



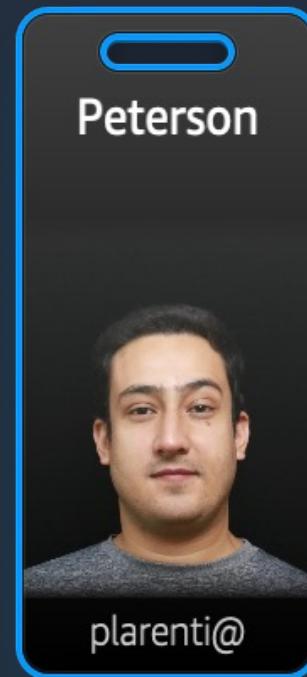
Habilitando negocios con arquitectura basada en eventos

Javier Cristancho
Sr. SA Manager for Colombia

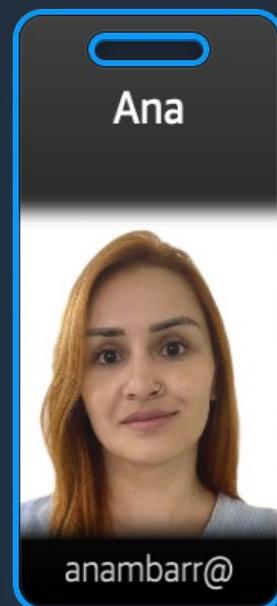
El Team Serverless para LATAM



Jaime Nagase
Sr. GTM Specialist



Peterson Larentis
Sr. Specialist SA



Serverless TFC
20+ y creciendo



Ayude a los equipos de campo y a los clientes estratégicos de AWS a impulsar y acelerar la migración y la modernización con arquitecturas sin servidor

Agenda

9:30 am	Breakfast
10:00 am	Kickoff - Welcome
10:45 am	Session 1: Building event-driven architectures on AWS
11:45 am	Session 2: Serverlesspresso presentation and demo
12:45 pm	Lunch
2:00 pm	Workshop: Serverlesspresso workshop
5:00 pm	Wrap Up

Integrando Sistemas

Al inicio es fácil utilizar arquitecturas acopladas ...



Conectar dos sistemas. ¿Qué tan difícil puede ser?

Flujo de Control o de Datos?

Pub-sub or point-to-point?

Modelo de Interacción?

Sync/async?

Polling?



Formatos de información?

Sistemas o instancias

Manejo de errores?

Idempotencia?

Fallas parciales?

Re-Intentos?

Back-off?

...pero la complejidad se multiplica al escalar



¿Cómo la **integración** de los componentes define las propiedades esenciales de tu sistema?

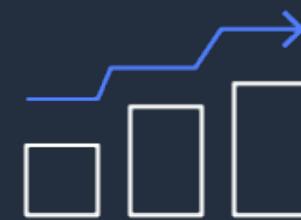
- ¿Cada componente puede **escalar y fallar independientemente**?
- ¿Los sistemas ascendentes soportan **respuestas lentas** de los sistemas descendentes ?
- ¿Los equipos interdependientes necesitan **coordinarse estrechamente** para lanzar nuevas funciones?
- ¿Cómo se manejan los errores y **reintentos**?
- ¿Cómo integran eventos desde **aplicaciones SaaS**?

¿Por qué construir arquitecturas basadas en eventos?

¿Por qué construir arquitecturas basada en eventos?



