goto;

GOTO
CHICAGO 2023

#GOTOchgo

Building distributed applications with event driven architecture

Eric Johnson

Principal Developer Advocate
AWS

Who am I?

- Principal Developer Advocate – AWS
- Solutions Architect for 10 years
- Developer for almost 30 years
- Father of 5, husband to a superstar
- Musician Drummer

Who am I?

- Principal Developer Advocate – AWS
- Solutions Architect for 10 years
- Developer for almost 30 years
- Father of 5, husband to a superstar
- ~~Musician~~ Drummer

Today’s agenda

Enterprise integration patterns

Event driven architecture (EDA)

Maintaining idempotency

Enterprise integration patterns

Coupling – Integration's magic word

- **Coupling is a measure of independent variability between connected systems**
- **Decoupling has a cost, both at design and runtime**
- **Coupling isn't binary**
- **Coupling isn't one-dimensional**

The many facets of coupling

Technology dependency:	Java vs. C++
Location dependency:	IP addresses, DNS
Data format dependency:	Binary, XML, JSON, ProtoBuf, Avro
Data type dependency:	int16, int32, string, UTF-8, null, empty
Semantic dependency:	Name, middle name, ZIP
Temporal dependency:	sync, async
Interaction style dependency:	messaging, RPC, query-style (GraphQL)
Conversation dependency:	pagination, caching, retries

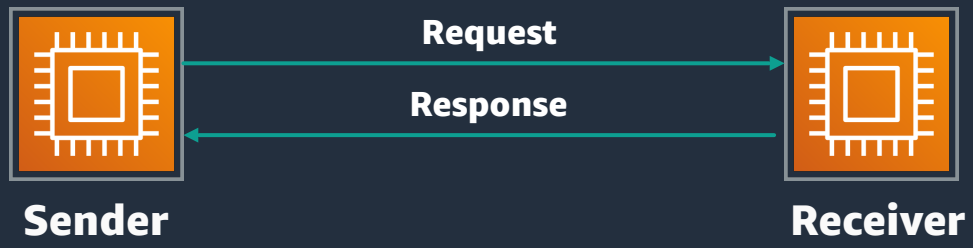
"The appropriate level of coupling depends on the level of control you have over the endpoints."

Gregor Hohpe

Co-author of "Enterprise Integration Patterns"

Synchronous request-response model

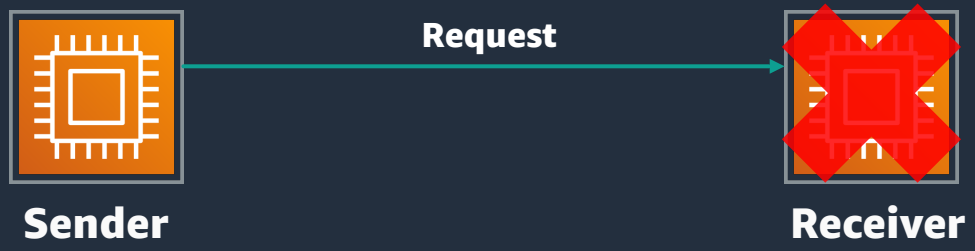
Synchronous request-response model



Advantages

- Low latency
- Simple
- Fail fast

Synchronous request-response model



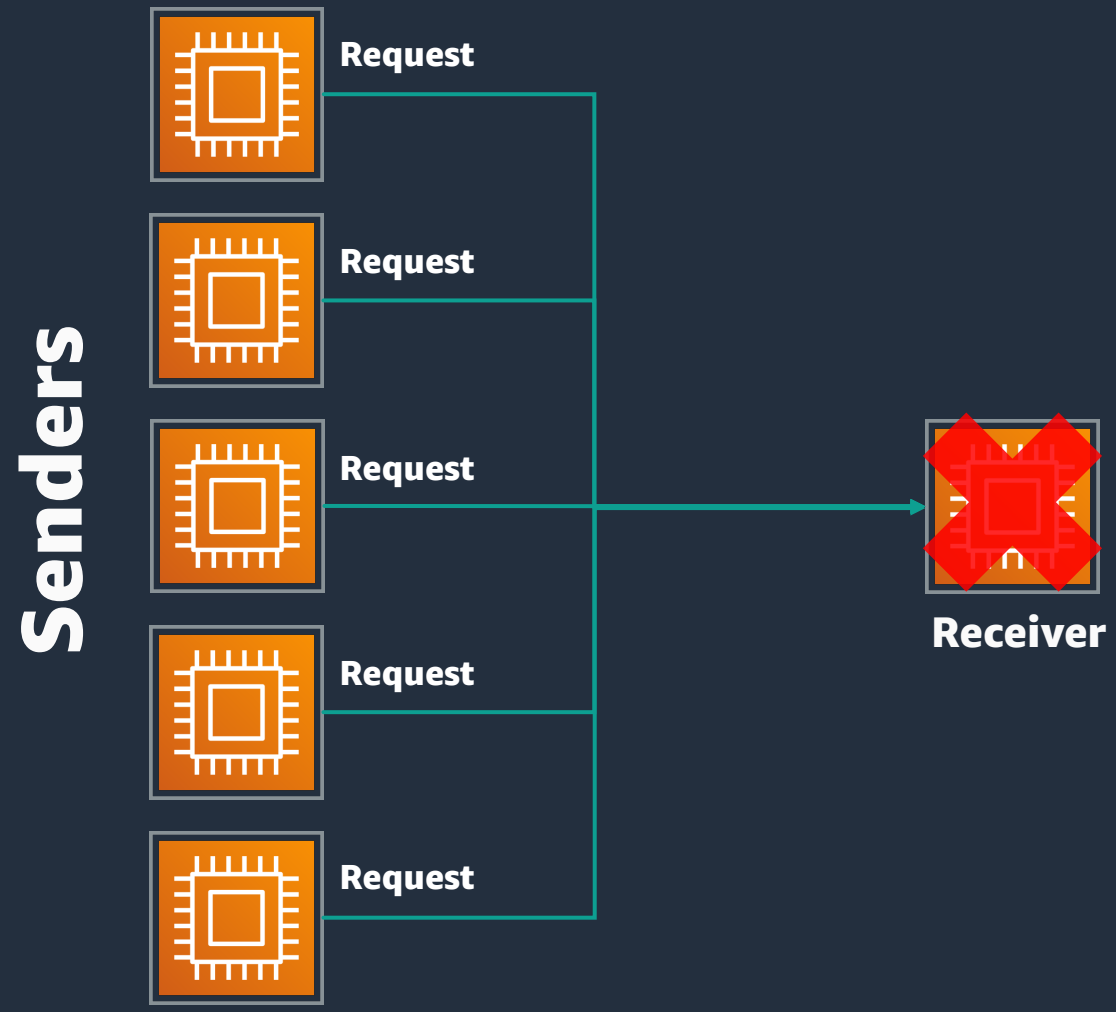
Advantages

- Low latency
- Simple
- Fail fast

Disadvantages

- Receiver failure

Synchronous request-response model



Advantages

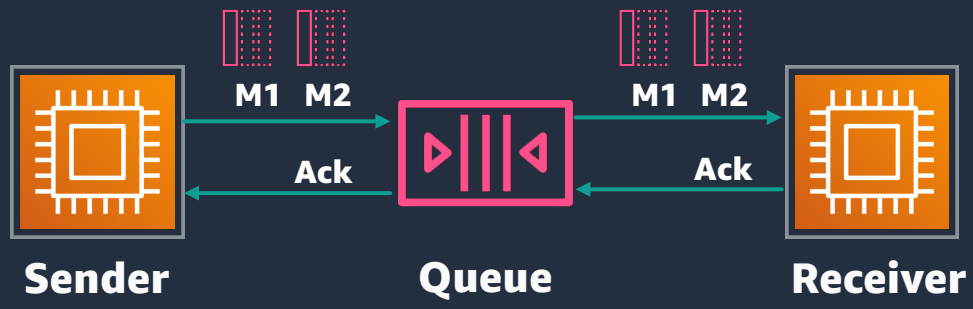
- Low latency
- Simple
- Fail fast

Disadvantages

- Receiver failure
- Receiver throttled

Asynchronous point-to-point model (queue)

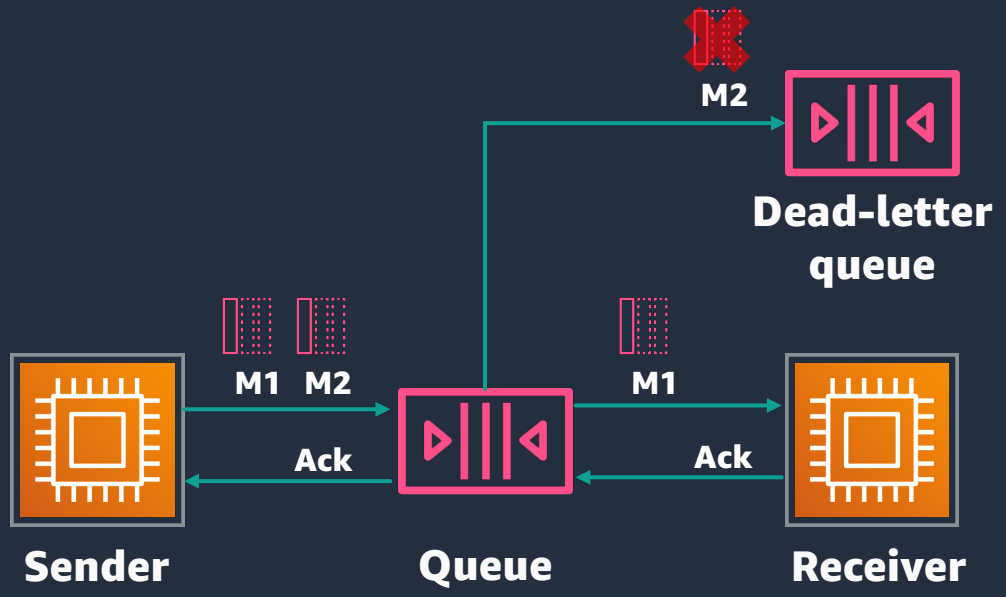
Asynchronous point-to-point model (queue)



Advantages

- Decreases temporal coupling
- Resilient to receiver failure
- Receiver controls consumption rate

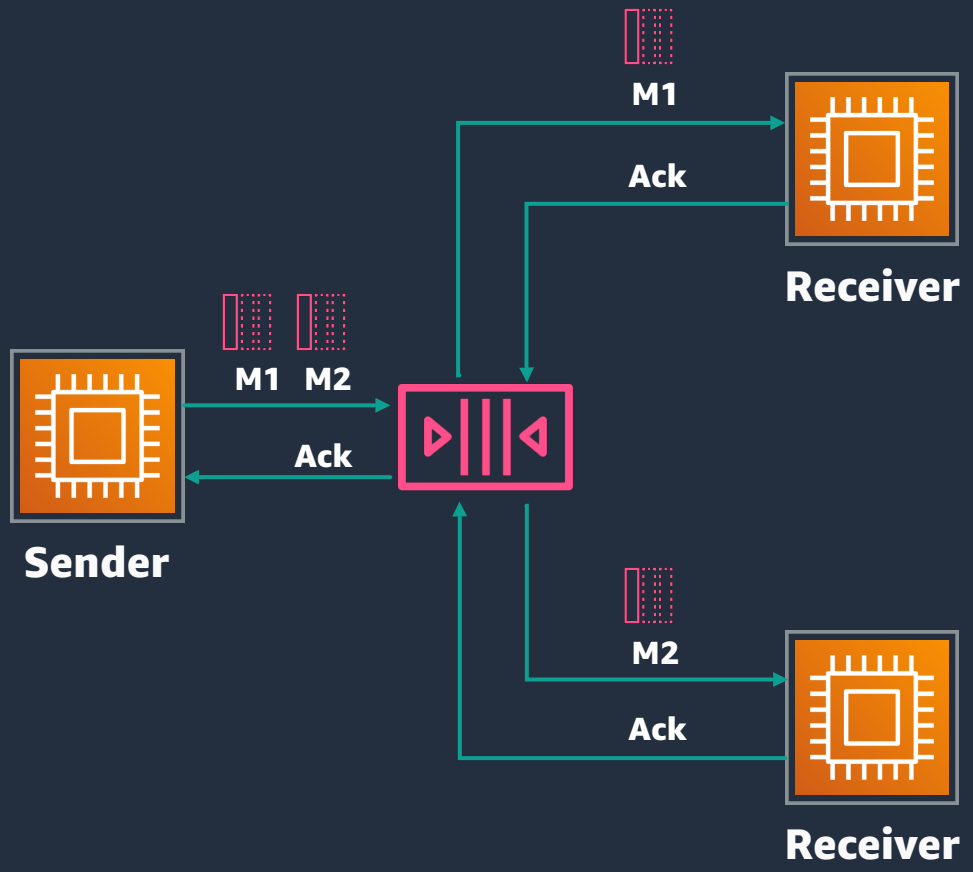
Asynchronous point-to-point model (queue)



Advantages

- ...
- Dead-letter queue (DLQ) for errors

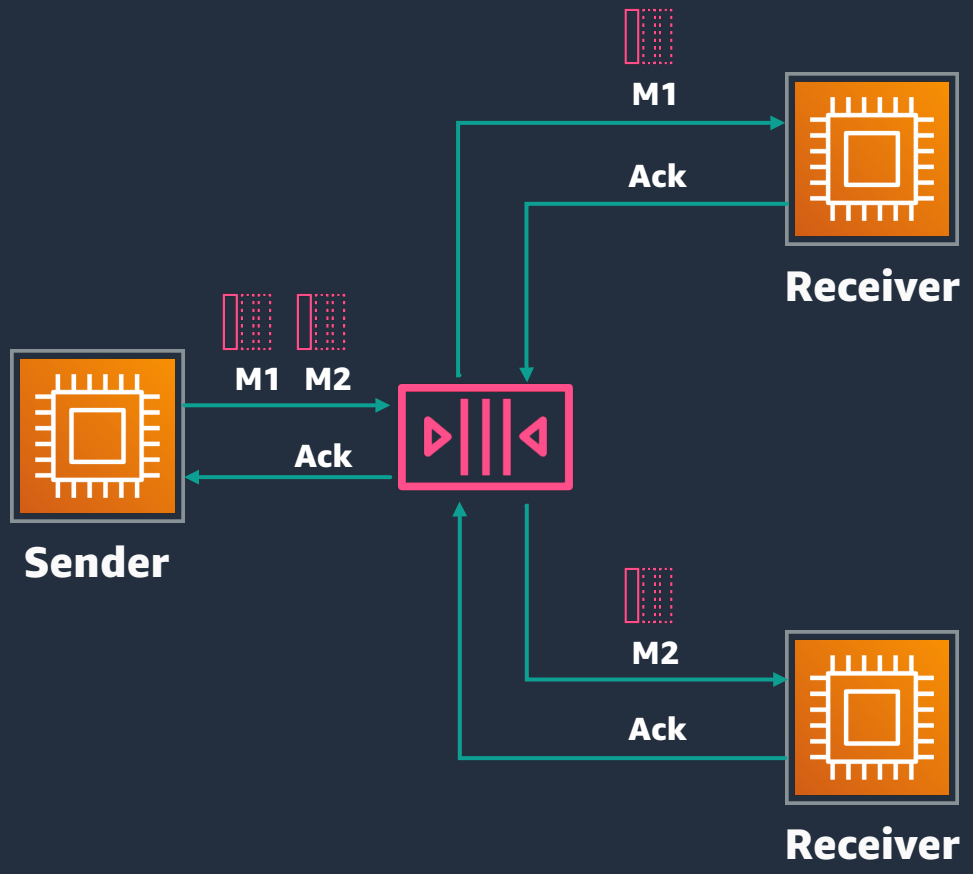
Asynchronous point-to-point model (queue)



