

CS2032 - Cloud Computing (Ciclo 2024-2)

Event-driven architecture

Semana 13 - Clase 1: Arquitectura basada en eventos

ELABORADO POR: GERALDO COLCHADO

Contenido

Event-driven architecture

1. Objetivo de la sesión
2. Event-driven architecture
3. Cuando utilizarla?
4. Cómo implementarla en AWS?
5. Patrones de diseño
6. Cierre

Objetivo de la sesión

Event-driven architecture

- Comprender qué es una arquitectura basada en eventos
- Identificar cuando es conveniente usar arquitectura basada en eventos
- Aprender el patrón de diseño pub/sub - fan out
- Conocer qué servicios de AWS se pueden usar para arquitectura basada en eventos

Contenido

Event-driven architecture

1. Objetivo de la sesión
2. **Event-driven architecture**
3. Cuando utilizarla?
4. Cómo implementarla en AWS?
5. Patrones de diseño
6. Cierre

Event-driven architecture

¿Qué es la arquitectura basada en eventos?

*“Sistemas **desacoplados** que se ejecutan en respuesta a **eventos**”*

*“La arquitectura basada en eventos utiliza **eventos** para **desencadenar** y establecer **comunicación entre servicios desacoplados**, y es común en las aplicaciones modernas creadas con **microservicios**. **Un evento es un cambio de estado**, o una actualización, como un elemento que se coloca en un carro de compras de un sitio web de comercio electrónico. Los eventos pueden llevar el estado (el elemento comprado, su precio y una dirección de entrega) o pueden ser identificadores (una notificación de que se envió una orden)”*

Event-driven architecture

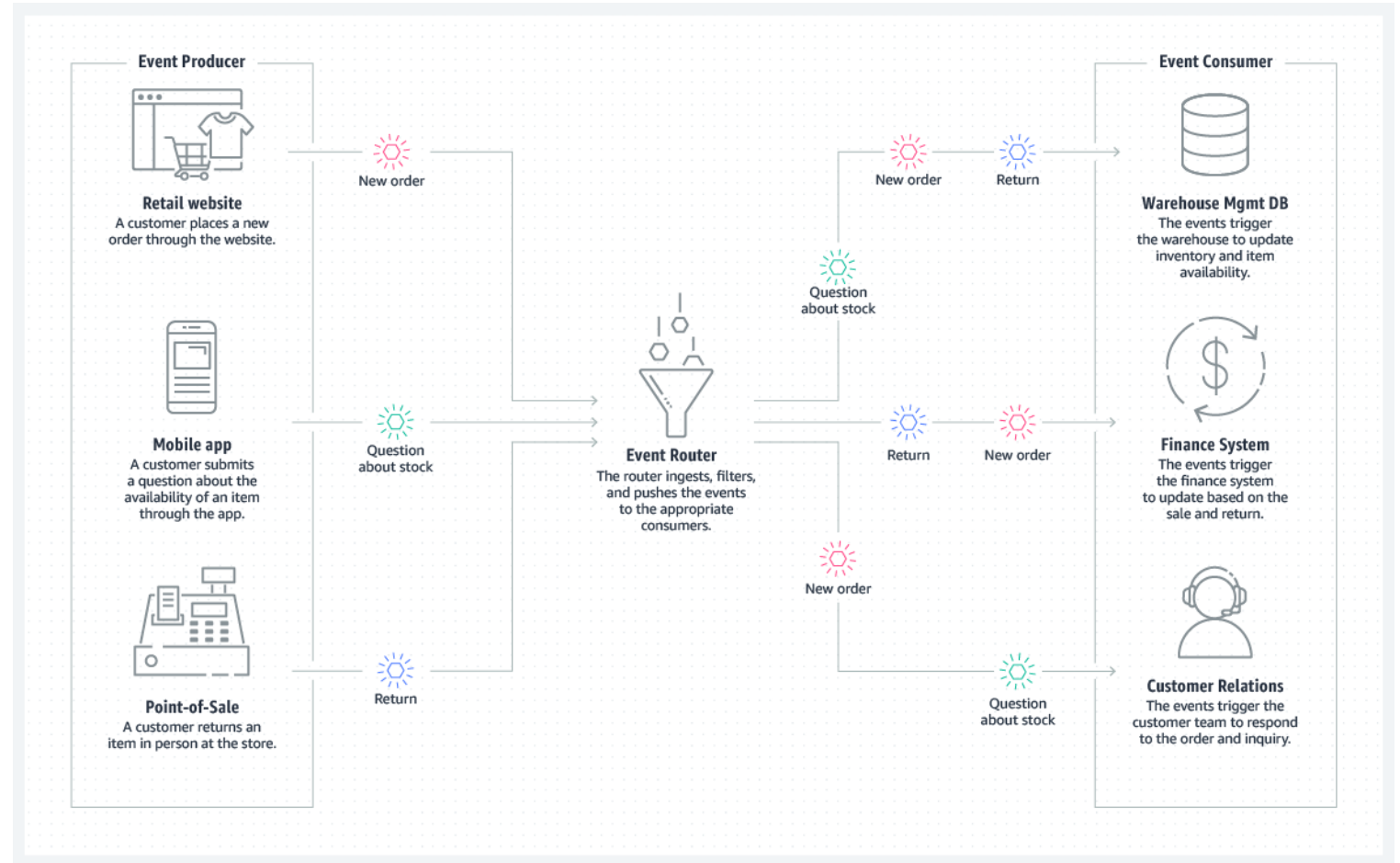
¿Qué es la arquitectura basada en eventos?