Tolerancia de
fallos



Escalabilidad

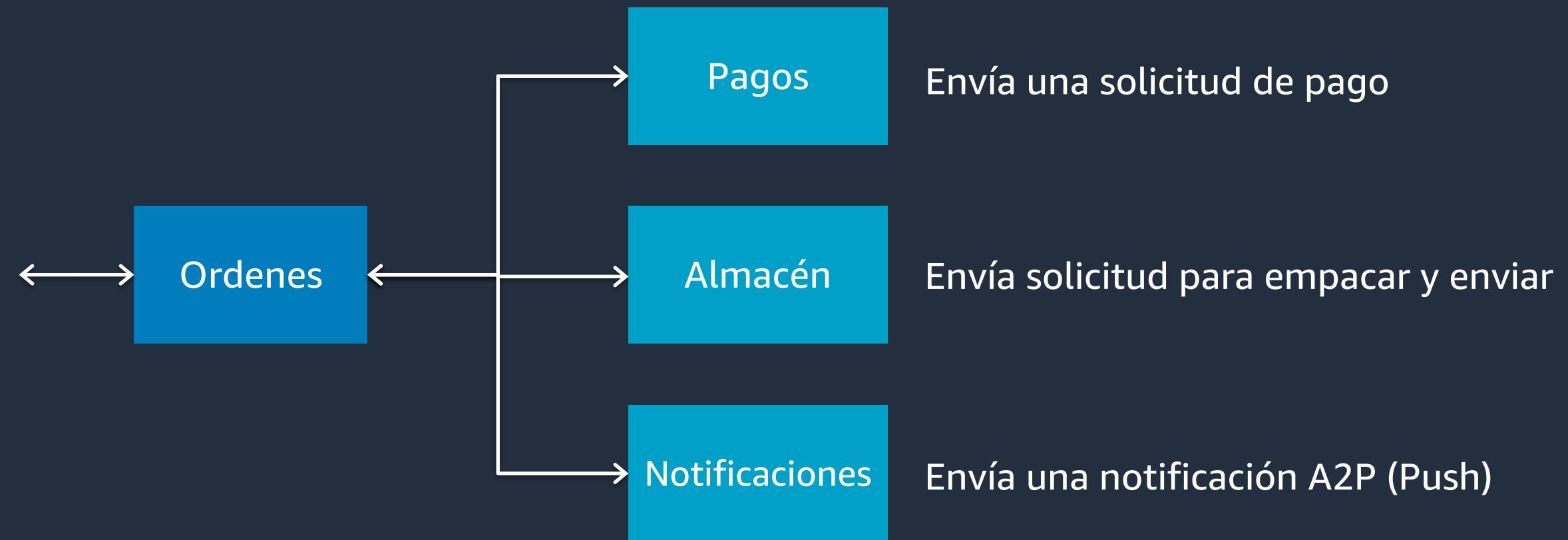


Extensible

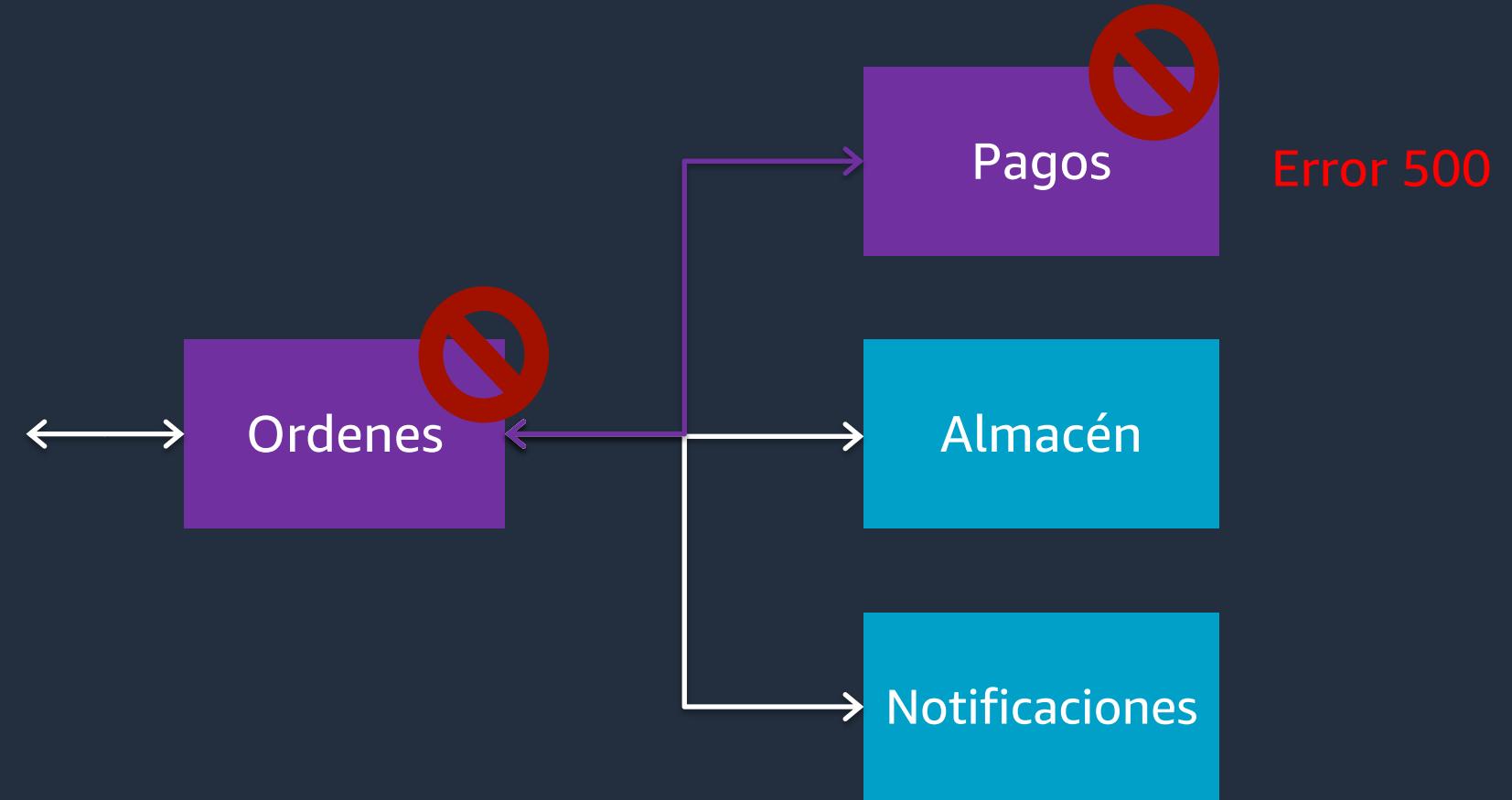


Agilidad

Ejemplo de arquitectura acoplada



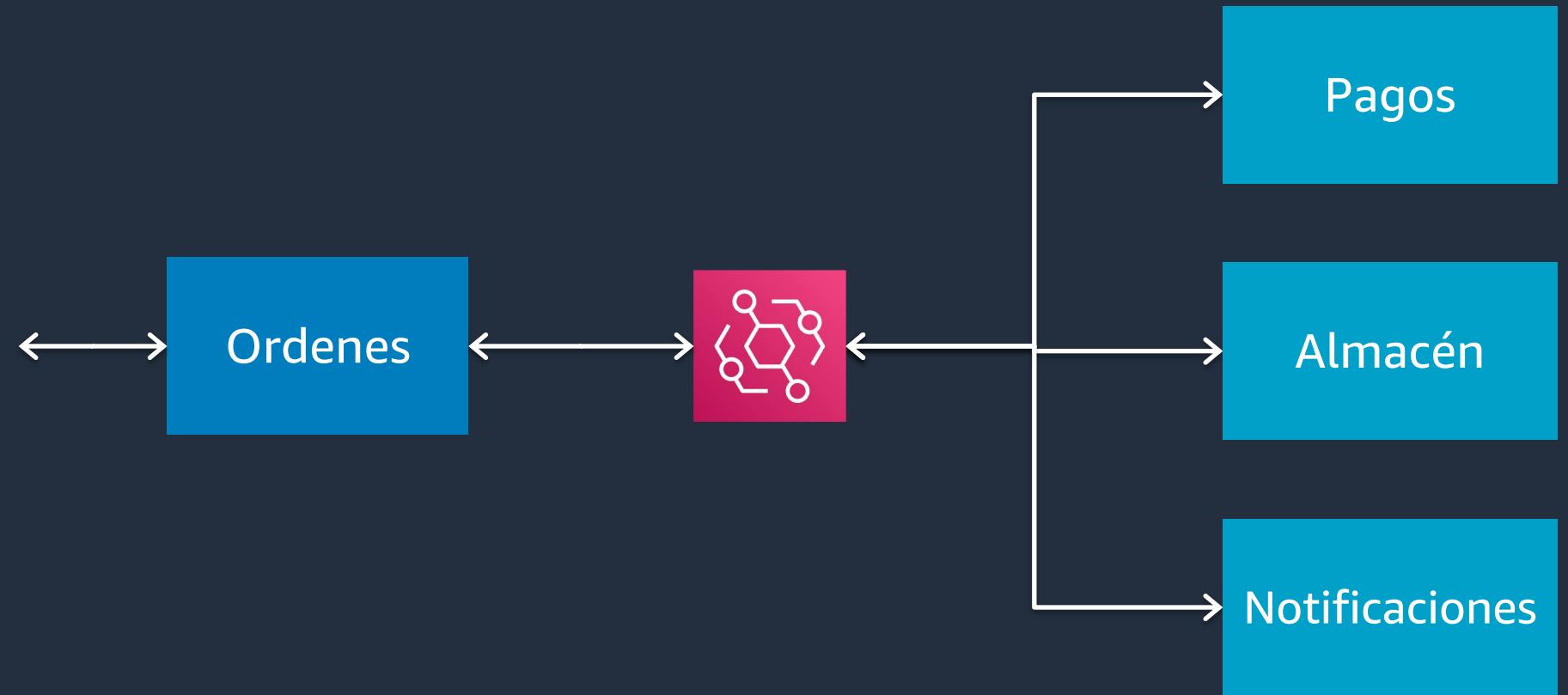
Ejemplo de arquitectura acoplada



Ejemplo de arquitectura desacoplada



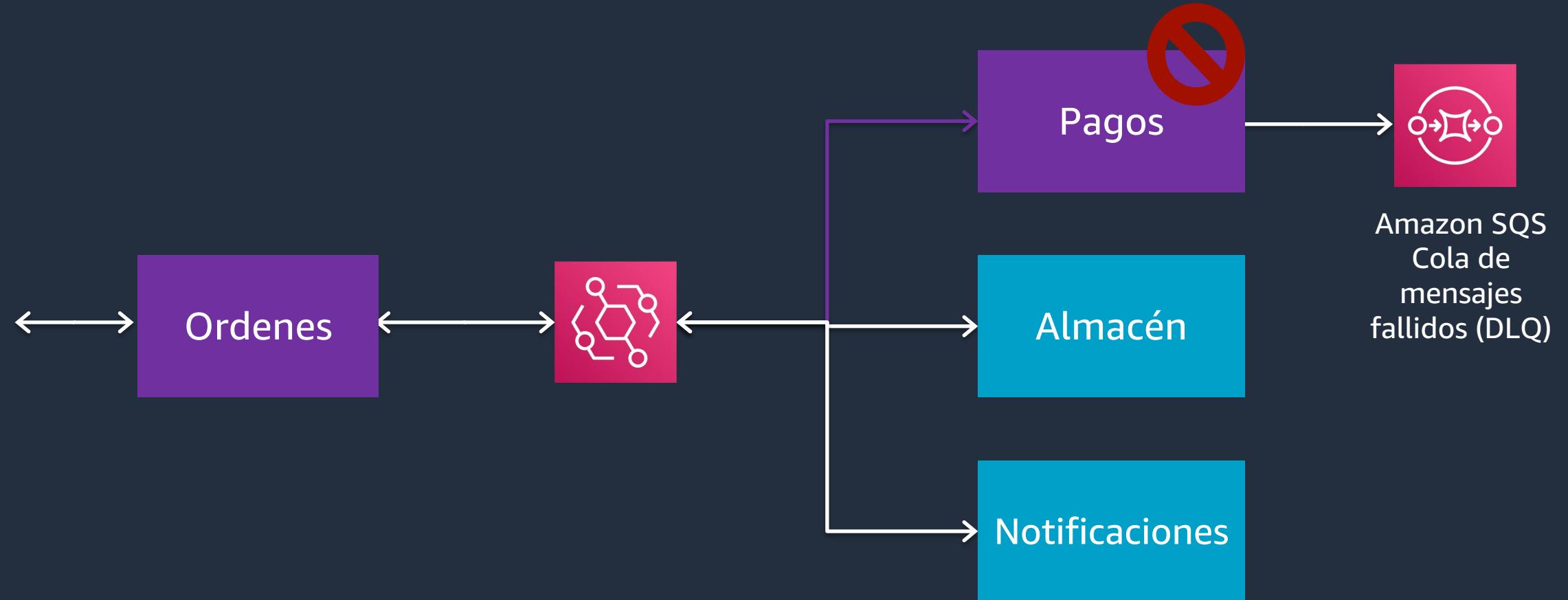
Tolerancia de fallos



El servicio de ordenes ya no tiene que esperar por la respuesta de los otros servicios.

- Alta disponibilidad del servicio de ordenes
- Baja latencia para los usuarios finales.
- No mas fallas en cascada.

Ejemplo de arquitectura desacoplada



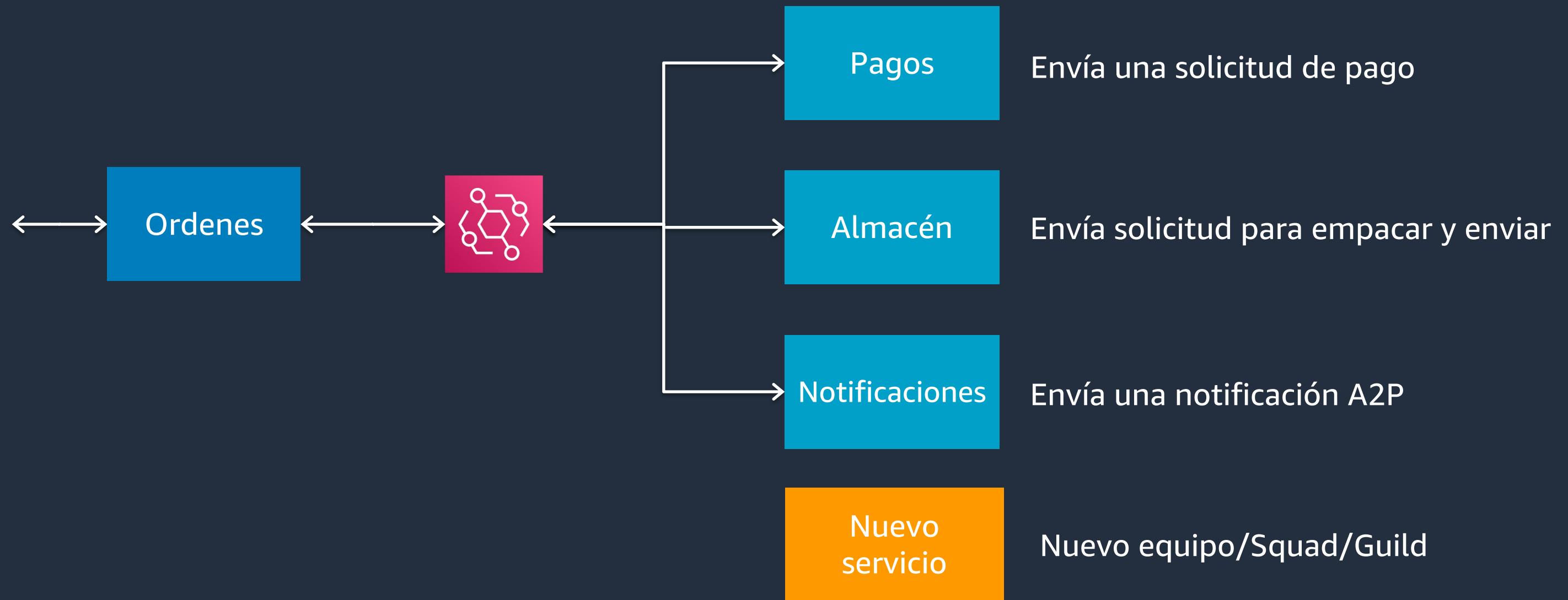
Ejemplo de arquitectura desacoplada (nuevo servicio)