Advantages

- ...
- **Only one receiver can consume each message**

Asynchronous point-to-point model (queue)



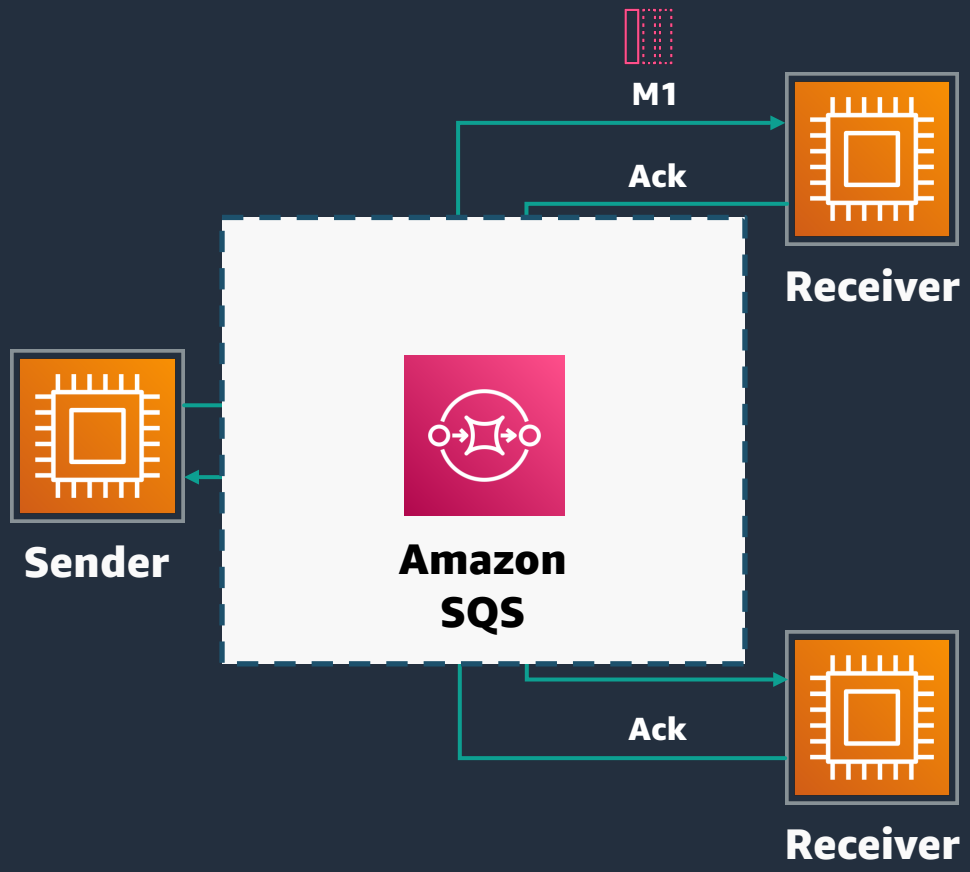
Advantages

- ...
- Only one receiver can consume each message

Disadvantages

- Response correlation
- Backlog recovery time
- Fairness in multi-tenant systems

Asynchronous point-to-point model (queue)

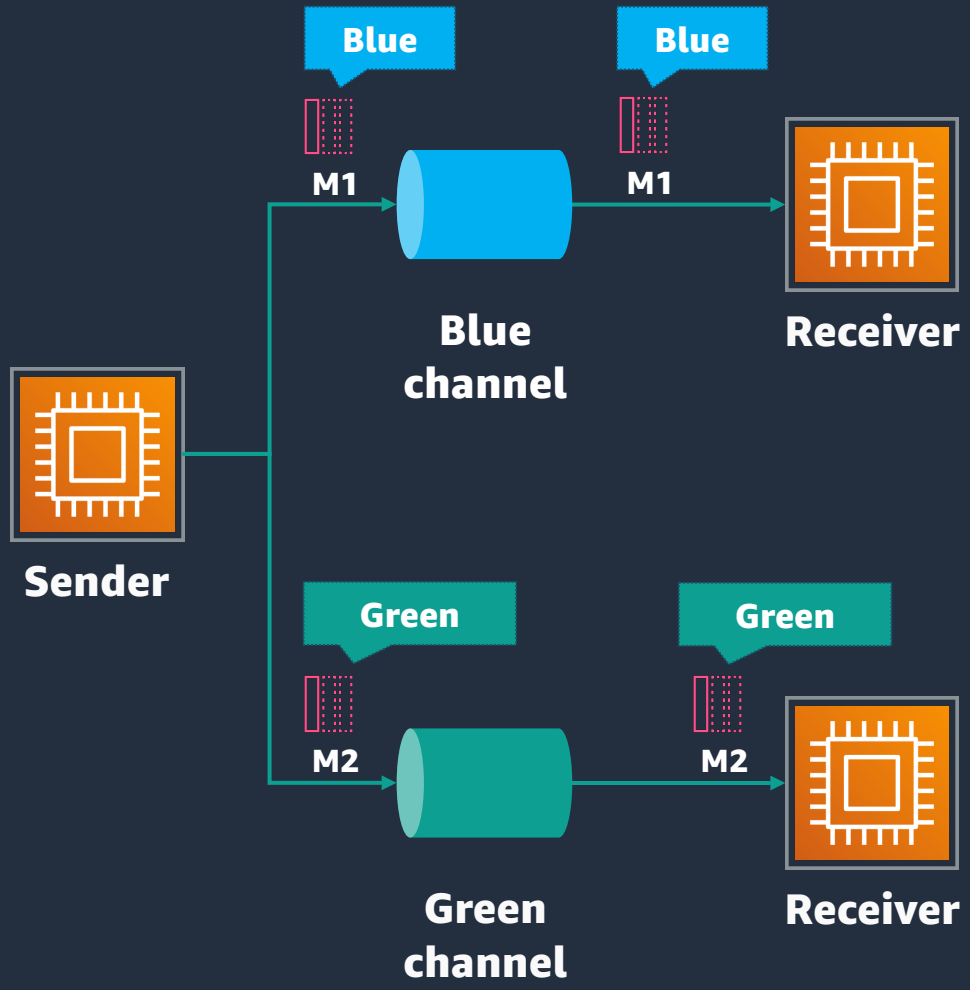


Amazon Simple Queue Service (Amazon SQS)

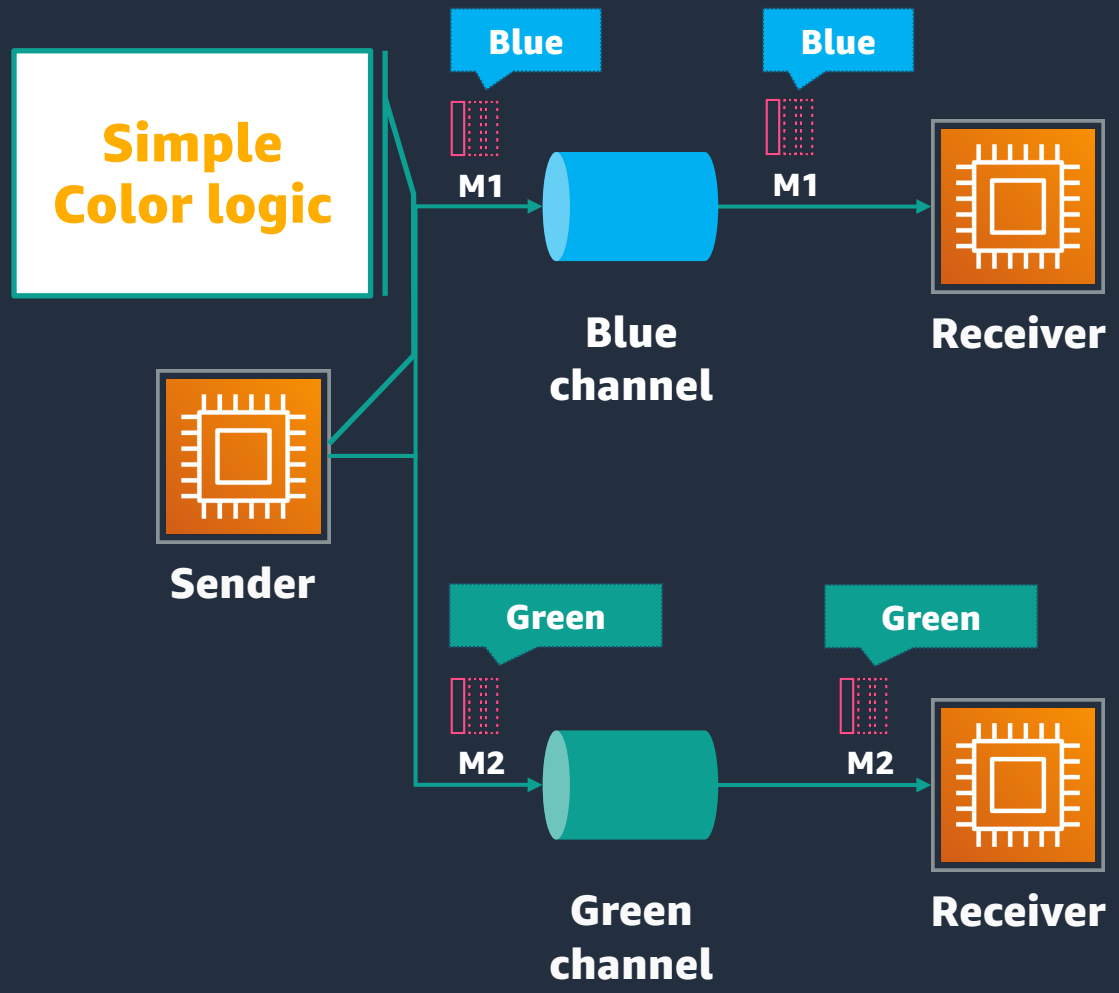
- Fully managed message queue
- Scales almost infinitely
- Simple, easy-to-use API
- DLQ support
- Standard and FIFO options

Asynchronous point-to-point model (router)

Asynchronous point-to-point model (router)



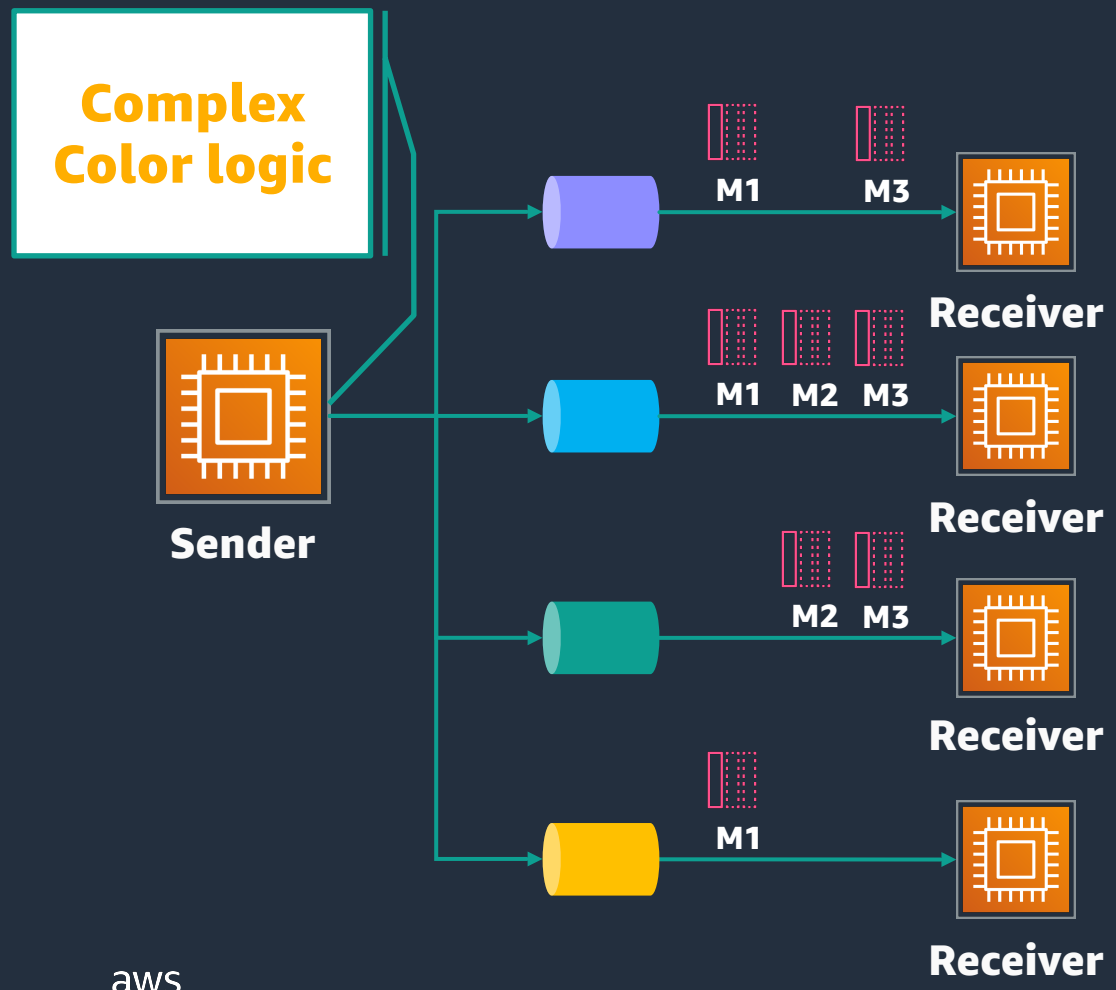
Asynchronous point-to-point model (router)



