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

API312

How to select the right application integration service

Christian Müller (he/him)
Principal Solutions Architect
Amazon Web Services

Dirk Fröhner (he/him)
Principal Solutions Architect
Amazon Web Services

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

Agenda

Motivation: Why is this topic important?

Integration Services: Overview, decision guidance, and differentiation

Open discussion: Share your integration challenges!

Resources and call-to-action

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

Related sessions

API002	Advanced patterns with Amazon EventBridge
API202	Decoupled microservices
API303	Application integration patterns for microservices
API306	Building event-driven architectures
API307	Designing event-driven integrations using Amazon EventBridge
API308	Are you integrating or building distributed applications?
SVS306	Serverlesspresso: Building an event-driven app from the ground up
SVS308	AWS serverless developer experience: A day in the life of a developer
SVS312	Building Serverlesspresso: Creating event-driven architectures

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

Motivation:
Why is this topic important?

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

Companies deal with integration scenarios



in many areas, on many layers

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

One fairly prominent example

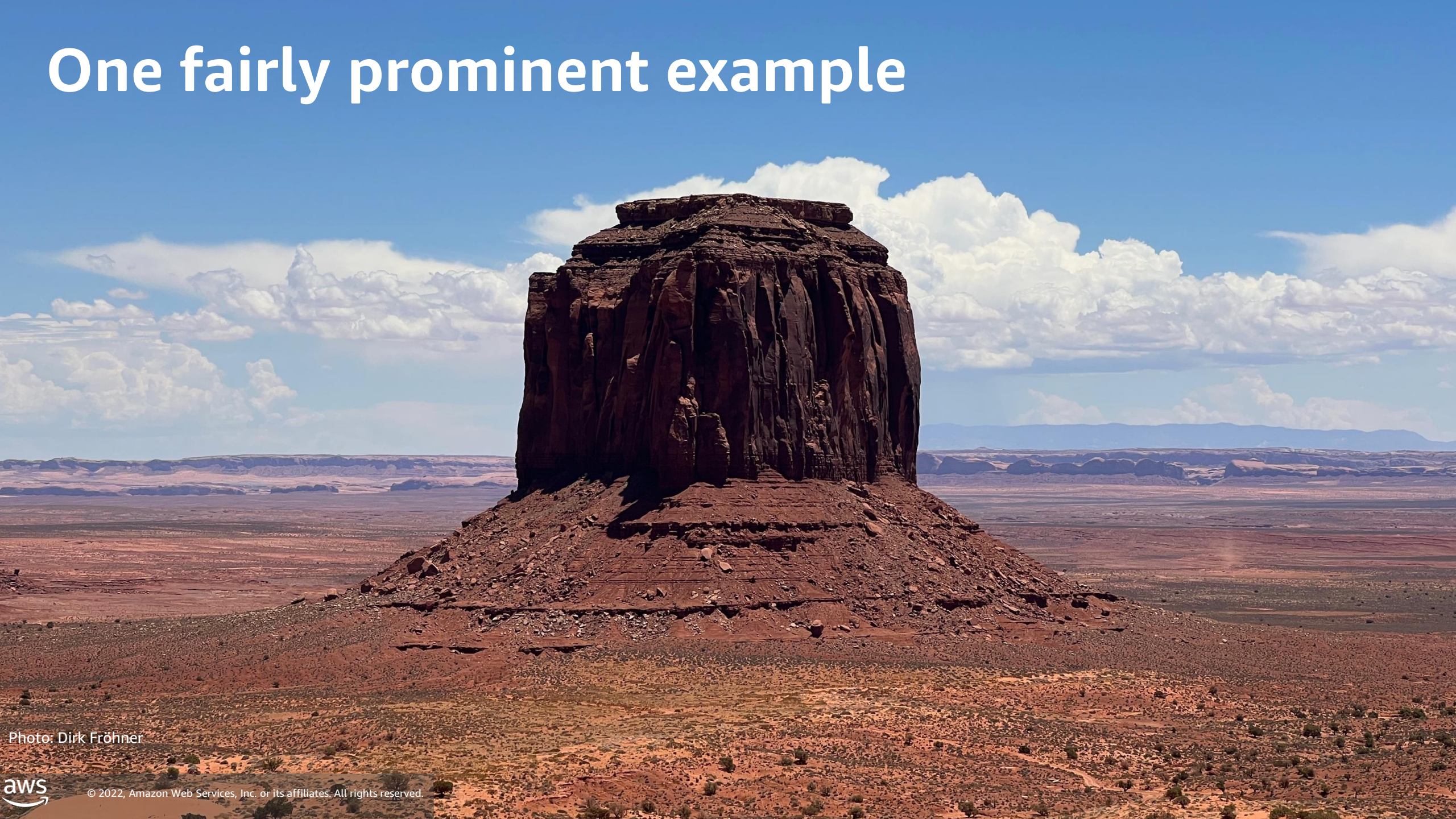


Photo: Dirk Fröhner

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

One fairly prominent example

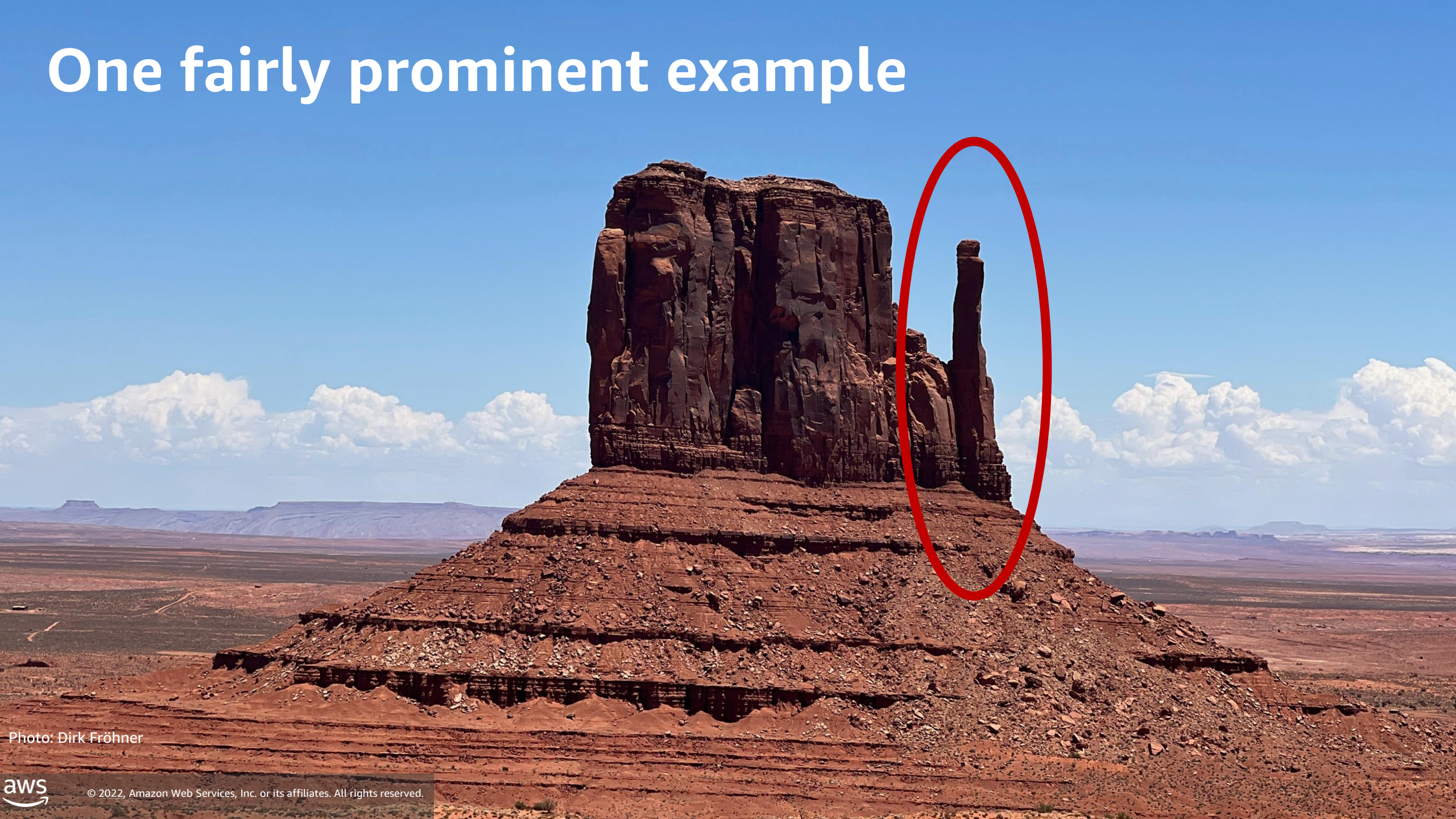


Photo: Dirk Fröhner

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

One fairly common scenario



Photo: Dirk Fröhner

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

One fairly common scenario



Photo: Dirk Fröhner

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

“In modern cloud applications, integration isn’t an afterthought.
It’s an integral part of the application architecture and the software delivery lifecycle.”

Gregor Hohpe

Author of “Enterprise Integration Patterns”, “Cloud Strategy”, and “The Software Architect Elevator”

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

Common application integration options

- File transfer
- Shared databases
- Connectors
- APIs (and all their cranky uncles)
- Messaging
- Streaming
- Orchestration

aws

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

Common application integration options

- File transfer
- Shared databases
- Connectors
- APIs (and all their cranky uncles)
- Messaging
- Streaming
- Orchestration



Our focus for today

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

Beware of the faith healer

"No architecture decision comes without trade-off.
Every architecture decision brings you pain.
Your job as software architect is to identify the least painful option on the table."

The guild of commonsensible software architects

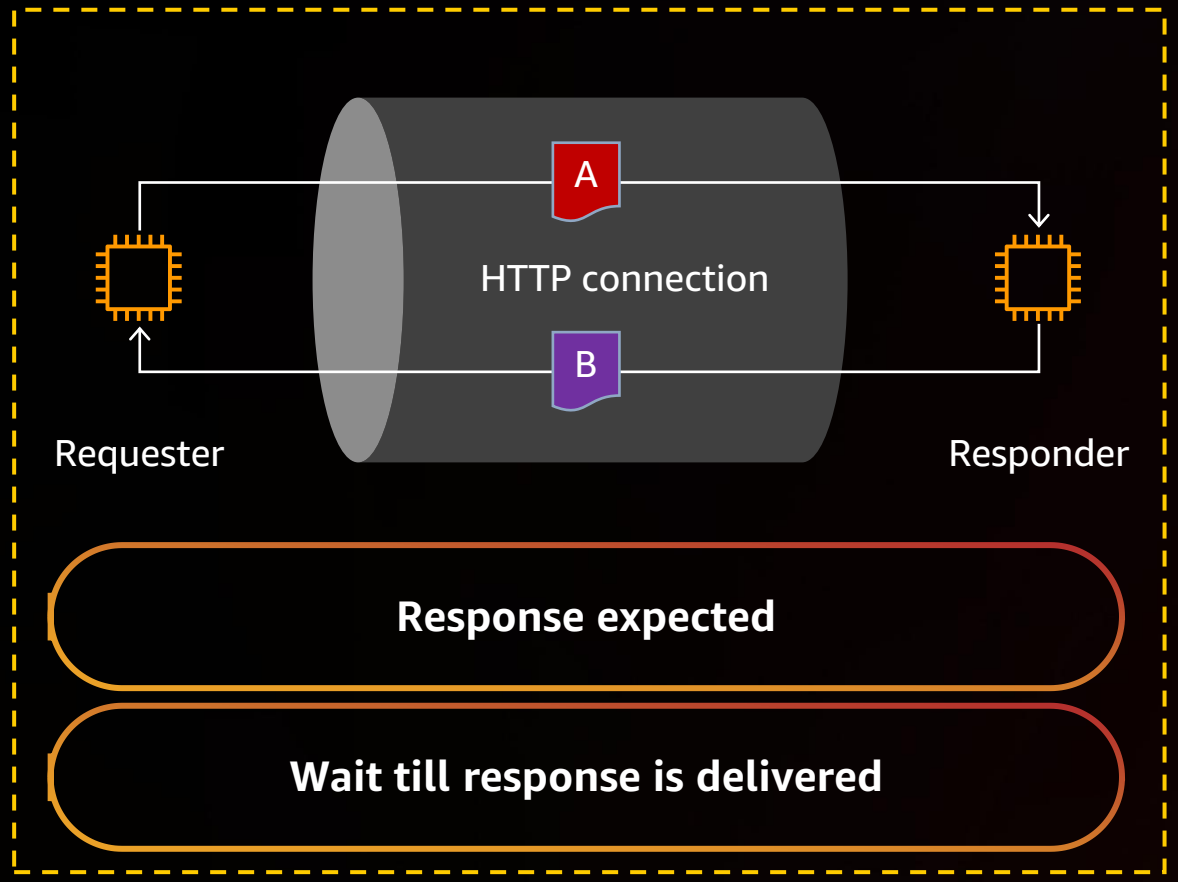
aws
© 2022, Amazon Web Services, Inc. or its affiliates. All rights reserved.