Agilidad



Extensible



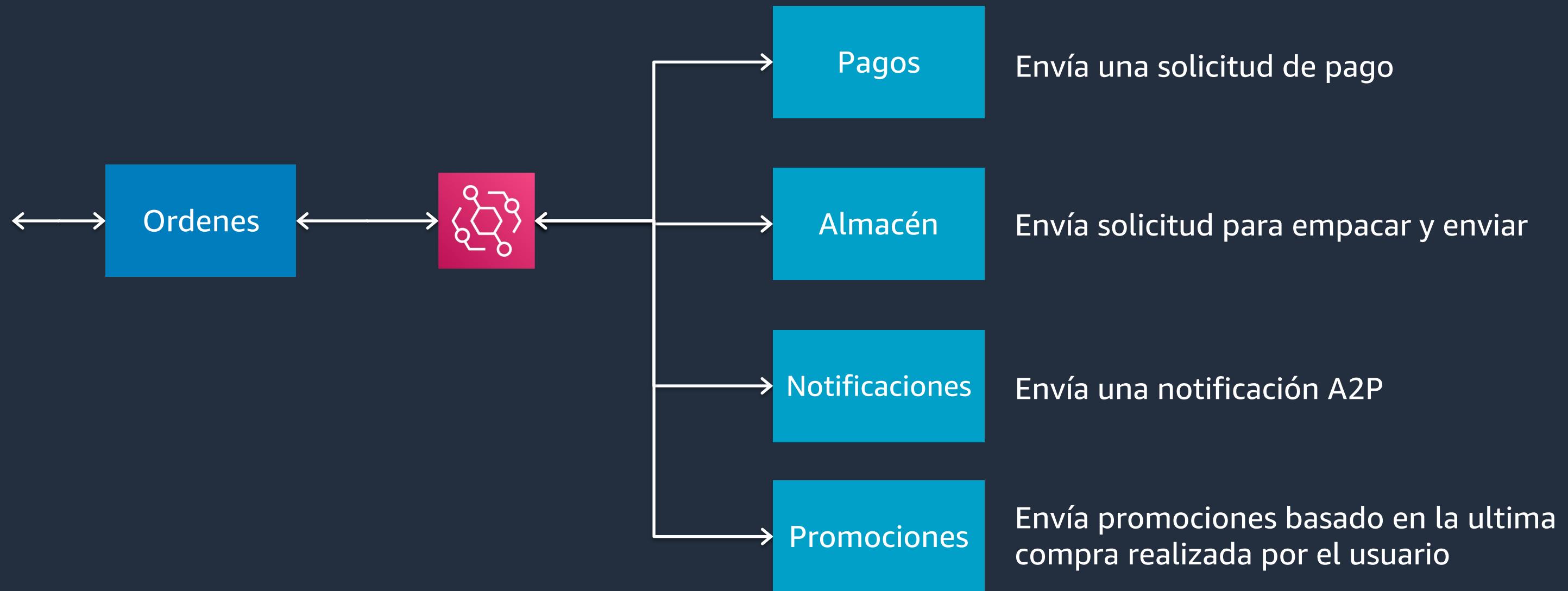
Ejemplo de arquitectura desacoplada (nuevo servicio)