Disadvantages

- Increases location coupling
- Sender maintains routing logic

Asynchronous point-to-point model (router)

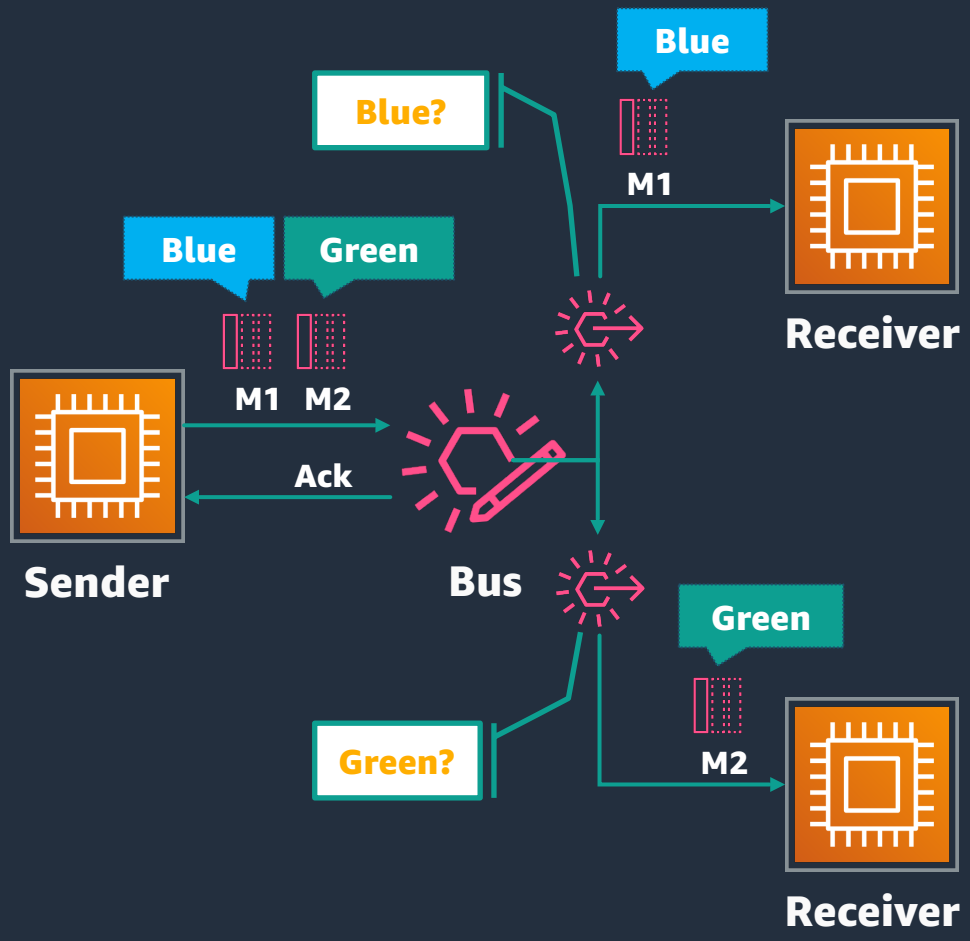


Disadvantages

- Increases location coupling
- Sender maintains routing logic
- Sender complexity increases with time

Asynchronous message-router model (bus)

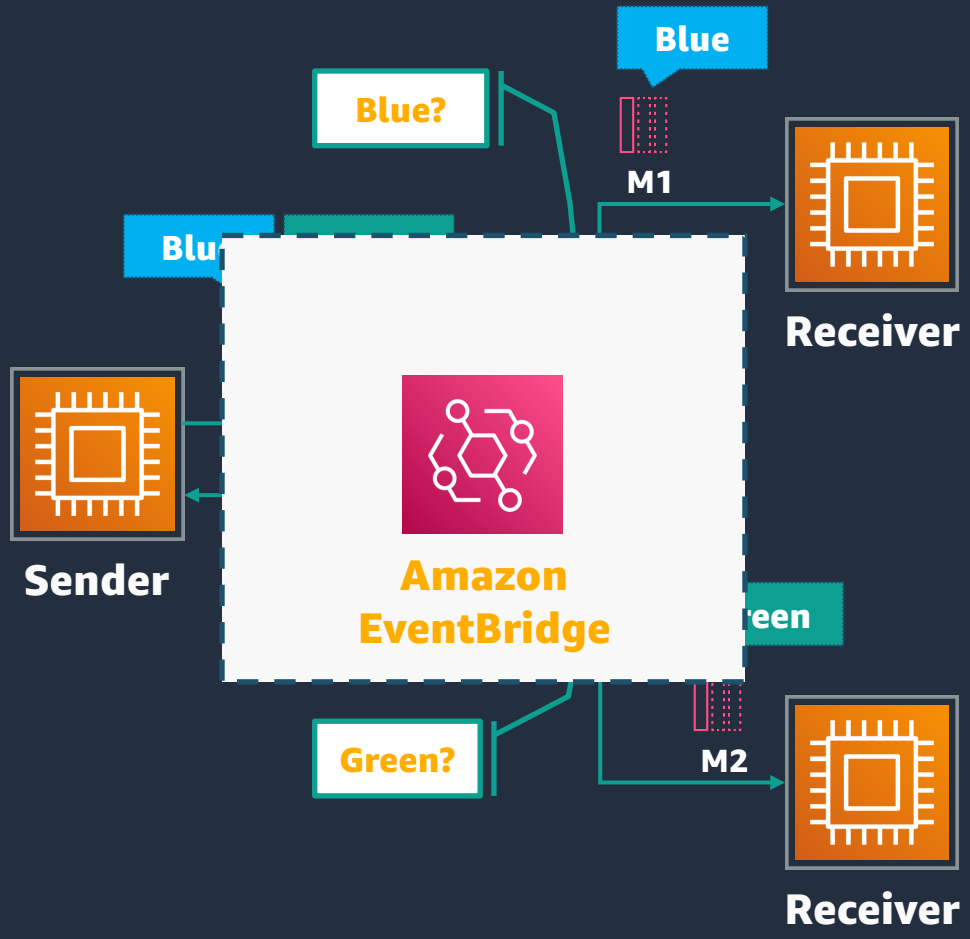
Asynchronous message-router (bus)



Advantages

- Reduces location coupling
- Efficient for senders and receivers

Asynchronous message-router (bus)

