aws modern apps

Building Event-Driven Architectures on AWS

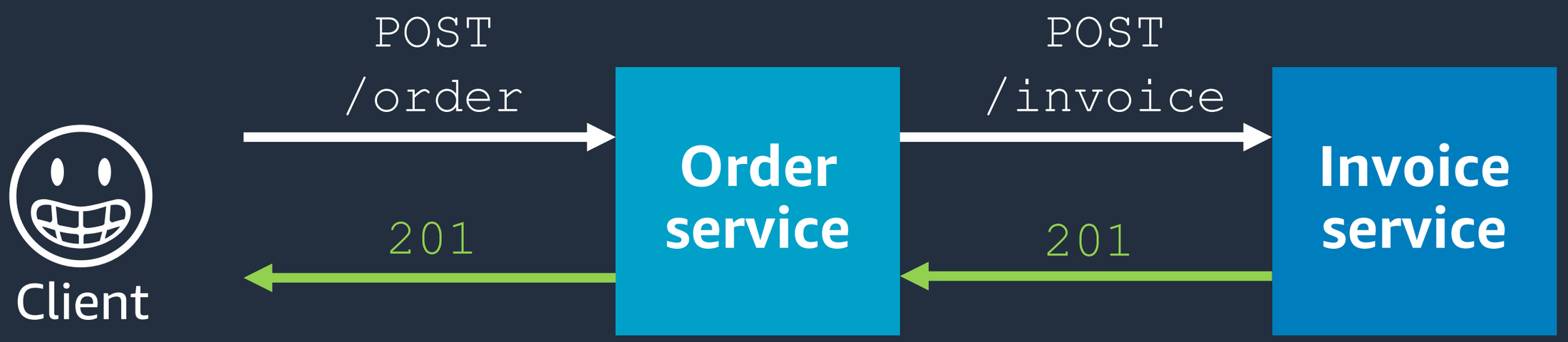
Sam Dengler
AWS Principal Solutions Architect

Agenda

- Challenges with distributed systems
- Journey to event-driven architectures
- Design considerations
- Next Steps