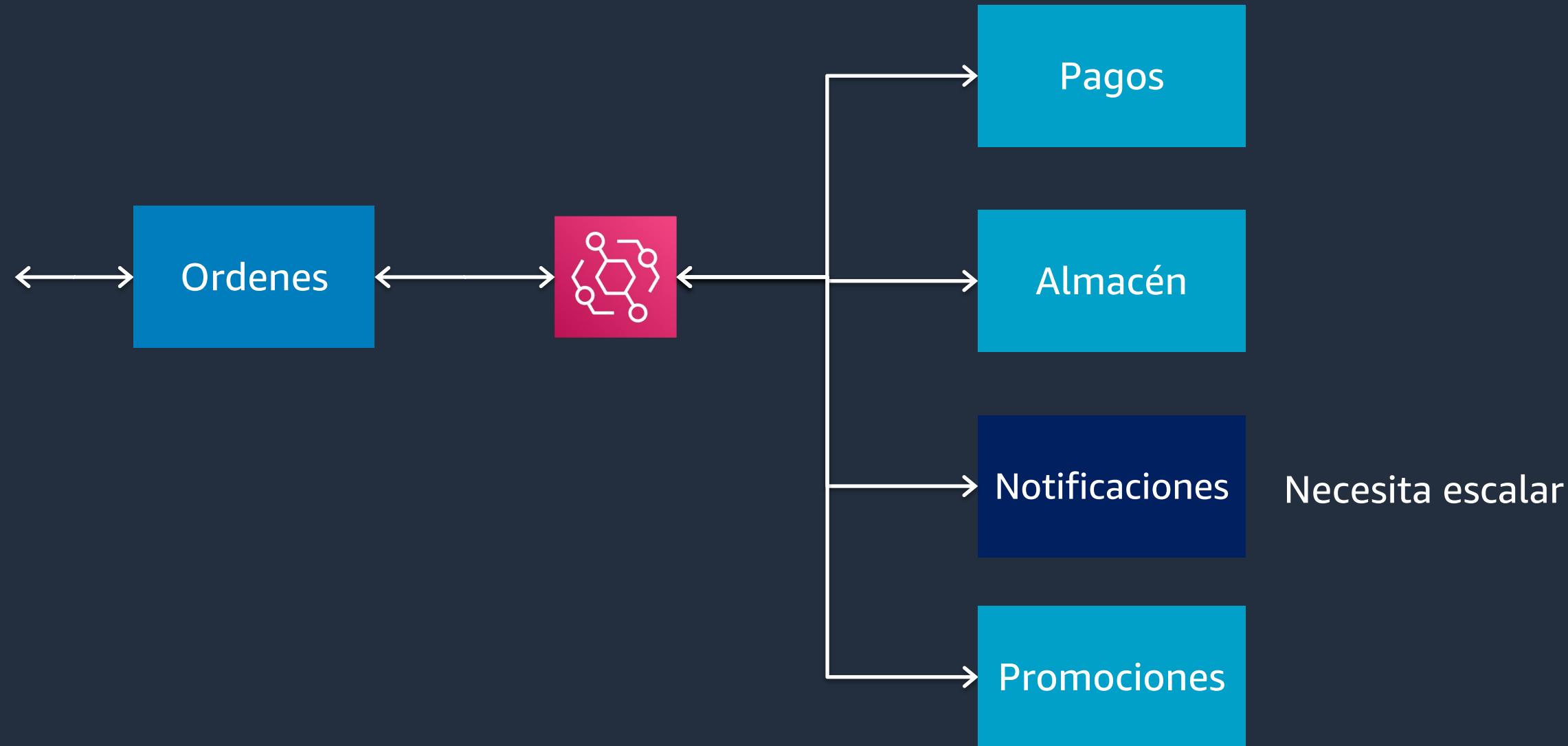
Agilidad



Extensible

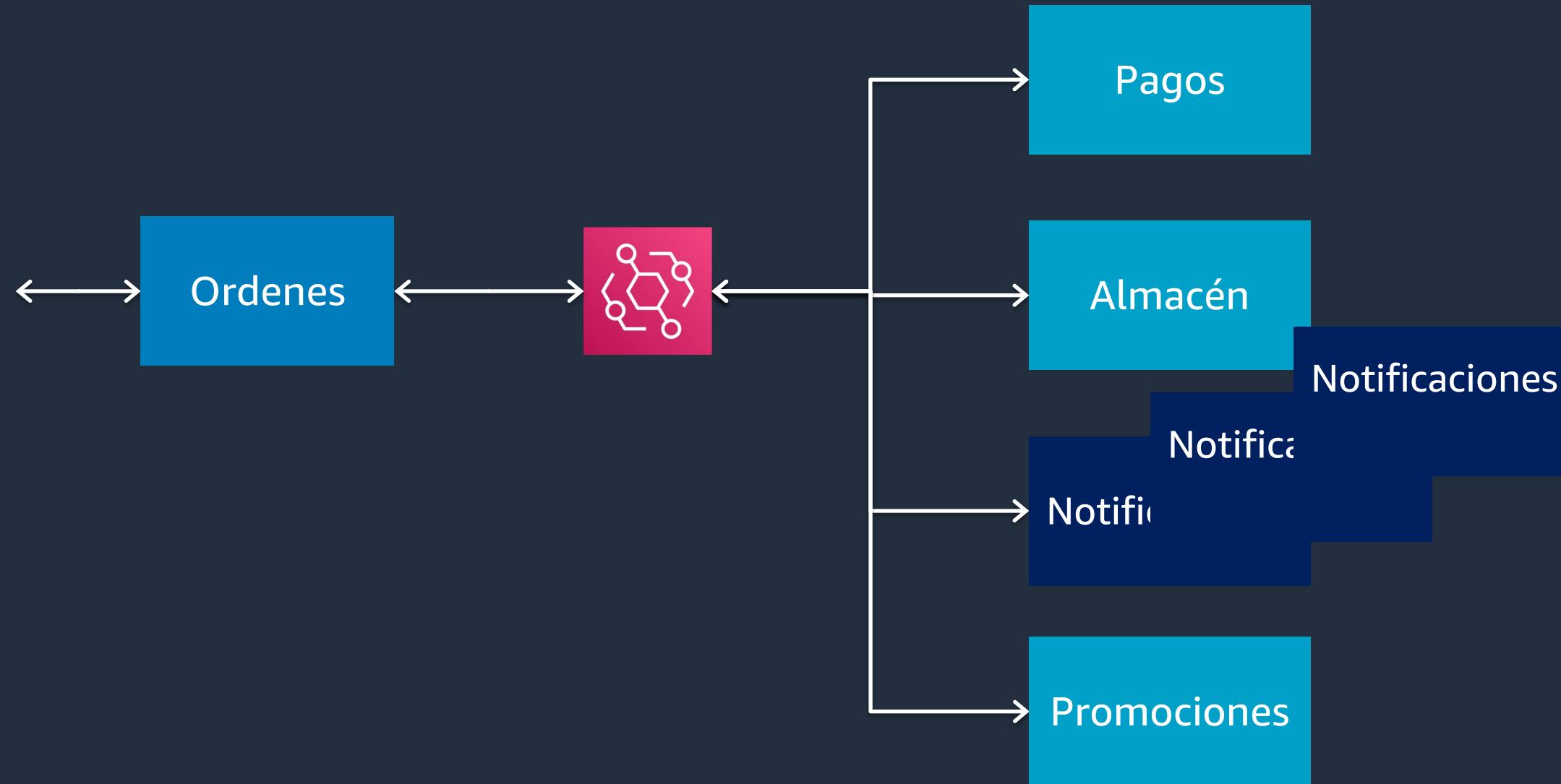


Ejemplo de arquitectura desacoplada



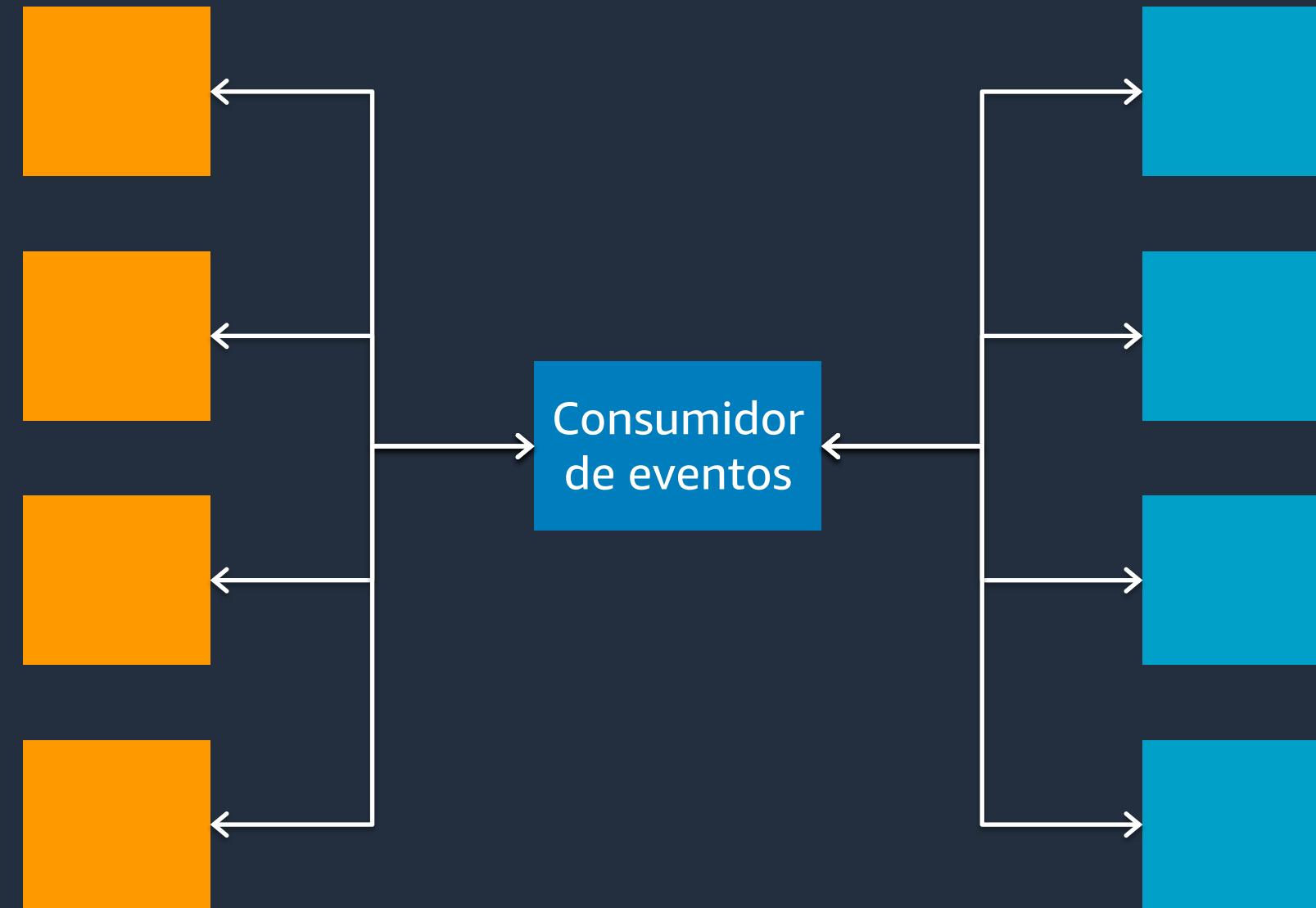
Necesita escalar

Ejemplo de arquitectura desacoplada



Solo esta escalando el servicio de notificaciones, no todo el sistema.

Las arquitecturas desacopladas permiten implementar aplicaciones resilientes y un desarrollo rápido de funciones



Un **evento** es un cambio en un estado o una actualización



Ventas

Se creo una orden nueva



Medios de comunicación
y entretenimiento



Servicios financieros



Sector Salud

Actualización de los datos médicos del paciente



Publicaciones

Documento subido para su traducción



Educación

Nuevo estudiante inscrito



Manufactura

Problema detectado en el sensor de IoT



Gobierno

Nueva solicitud enviada

¿Qué es una arquitectura Orientada a Eventos?



En una arquitectura orientada a **eventos**, los eventos se **observan**, no se **dirigen**.

Los eventos se observan, no se dirigen



Comandos directos

Los eventos se observan, no se dirigen



Comandos directos

Cliente X
acaba de
ordenar



Eventos observados

Manejo de eventos



Enrutador de eventos

Filtra y enruta los eventos para enviarlos a los consumidores que están suscritos.



Transmisión de eventos

Transmisión continua de eventos donde los consumidores pueden extraer eventos

¿Qué servicios de AWS se pueden utilizar en una arquitectura desacoplada?

La arquitectura basada en eventos se divide en tres partes:



Productor de eventos

Publica eventos
(web o aplicaciones
móviles,
microservicios,
dispositivos IoT)



Ingestión de eventos



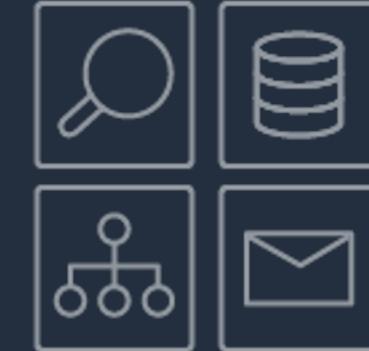
Consumidor de eventos

Procesa los eventos
(Actualiza la base de
datos, envía una
notificación, ejecuta
análisis de ML)

La arquitectura basada en eventos se divide en tres partes:



Ingestión de eventos



Consumidor de eventos

Procesa los eventos
(Actualiza la base de datos, envía una notificación, ejecuta análisis de ML)

Productores de eventos



Amazon Simple
Storage Service (S3)



Amazon API Gateway



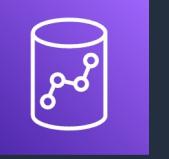
Amazon DynamoDB



AWS Lambda



AWS AppFlow



Amazon Redshift



Amazon Translate



AWS Step Functions

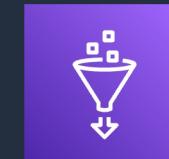
Cualquiera de los mas de 200+ Servicios de AWS produce eventos,
O tus eventos personalizados.