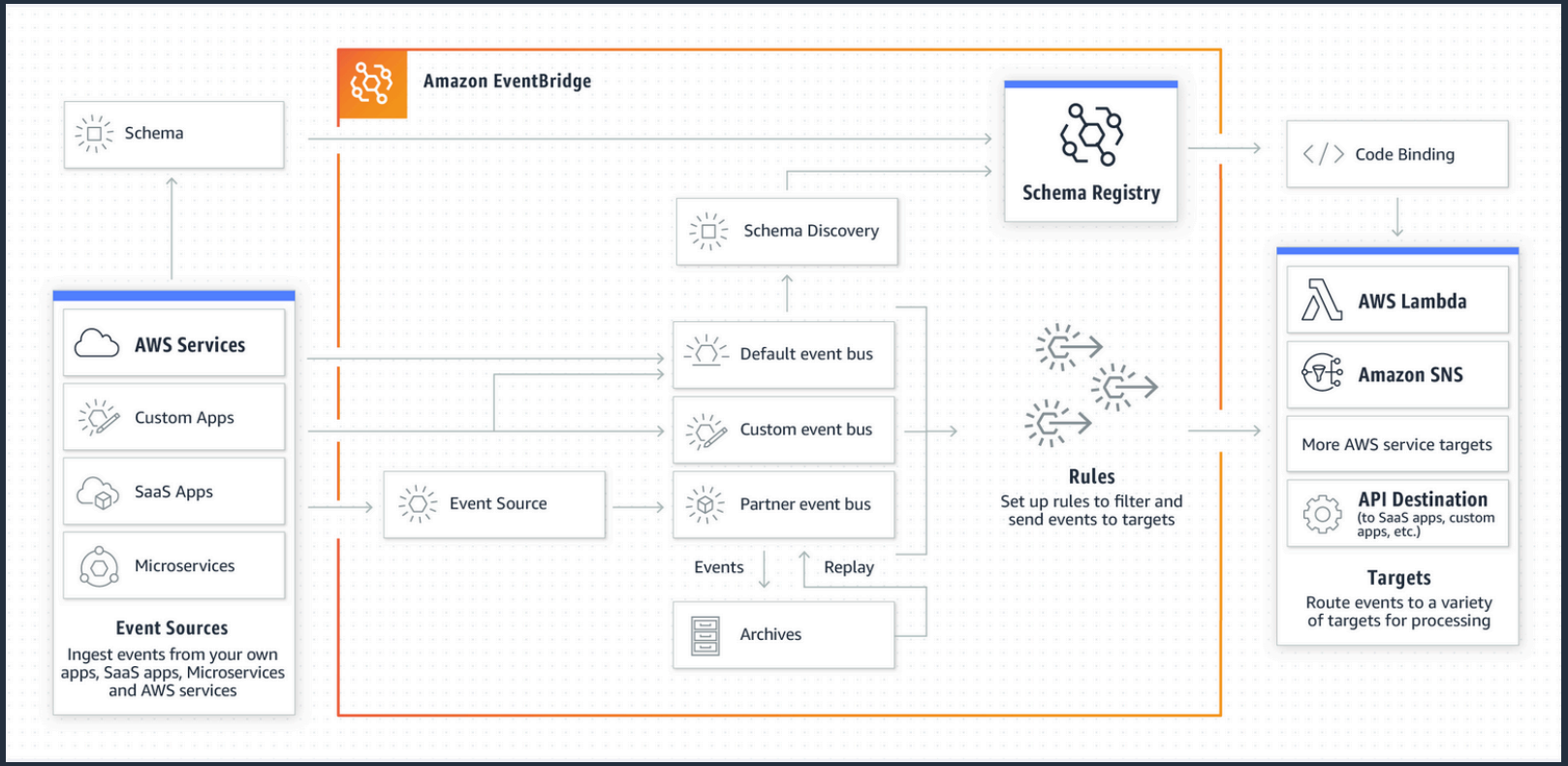


Advantages

- Reduces location coupling
- Efficient for senders and receivers

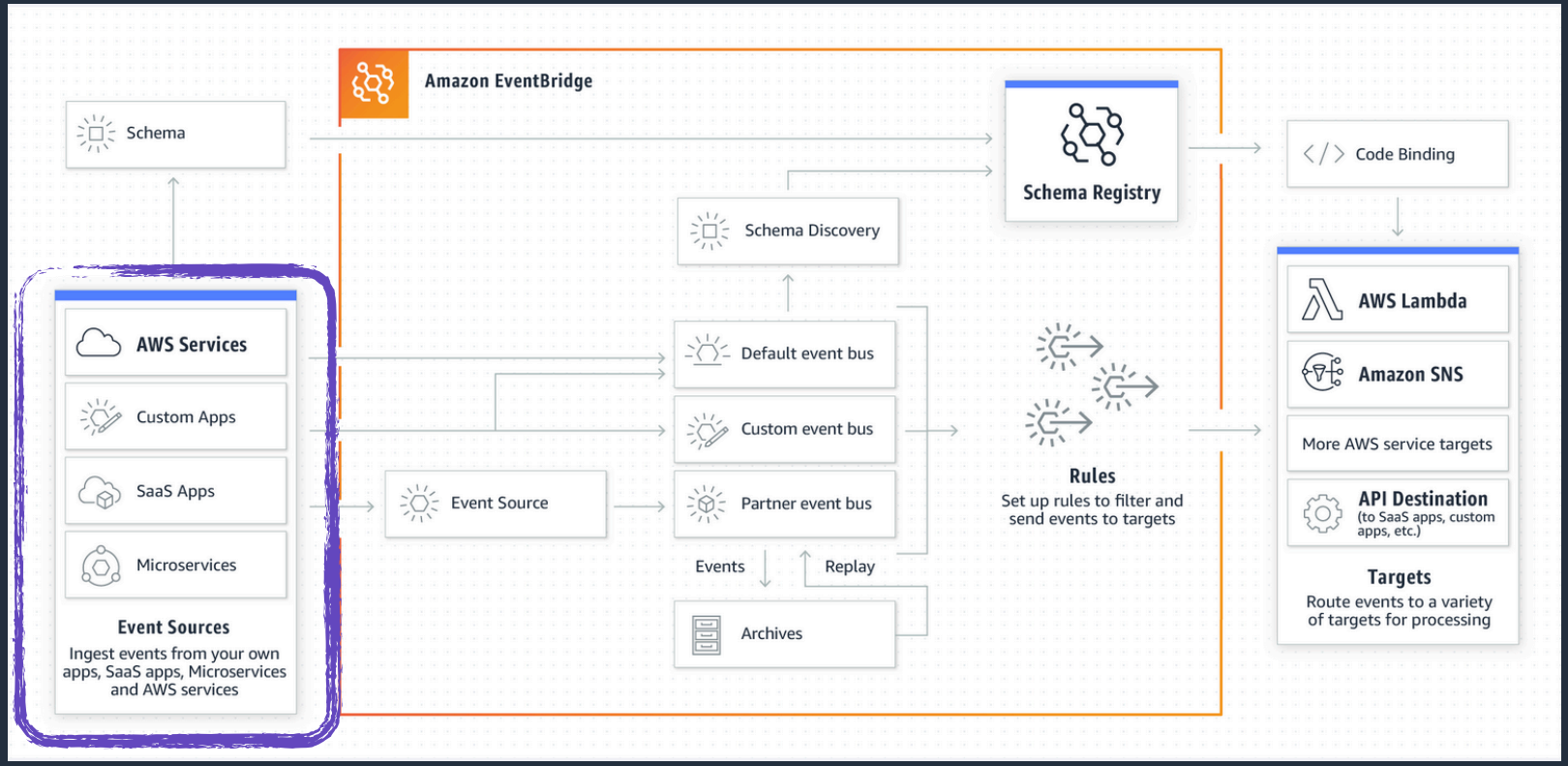
Amazon EventBridge

EventBridge is a 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**



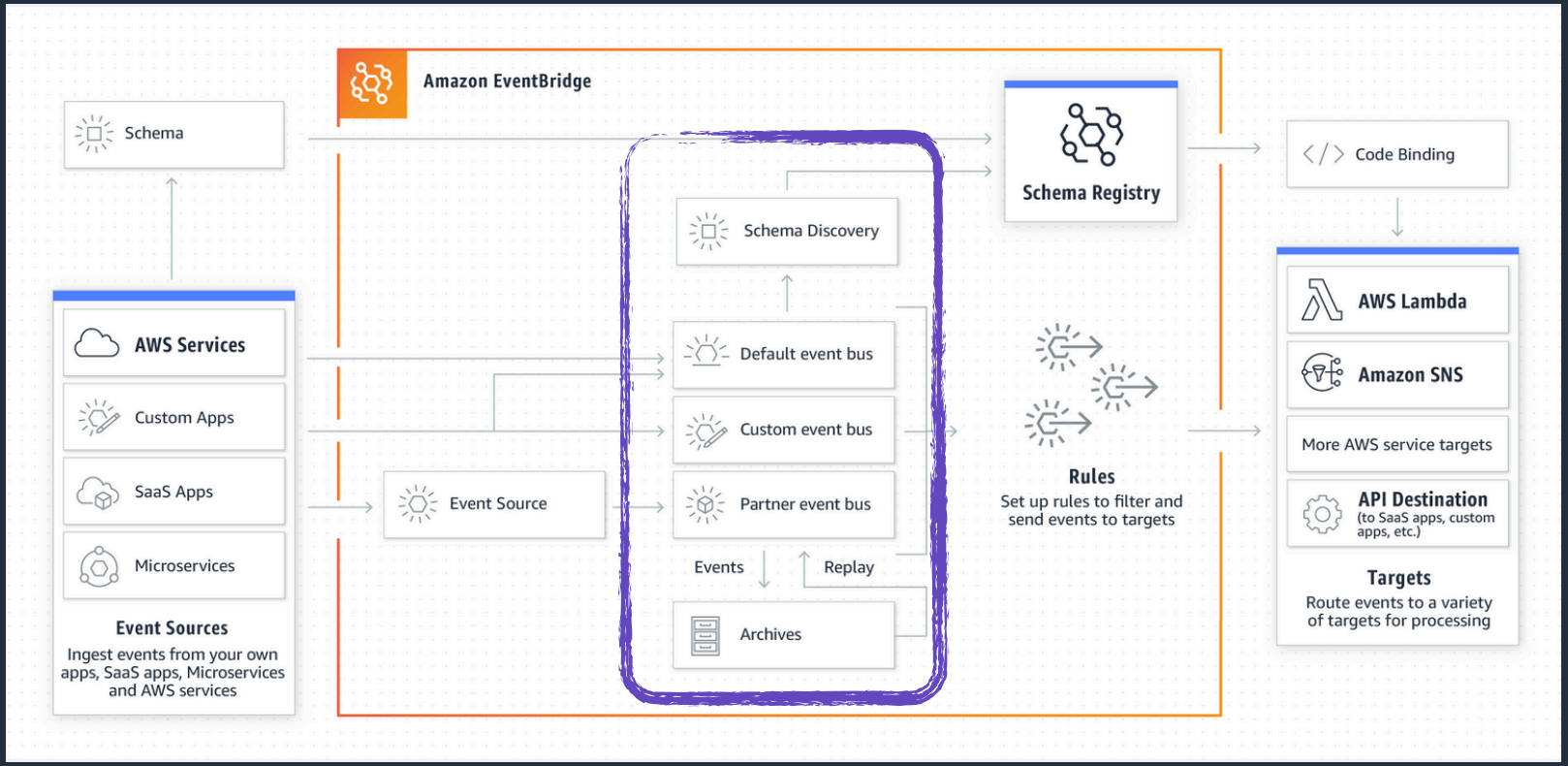
Amazon EventBridge

EventBridge is a 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**



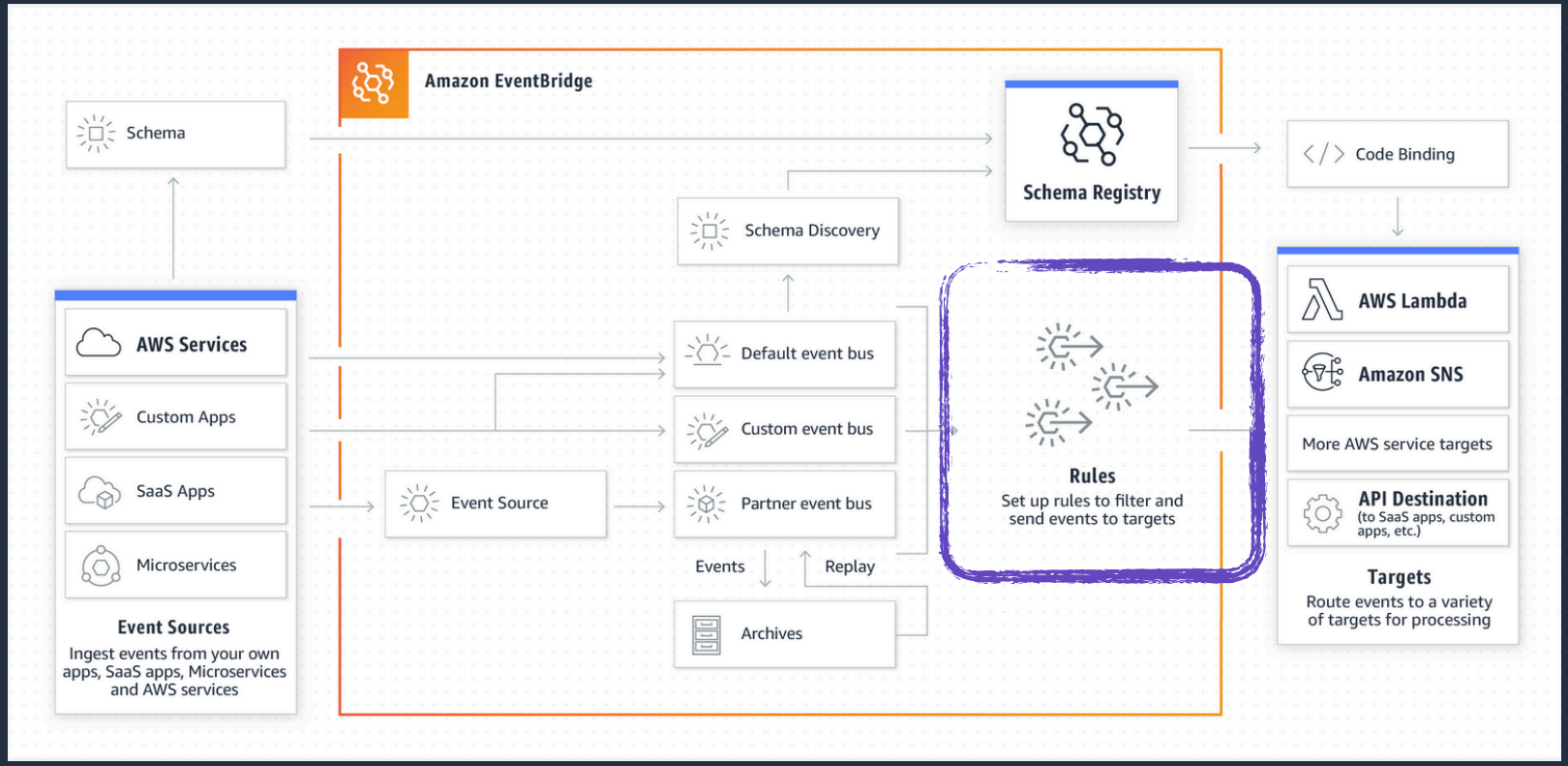
Amazon EventBridge

EventBridge is a 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**



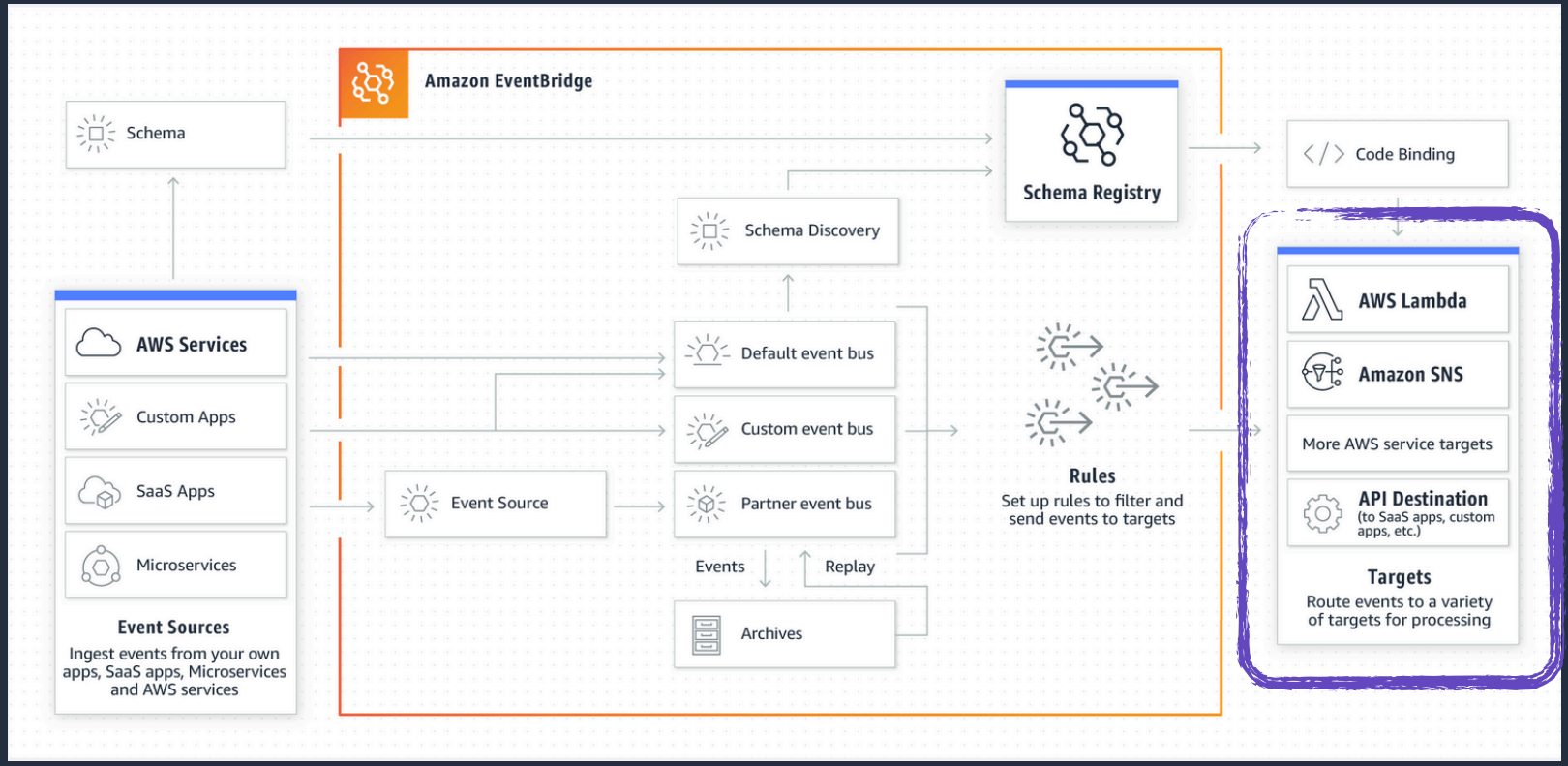
Amazon EventBridge

EventBridge is a 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**



Amazon EventBridge

EventBridge is a 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**



EventBridge content-based routing rules

EventBridge example event

```
{
  "source": "com.orders",
  "detail-type": "OrderCreated",
  "detail": {
    "metadata": {
    },
    "data": {
      "order-id": "1073459984",
      "created-at": "2021-11-26T16:05:09-04:00",
      "price": 24.62,
      "currency": "AU",
    }
  }
}
```

EventBridge content-based routing rules

EventBridge example event


```
{
  "source": "com.orders",
  "detail-type": "OrderCreated",
  "detail": {
    "metadata": {
    },
    "data": {
      "order-id": "1073459984",
      "created-at": "2021-11-26T16:05:09-04:00",
      "price": 24.62,
      "currency": "AU",
    }
  }
}
```


EventBridge example rule


```
{
  "detail": {
    "data":  {
      "currency": ["AU", "NZ"]
    }
  }
}
```

EventBridge content-based routing rules

EventBridge example event

```
{
  "source": "com.orders",
  "detail-type": "OrderCreated",
  "detail": {
    "metadata": {
    },
    "data": {
      "order-id": "1073459984",
      "created-at": "2021-11-26T16:05:09-04:00",
      "price": 24.62,
      "currency": "AU",
    }
  }
}
```

EventBridge example rule

```
{
  "detail": {
    "data":  {
      "currency": ["AU", "NZ"]
    }
  }
}
```

EventBridge content-based routing rules

EventBridge example event

```
{
  "source": "com.orders",
  "detail-type": "OrderCreated",
  "detail": {
    "metadata": {
    },
    "data": {
      "order-id": "1073459984",
      "created-at": "2021-11-26T16:05:09-04:00",
      "price": 24.62,
      "currency": "AU",
    }
  }
}
```

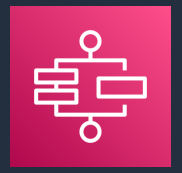
EventBridge example rule

```
{
  "detail": {
    "data":  {
      "currency": ["AU", "NZ"]
    }
  }
}
```

Match!

Target 20+ AWS services and API destinations


AWS Lambda


AWS Step Functions


Amazon Kinesis Data Streams


Amazon CloudWatch


AWS CodePipeline

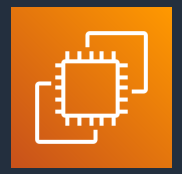

Amazon ECS


Amazon SNS


Amazon Kinesis Data Firehose


AWS Systems Manager


AWS CodeBuild


Amazon EC2

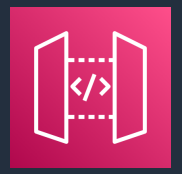

Amazon SQS


AWS Glue


Incident Manager, a capability of AWS Systems Manager


Amazon SageMaker


AWS Batch


Amazon API Gateway


Amazon Redshift


Amazon Inspector


API destinations

Event-driven architecture

Properties of events

```
{
  "source": "com.orders",
  "detail-type": "OrderCreated",
  "detail": {
    "metadata": {
      "idempotency-key": "c1b95b88”,
    },
    "data": {
      "order-id": "1073459984"
    }
}
```

Sparse events vs. full state descriptions

Order 123 was created at 10:47 AM by customer 456

Order 123 was created at 10:47 AM by customer 456. The current status is Open, the total was $237.51, ...