*“Las arquitecturas impulsadas por eventos tienen **tres componentes clave**: **productores** de eventos, **enrutadores** de eventos y **consumidores** de eventos. **Un productor publica un evento para el enrutador, que filtra y envía los eventos a los consumidores.** Los **servicios del productor** y los **servicios del consumidor** se **desacoplan**, lo que les permite escalarse, actualizarse e implementarse de manera independiente.”*

Event-driven architecture

¿Qué es la arquitectura basada en eventos?

“Este es un ejemplo de arquitectura basada en eventos para un sitio de comercio electrónico. Esta arquitectura permite que el sitio reaccione a los cambios procedentes de diversas fuentes durante los momentos de mayor demanda, sin que se bloquee la aplicación ni se sobreaprovisionen recursos.”



Contenido

Event-driven architecture

1. Objetivo de la sesión
2. Event-driven architecture
3. **Cuando utilizarla?**
4. Cómo implementarla en AWS?
5. Patrones de diseño
6. Cierre

Event-driven architecture

¿Cuándo utilizarla?

- *Supervisión del estado de los recursos y alertas*
- *Procesamiento en paralelo y distribución ramificada*
- *Integración de sistemas heterogéneos*
- *Replicación de datos entre cuentas y regiones*

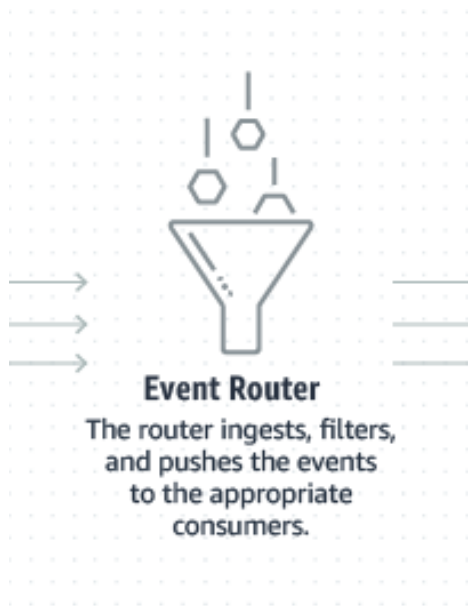
Contenido

Event-driven architecture

1. Objetivo de la sesión
2. Event-driven architecture
3. Cuando utilizarla?
4. **Cómo implementarla en AWS?**
5. Patrones de diseño
6. Cierre

Event-driven architecture

¿Cómo implementarla en AWS?



¿Por dónde comenzar?

Existen dos tipos principales de enrutadores que se utilizan en las arquitecturas basadas en eventos: los buses de eventos y los temas de eventos. En AWS, ofrecemos [Amazon EventBridge](#) para crear buses de eventos y [Amazon Simple Notification Service \(SNS\)](#) para crear temas de eventos.

launched in July 2019



Amazon EventBridge se recomienda cuando desee crear una aplicación que reaccione a los eventos de las aplicaciones SaaS, los servicios de AWS o las aplicaciones personalizadas. EventBridge usa un esquema predefinido para los eventos y le permite crear reglas que se aplican a todo el cuerpo del evento para filtrarlo antes de enviarlo a los consumidores.