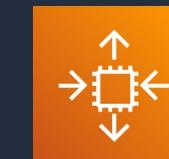
AWS Batch



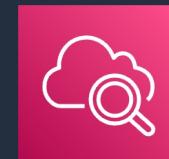
Amazon Elastic
Container Service



AWS Glue



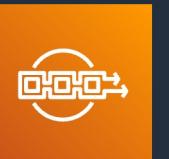
Amazon EC2
Auto Scaling



Amazon CloudWatch



AWS IoT Greengrass



Amazon Managed
Blockchain



Amazon Athena

Recibir eventos SaaS a través de Amazon Eventbridge



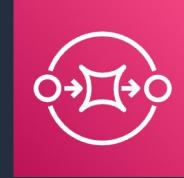
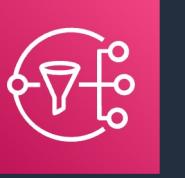
Más de 42 integraciones SaaS en Amazon EventBridge



La arquitectura basada en eventos se divide en tres partes:



Manejo de eventos

	Almacenador de eventos		Enrutador de eventos	
	Colas	Transmisión	Temas	Bus de eventos
Nativos de AWS	 Amazon SQS	 Amazon Kinesis	 Amazon SNS	 Amazon EventBridge
Manejado por código abierto	 Amazon MQ	 Amazon MSK		 Amazon MQ

Comparando servicios para manejo de eventos: Operaciones



Amazon EventBridge



Amazon SNS



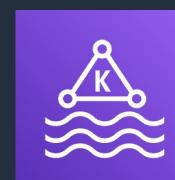
Amazon MQ



Amazon SQS



Amazon Kinesis
Data Streams



Amazon MSK



Comparando servicios para el manejo de eventos: Ordenamiento

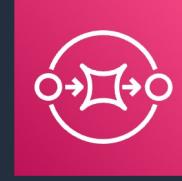
Garantiza orden de eventos



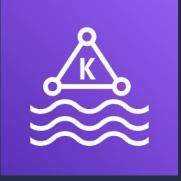
Amazon MQ



Amazon Kinesis
Data Streams



Amazon SQS
FIFO



Amazon MSK

No garantiza orden de eventos



Amazon EventBridge

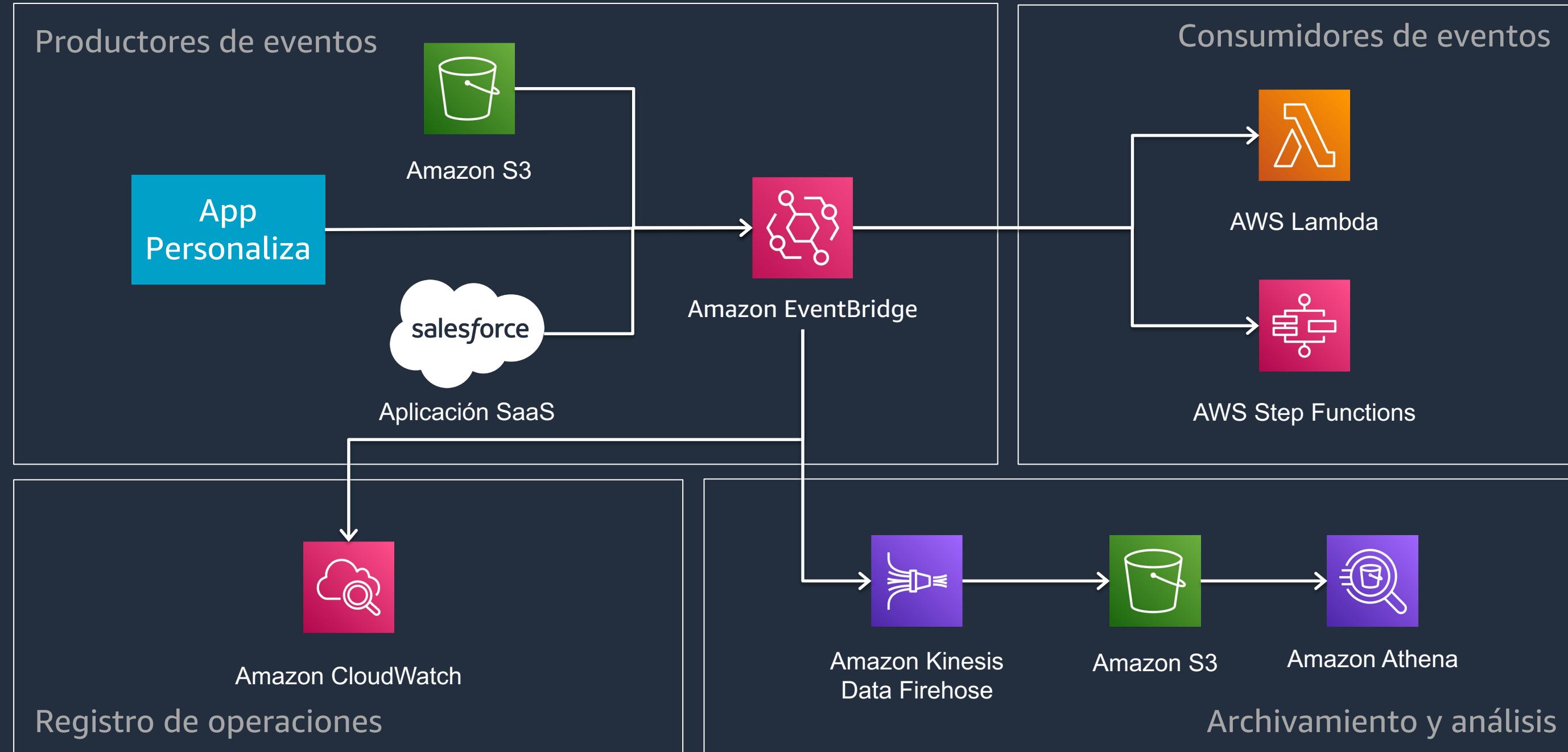


Amazon SNS



Amazon SQS

Comparando servicios para el manejo de eventos : Integración



Combina eventos para tu caso de uso



La arquitectura basada en eventos se divide en tres partes:



Productor de eventos

Publica eventos
(web o aplicaciones
móviles,
microservicios,
dispositivos IoT)



Ingestión de eventos



Consumidor
de eventos

Procesa los eventos
(Actualiza la base de
datos, envía una
notificación, ejecuta
análisis de ML)

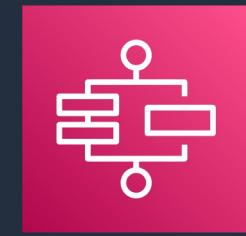
Consumidor de eventos

Servicios Serverless:

- Tiene integración nativa con otros servicios.
- Solo se ejecuta cuando existan eventos para ser procesados.
- Escala o reduce su capacidad automáticamente dependiente del volumen de eventos.



Invoca una función de Lambda para procesar el evento o lo envía hacia otro servicio de AWS.