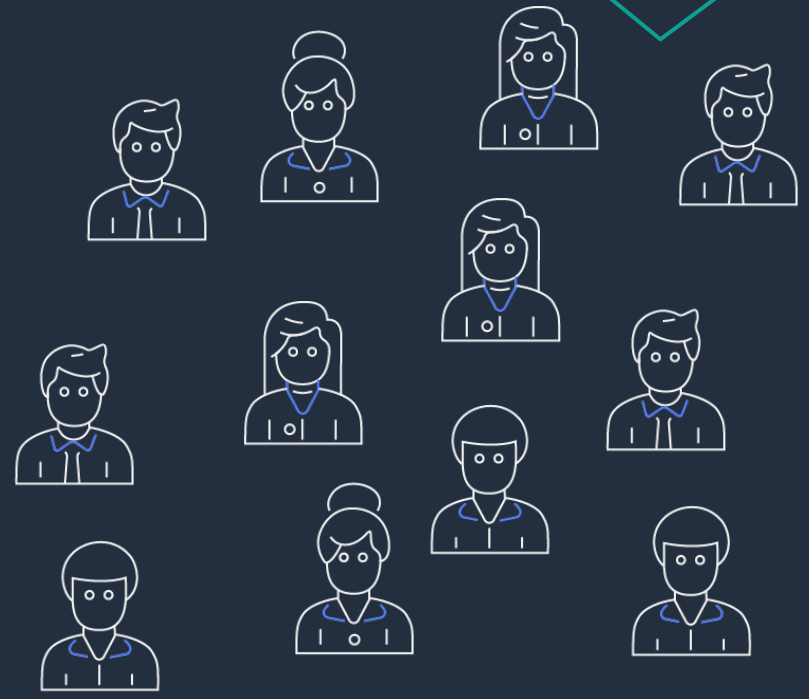
Sparse events

Full state description

Considerations with sparse events

Order 123 was created at 10:47 AM by customer 456

What are the details for order 123?



Events

Considerations with full state descriptions

```
{
  "source": "com.orders",
  "detail-type": "OrderCreated",
  "detail": {
    "metadata": {
      "idempotency-key": "c1b95b88”,
    },
    "data": {
      "order-id": "1073459984",
      "status": "Open",
      "total": "237.51"
    }
}
```

- **Event schemas should be backward compatible**

Considerations with full state descriptions

```
{
  "source": "com.orders",
  "detail-type": "OrderCreated",
  "detail": {
    "metadata": {
      "idempotency-key": "c1b95b88”,
    },
    "data": {
      "order-id": "1073459984",
      "status": "Open",
      "total": "237.51"
    }
}
```

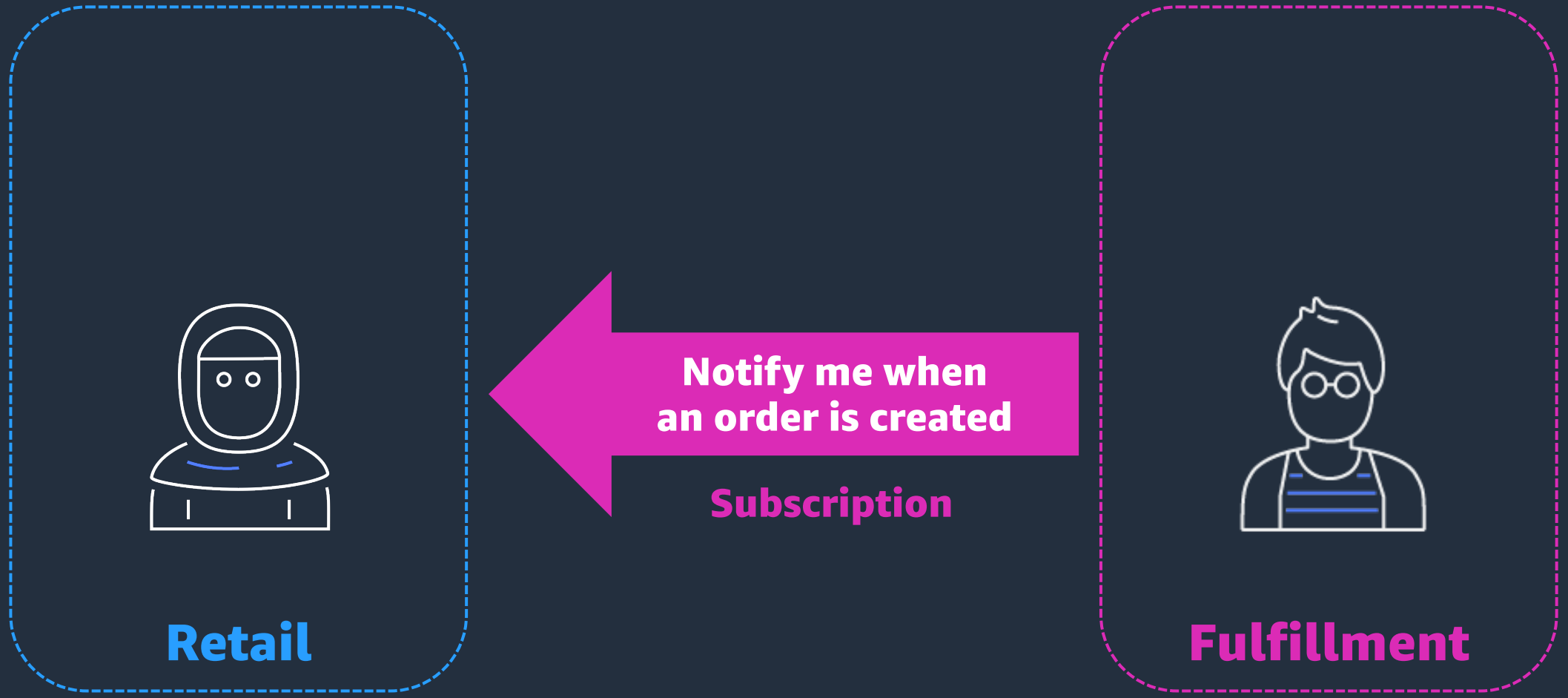
- **Event schemas should be backward compatible**
- **Cost to calculate values can increase over time**

Choreograph events between domains using subscriptions

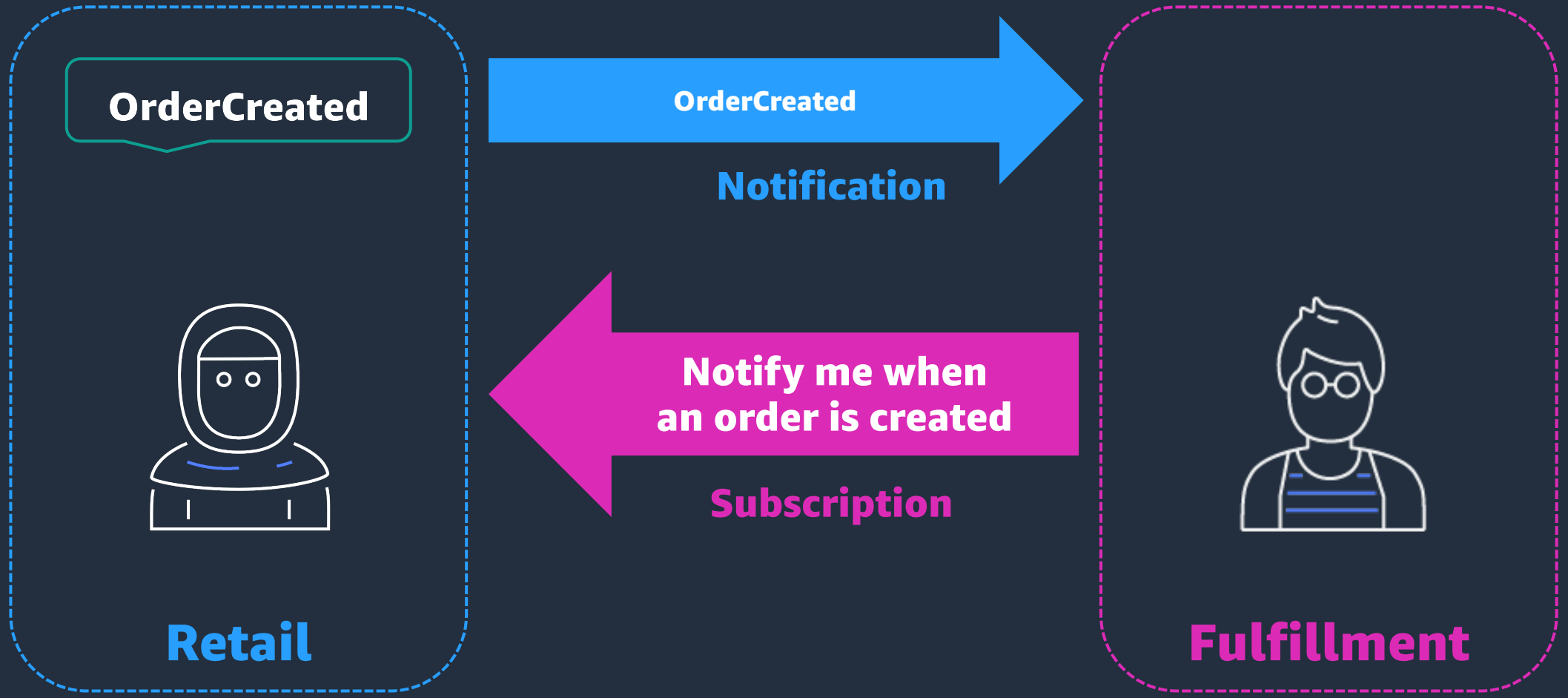
Retail

Fulfillment

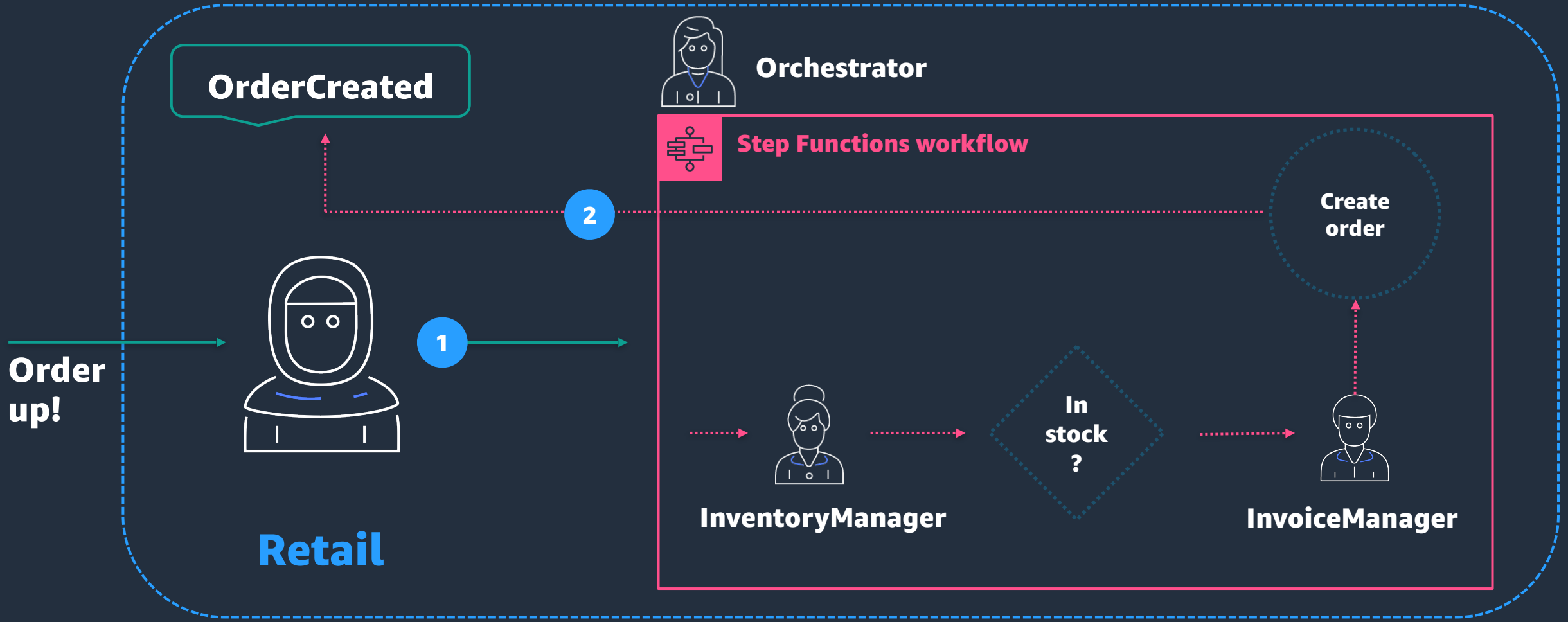
Choreograph events between domains using subscriptions