Synchronous vs. asynchronous communication

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

Communication exchange

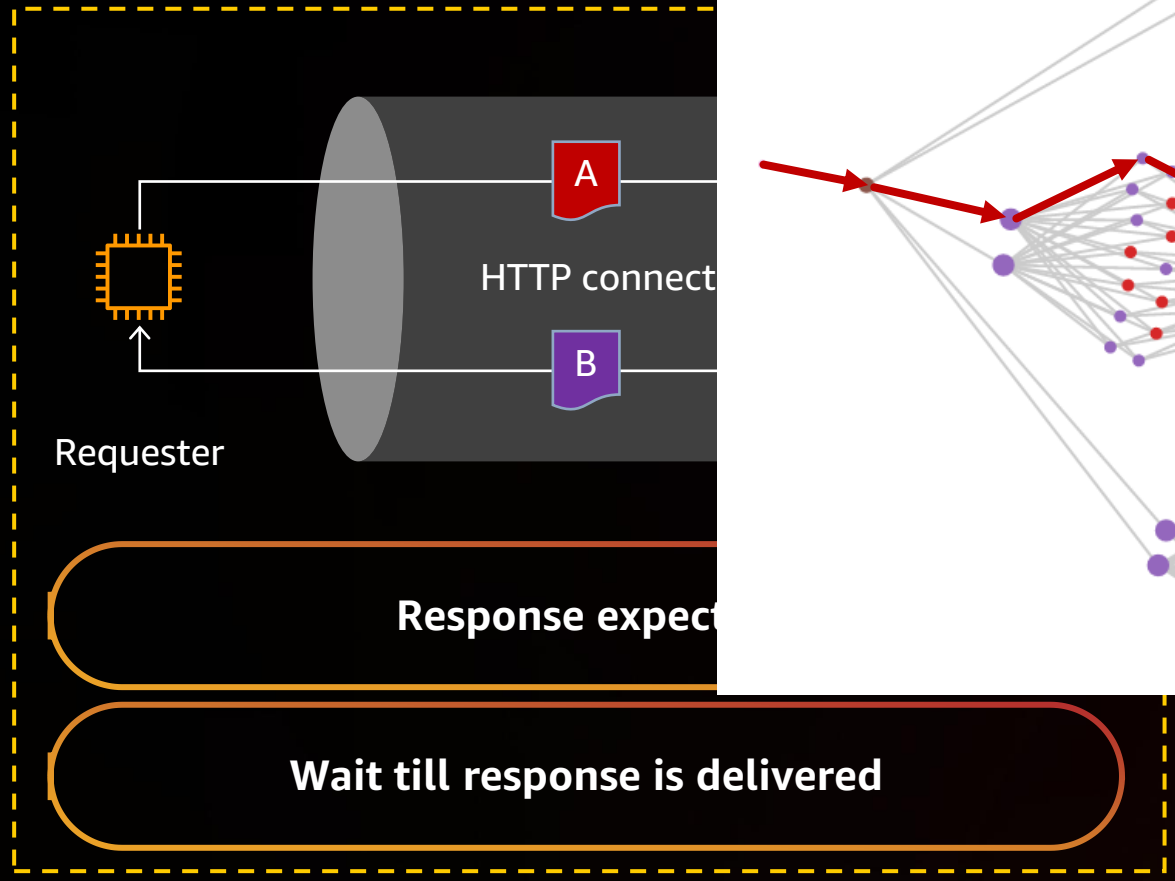
Synchronous request-response



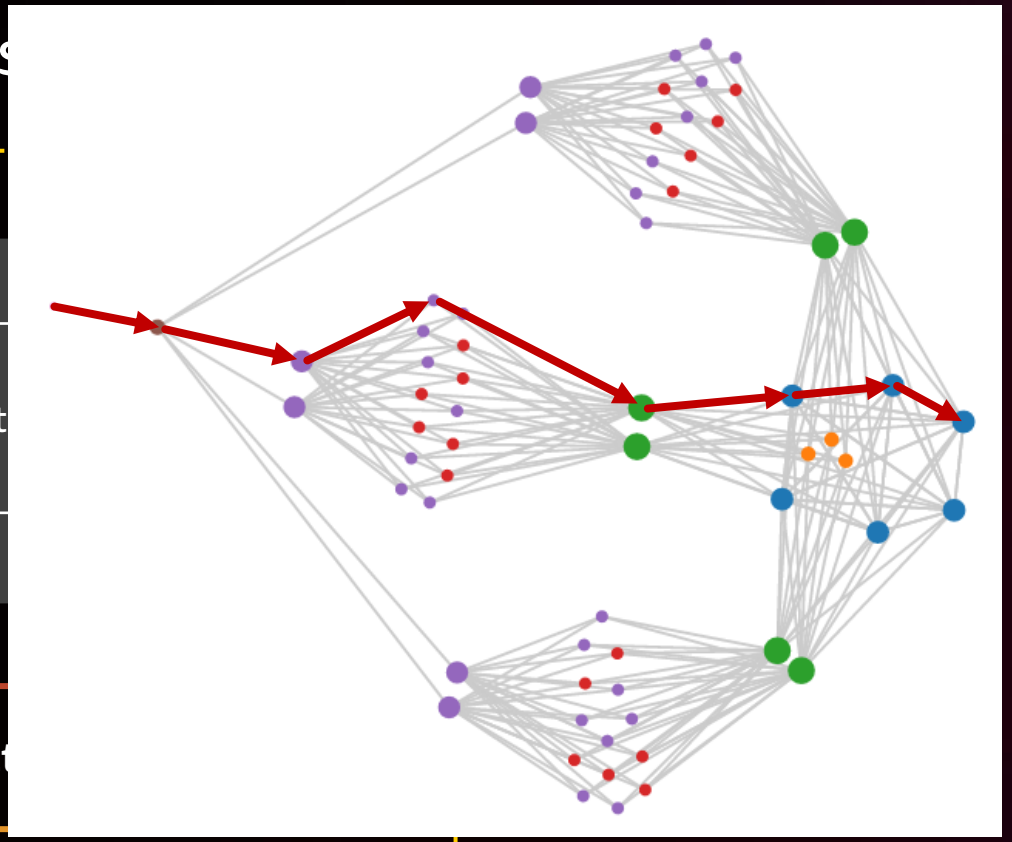
© 2022, Amazon Web Services, Inc. or its affiliates. All rights reserved.

Communication exchange

Synchronous request-res



Example of a dependencies graph at Netflix

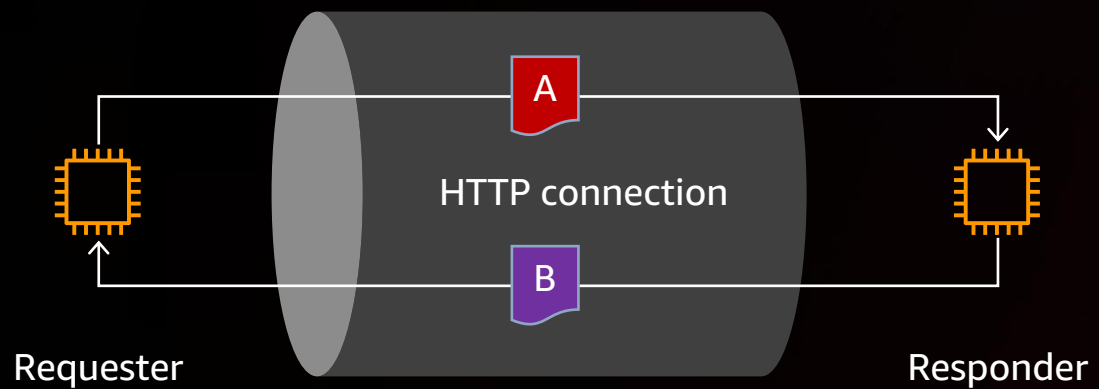


© 2022, Amazon Web Services, Inc. or its affiliates. All rights reserved.

