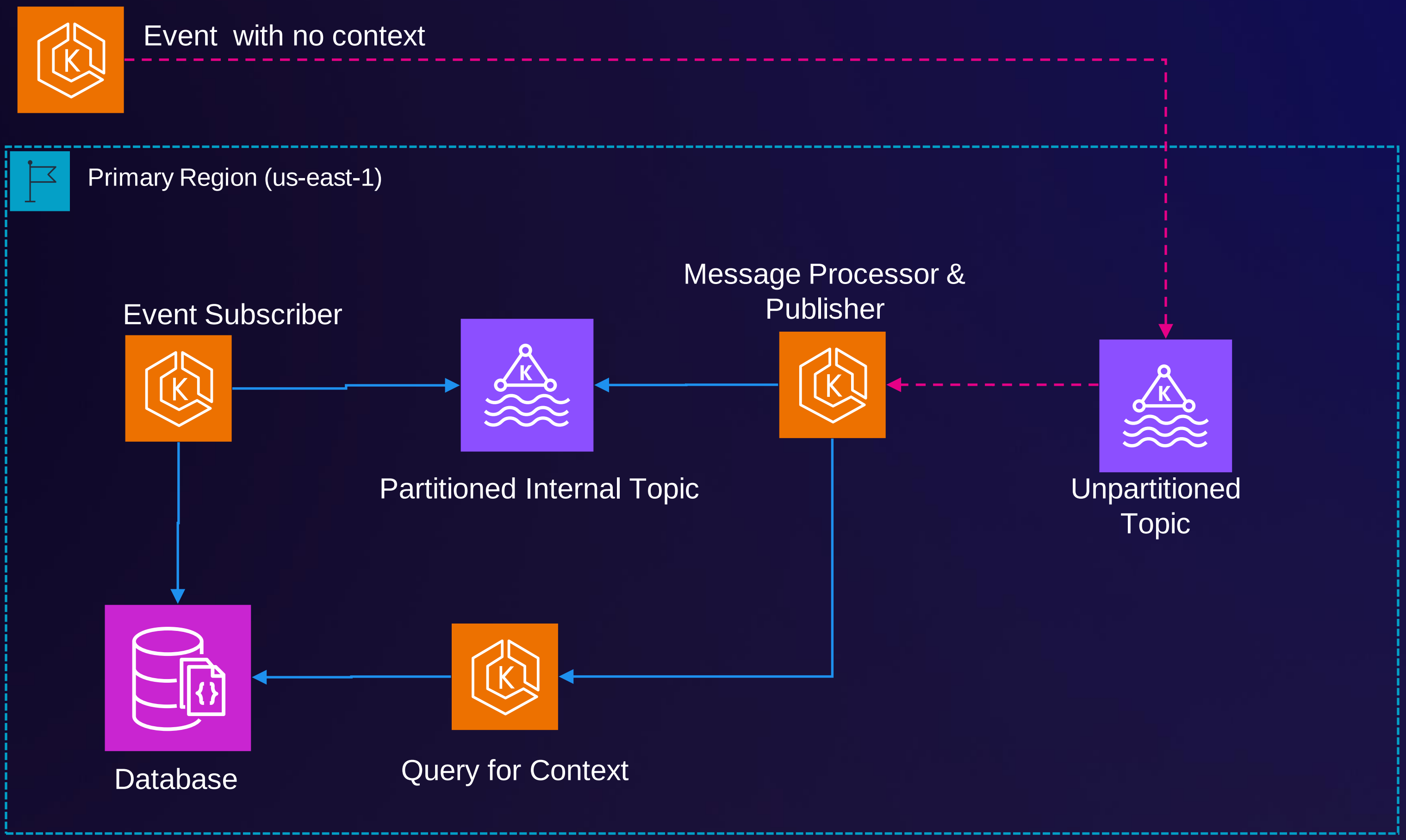
© 2024, Amazon Web Services, Inc. or its affiliates. All rights reserved.