Choreograph events between domains using subscriptions



Orchestrate a business process within a domain

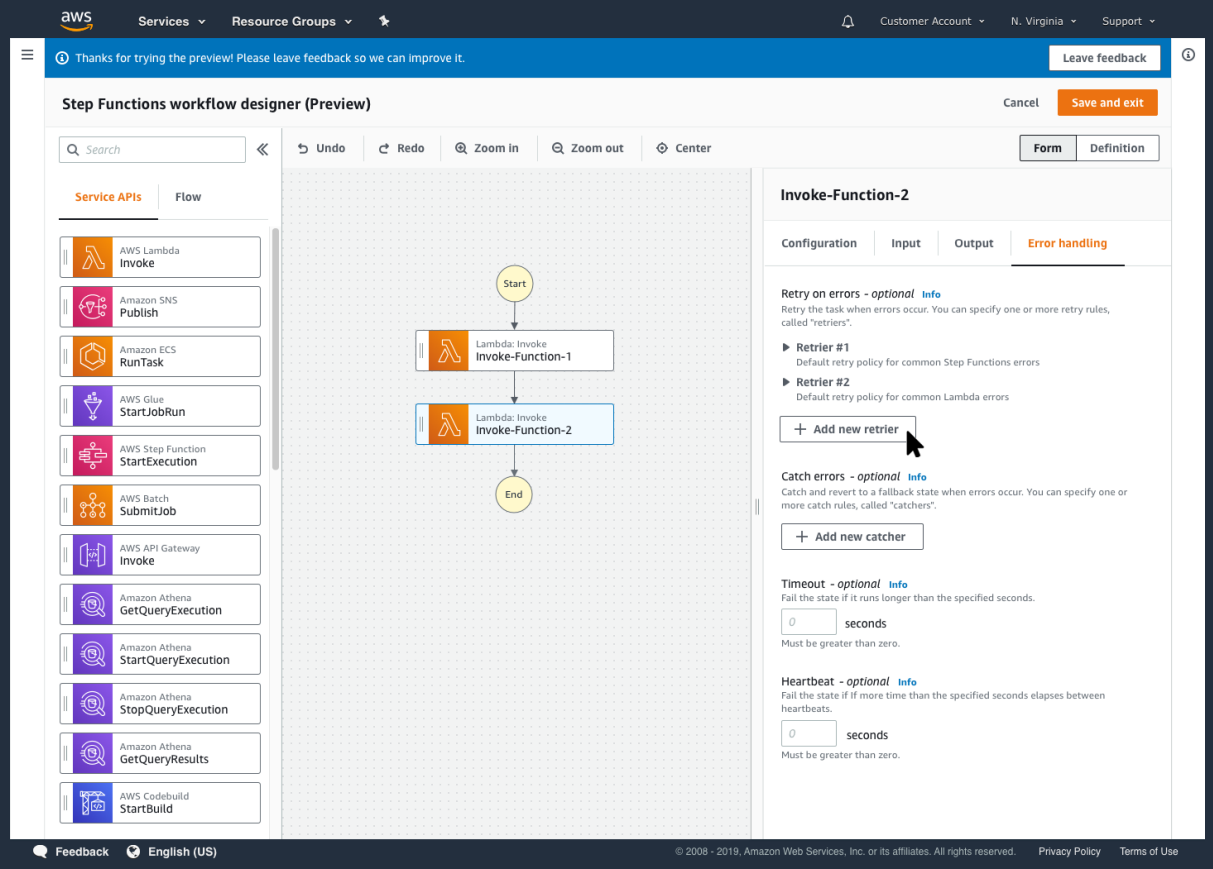


Step Functions

The **workflows** you build with Step Functions are called **state machines**, and **each step** of your workflow is called a **state**

When you execute your state machine, **each move** from one state to the next is called a **state transition**

You can **reuse components**, easily edit the sequence of steps, or swap out the code called by task states as your needs change



Step Functions Workflow Studio

Visual workflows

Define

JSON – Amazon States Language

Code

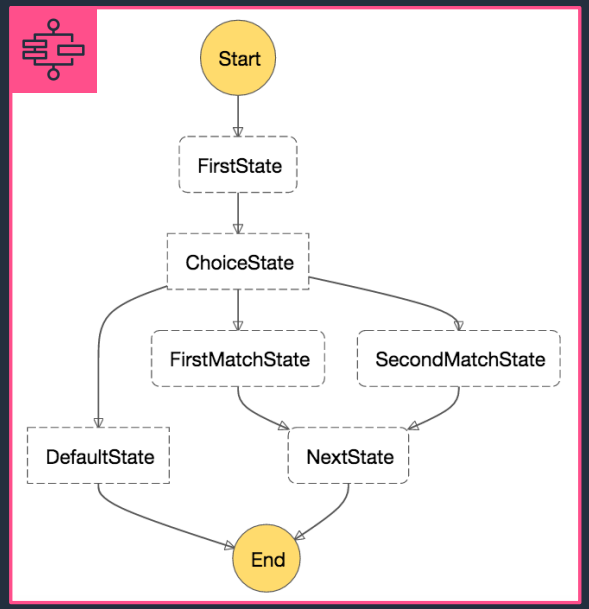
```
{
  "Comment": "An AWL example using a choice state.",
  "StartAt": "FirstState",
  "States": {
    "FirstState": {
      "Type": "Task",
      "Resource": "arn:aws:lambda:REGION:ACCOUNT_ID:function:FUNCTION_NAME",
      "Next": "ChoiceState"
    },
    "ChoiceState": {
      "Type" : "Choice",
      "Choices": [
        {
```

CDK
SAM
TypeScript, JavaScript, Python, Java, C#

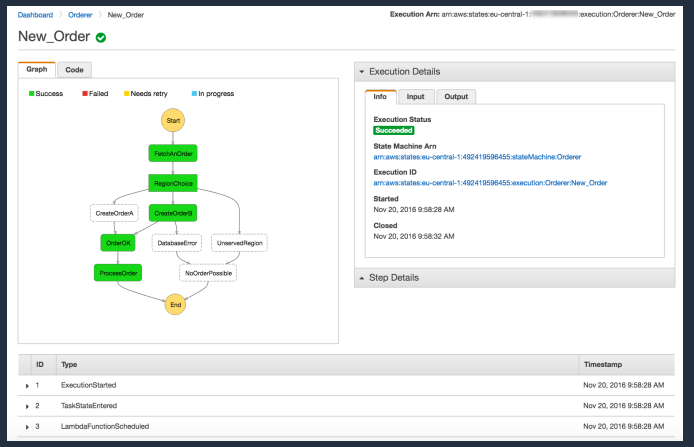
Data science SDK

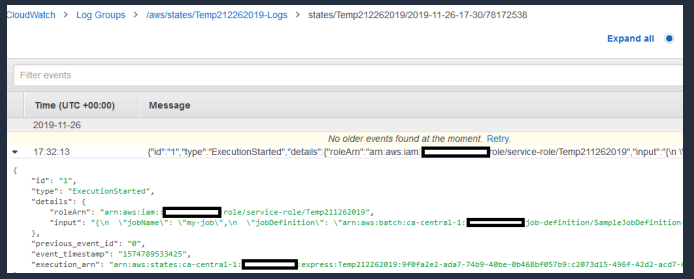
Python, Jupiter

Visualize



Execute & monitor





Step Functions integration types

Custom integrations

Customized to simplify the usage of 17 AWS services

Supported integration patterns

- **Request Response**
- **Wait for a Callback (.waitForTaskToken)**
- **Run a Job (.sync)**

AWS SDK integrations

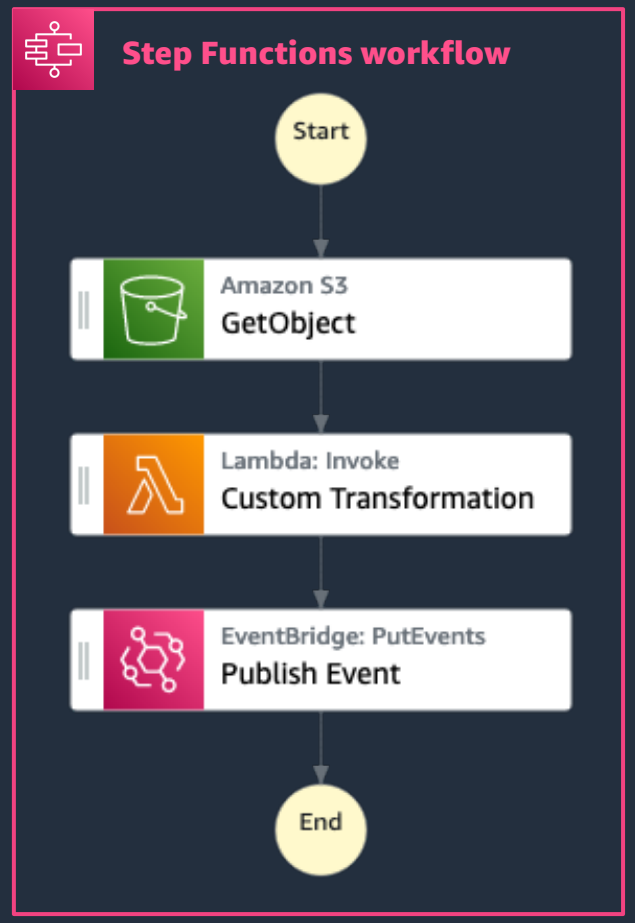
Call 200+ AWS services directly (10,000+ API actions)

Supported integration patterns

- **Request Response**
- **Wait for a Callback (.waitForTaskToken)**

Request Response

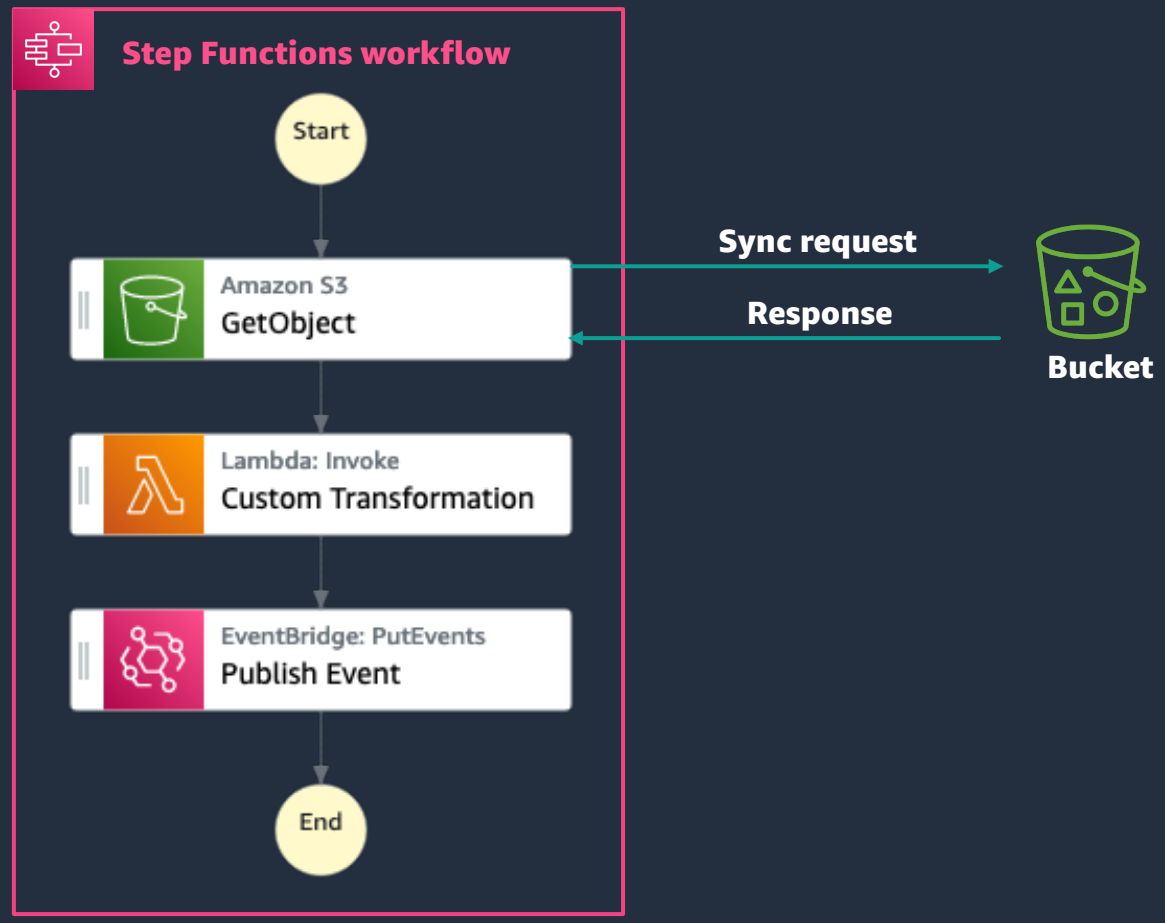
SERVICE INTEGRATION PATTERN



- Amazon EMR
- Lambda
- Amazon SQS
- AWS Glue
- Amazon ECS/ AWS Fargate
- Amazon SNS
- Amazon Athena
- Amazon EKS
- Step Functions
- SageMaker
- AWS Batch
- EventBridge
- Amazon DynamoDB
- AWS SDK
- API Gateway

Request Response

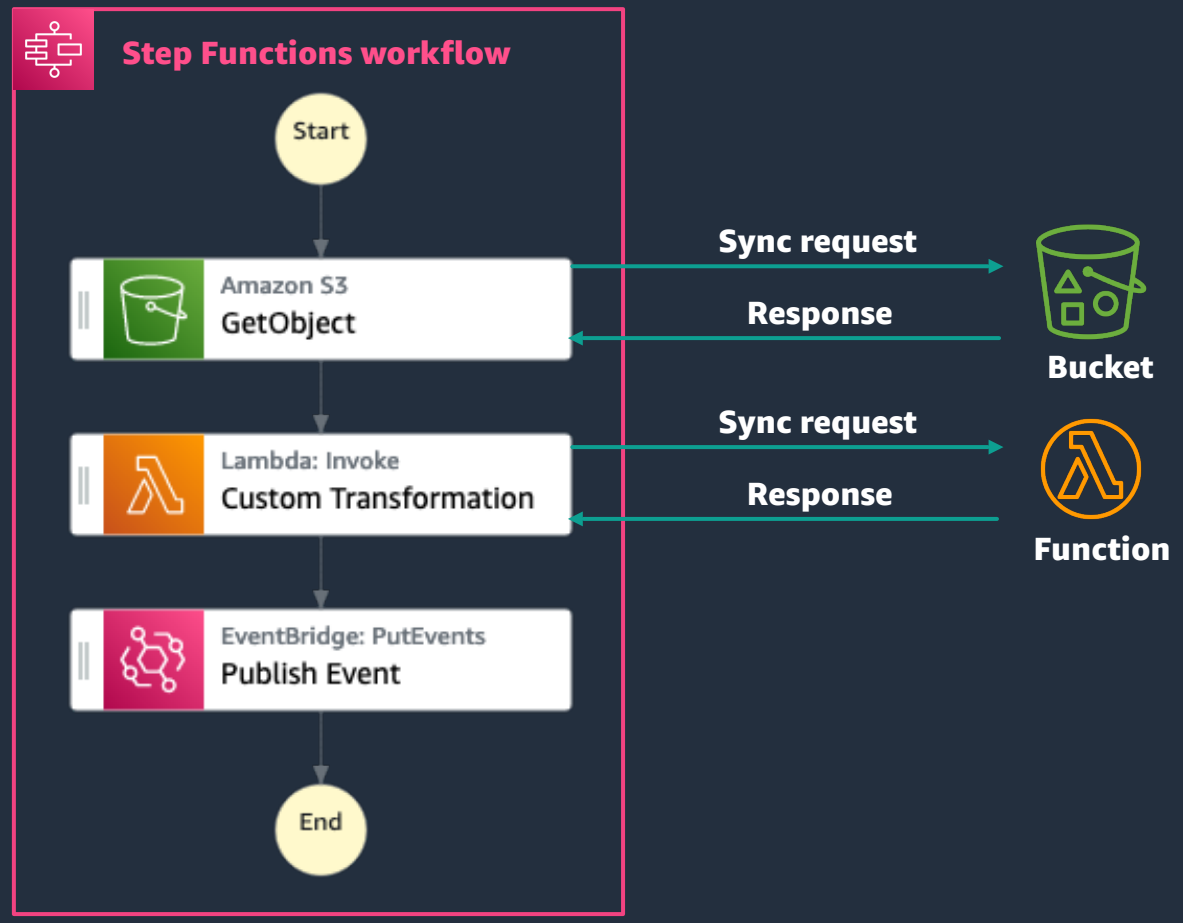
SERVICE INTEGRATION PATTERN



- Amazon EMR
- Lambda
- Amazon SQS
- AWS Glue
- Amazon ECS/ AWS Fargate
- Amazon SNS
- Amazon Athena
- Amazon EKS
- Step Functions
- SageMaker
- AWS Batch
- EventBridge
- Amazon DynamoDB
- AWS SDK
- API Gateway