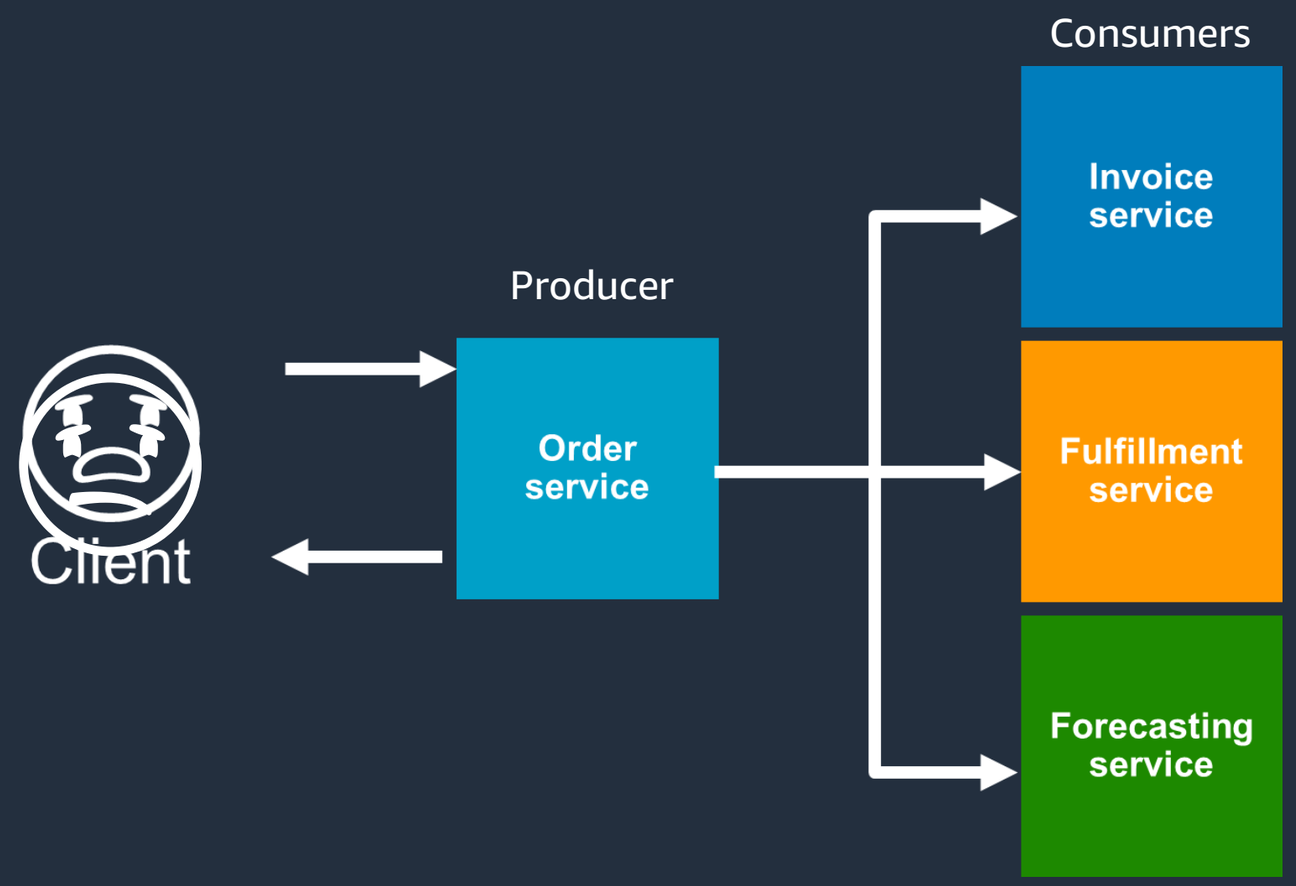
Challenges with Distributed Systems

Microservices start simple



aws modern apps

Synchronous API challenges over time



Coupling between producers and consumers

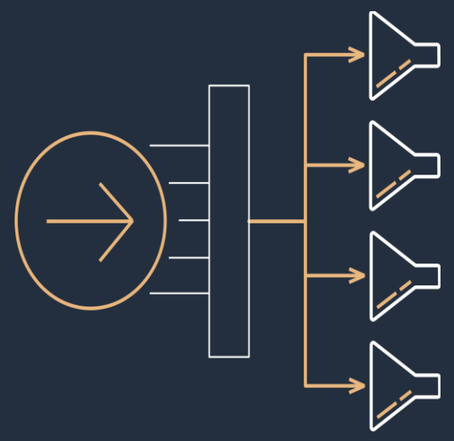
Multiple points of failure

Varying degree of quality of service

External dependencies

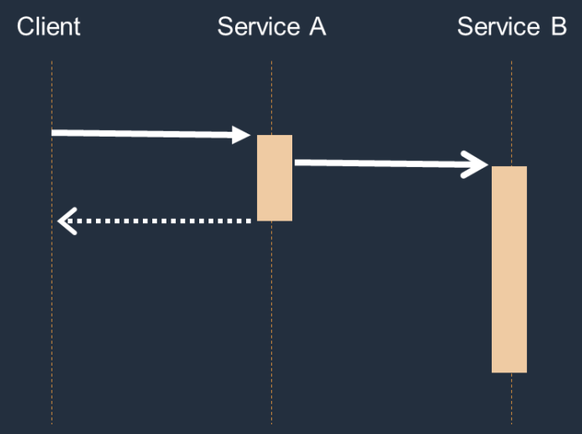
aws modern apps

Event-driven architectures drive reliability and scalability



Event Routers

Abstract producers and consumers from each other



Asynchronous Events

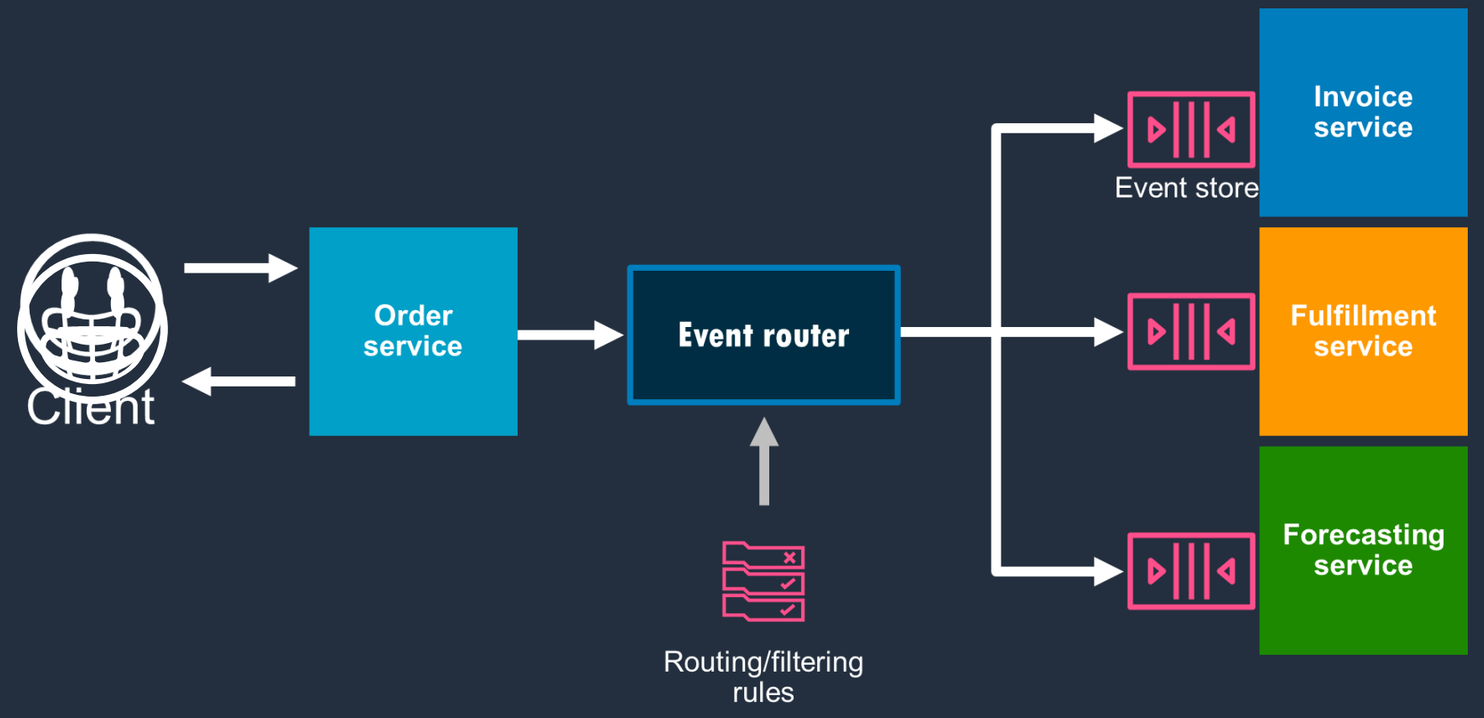
Improve responsiveness and reduce dependencies

Event Stores

Buffer messages until services are available to process

aws modern apps

Reliable, resilient, and independently scalable



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

Tim Bray
General-purpose Internet-software geek

aws modern apps

Journey to Event-Driven Architectures

aws modern apps

Journey to event-driven architectures

STEP 1

Start with the domain

aws modern apps

event

[i-’vent] **noun**

A signal that a system’s state has changed.