Amazon SNS se recomienda cuando quiera crear una aplicación que reaccione a eventos de alto rendimiento y baja latencia publicados por otras aplicaciones, microservicios o servicios de AWS, o para aplicaciones que necesiten una gran difusión (miles o millones de puntos de conexión). Los temas de SNS son independientes del esquema de eventos que se publiquen.

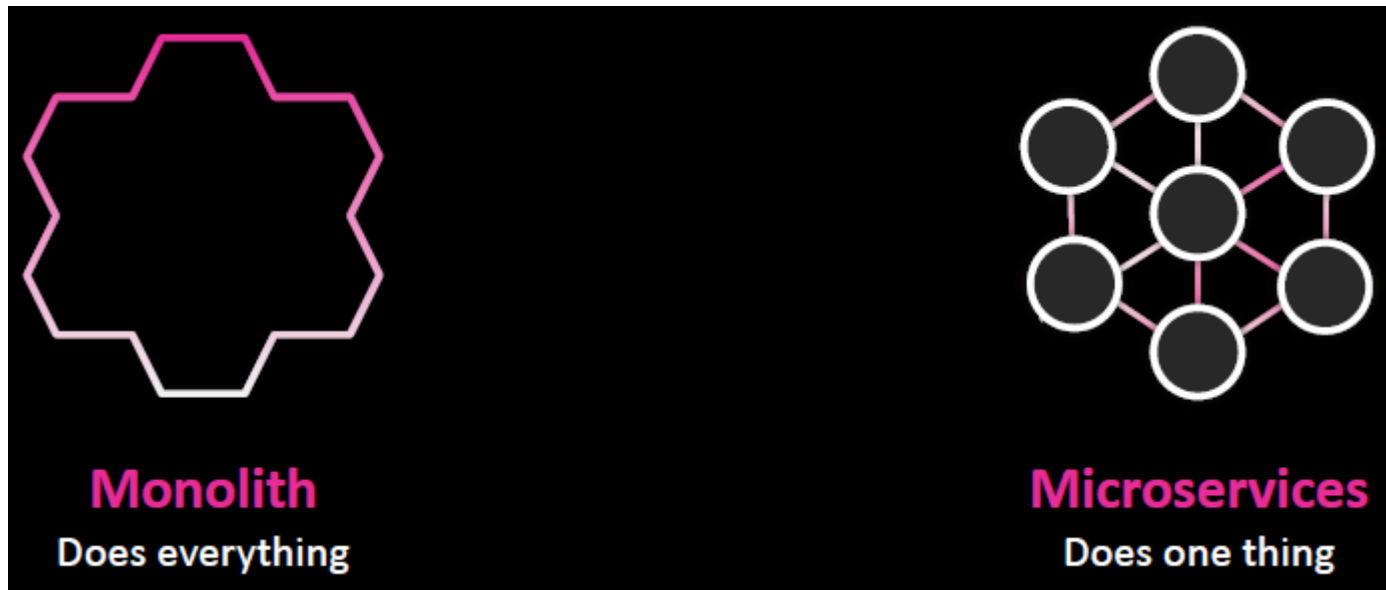
Contenido

Event-driven architecture

1. Objetivo de la sesión
2. Event-driven architecture
3. Cuando utilizarla?
4. Cómo implementarla en AWS?
5. **Patrones de diseño**
6. Cierre

Patrones de diseño

Monoliths and microservices



Patrones de diseño

microservices

- There are two patterns of application communication

1) Synchronous communications
(application to application)