Activa un flujo de trabajo en Step Functions

Objetivos de Amazon EventBridge



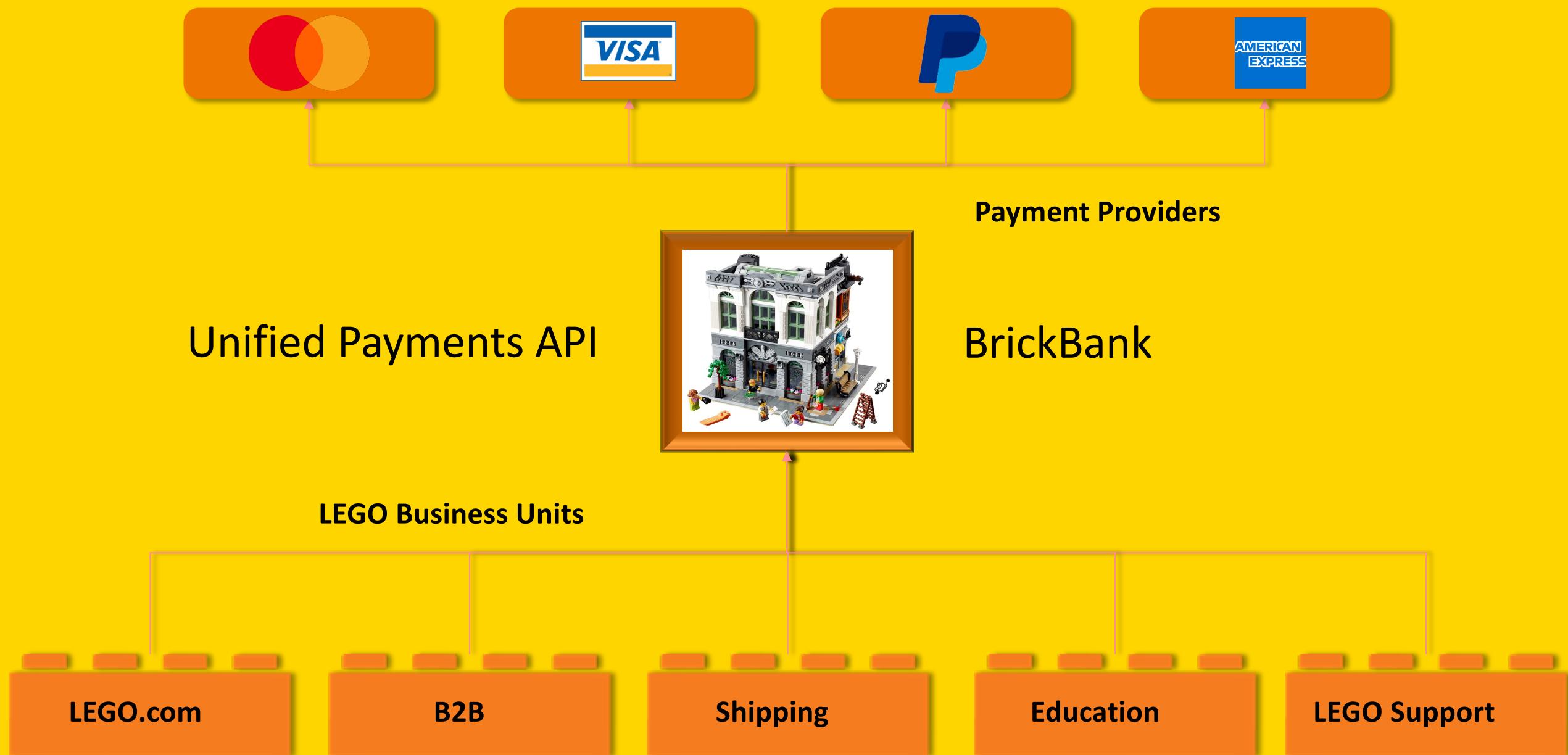
- AWS Lambda
- Amazon Kinesis
- AWS Step Functions
- Amazon API Gateway
- Amazon Redshift
- Amazon SNS
- Amazon SQS
- Amazon CloudWatch
- EventBridge bus de eventos en una región o cuenta diferente
- Y aun mas!

¿Qué están construyendo los clientes con arquitecturas basadas en eventos ?

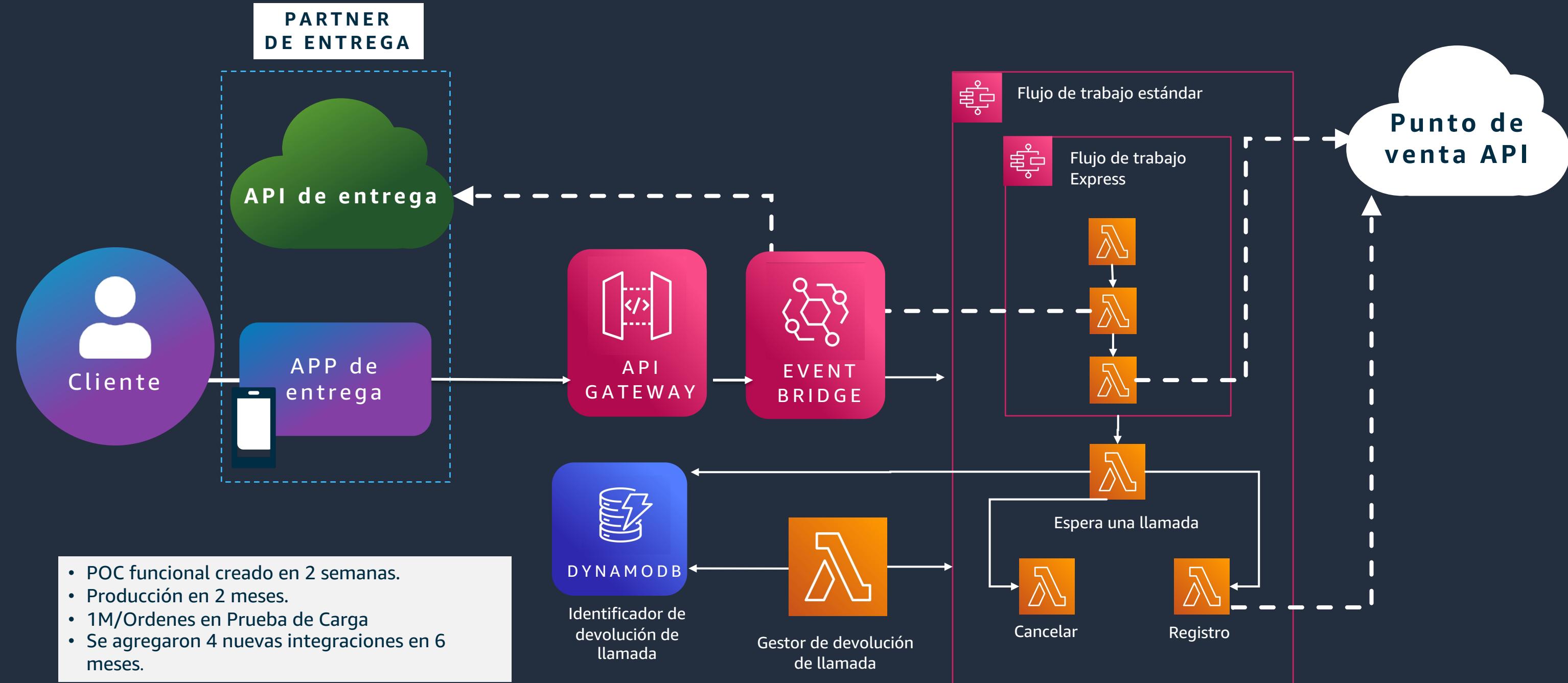
Lego utiliza una arquitectura basada en eventos diseñada para ser escalable.



Serverless Payments Platform



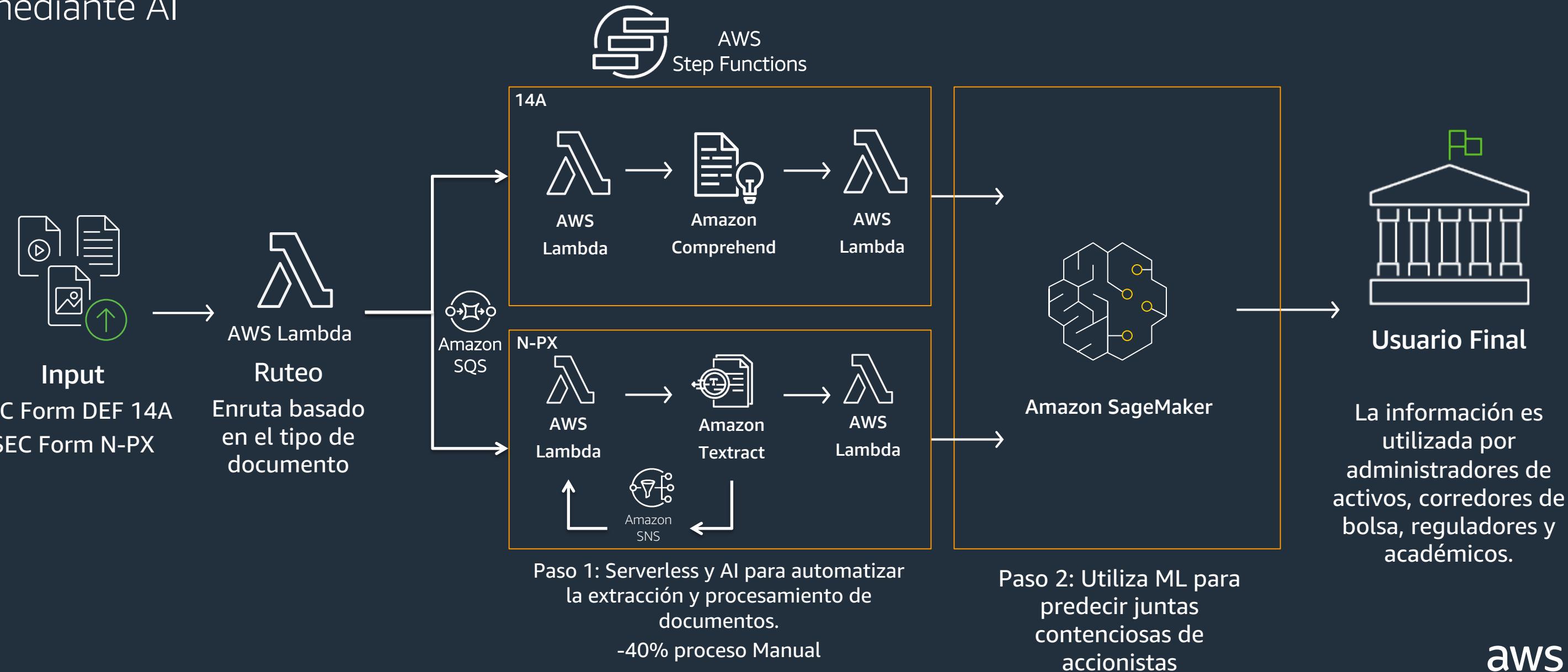
Taco Bell – Solución partner middleware para órdenes





Broadridge®

Proceso orquestado actualizado para extraer datos de formularios de votación por representación mediante AI



Disney+ potencia la ubicuidad de los datos con Amazon Kinesis

Desafío

Disney+, que disponía de datos lentos y limitados a partir de los silos de datos y el procesamiento por lotes, necesitaba ofrecer a sus equipos un acceso rápido a los datos para poder mejorar la experiencia de los clientes de Disney+ a gran escala.