aws modern apps

Event-driven modeling

1. Identify business events, processes, actors, etc.

2. Cluster-related concepts

3. Define bounded contexts and sub-domains

aws modern apps

Event-driven architectures start with business events

OrderCreated

OrderPicked

OrderShipped

ReturnRequested

ReturnReceived

aws modern apps

Map events to business domains using attributes, not entities

OrderCreated

Retail

OrderPicked

OrderShipped

Fulfillment

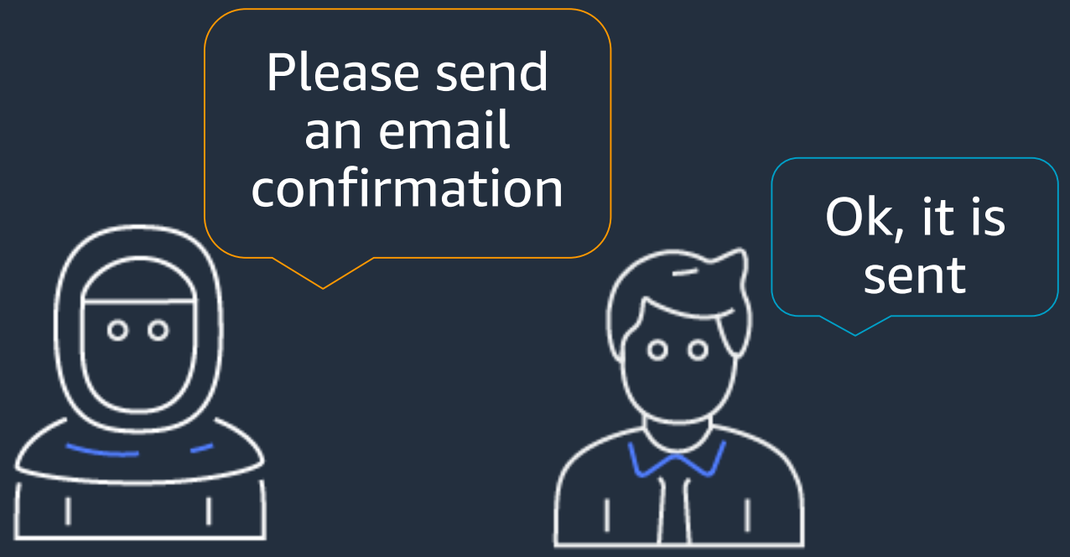
ReturnRequested

ReturnReceived

Support

aws modern apps

Events are observable, not directed

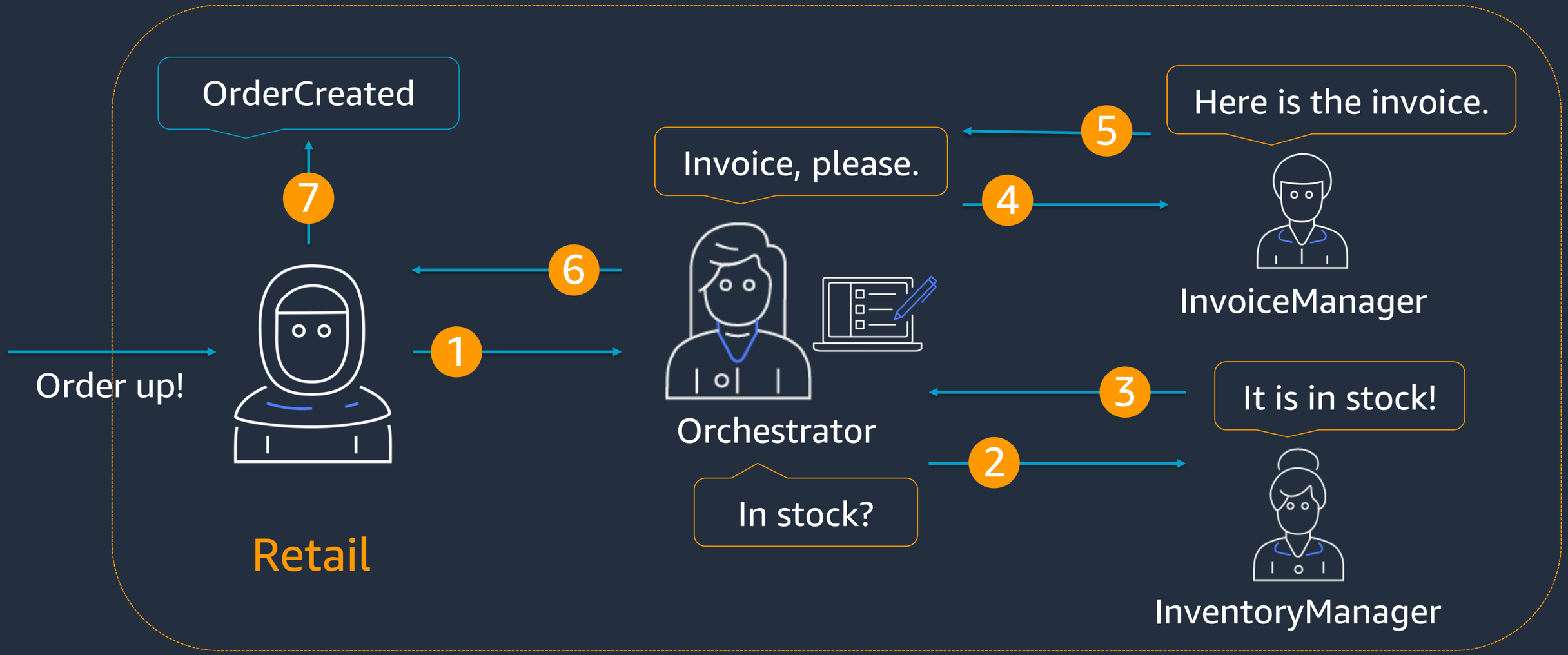


Directed commands

Observable events

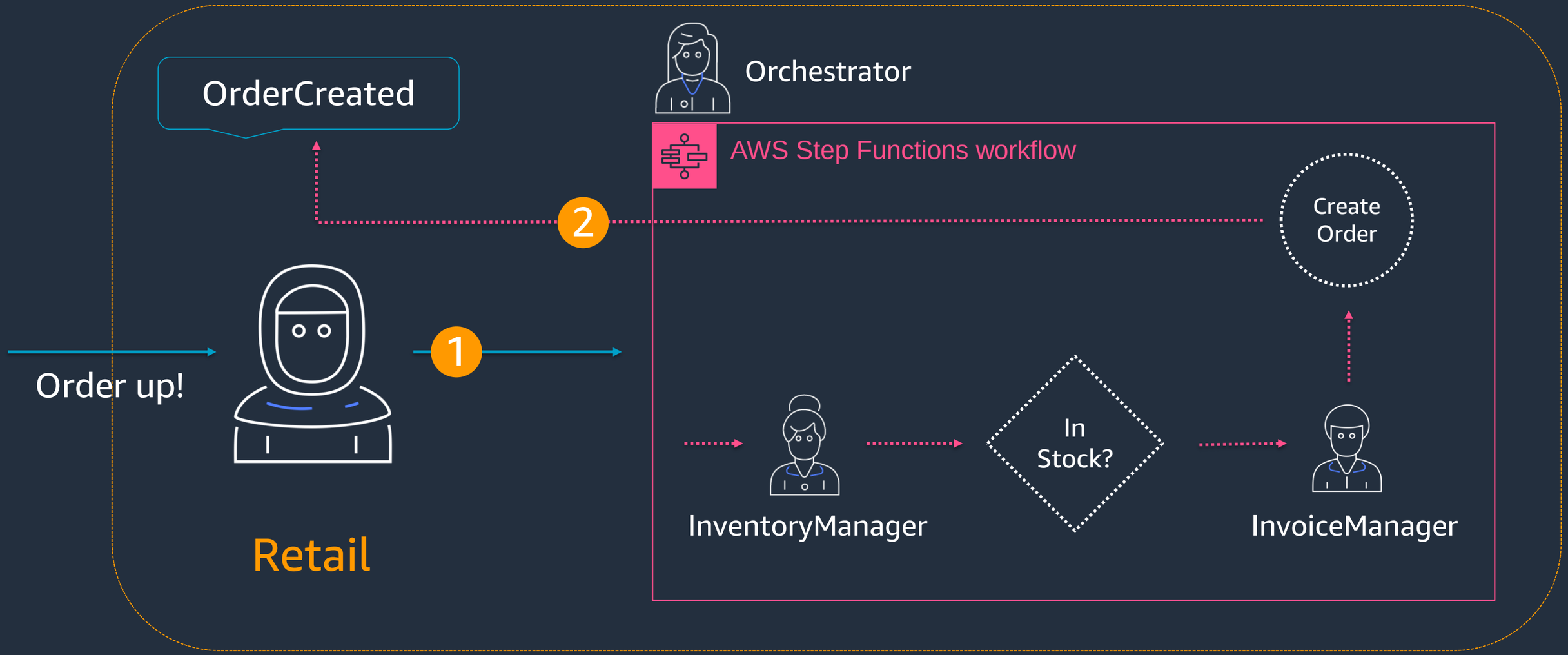
aws modern apps

Orchestrate a business process *within a domain*, resulting in a published event



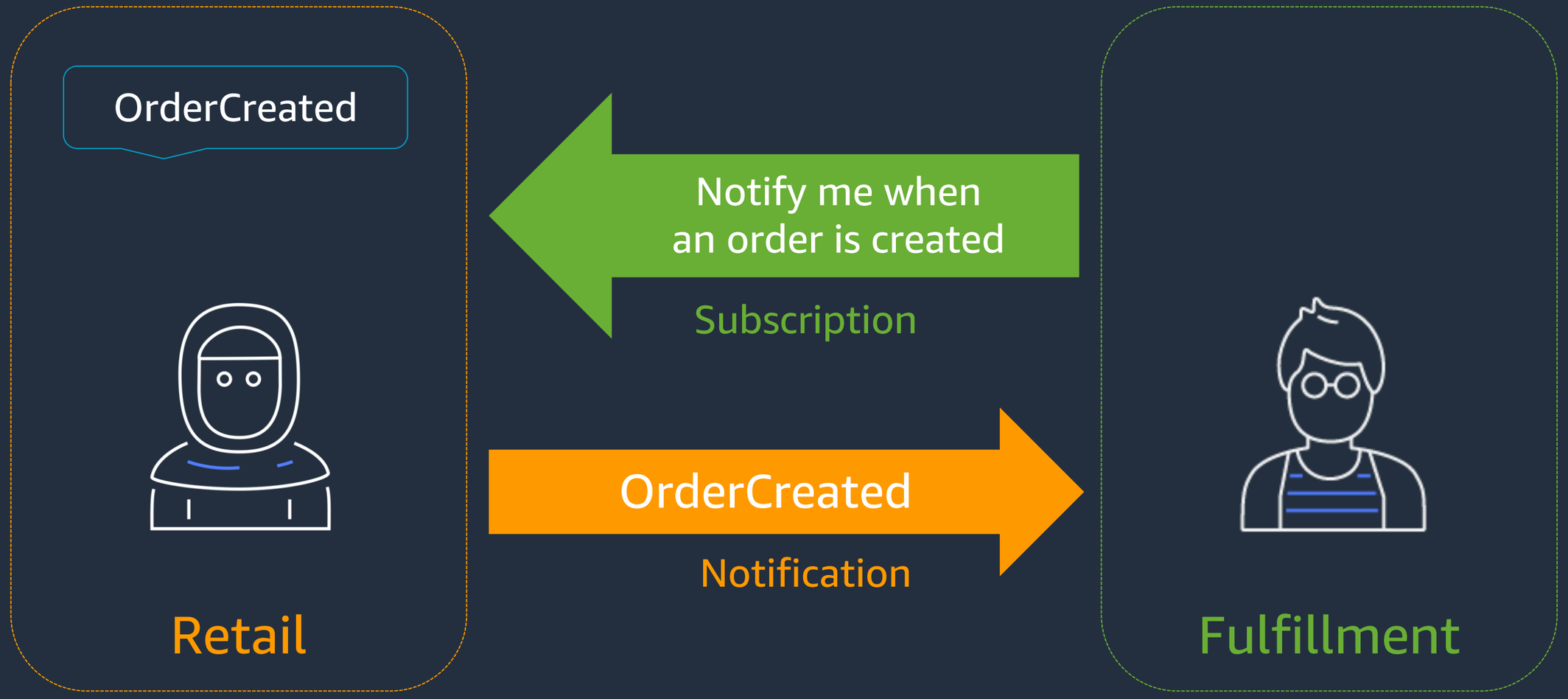
aws modern apps

Orchestrate a business process *within a domain*, resulting in a published event