https://github.com/adrianco/spigo/blob/master/png/netflixoss.png

Communication exchange

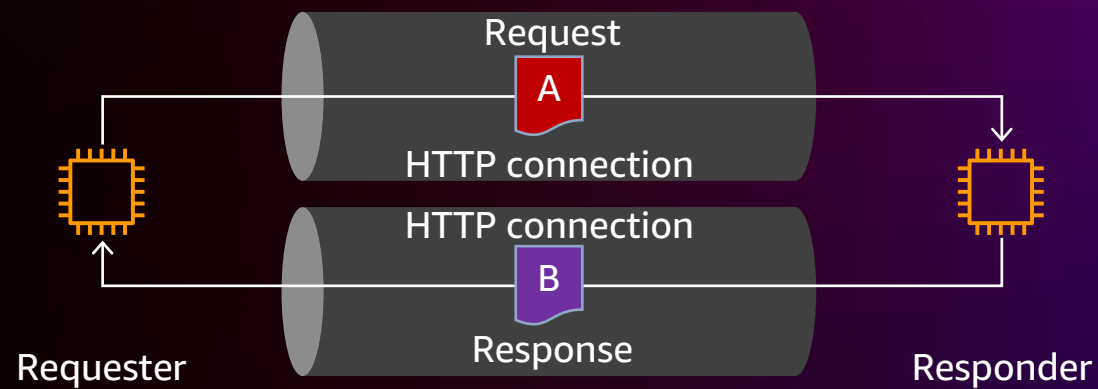
Synchronous request-response



Response expected

Wait till response is delivered

Asynchronous request-response



Return address, correlation ID

No active waiting for response

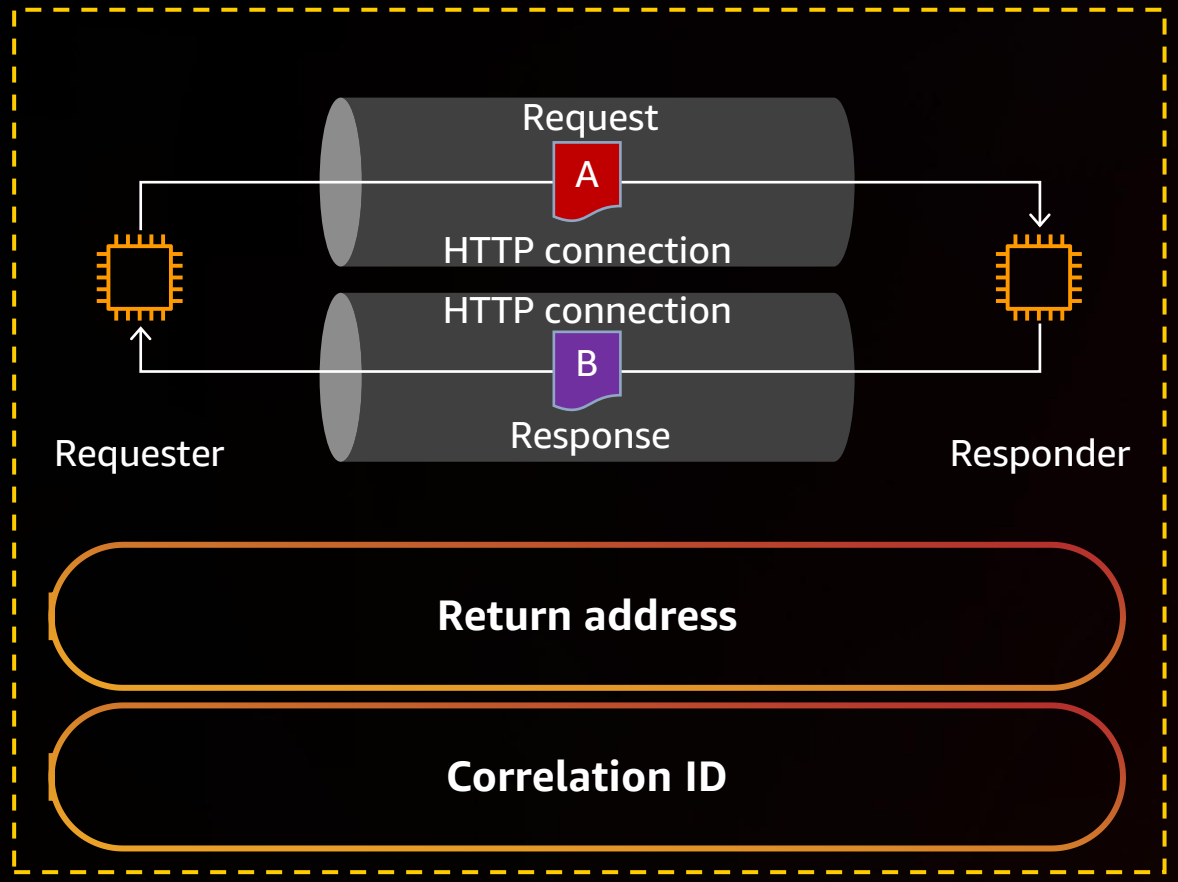
© 2022, Amazon Web Services, Inc. or its affiliates. All rights reserved.

APIs vs. messaging

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

APIs vs. messaging

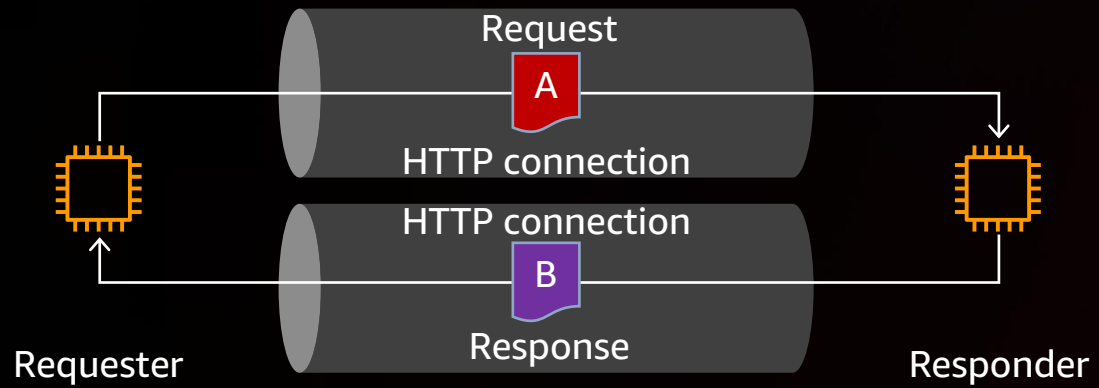
Asynch request-response (APIs)



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

APIs vs. messaging

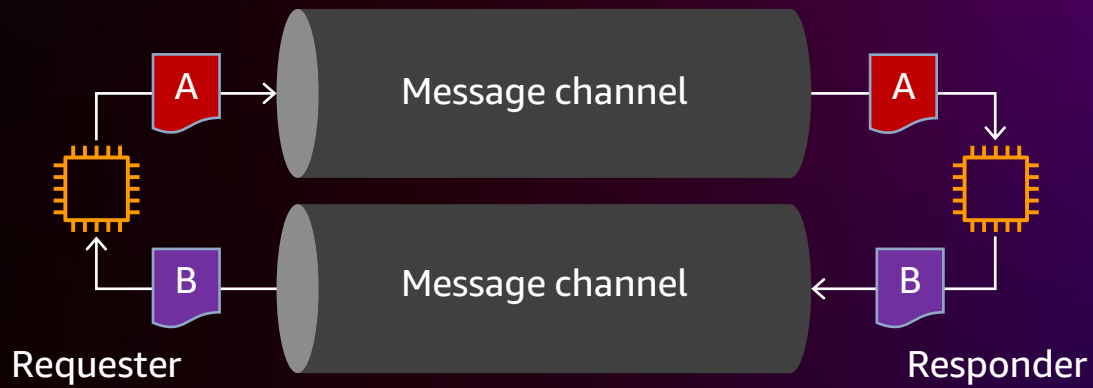
Asynch request-response (APIs)



Return address

Correlation ID

Asynch request-response (messaging)



Point-to-point (queue)

Publish-subscribe / fan-out (topic or bus)

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

Patterns

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

Patterns

Gregor Hohpe and Bobby Woolf:

Enterprise Integration Patterns

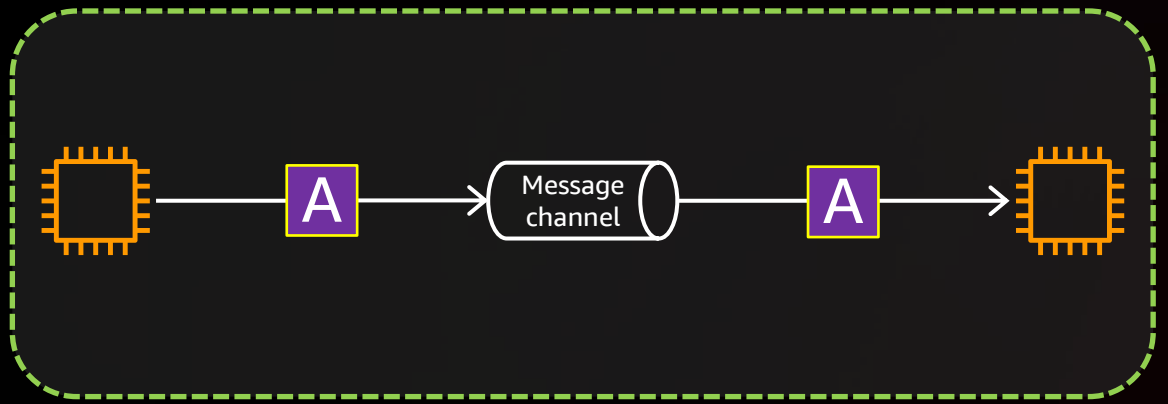
Designing, Building, and Deploying Messaging Solutions

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

Message exchange

Integration pattern

One-way



Sender Receiver

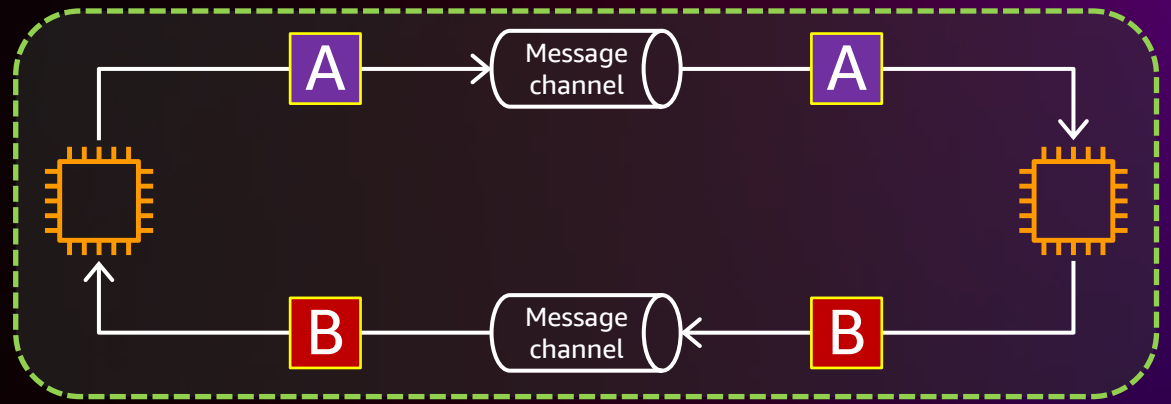
No response expected

Message channel decouples parties

Message intent: Command-, document-, event-message

Conversation pattern

Asynchronous request-response



Requester Responder

Response expected

Message channels decouple parties

Return address

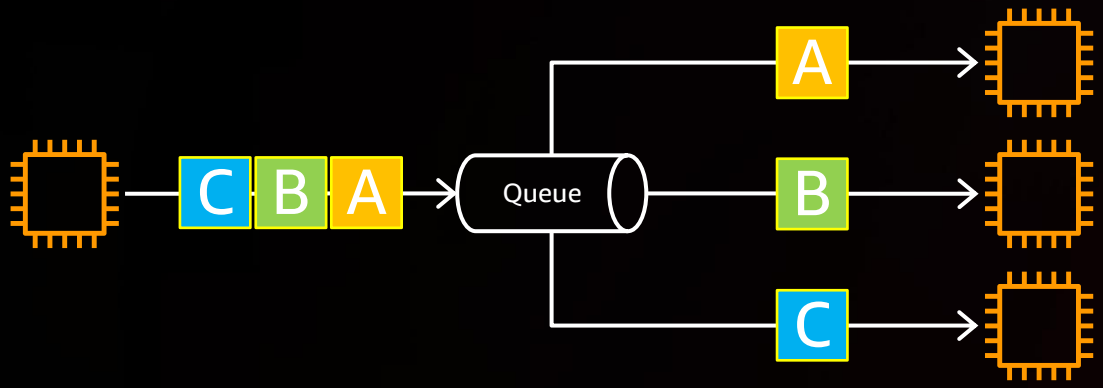
Correlation ID

aws

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

Message channels

Point-to-point (queue)

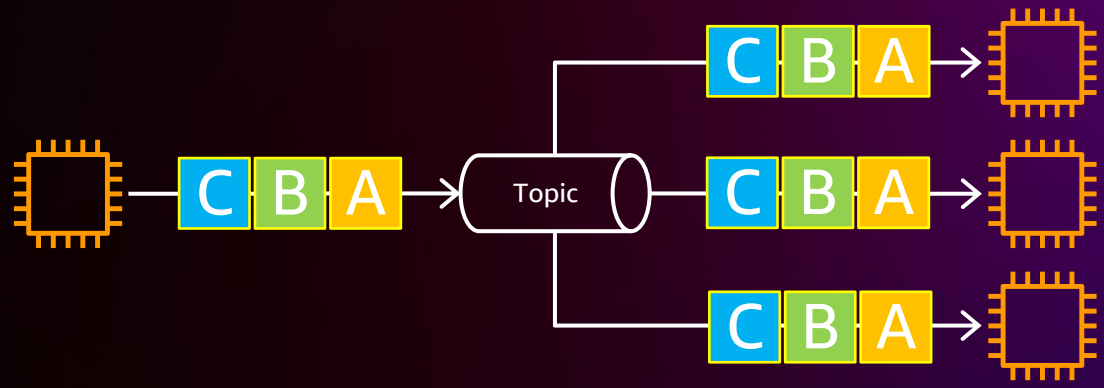


Each message consumed by one receiver

Competing consumers

Buffering load balancer

Publish-subscribe / fan-out (topic)



Each message consumed by each subscriber

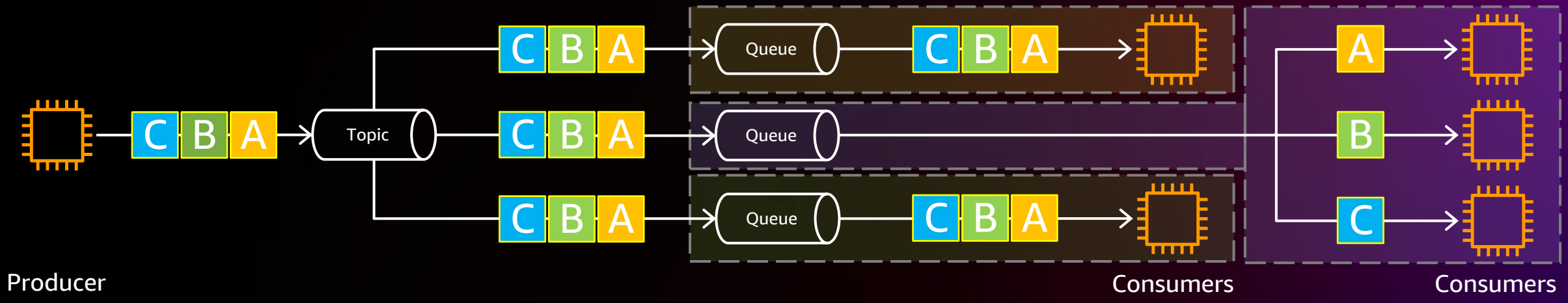
How to scale consumers

How to not miss a message when down

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

Message channels

Composite pattern: Topic-queue-chaining



Durable subscriber pattern: How to avoid missing messages while not listening?