Pattern 4 - Context injection

Augment context to process event sequentially

Provides scalability during IRROPS events

Enable low latency updates for downstream systems

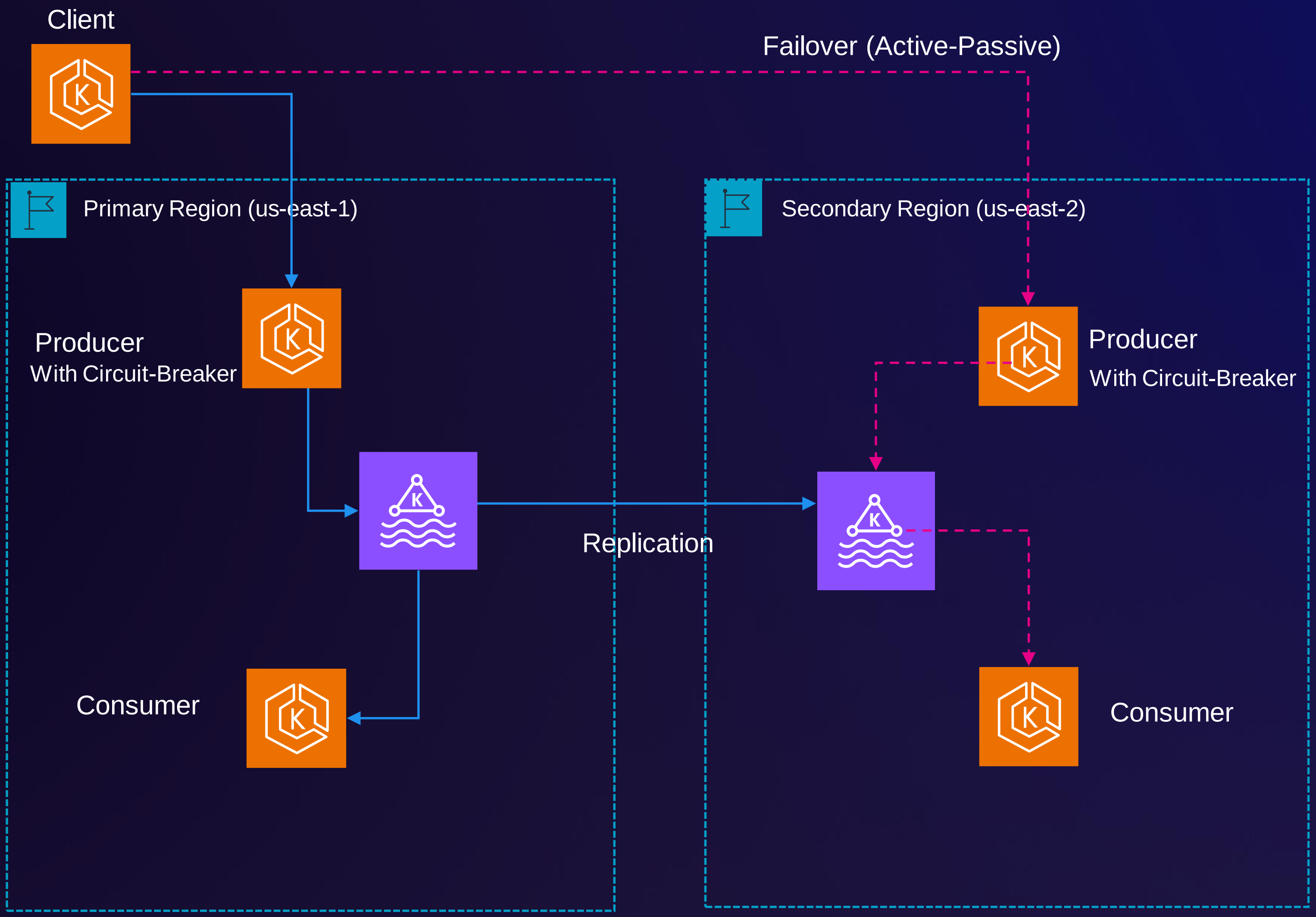

Event with no context
Primary Region (us-east-1)
Message Processor & Publisher
Event Subscriber
Partitioned Internal Topic
Unpartitioned Topic
Database
Query for Context

© 2024, Amazon Web Services, Inc. or its affiliates. All rights reserved.