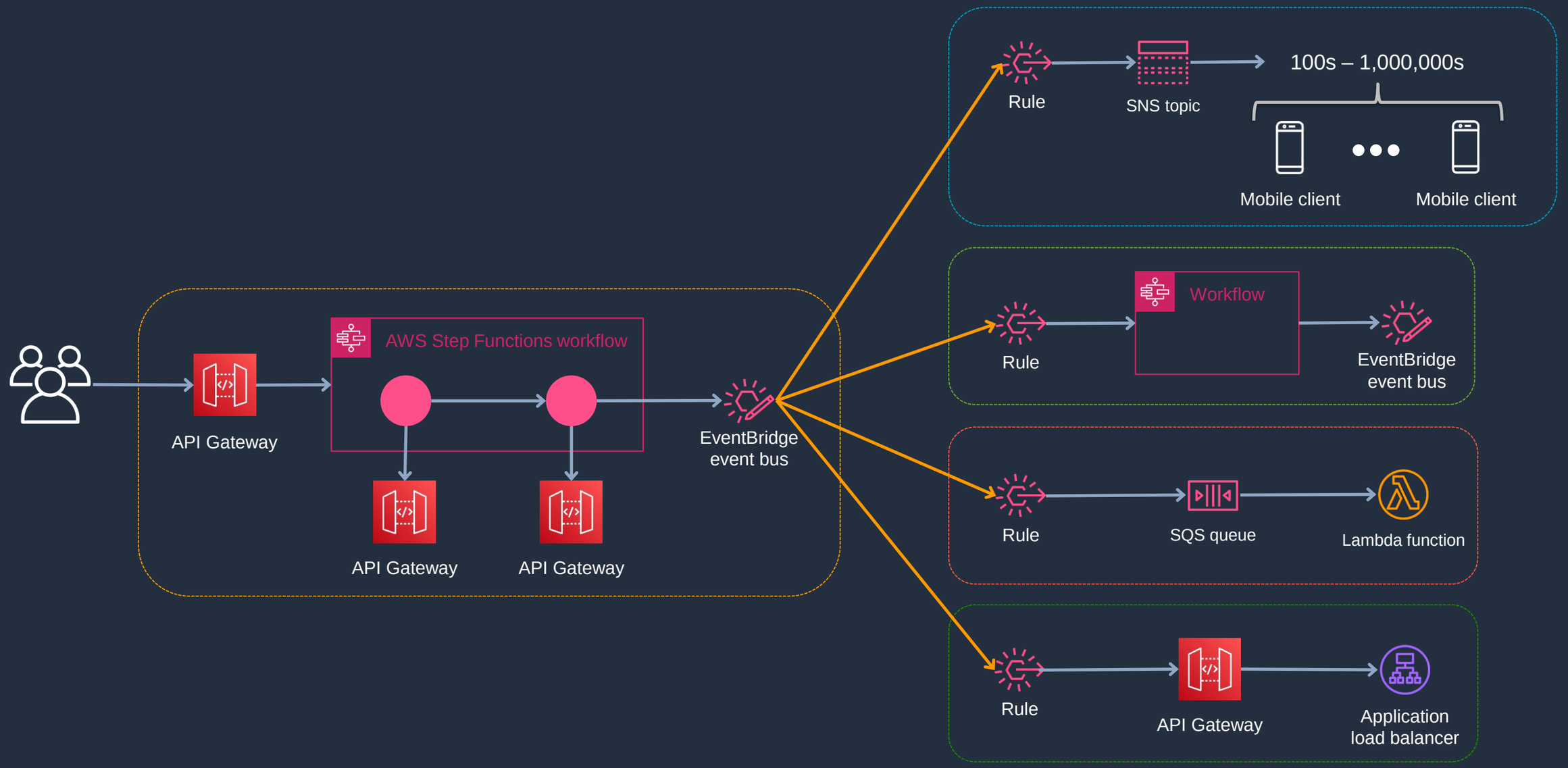
aws modern apps

Choreograph events *between domains* using subscriptions



aws modern apps

Better together: Orchestration + Choreography



aws modern apps

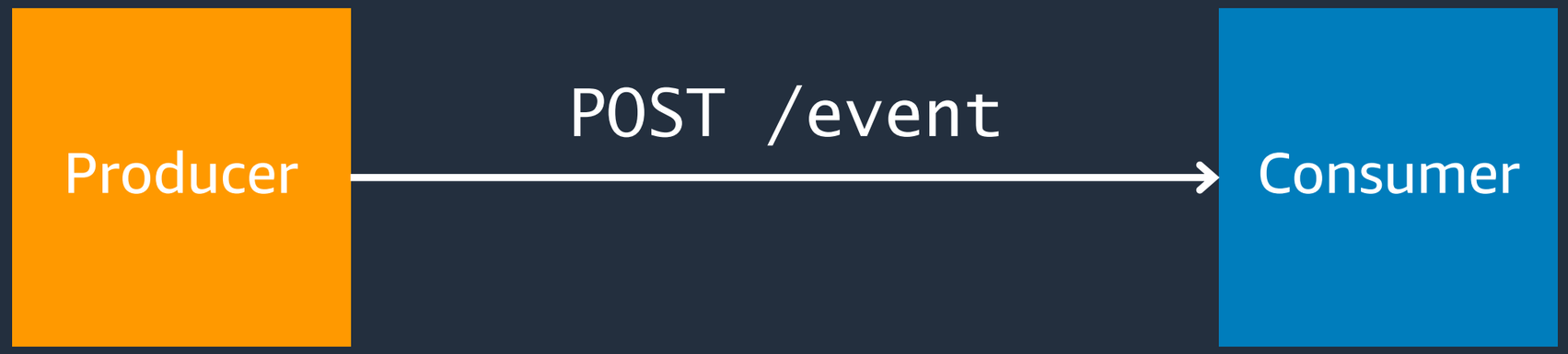
Journey to event-driven architectures

STEP 1
Start with the domain

STEP 2
Pick an event router

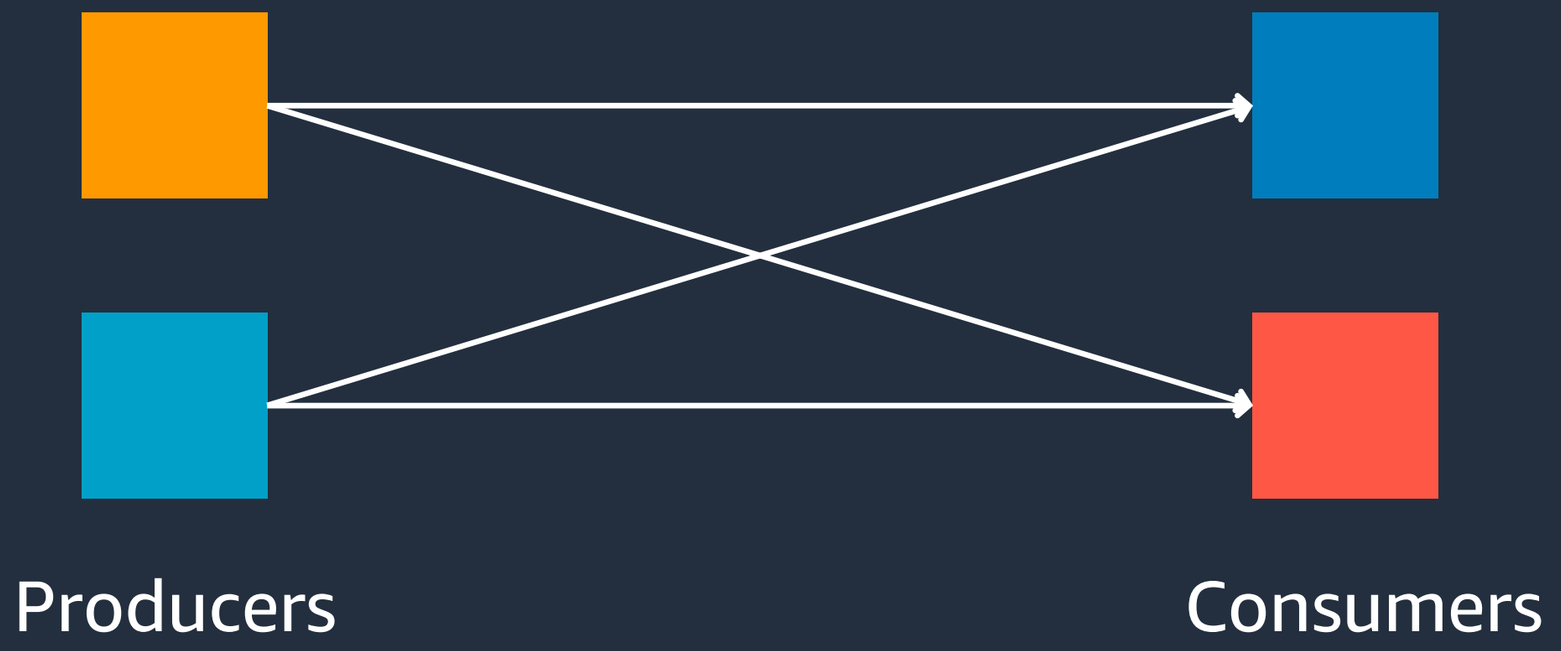
aws modern apps

Why do you need a router?