Concurrent consumer pattern: How to scale consumers of publish-subscribe channels?

Best of both worlds: fan-out & consumer scale-out & buffering load balancing all at the same time

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

Message channels

Publish-subscribe / fan-out (message bus)



Scope: components & applications

Prosumers over producers vs consumers

Additional functionality, but also constraints, e.g.:
content-based router, schema validation, payload format constraint, adapters for 3P software

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

Messaging vs. streaming

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

Messaging vs. streaming

A stream is yet another fan-out pattern, but:

Message processing



Each individual message is a unit of work

Messages **deleted** after consumption

Stream processing



Batches of messages are units of work

Messages **retained** after consumption (until expiration)

aws

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

Integration services: Overview

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

Integration Services

Messaging
Serverless



AWS IoT Core



Amazon SNS



Amazon SQS



Amazon EventBridge

Messaging
Managed



Amazon MQ
(ActiveMQ / RabbitMQ)



Amazon ElastiCache
(Redis)

Streaming
Managed



Amazon MSK

Streaming
Serverless



Amazon Kinesis



Amazon MSK
Serverless

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

Integration Services: Decision guidance

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

Which messaging/streaming service should I use?





Amazon MQ



Amazon ElastiCache (Redis)



AWS IoT Core



Amazon EventBridge



Amazon SNS (Standard)



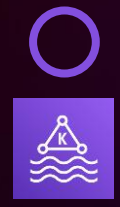
Amazon SNS (FIFO)



Amazon SQS (Standard)



Amazon SQS (FIFO)



Amazon MSK



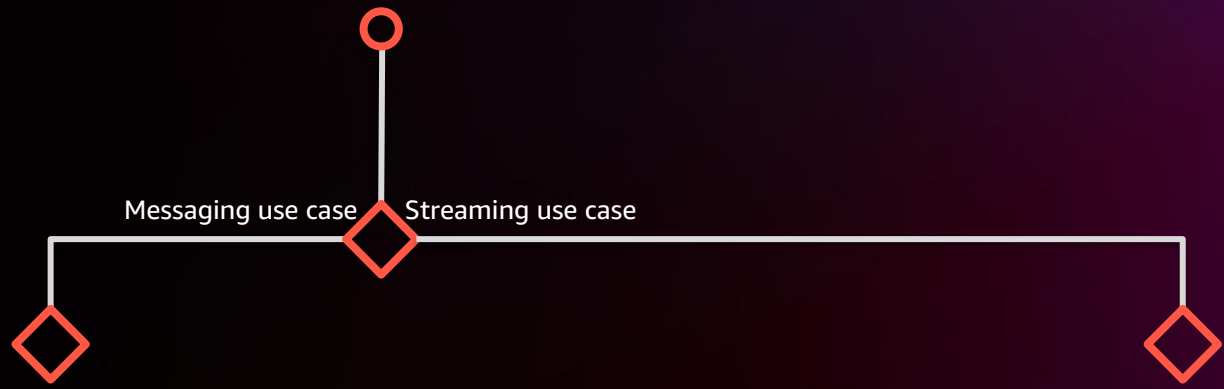
Amazon Kinesis



Amazon Compute/Container

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

Which messaging/streaming service should I use?



Amazon MQ

Amazon ElastiCache (Redis)

AWS IoT Core

Amazon EventBridge

Amazon SNS (Standard)

Amazon SNS (FIFO)

Amazon SQS (Standard)

Amazon SQS (FIFO)

Amazon MSK

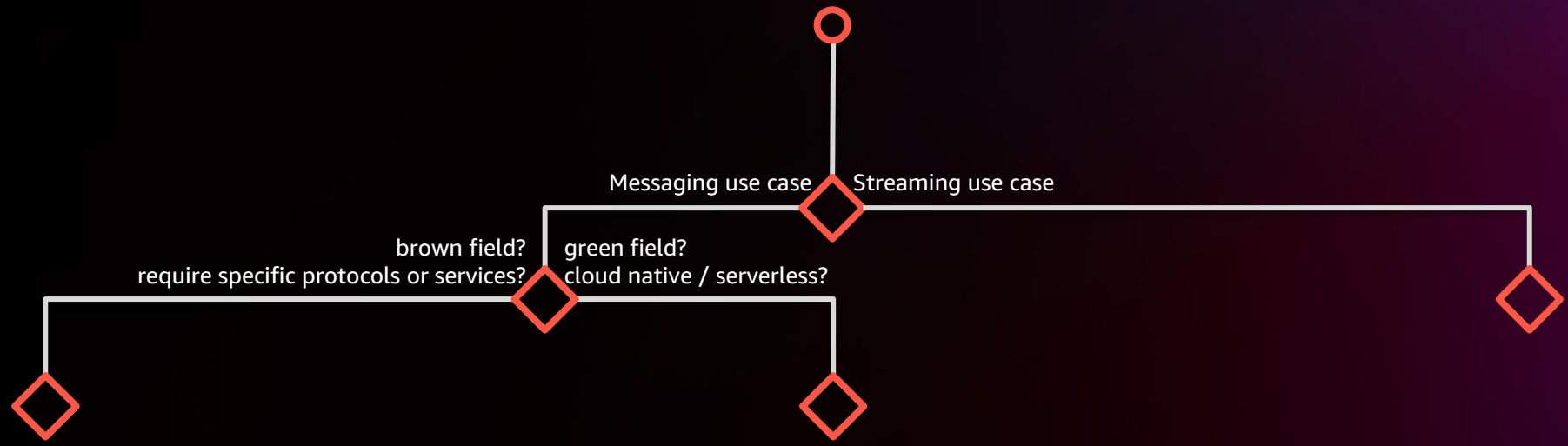
Amazon Kinesis

Amazon Compute/Container

aws

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

Which messaging/streaming service should I use?



Amazon MQ

Amazon ElastiCache (Redis)

AWS IoT Core

Amazon EventBridge

Amazon SNS (Standard)

Amazon SNS (FIFO)

Amazon SQS (Standard)

Amazon SQS (FIFO)

Amazon MSK

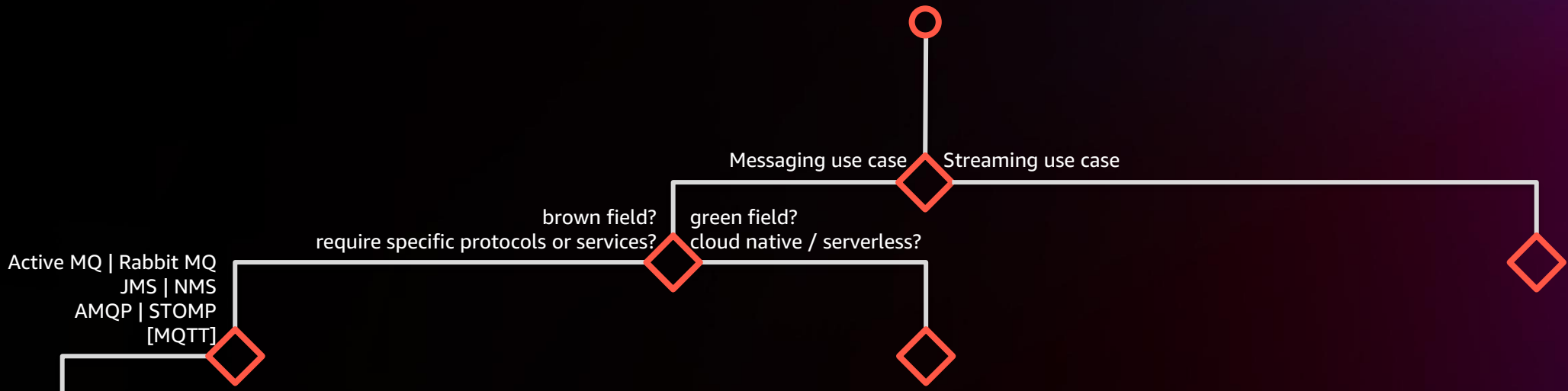
Amazon Kinesis

Amazon Compute/Container

aws

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

Which messaging/streaming service should I use?

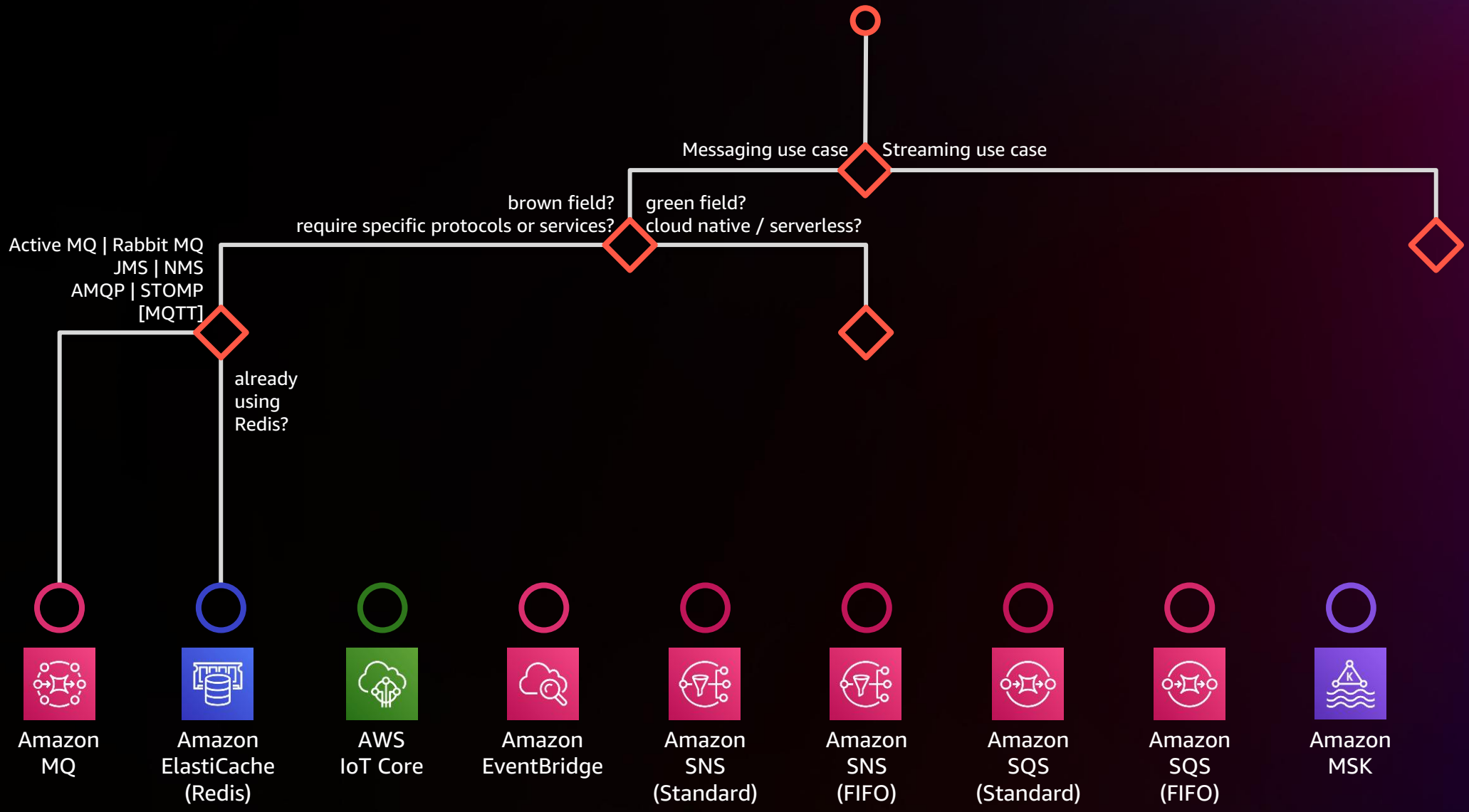




Amazon MQ | Amazon ElastiCache (Redis) | AWS IoT Core | Amazon EventBridge | Amazon SNS (Standard) | Amazon SNS (FIFO) | Amazon SQS (Standard) | Amazon SQS (FIFO) | Amazon MSK | Amazon Kinesis | Amazon Compute/Container