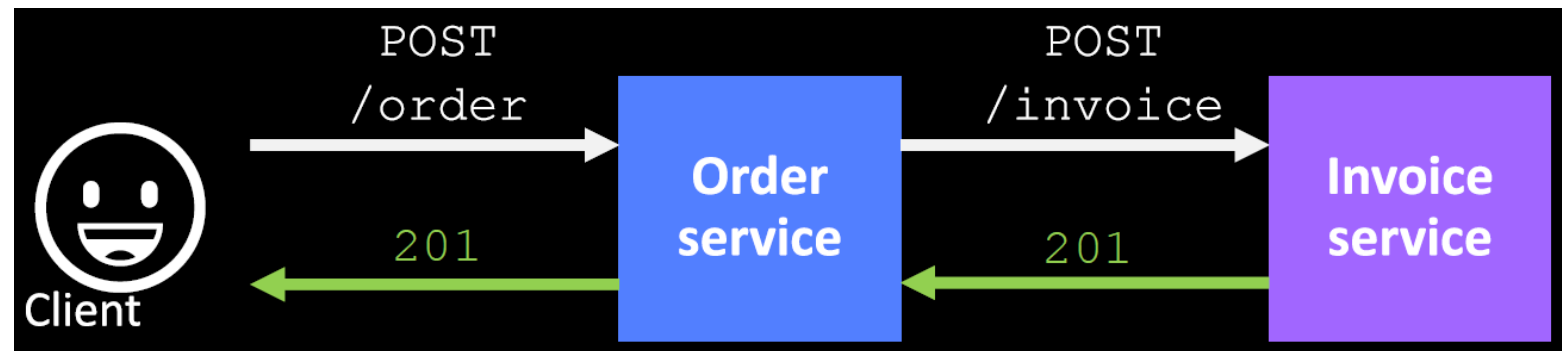
2) Asynchronous / Event based
(application to queue to application)