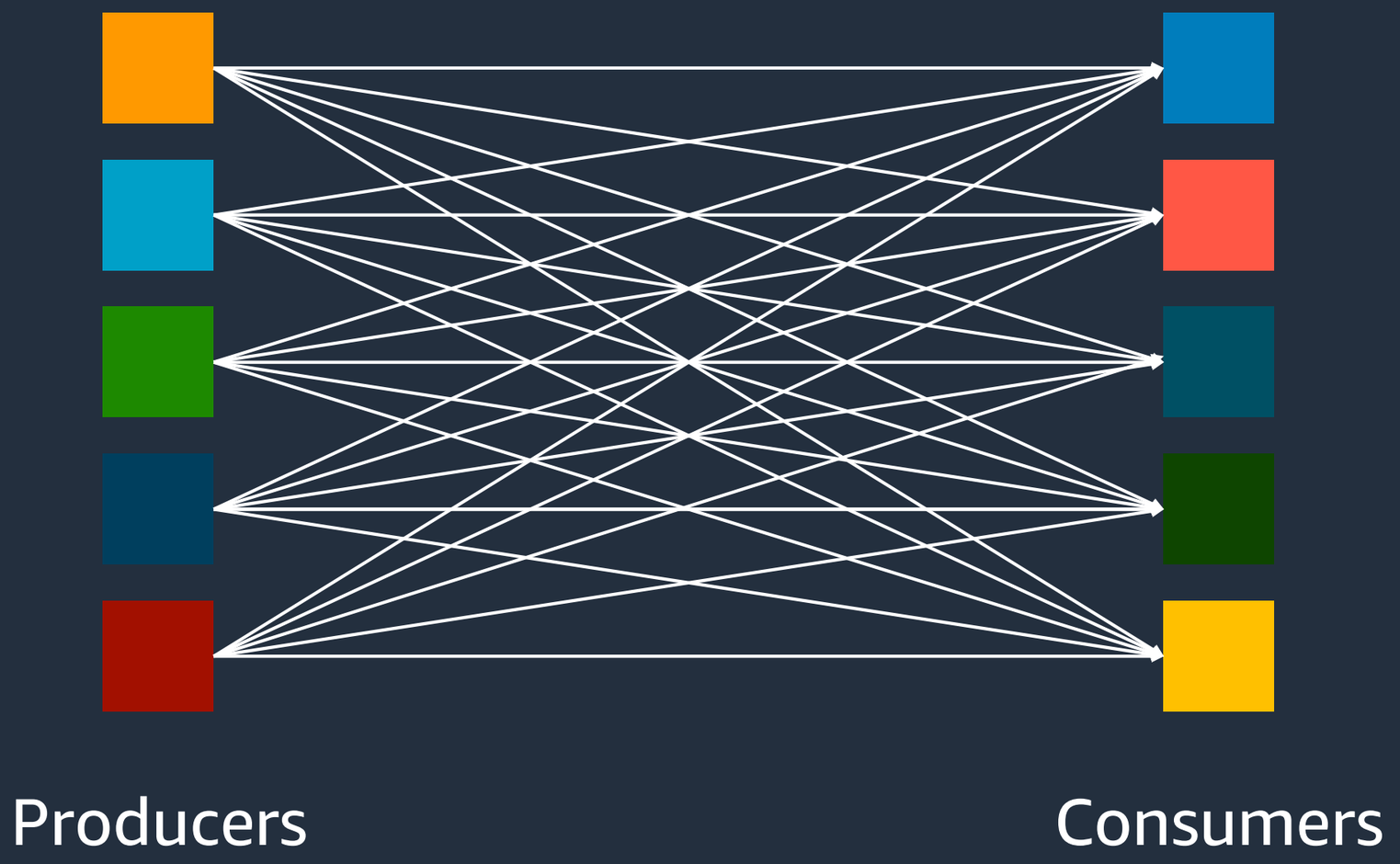
aws modern apps

Why do you need a router?