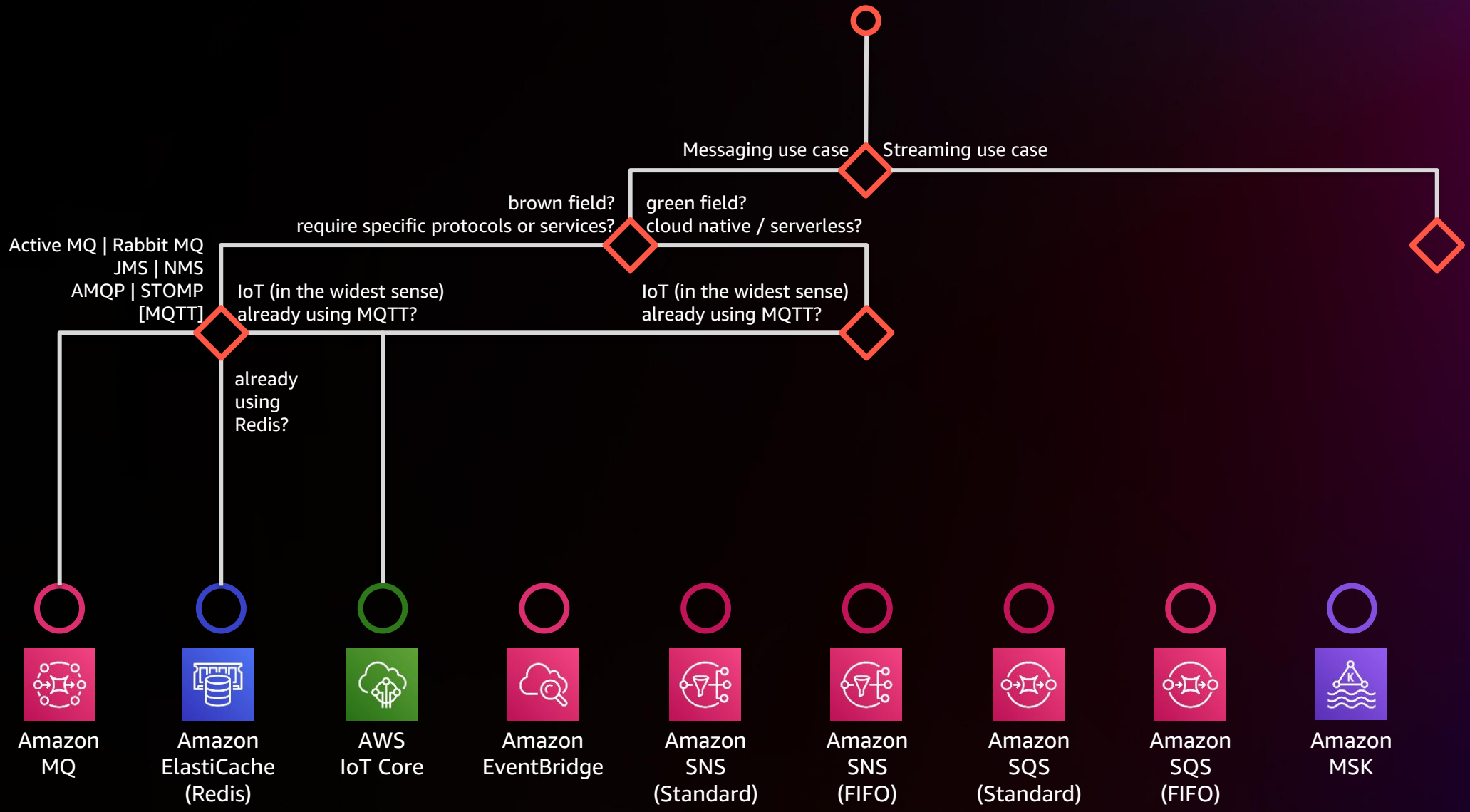
© 2022, Amazon Web Services, Inc. or its affiliates. All rights reserved.

Which messaging/streaming service should I use?



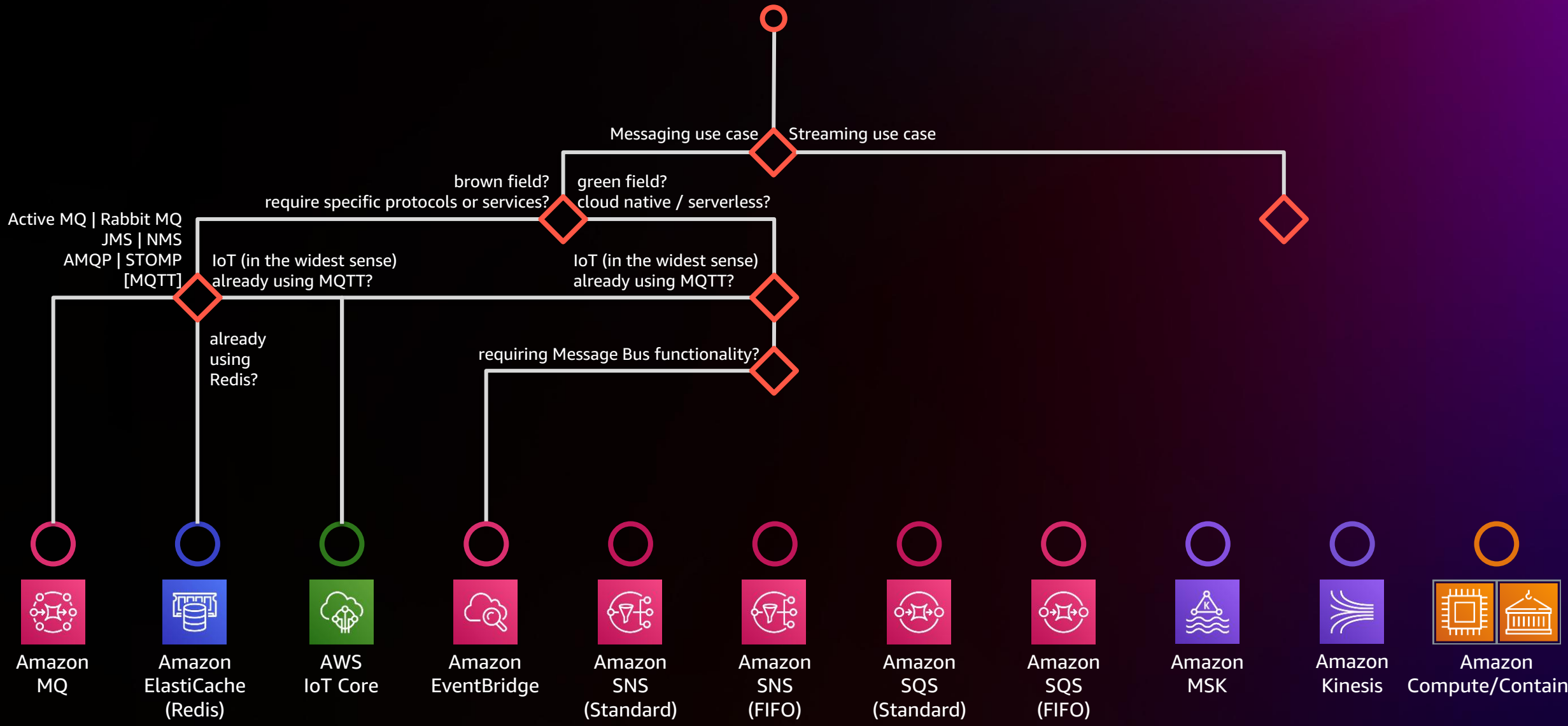
© 2022, Amazon Web Services, Inc. or its affiliates. All rights reserved.

Which messaging/streaming service should I use?



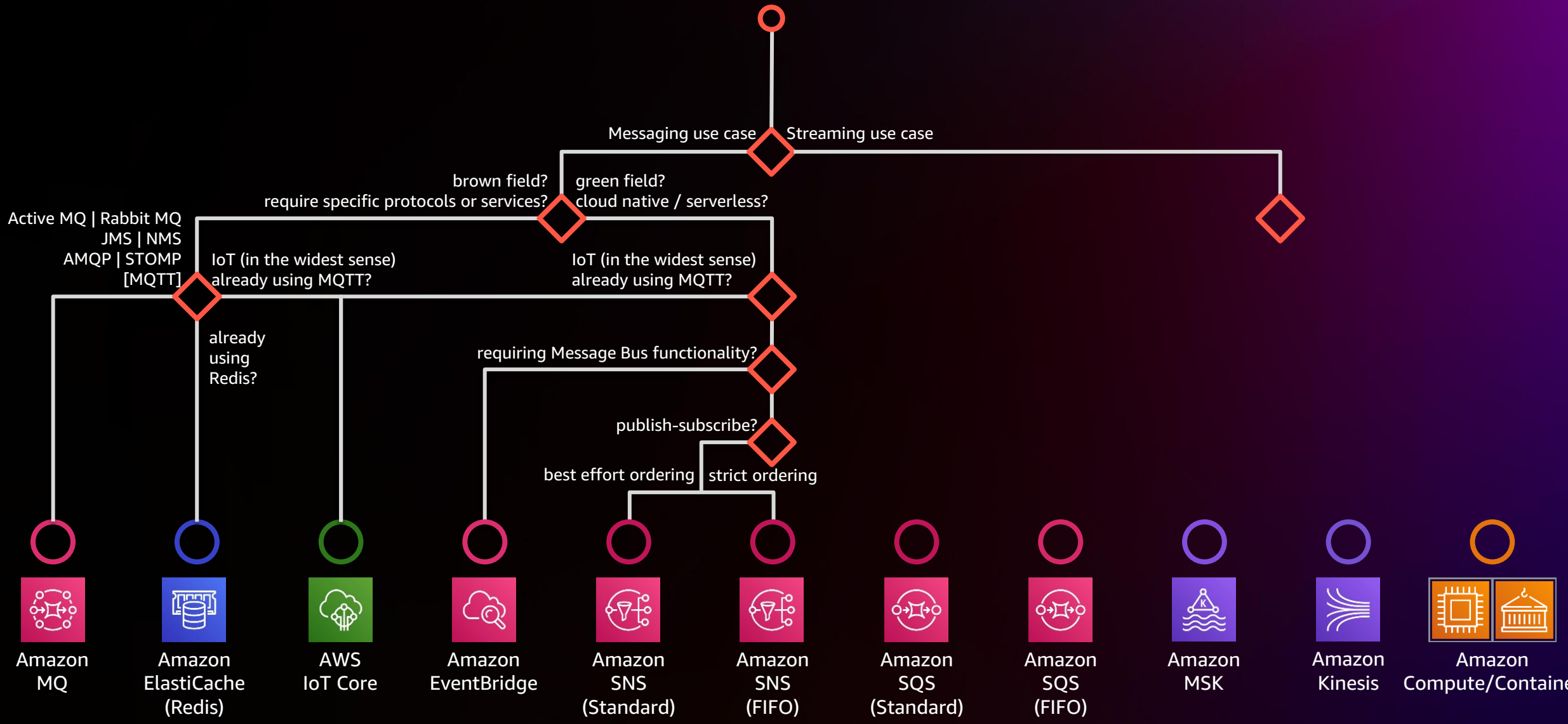
© 2022, Amazon Web Services, Inc. or its affiliates. All rights reserved.