Solución

Para lograr una democracia de datos rápida y obtener información casi en tiempo real, Disney+ creó una plataforma de transmisión de datos mediante los servicios de AWS Analytics, incluidos Amazon Kinesis Data Streams y Amazon Kinesis Data Firehose.

Resultado

Utilizando AWS, Disney+ ahora apoya una cultura basada en los datos que proporciona datos e información casi en tiempo real basados en miles de millones de eventos para mejorar la experiencia de decenas de millones de usuarios de manera confiable y rentable.



Primeros pasos

Recursos

Charlas:

- re:Invent 2021: [Building next-gen applications with event-driven architectures](#)
- re:Invent 2021: [Building modern applications? Think integration](#)

Tutoriales:

- Serverless Land: [Build Decoupled, Event-Driven Architectures](#)
- AWS Skill Builder: [Building Serverless Applications with an Event-Driven Architecture](#)

¿Dónde está tu equipo el día de hoy ?

Ponte en contacto con tu equipo de cuenta de AWS para conectarte con expertos de arquitectura basada en eventos.

Escenario	Siguiente paso
Necesitamos entrenamiento	Definamos un AWS-led Workshop.
Intersados en construir una aplicación basada en eventos.	Desarrollemos una PoC en AWS.
Estamos actualmente construyendo una arquitectura basada en eventos y buscamos ayuda en servicios y diseño de decisiones.	Revisión de arquitectura con un experto de AWS.

Muchas gracias!

Integración vs. Sistemas distribuidos

Integración? Sistemas distribuidos?



Las ventajas y desventajas de la arquitectura dependen en gran medida del contexto: organización, cronograma y nivel de control

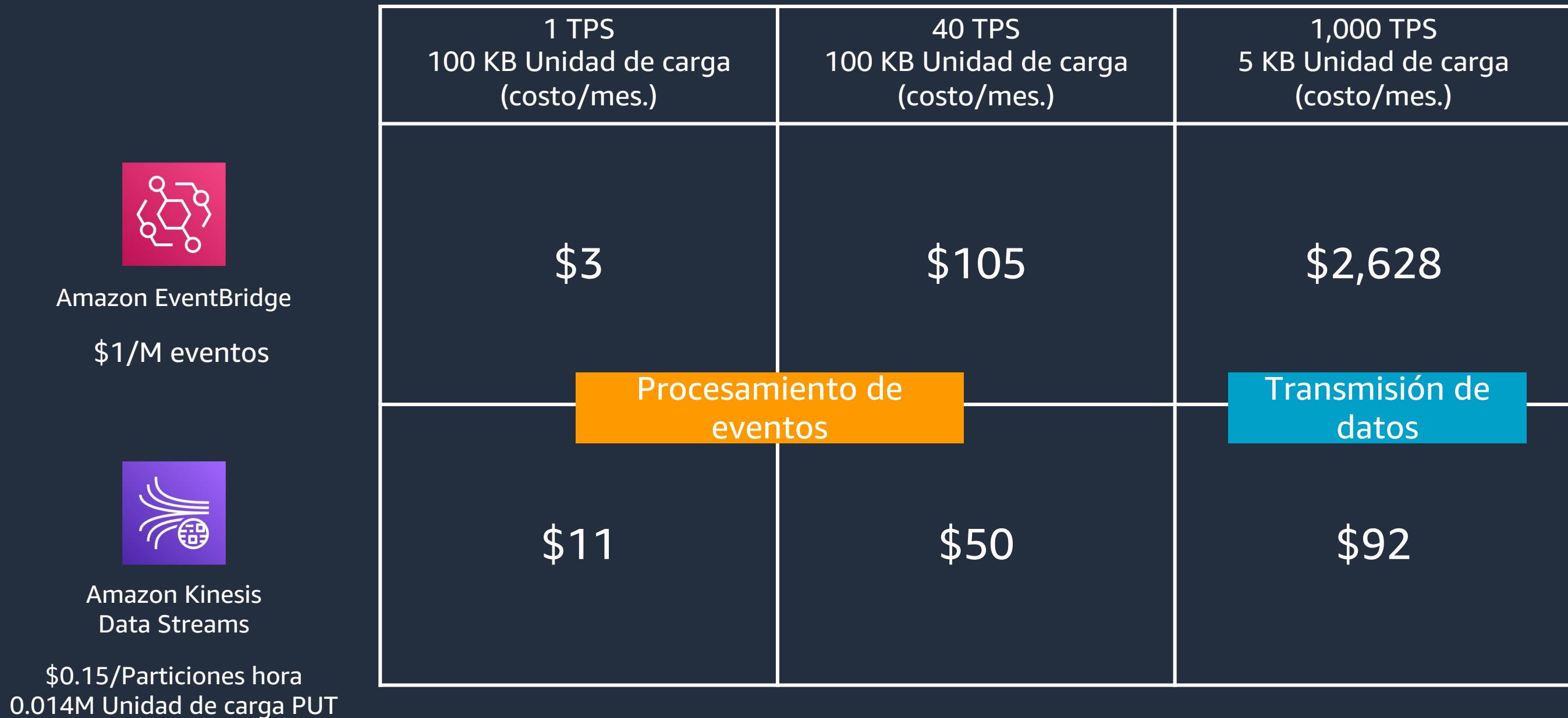
Abarcando equipos, tiempo y enfoque de control

Enfoque	Nivel de control	Ciclo de vida de la entrega	Equipo	
Migración	Bajo	Una vez	Una vez	
Sincronización de datos/integración tradicional	Bajo	Largo	Dedicado	
Bus de servicio empresarial	Alguno	Más lento que el desarrollo de componentes	Possiblemente dedicado	
Aplicaciones en la nube distribuida	Alto	Igual que el desarrollo de componentes	embebido	

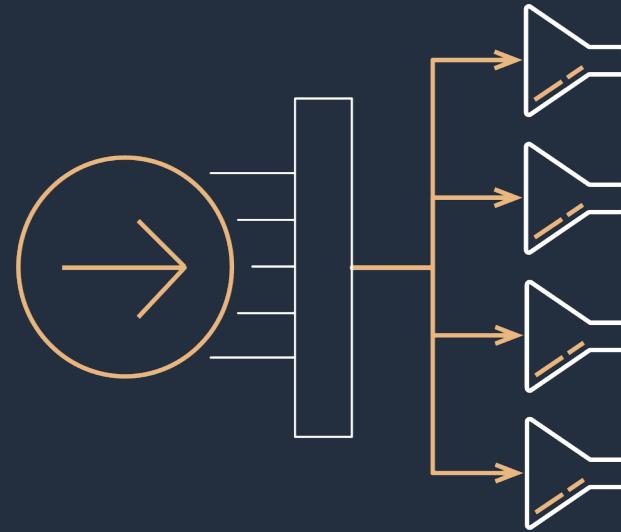
Abarcando equipos, tiempo y enfoque de control

Enfoque	Nivel de control	Ciclo de vida de la entrega	Equipo	Herramienta (indicativa)
Migración	Bajo	Una vez	Una vez	Amazon AppFlow
Sincronización de datos/integración tradicional	Bajo	Largo	Dedicado	Amazon AppFlow
Bus de servicio empresarial	Alguno	Más lento que el desarrollo de componentes	Possiblemente dedicado	Amazon MQ
Aplicaciones en la nube distribuida	Alto	Igual que el desarrollo de componentes	embebido	Amazon EventBridge AWS Lambda Destinations

Comparación de los servicios de ingestión de eventos: costo

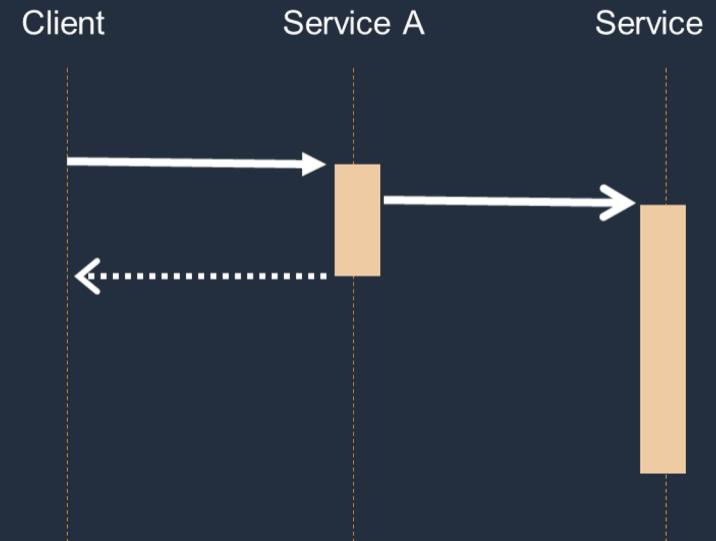


Las arquitecturas basadas en eventos aumentan la confiabilidad y la escalabilidad



Ingestión de eventos

Productores y consumidores abstractos unos de otros



Eventos asíncronos

Mejore la capacidad de respuesta y reduce las dependencias



Almacenamiento de eventos

Almacene los mensajes en búfer hasta que los servicios estén disponibles para procesarlos

Un **evento** Es un cambio en un estado o una actualización



Ventas

Se creo una orden nueva



Medios de comunicación
y entretenimiento



Servicios financieros



Sector Salud

Actualización de los datos médicos del paciente



Publicaciones

Documento subido para su traducción



Educación

Nuevo estudiante inscrito



Manufactura

Problema detectado en el sensor de IoT



Gobierno

Nueva solicitud enviada

En una arquitectura basada en eventos, los eventos se observan, no se dirigen.