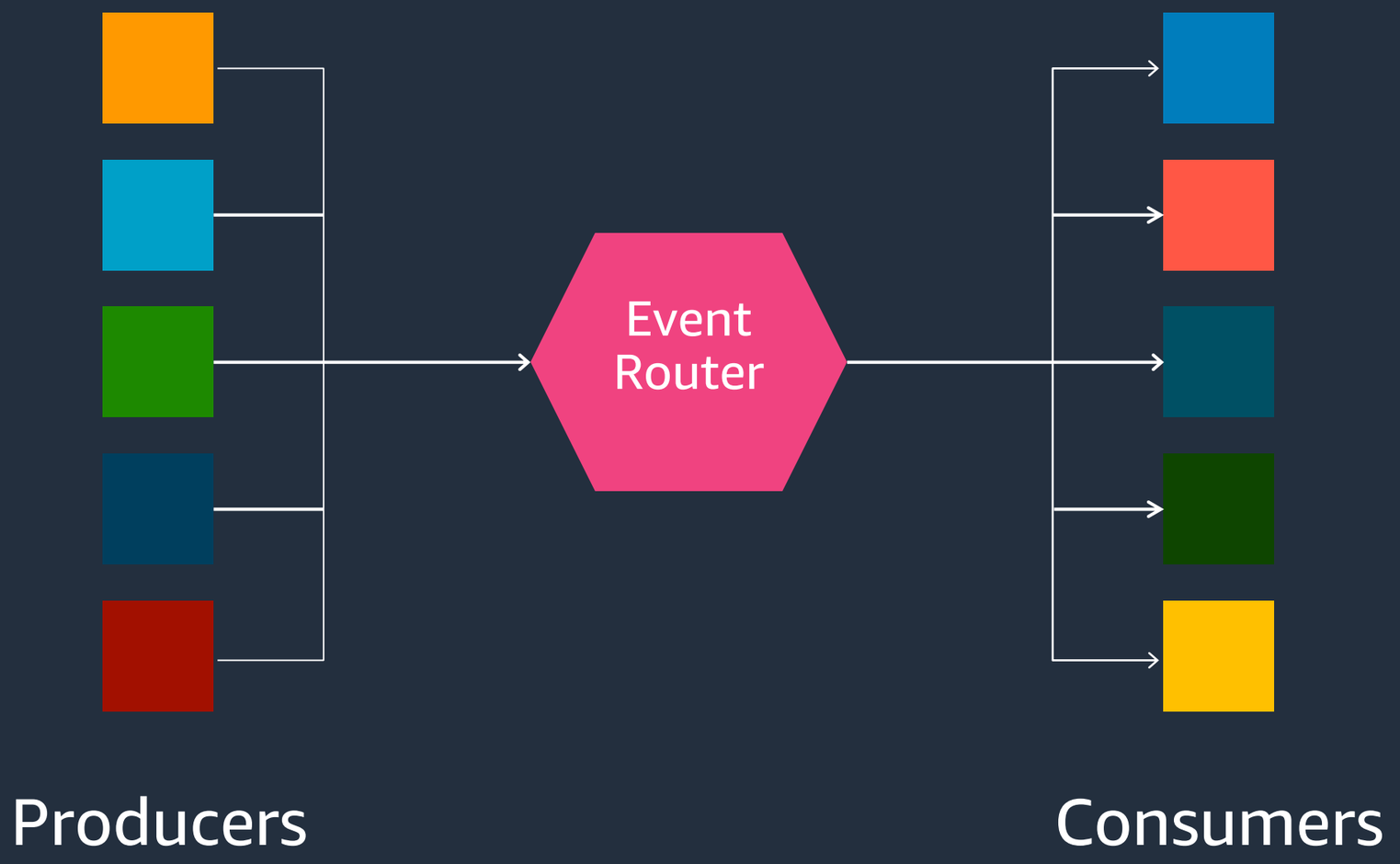
aws modern apps

Why do you need a router?



aws modern apps

Why do you need a router?



aws modern apps

AWS Event Routers

	Event Store		Event Router	
	Queues	Streams	Topics	Event Bus
AWS Native	Amazon SQS	Amazon Kinesis	Amazon SNS	Amazon EventBridge
Managed Open Source	Amazon MQ	Amazon MSK	Amazon MQ	

aws modern apps

AWS operational responsibility models

Less → More

On-premises	Managed	Native
RabbitMQ Active MQ	Amazon MQ	Amazon EventBridge Amazon SNS

Routers

Amazon MQ

WHAT IT IS

Managed message broker service for Apache ActiveMQ or RabbitMQ that makes it easy to set up and operate message brokers in the cloud.

USE CASE

Migrations and hybrid cloud architectures to support application replatforming to AWS.

COOL CAPABILITIES

Reduces operational load by managing the provisioning, setup, and maintenance of ActiveMQ, a popular open-source message broker.

aws modern apps

Amazon EventBridge

WHAT IT IS

Simple, flexible, fully managed, pay as you go, event bus service that makes it easy to ingest and process data from AWS services, your own applications, and SaaS applications.

USE CASE

Remove friction of having to write point-to-point integrations between services. Take action on SaaS messages, run workflows, apply intelligence, audit and analyze, and synchronize data.

COOL CAPABILITIES

28 Targets including Lambda, SQS, SNS, REST API, Kinesis, Step Functions, and API Destinations.

Schema Registry stores a collection of schemas a and allows developers to SEARCH/FIND/TRACK different schemas that are used for applications

Event Source

aws modern apps

Routing events using Amazon EventBridge rules

Example event:

```
{
  "detail-type": "Ticket Created",
  "source": "aws.partner/example.com/orders",
  "detail": {
    "ticketId": ”987654321",
    "department": "billing",
    "creator": "user12345"
    ...
  }
}
```

Example rule:

```
{
  "detail": {
   "department": ["billing", "fulfillment"]
  }
}
```


aws modern apps

Amazon Simple Notification Service (SNS)

WHAT IT IS

Simple, flexible, fully managed publish/subscribe messaging and mobile push notification service for high throughput, highly reliable message delivery

USE CASE

Notify multiple subscribed applications

Replicate data across regions

Invoke multiple steps in workloads

Parallel processing

Trigger serverless actions

COOL CAPABILITIES

Highly reliable delivery of any volume of messages to any number of recipients across multiple protocols

Event Source

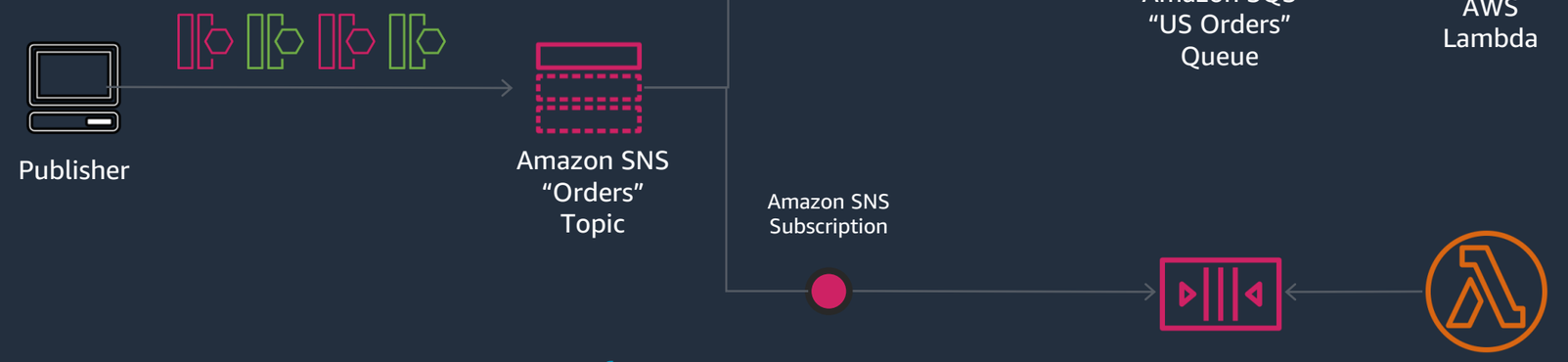
aws modern apps

Amazon SNS Message Filters

- Publishers do not need to route message
- Subscribers do not need to filter for message of interest
- Lowers cost

Message Attributes

```
{
  "location": "eu-west"
}
```



Filter Policy

```
{
  "location":
    ["us-west", "us-east"]
}
```

```
{
  "location":
    ["eu-west", "eu-east"]
}
```


aws modern apps

Amazon EventBridge or Amazon SNS

	Amazon EventBridge	Amazon SNS
Fanout	5 targets per rule	Millions of subscriptions per topic
Rules/Filters	2000 rules per bus	200 filters per account
Routing Logic	JSON payload	10 key-value attributes per message
Consistency Latency	Up to 5 minutes	Up to 15 minutes
Sources	65 AWS services + CloudTrail	31 AWS services
Targets	28 AWS services, including SNS	9 AWS services, including SMS
HTTP Target	API Destinations	HTTP/S Endpoint
Encryption	Server-side encryption via KMS	Server-side encryption via KMS

aws modern apps

Journey to event-driven architectures

STEP 1
Start with the domain

STEP 3
Pick an event store

STEP 2
Pick an event router

aws modern apps

Improve resiliency and scalability with event stores

Buffer messages until service are available to process

aws modern apps

Event stores handle messages and streams

Message Processing

- The individual message is the unit of work
- Computation/processing per message
- Message occurrence varies
- Message are deleted after consumption
- No need to track the position
- DLQ functionality built-in