Patrones de diseño

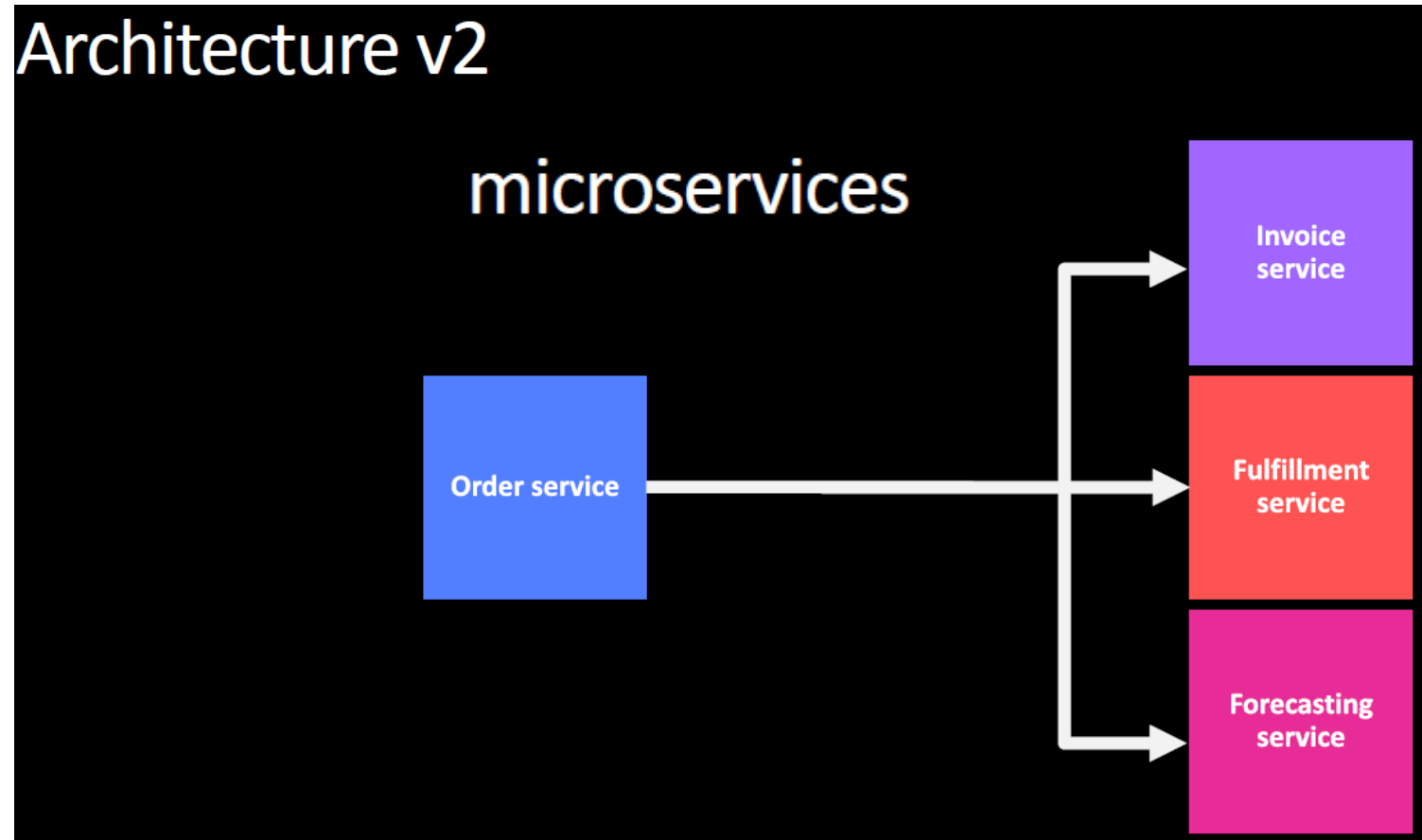
microservices

- Synchronous
- Coupled



Patrones de diseño