Los eventos se observan, no se dirigen



Comandos directos

Los eventos se observan, no se dirigen



Comandos directos

Cliente X acaba de ordenar un widget



Eventos observados

Atributos de los eventos

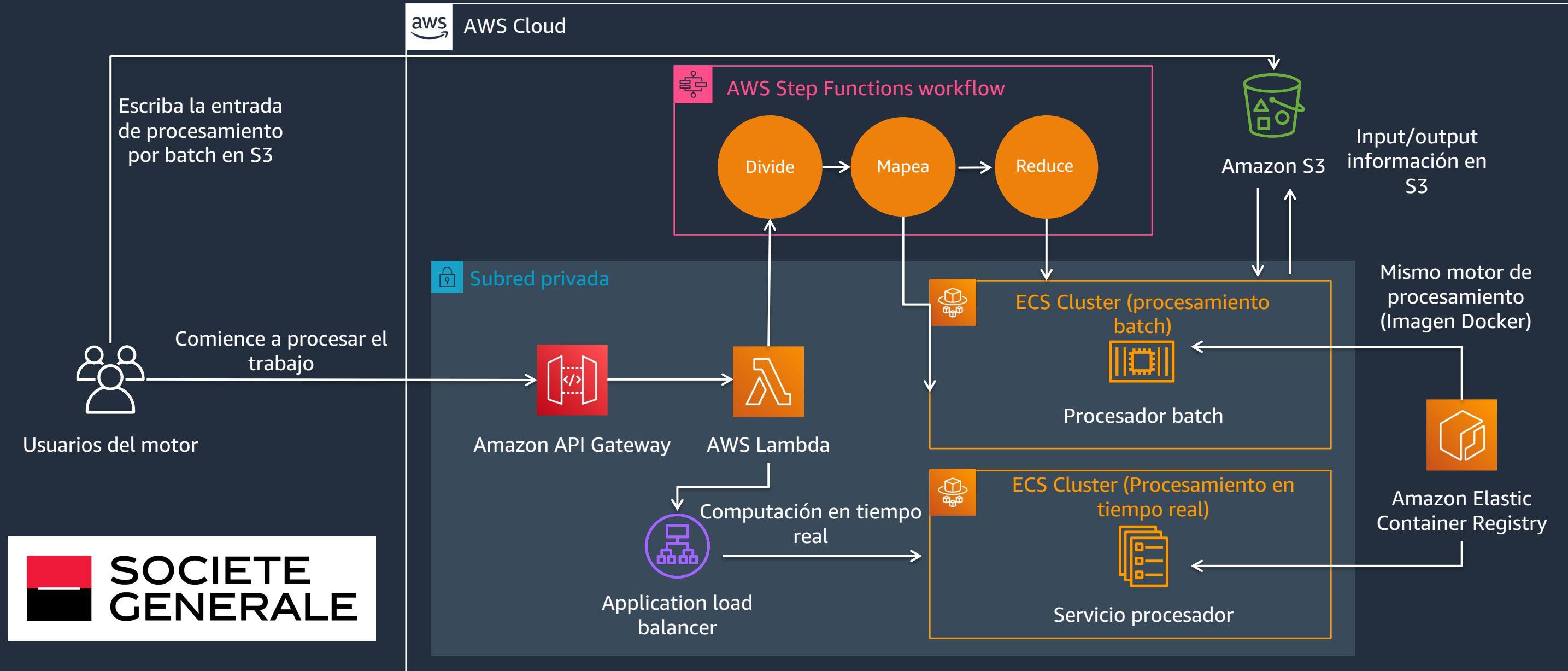
El evento se convierte en el principal mecanismo para **compartir información entre los servicios** sobre el cambio de estado.

Los eventos son **inmutables**: los eventos pasados no se pueden cambiar.

Los eventos tienen una intención semántica, que se representa como **verbos en tiempo pasado**, por ejemplo, «cliente_creado».

Los eventos son **livianos** y se correlacionan mediante propiedades que son comunes en los contextos acotados, por ejemplo, «cliente_id». Los eventos pueden apuntar a archivos más grandes almacenados en otro lugar.

Société Générale amplía y reduce AWS Fargate para realizar análisis de riesgo crediticio muy paralelos



Creación de arquitecturas basadas en eventos en AWS



TACO BELL™

Taco Bell's middleware de pedidos se puso en producción en 2 meses y puede ampliarse a millones de pedidos por hora.



Broadridge®

Broadridge automatiza miles de horas de extracción y validación manual de datos.

Hudson

Great people. Great performance.

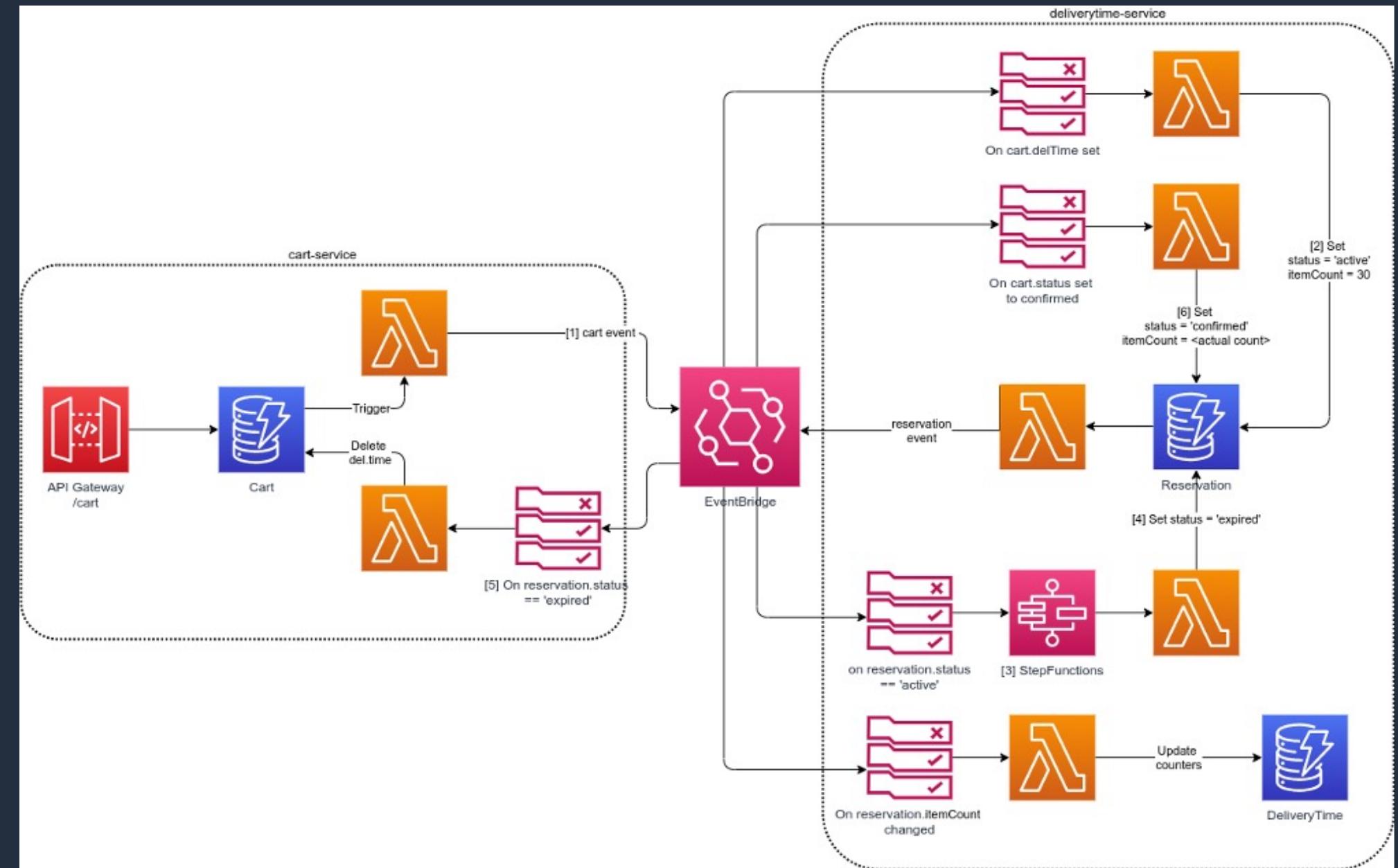
Hudson amplió su aplicación SaaS 10 veces y ahorró cientos de miles de dólares al año en costos.

MathEm

MathEm aceleró el tiempo de lanzamiento de las funciones entre 5 y 10 veces, creando un MVP de la función con ventana de entrega en 3 días.

Acelerando el lanzamiento de nuevas funciones en MatHem

MatJtem



¿Qué son las arquitecturas basadas en eventos?