Stream Processing

- The message stream is the unit of work
- Complex computation on many messages
- Constant stream of messages
- Messages are available after consumption until expiration
- Each client needs to track the current position in the stream
- No built-in DLQ functionality

aws modern apps

AWS Event Stores

	Event Store		Event Router	
	Queues	Streams	Topics	Event Bus
AWS Native	Amazon SQS	Amazon Kinesis	Amazon SNS	Amazon EventBridge
Managed Open Source	Amazon MQ	Amazon MSK	Amazon MQ	

aws modern apps

AWS operational responsibility models

Less → More

	On-premises	Managed	Native
Queues	RabbitMQ Active MQ	Amazon MQ	Amazon SQS
Streams	Kafka	Amazon MSK	Amazon Kinesis

Amazon Simple Queue Service (SQS)

WHAT IT IS

Simple, flexible, fully managed message queuing service for reliably and continuously exchanging any volume of messages from anywhere

(Standard and FIFO)

USE CASE

Build decoupled, highly scalable microservices, distributed systems, and serverless applications in the cloud

COOL CAPABILITIES

Nearly infinite scalability without pre-provisioning capacity

Event Source

25B messages per hour

amazon BLACK FRIDAY

amazon.com cyber monday

aws modern apps

Amazon Kinesis

WHAT IT IS

Enables you to ingest, buffer, and process streaming data in real-time, so you can derive insights in seconds or minutes instead of hours or days.

USE CASE

Ingest real-time data such as video, audio, application logs, website clickstreams, and IoT telemetry data for machine learning, analytics.

COOL CAPABILITIES

Can handle any amount of streaming data and process data from hundreds of thousands of sources with very low latencies.

Event Source

aws modern apps

Journey to event-driven architectures

STEP 1
Start with the domain

STEP 3
Pick an event store

STEP 2
Pick an event router

STEP 4
Structure events

aws modern apps

Events vs. full state descriptions

Order 123 was created at 10:47 a.m. by customer 456

Events

Order 123 was created at 10:47 a.m. by customer 456. The current status is Open, the total was $237.51, the items were …

Full state description

aws modern apps

Considerations with sparse events

What are the details for order 123?

Order 123 was created by customer 456

aws modern apps

Anatomy of an event

```
{
    "version": "0",
    "id": "adeacade-c34c-ce58-c4a0-74f106398c4e",
    "account": "123456789012",
    "region": "us-east-1",
    "time": "2019-12-02T21:46:19Z",
    "source": "order-service",
    "detail-type": "New Order",
    "resources": [],
    "detail": {
        "orderId": "cfb2ae566f9b",
        "customerId": "C12345",
        ...
    }
}
```

Envelope metadata

Payload

aws modern apps

Anatomy of an event

```
{
    "version": "0",
    "id": "adeacade-c34c-ce58-c4a0-74f106398c4e",
    "account": "123456789012",
    "region": "us-east-1",
    "time": "2019-12-02T21:46:19Z",
    "source": "order-service",
    "detail-type": "New Order",
    "resources": [],
    "detail": {
        "orderId": "cfb2ae566f9b",
        "customerId": "C12345",
        ...
    }
}
```

Managed attributes

aws modern apps

Anatomy of an event

```
{
    "version": "0",
    "id": "adeacade-c34c-ce58-c4a0-74f106398c4e",
    "account": "123456789012",
    "region": "us-east-1",
    "time": "2019-12-02T21:46:19Z",
    "source": "order-service",
    "detail-type": "New Order",
    "resources": [],
    "detail": {
        "orderId": "cfb2ae566f9b",
        "customerId": "C12345",
        ...
    }
}
```

Service that created the event

aws modern apps

Anatomy of an event

```
{
    "version": "0",
    "id": "adeacade-c34c-ce58-c4a0-74f106398c4e",
    "account": "123456789012",
    "region": "us-east-1",
    "time": "2019-12-02T21:46:19Z",
    "source": "order-service",
    "detail-type": "New Order",
    "resources": [],
    "detail": {
        "orderId": "cfb2ae566f9b",
        "customerId": "C12345",
        ...
    }
}
```

Event type

aws modern apps

Anatomy of an event

```
{
    "version": "0",
    "id": "adeacade-c34c-ce58-c4a0-74f106398c4e",
    "account": "123456789012",
    "region": "us-east-1",
    "time": "2019-12-02T21:46:19Z",
    "source": "order-service",
    "detail-type": "New Order",
    "resources": [],
    "detail": {
        "orderId": "cfb2ae566f9b",
        "customerId": "C12345",
        ...
    }
}
```

Any valid JSON object

aws modern apps

Design Considerations

Delivery semantics

At-least once delivery

Events can be delivered to a target more than once. Include logic to detect duplicate events by tracking the state of processed events (idempotent).

- Amazon EventBridge
- Amazon SNS Standard
- Amazon SQS Standard
- Amazon Kinesis

Exactly-once delivery*

Specify an identifier used by the AWS service for deduplication.

- Amazon MQ
- Amazon MSK
- Amazon SNS FIFO
- Amazon SQS FIFO

** outside of application error handling*

aws modern apps

Ordering semantics*

Unordered

Events can be delivered out of order.

- Amazon EventBridge
- Amazon SNS Standard
- Amazon SQS Standard

Ordered

Events are delivered in order within a partition, message group, etc (no global order).

- Amazon MQ
- Amazon MSK
- Amazon Kinesis
- Amazon SNS FIFO
- Amazon SQS FIFO

** Out-of-order event handling logic is highly application specific. If the application cannot be designed to handle out-of-order events, consider orchestration instead.*

aws modern apps

Next Steps

Schedule a follow up meeting with your Serverless team

Event-driven architecture best practices
https://aws.amazon.com/event-driven-architecture/

Digital Course: Architecting Serverless Solutions
https://www.aws.training/Details/eLearning?id=42594

Event-driven reference architecture
https://github.com/aws-samples/aws-serverless-ecommerce-platform

Building event-driven architectures on AWS workshop
http://event-driven-architecture.workshop.aws

aws modern apps