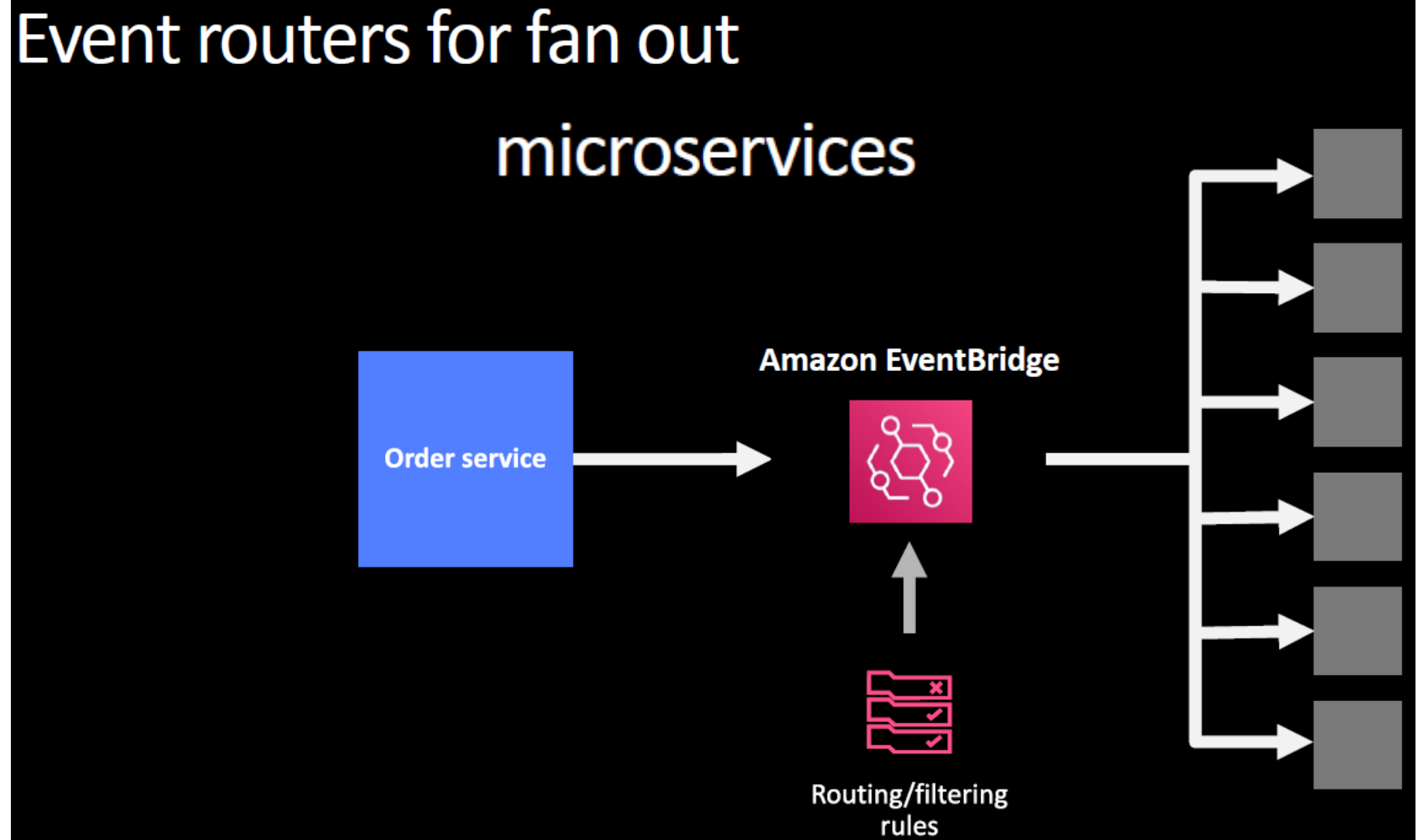
- Synchronous
- Coupled



Event-driven architecture

Patrones de diseño (Fan Out - Abanico)