Pattern 5 – Circuit breaker

- **Scenario with Active-Passive setup**
  Messaging or Database failure

- **MSK Broker Failure**
  Circuit Breaker setup to initiate cross region failover

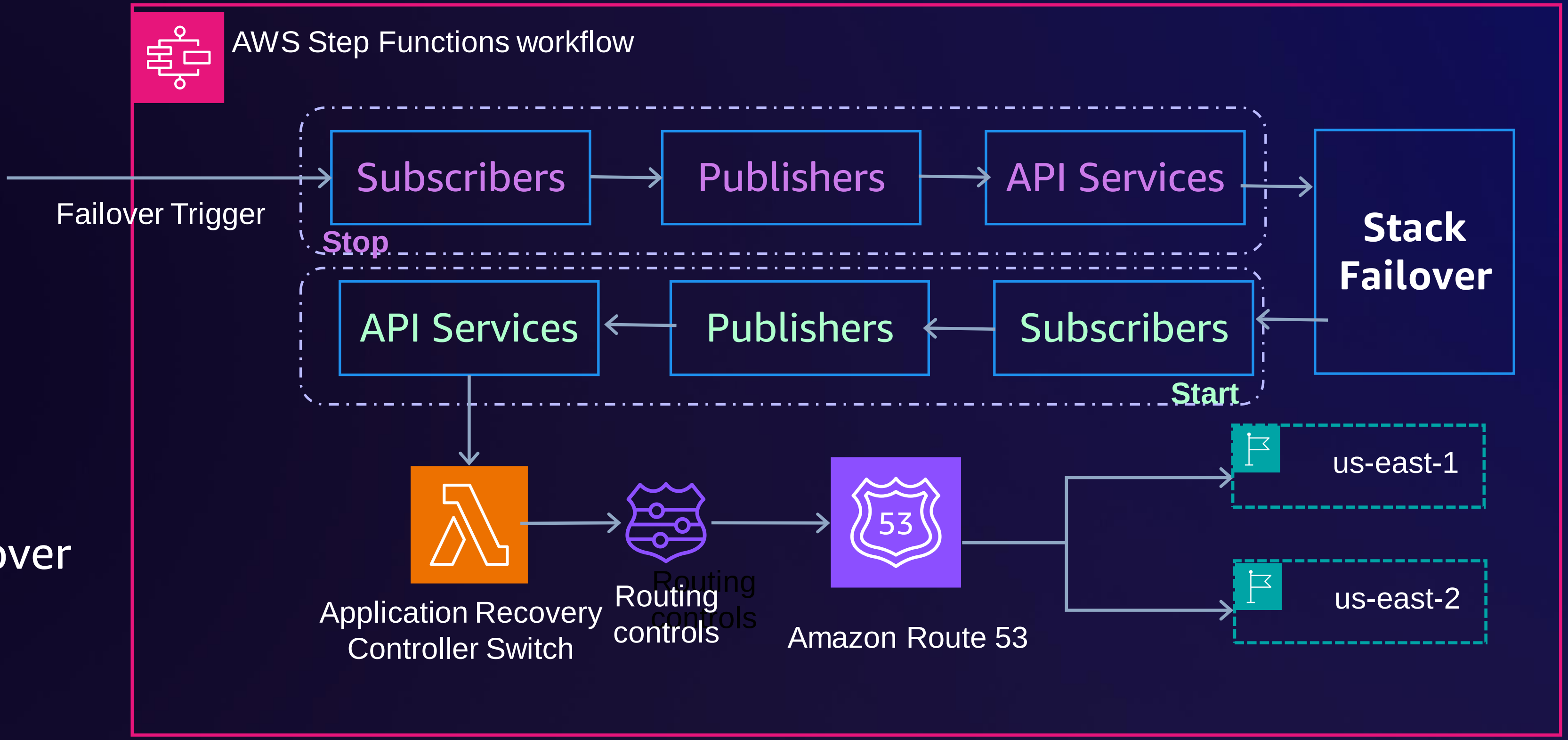


© 2024, Amazon Web Services, Inc. or its affiliates. All rights reserved.

Failover automation process

- **AWS Step function**
  Minimize data loss by restart services in sequence

- **Application Recovery Controller**
  Re-route API traffic post failover



© 2024, Amazon Web Services, Inc. or its affiliates. All rights reserved.

Guiding principles for resilient architecture

- Microservices design
- Event-driven architecture
- Leverage platform capabilities
- Data-driven integrations
- Design flexible interfaces
- Use cloud-native technologies
- Loose coupling of components
- Transition away from batch processing
- Establish systems of record with SOC
- Build for today, design for the future
- Comply with UAL security standards
- Iterate and continuously improve

© 2024, Amazon Web Services, Inc. or its affiliates. All rights reserved.

Summary

Event-driven architecture is key enabler for resilient PSS system

Data generated from PSS system is fueling Innovative capabilities enhancing your travel experience

© 2024, Amazon Web Services, Inc. or its affiliates. All rights reserved.