Which messaging/streaming service should I use?



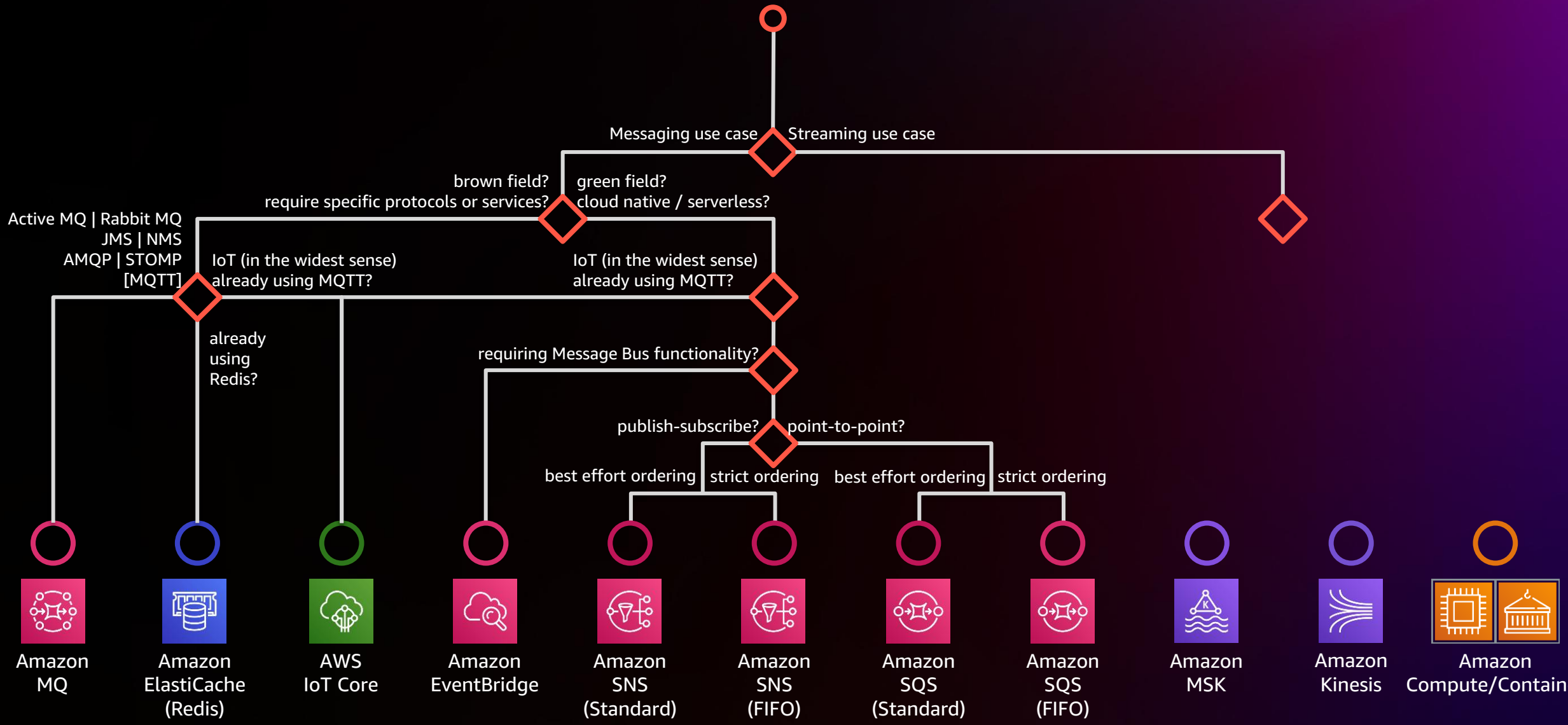
© 2022, Amazon Web Services, Inc. or its affiliates. All rights reserved.

Which messaging/streaming service should I use?



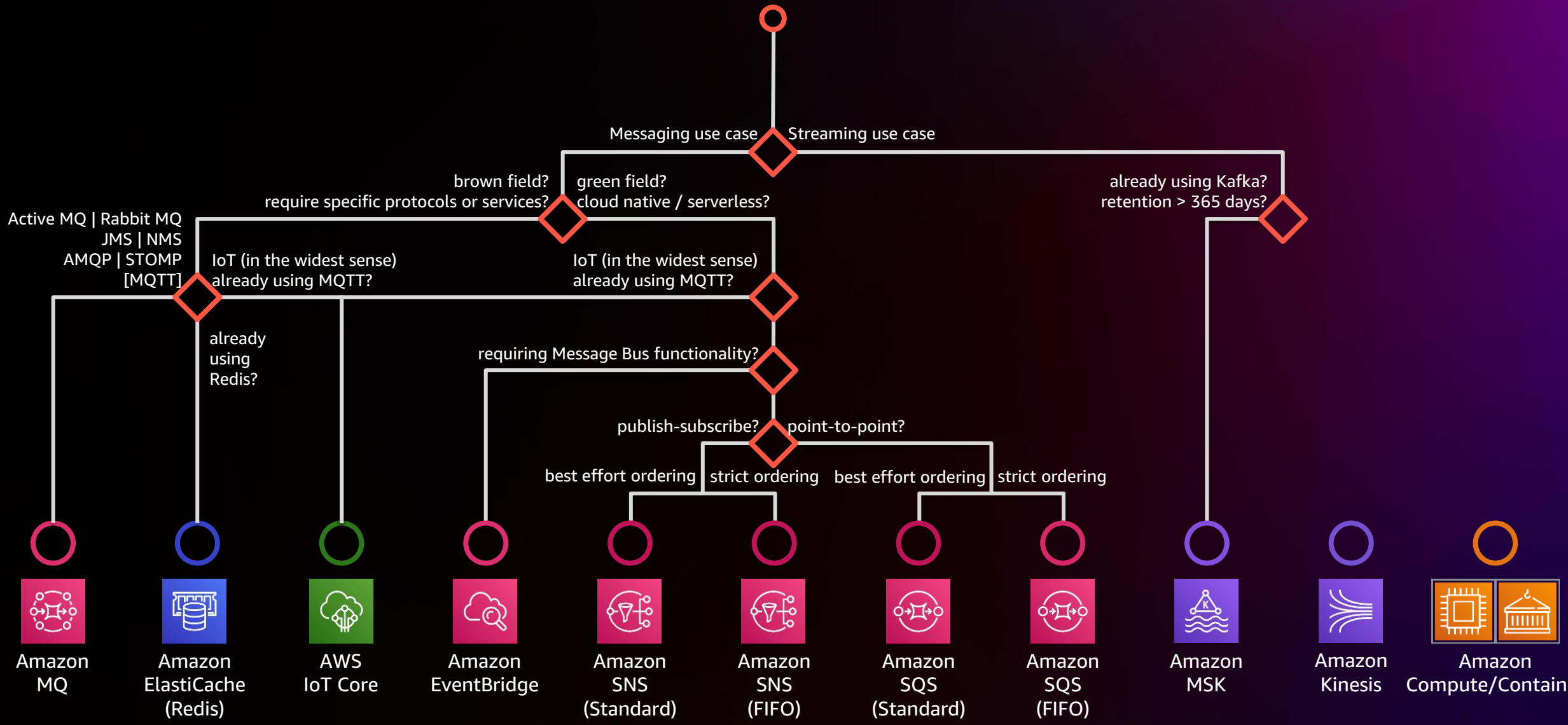
© 2022, Amazon Web Services, Inc. or its affiliates. All rights reserved.

Which messaging/streaming service should I use?



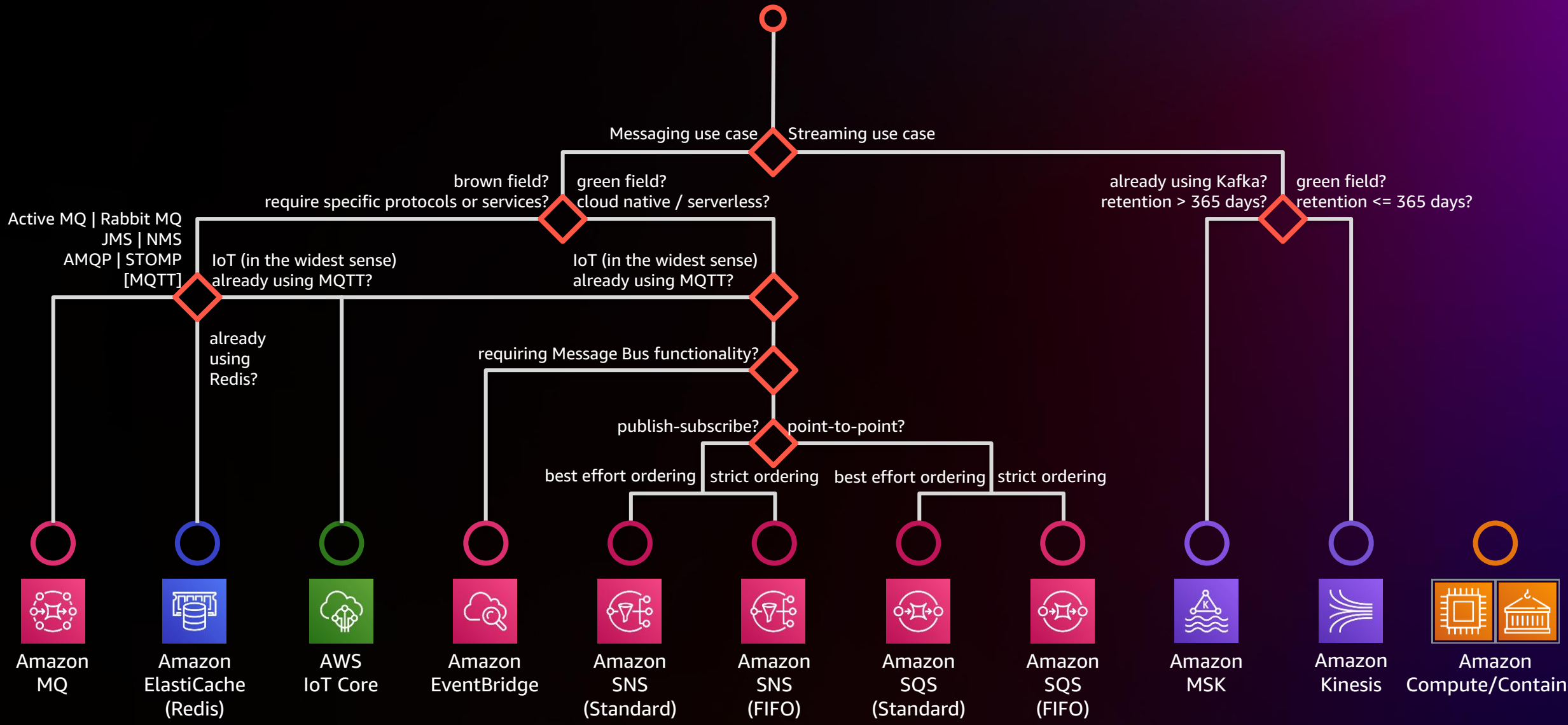
© 2022, Amazon Web Services, Inc. or its affiliates. All rights reserved.

Which messaging/streaming service should I use?