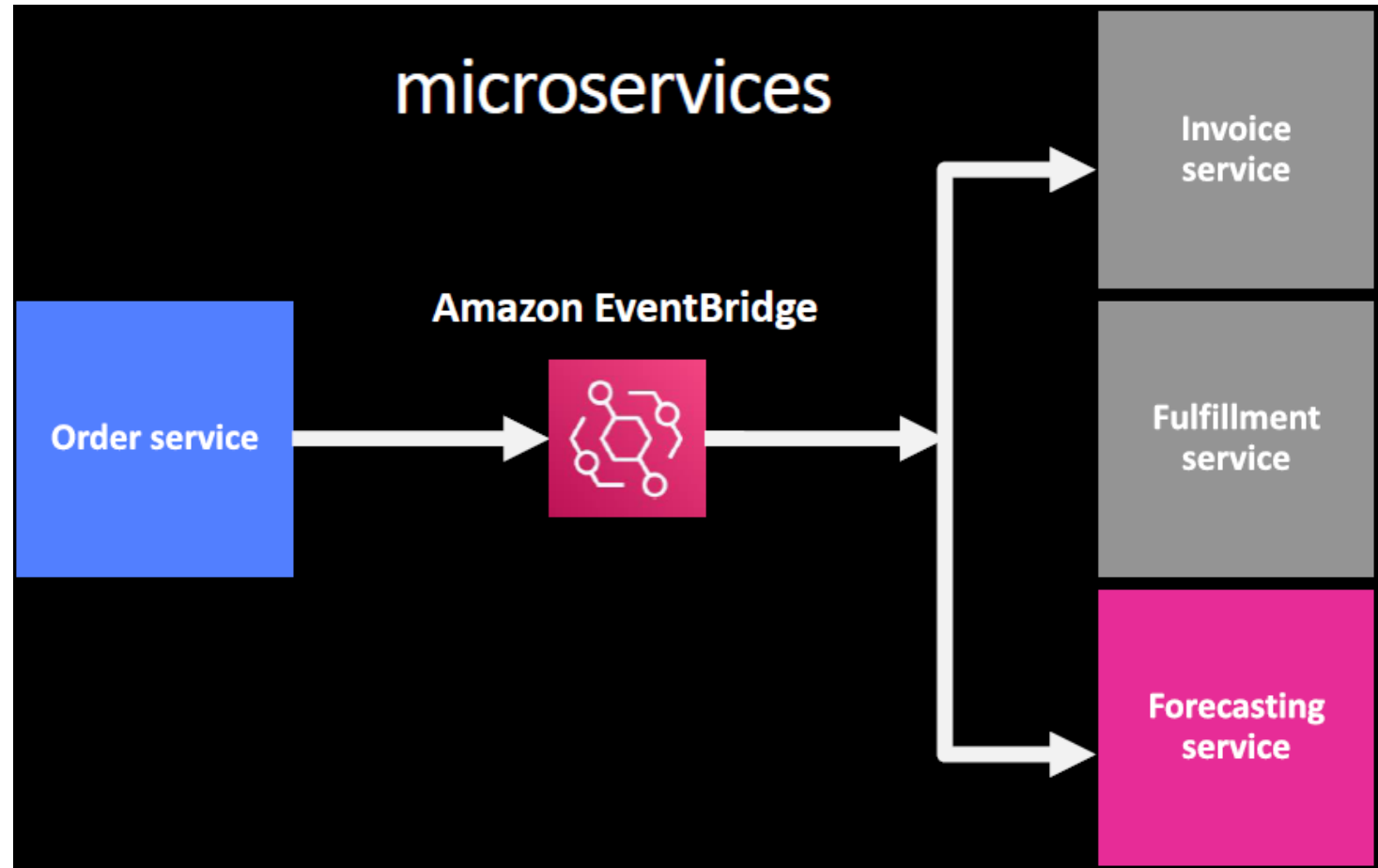
- Asynchronous
- Decoupled



Event-driven architecture

Patrones de diseño (Fan Out - Abanico)

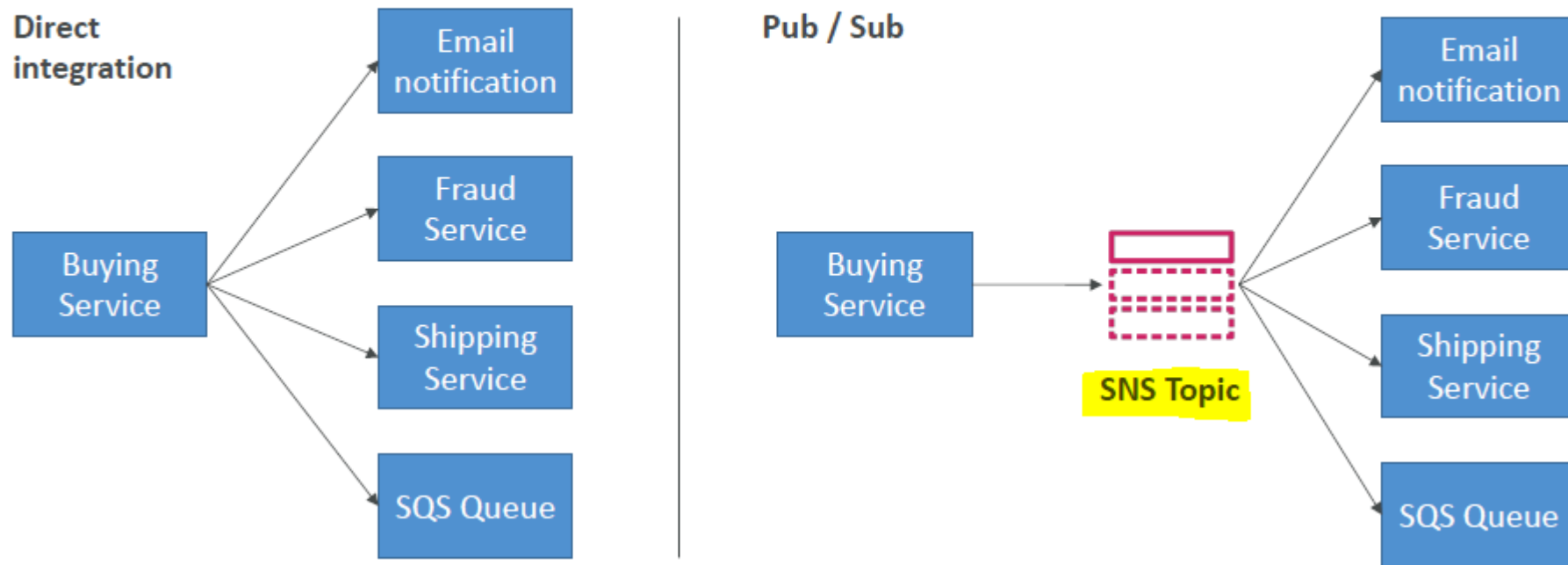
- Asynchronous
- Decoupled



Event-driven architecture

Patrones de diseño (Pub/Sub - Fan Out - Abanico)