Request Response

SERVICE INTEGRATION PATTERN



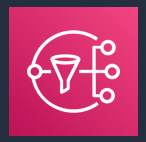

Amazon EMR


Lambda


Amazon SQS


AWS Glue


Amazon ECS/ AWS Fargate


Amazon SNS


Amazon Athena


Amazon EKS


Step Functions


SageMaker


AWS Batch


EventBridge

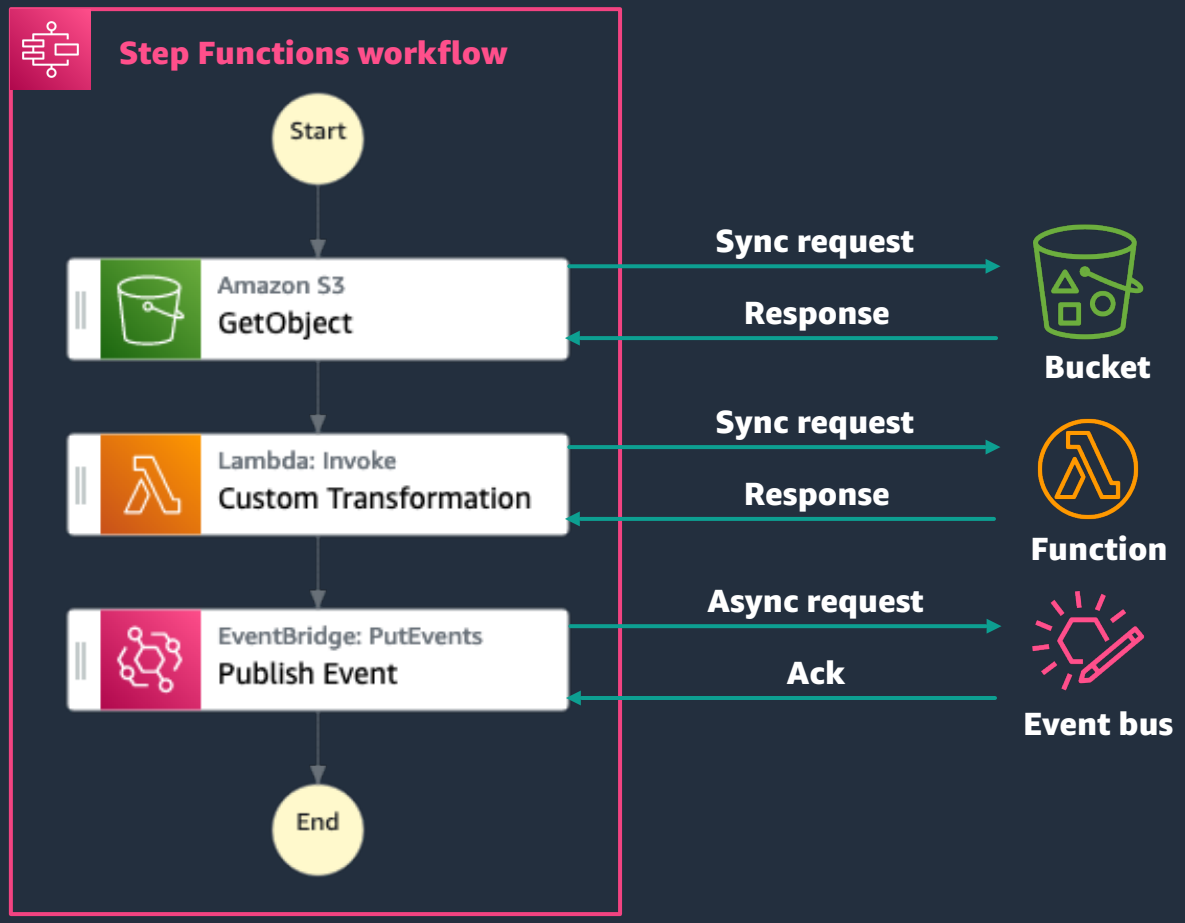

Amazon DynamoDB

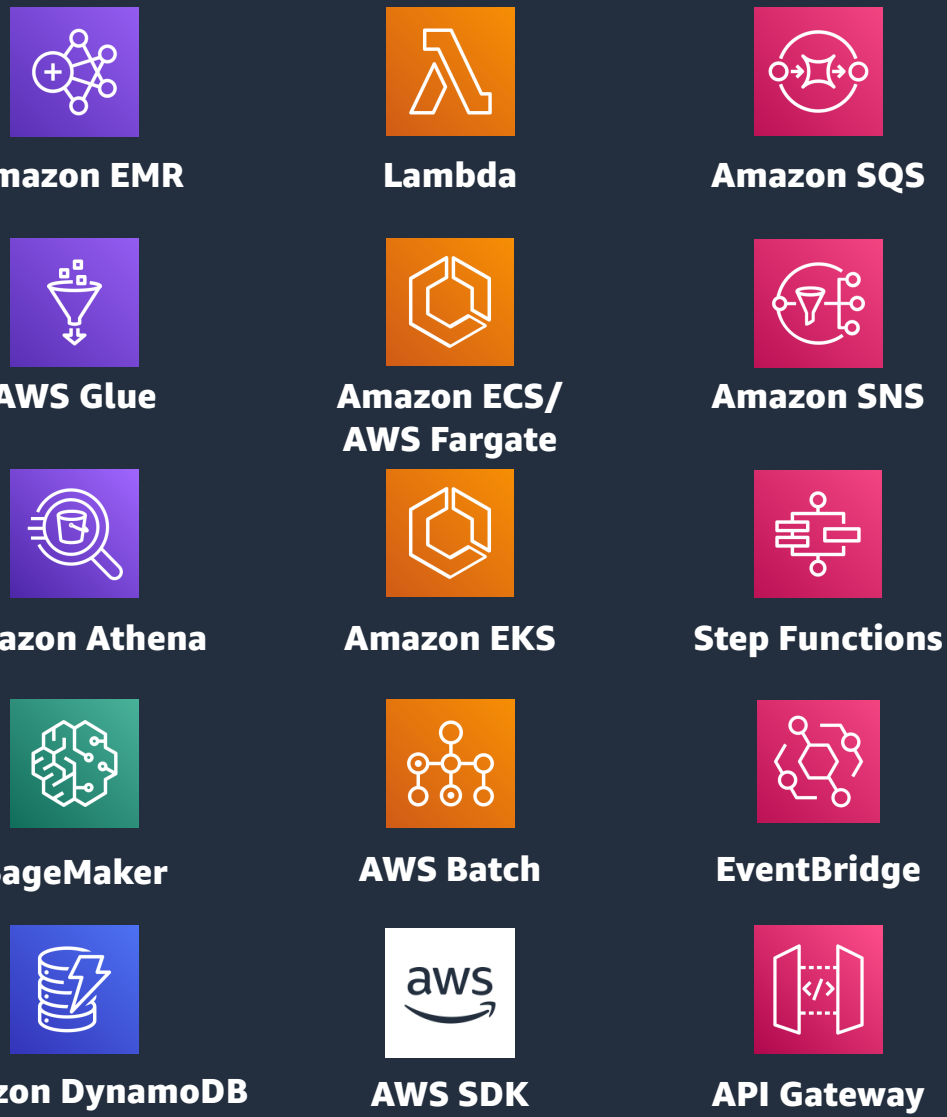
AWS SDK


API Gateway

Request Response

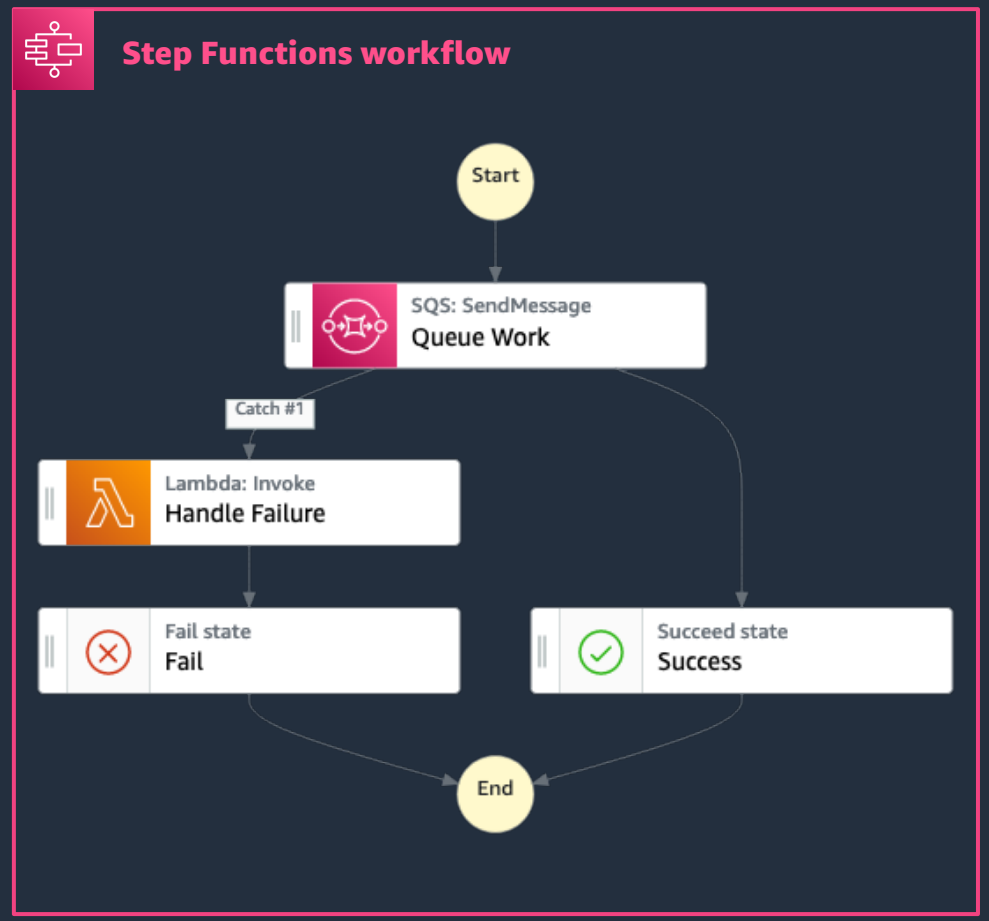
SERVICE INTEGRATION PATTERN





Wait for a Callback (.waitForTaskToken)