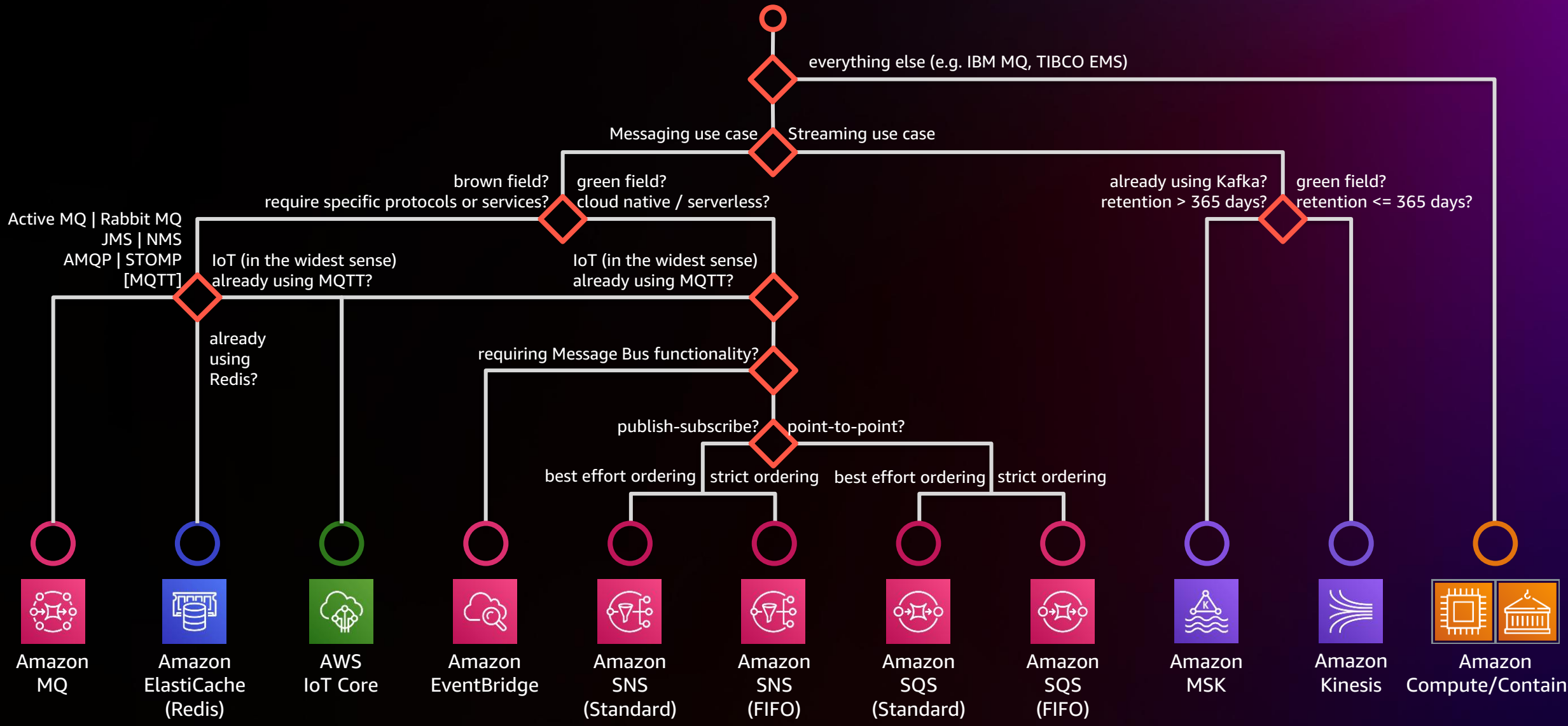
© 2022, Amazon Web Services, Inc. or its affiliates. All rights reserved.

Which messaging/streaming service should I use?



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

Which messaging/streaming service should I use?



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

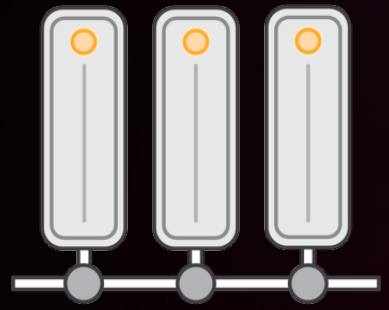
Integration services: Differentiation

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

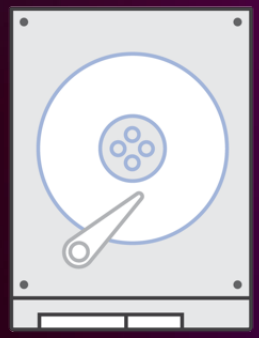
Integration services: Differentiation



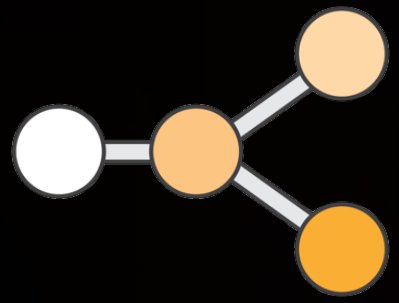
Scalability



Durability



Persistence



Consumption models



Retries



Pricing

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

Scalability

Service	How to scale
Amazon SNS	Standard: Auto – virtually unlimited FIFO: Auto - up to 300 transactions per second
Amazon EventBridge	Auto - up to 10,000 event ingestions per second in selected regions
Amazon SQS	Standard: Auto – virtually unlimited FIFO: Auto - up to 6,000 transactions per second in selected regions – otherwise 3,000 transactions
Amazon Kinesis Data Streams	Provisioned Mode: Horizontally - add or remove shards to / from the stream Capacity per shard: write <= 1MB or 1k ops / second; read <= 2MB / second On-Demand Mode: Auto
Amazon MQ	Vertically: Choose a larger or smaller instance type Horizontally: Set up a Network of Brokers
Amazon Managed Streaming for Apache Kafka (MSK)	Provisioned Mode: Vertically: Increase storage, and change instance type Horizontally: Expand cluster by adding broker nodes Serverless Mode: Auto
AWS IoT Core Message Broker	Auto
Amazon ElastiCache (Redis)	Vertically: Choose a larger or smaller instance type Horizontally: Add additional shards

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

Durability

Service	Durability of “in flight” messages in the respective channels
Amazon SNS	Replicated across multiple Availability Zones
Amazon EventBridge	Replicated across multiple Availability Zones
Amazon SQS	Replicated across multiple Availability Zones
Amazon Kinesis Data Streams	Replicated across 3 Availability Zones
Amazon MQ	Replicated across multiple Availability Zones (when using persistent messages)
Amazon Managed Streaming for Apache Kafka (MSK)	Kafka: Data is replicated to multiple nodes in a cluster (default is 3) Amazon EBS: Data is replicated across multiple servers in an Availability Zone
AWS IoT Core Message Broker	Replicated across multiple Availability Zones
Amazon ElastiCache (Redis)	Redis: Data can be replicated to multiple shards (0-5) Amazon EBS: Data is replicated across multiple servers in an Availability Zone

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

Persistence

Service	Persistence
Amazon SNS	No formal persistence model; delivery retries up to 23 days (different per protocol) Supports DLQs
Amazon EventBridge	No formal persistence model; delivery retry potentially up to 24 hours Supports DLQs (Note: Option to archive and replay events not persistence in this sense)
Amazon SQS	Up to 14 days; default is 4 days
Amazon Kinesis Data Streams	Up to 365 days (with additional cost); default is 24 hours
Amazon MQ	Storage capacity per broker is 200 GB (20 GB for mq.t2.micro)
Amazon Managed Streaming for Apache Kafka (MSK)	As provisioned (between 1,000 GB and 16,384 GB)
AWS IoT Core Message Broker	Persistent Sessions store subscription information and pending QoS 1 messages should the devices become disconnected.
Amazon ElastiCache (Redis)	No formal persistence model, as it's an in-memory database

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

Consumption

Service	Invocation model
Amazon SNS	Push-based Targets: Lambda, SQS, Kinesis Data Firehose, HTTP, SMS, mobile push, SMTP
Amazon EventBridge	Push-based Targets: 28 AWS service targets, including Lambda, SQS, SNS, ECS, Step Functions, etc.
Amazon SQS	Pull-based Integrated with Lambda
Amazon Kinesis Data Streams	Pull-based, clients have to maintain a checkpoint or cursor in the stream Integrated with Lambda
Amazon MQ	Pull- or push-based, typically with protocols like JMS, NMS, AMQP, or MQTT Integrated with Lambda
Amazon Managed Streaming for Apache Kafka (MSK)	Pull-based, clients have to maintain a checkpoint or cursor in the stream Integrated with Lambda
AWS IoT Core Message Broker	Push-based Targets: Devices (in the widest sense), Device Shadows, Rules Engine
Amazon ElastiCache (Redis)	Pull- or push-based

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

Retries & failure handling

Service	Failure handling
Amazon SNS	Different retry phases (immediate, pre-backoff, backoff, post-backoff) Kinesis Data Firehose, Lambda, SQS: <= 100,015 times, over <= 23 days SMTP, SMS, mobile push: Up to a total of 50 attempts, over 6 hours HTTP: custom
Amazon EventBridge	Retries for <= 24 hrs & <= 185 times w/ exponential back off + jitter (randomized delay) If DLQ configured: hopeless cases (e.g. no permission, target !exists) directly go to DLQ
Amazon SQS Amazon MQ	Messages remain in queue until deleted, or expired Consumer's responsibility to retry message retrieval
Amazon Kinesis Data Streams Amazon Managed Streaming for Apache Kafka (MSK)	Messages remain in stream until expired Consumer's responsibility to move on with checkpoint or cursor
AWS IoT Core Message Broker	AWS IoT Core retries delivery of unacknowledged quality of service 1 (QoS 1) publish requests to a client for up to one hour.
Amazon ElastiCache (Redis)	Messages remain in queue until consumed, or expired Consumer's responsibility to retry message retrieval

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

Pricing

Service	Pricing model & cost (us-east-1)
Amazon SNS	Standard: Per API req (pub per 64KB chunks, topic ops, subscription ops) - $0.50 per 1M FIFO: Per pub API requests (any size) - $0.30 per 1M and $0.017 per GB of payload data
Amazon EventBridge	Per events published $1.00 / million events published (64 KB chunks) – AWS service events are free
Amazon SQS	Standard: Per API requests (64 KB chunks for messages) – <= $0.40 per million requests FIFO: Per API requests (64 KB chunks for messages) – <= $0.50 per million requests
Amazon Kinesis Data Streams	Provisioned Mode: Per shard hour ($0.015) + PUT payload (256 KB chunks, 1 million units: $0.014) Additional charges for (extended | long-term) data retention and enhanced fan-out On-Demand Mode: Per stream hour ($0.04) + data ingest ($0.08 per GB) + data retrievals ($0.04) Additional charges for (extended | long-term) data retention and enhanced fan-out
Amazon MQ	Per instance-hour and monthly storage usage (for EFS: average w/ hourly granularity) Instance-hour: $0.02704 to $2.304 – Storage: $0.10 (EBS) to $0.30 (EFS) per GB-month
Amazon Managed Streaming for Apache Kafka (MSK)	Provisioned Mode: Per instance-hour and storage provisioned Instance-hour: from $0.0456 to $10.08 – Storage provisioned: $0.10 per GB-month Serverless Mode: Per cluster hour ($0.75) + partition hour ($0.0015) + data ingest ($0.10 per GB) + data retrievals ($0.05)
AWS IoT Core Message Broker	Per connected minute and per message Connectivity: $0.08 per million minutes – MQTT and HTTP messages: $1.00 per million
Amazon ElastiCache (Redis)	Per instance-hour Instance-hour: $0.016 to $12.477