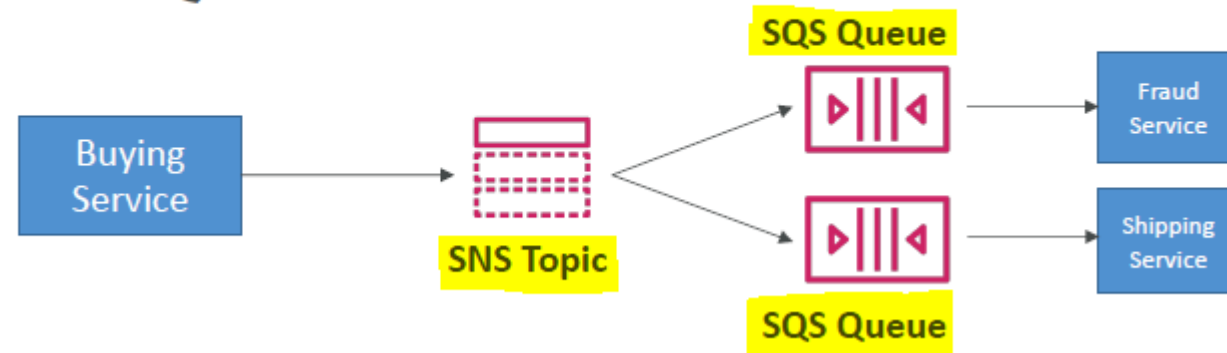
- What if you want to send one message to many receivers?



Event-driven architecture

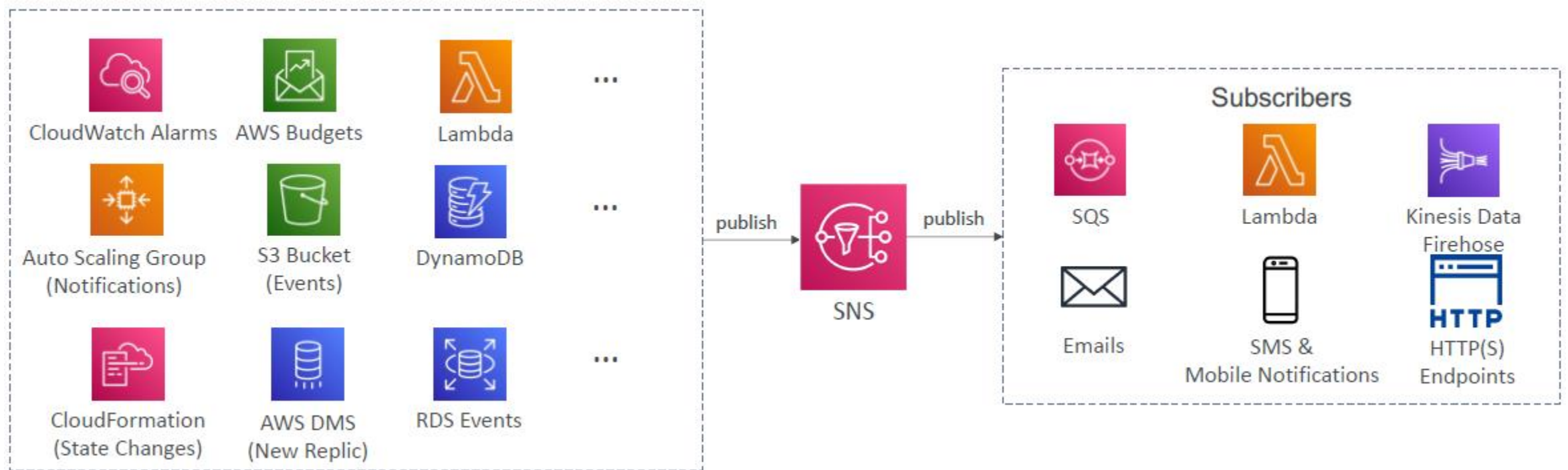
Patrones de diseño (Pub/Sub - Fan Out - Abanico)

SNS + SQS: Fan Out



Event-driven architecture

Patrones de diseño (Pub/Sub - Fan Out - Abanico)



Contenido

Event-driven architecture

1. Objetivo de la sesión
2. Event-driven architecture
3. Cuando utilizarla?
4. Cómo implementarla en AWS?
5. Patrones de diseño
6. **Cierre**

Cierre

Explique con sus propias palabras

- Qué es una arquitectura basada en eventos?
- Cuando es conveniente usar arquitectura basada en eventos?
- Explique el patrón de diseño pub/sub - fan out
- Qué servicios de AWS se pueden usar para arquitectura basada en eventos?

Gracias

Docente: Geraldo Colchado