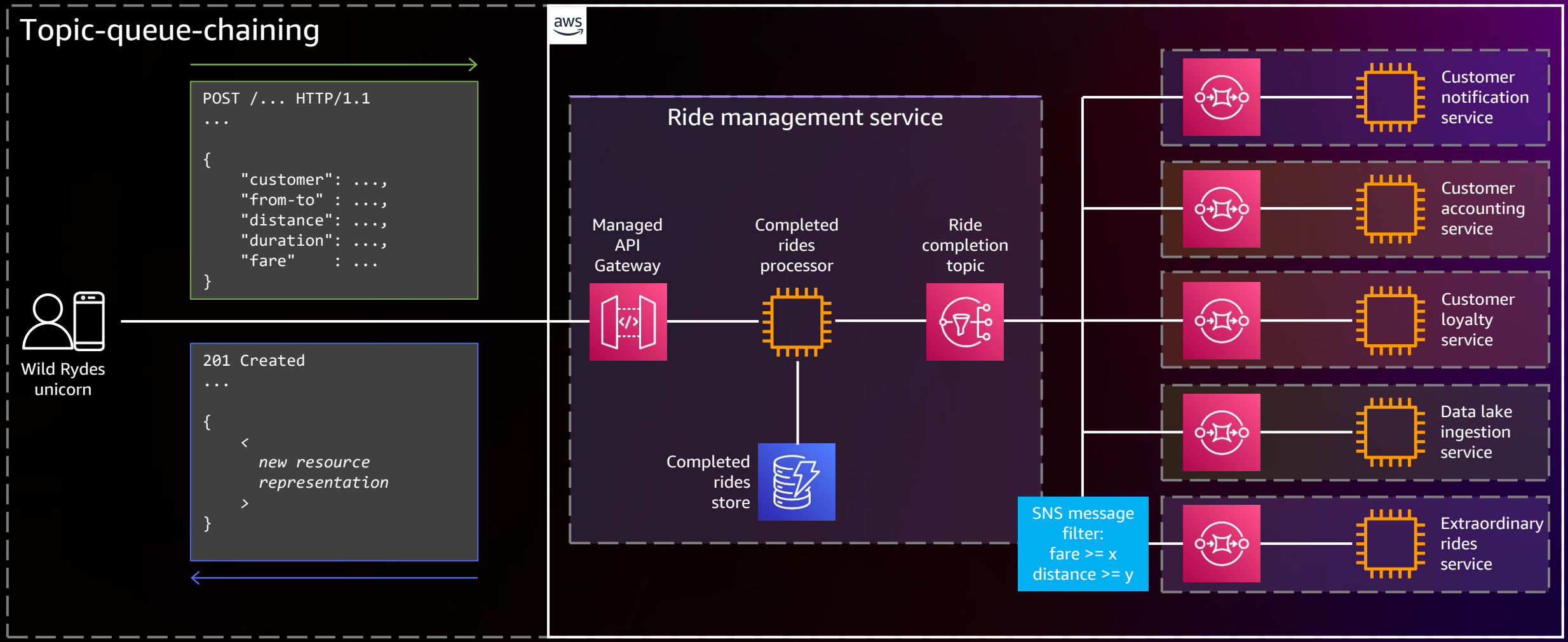
aws
© 2022, Amazon Web Services, Inc. or its affiliates. All rights reserved.

Open discussion:
Share your integration challenges!

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

Submit ride completion

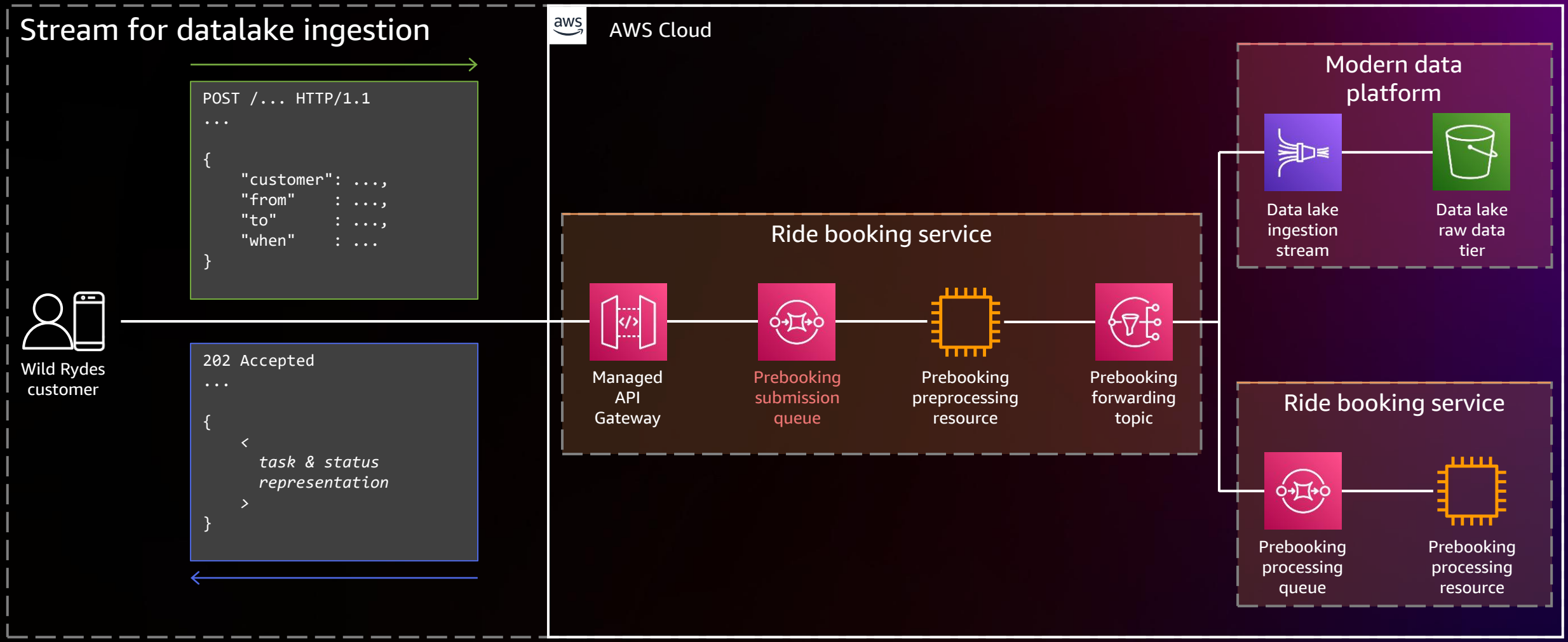
USE CASE



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

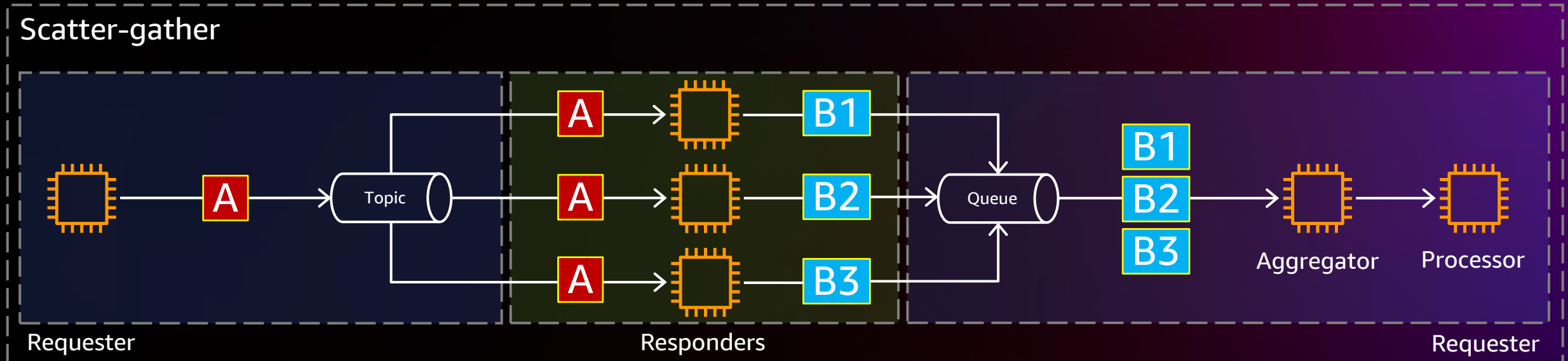
Prebooking campaigns

USE CASE



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

Message routing



How to distribute a request to relevant/interested parties and capture their individual responses?

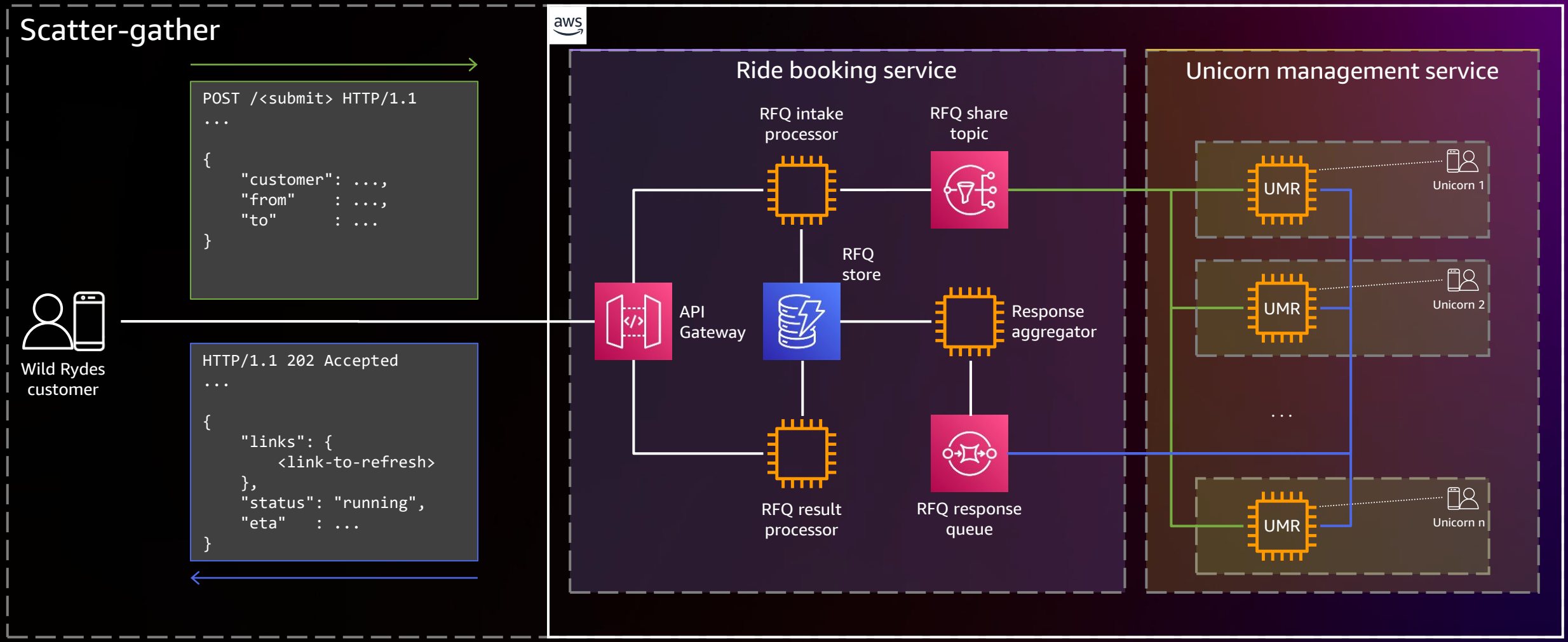
For election or parallel processing scenarios, i.e., search for best response or accumulate responses

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

Instant ride RFQ

UMR: Unicorn management resource

USE CASE



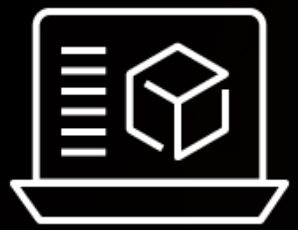
© 2022, Amazon Web Services, Inc. or its affiliates. All rights reserved.

Resources and call-to-action

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

Continue your AWS Serverless learning

Learn at your own pace



Expand your Serverless skills with our Learning Plan on **AWS Skill Builder**

Increase your knowledge



Use our **Ramp-Up Guides** to build your Serverless knowledge

Earn AWS Serverless badge



Demonstrate your Knowledge by achieving **digital badges**

https://s12d.com/serverless-learning

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

Resources and call to action

AWS blog series

Read again about the use cases and patterns from this talk (and more):

AWS architecture blog

Various blog posts in the application integration category:

Reach out to your friendly
AWS solutions architect

and keep in mind

Loose coupling is **always better** than **lousy** coupling

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

Thank you!

Dirk Fröhner

Twitter: @dirk_f5r

LinkedIn:

Christian Müller

Twitter: @ChristianM

LinkedIn:



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

aws

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