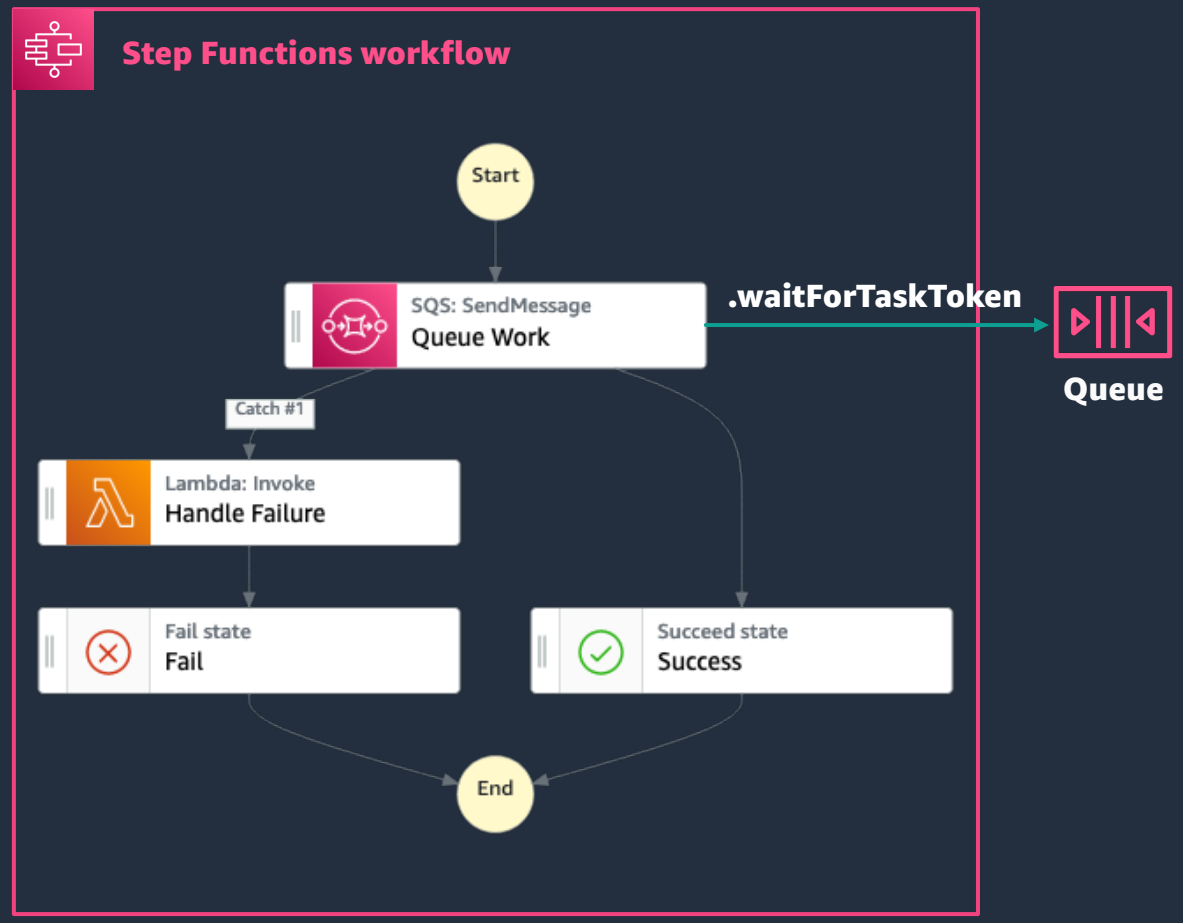
SERVICE INTEGRATION PATTERN

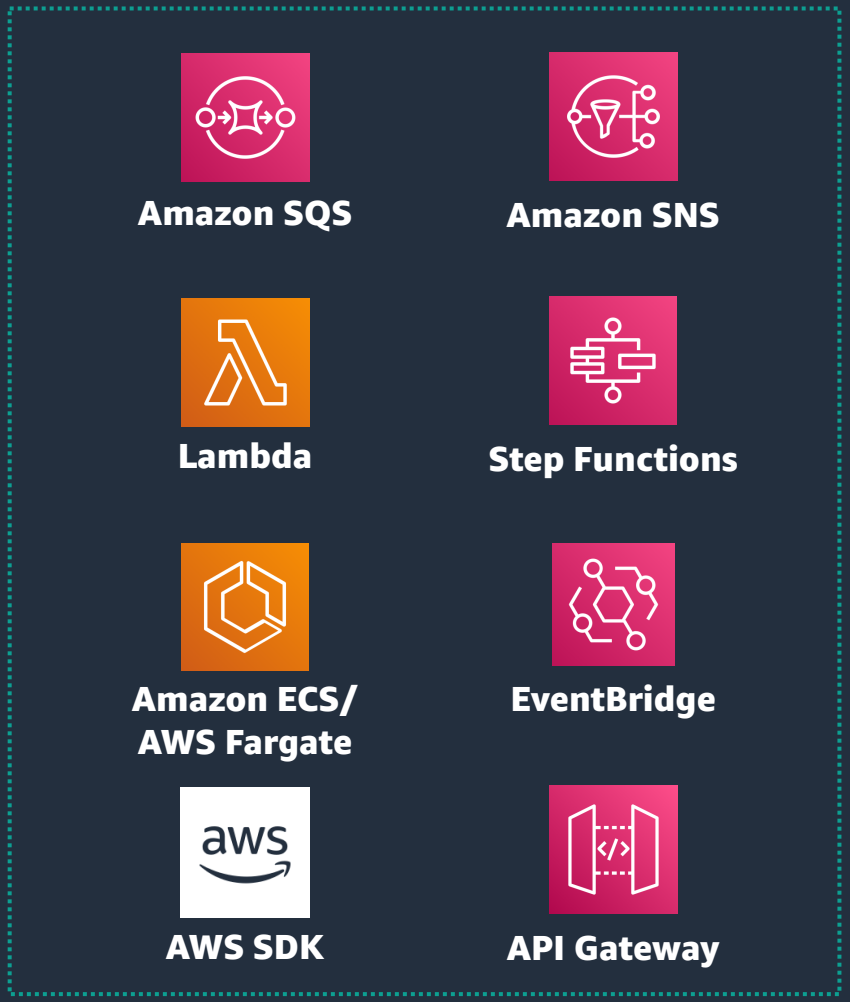


- Amazon SQS
- Amazon SNS
- Lambda
- Step Functions
- Amazon ECS/ AWS Fargate
- EventBridge
- AWS SDK
- API Gateway

Wait for a Callback (.waitForTaskToken)

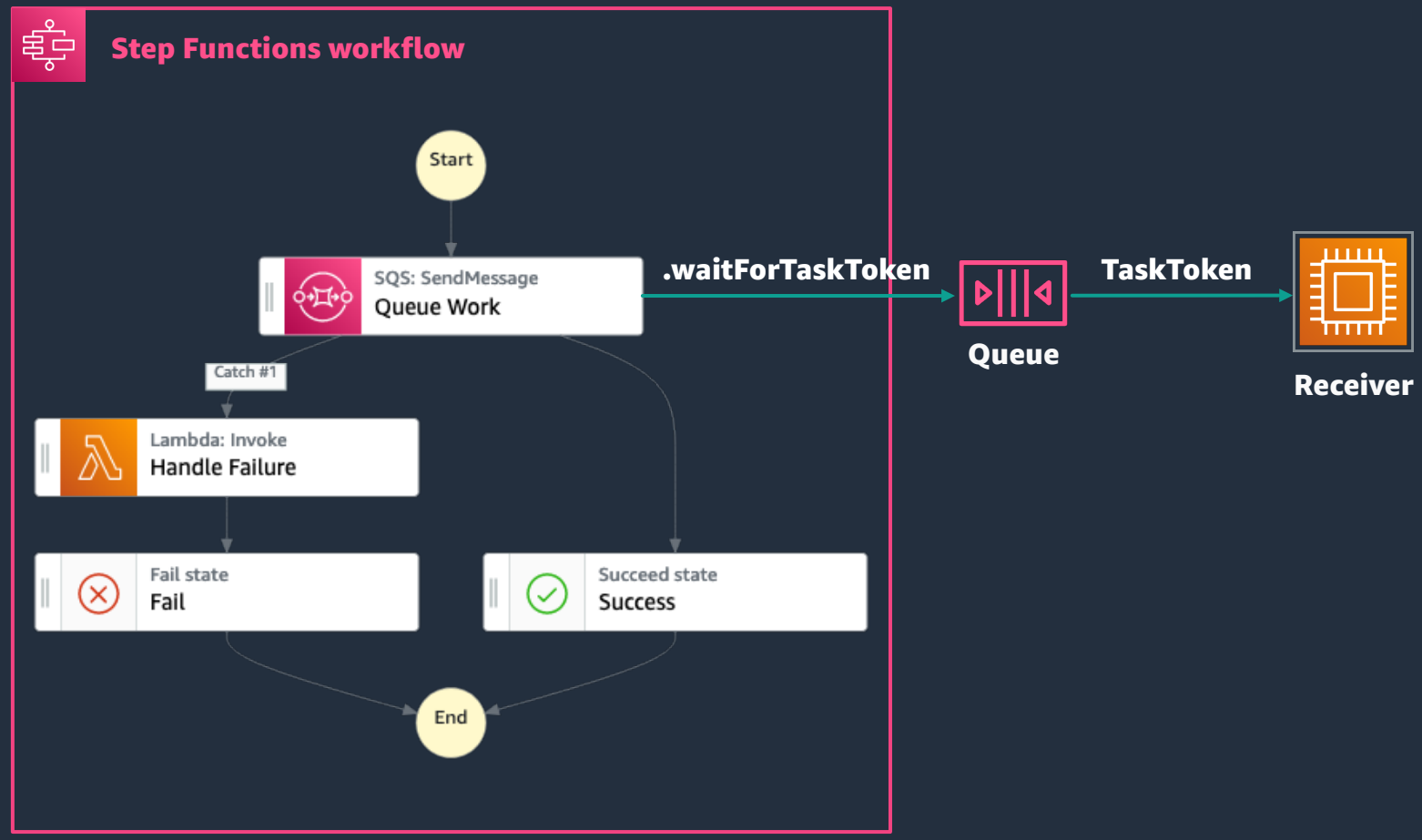
SERVICE INTEGRATION PATTERN





Wait for a Callback (.waitForTaskToken)

SERVICE INTEGRATION PATTERN



Queue

.waitForTaskToken

TaskToken

Receiver

Amazon SQS

Amazon SNS

Lambda

Step Functions

Amazon ECS/
AWS Fargate

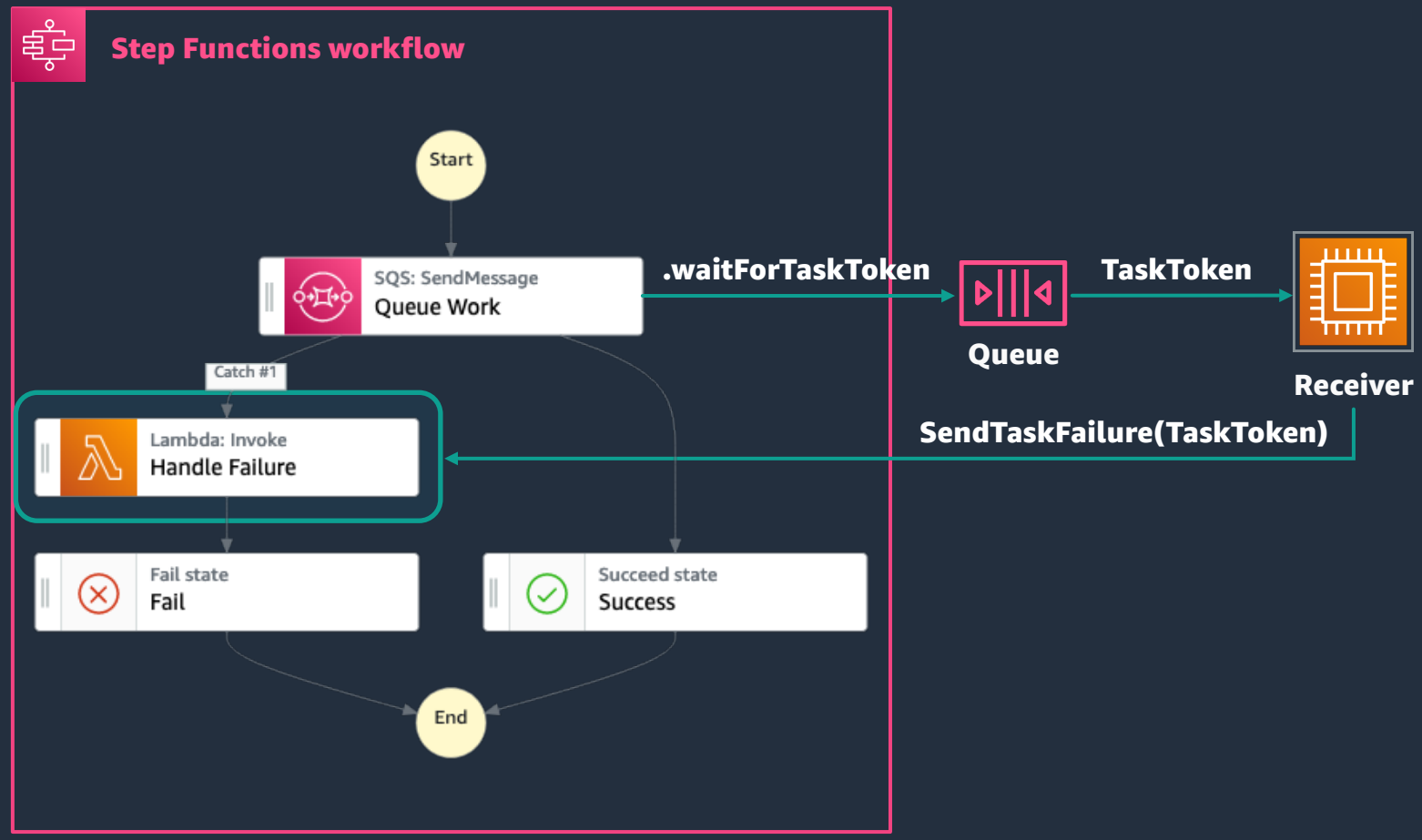
EventBridge

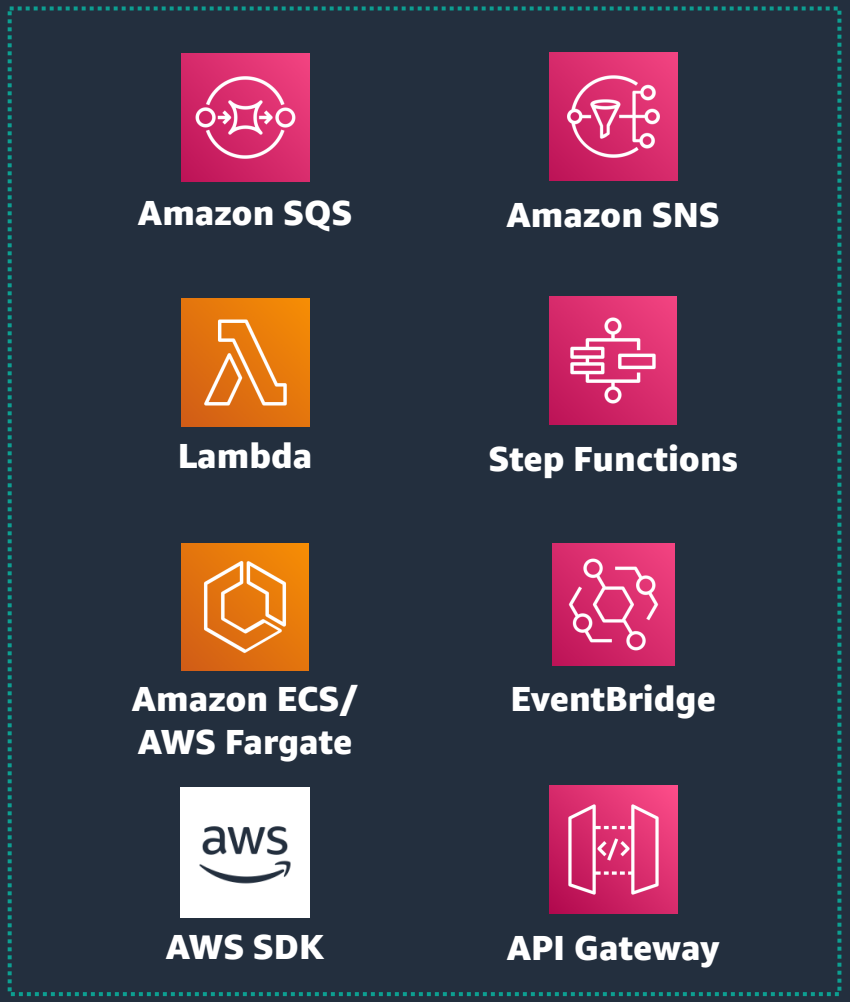
AWS SDK

API Gateway

Wait for a Callback (.waitForTaskToken)

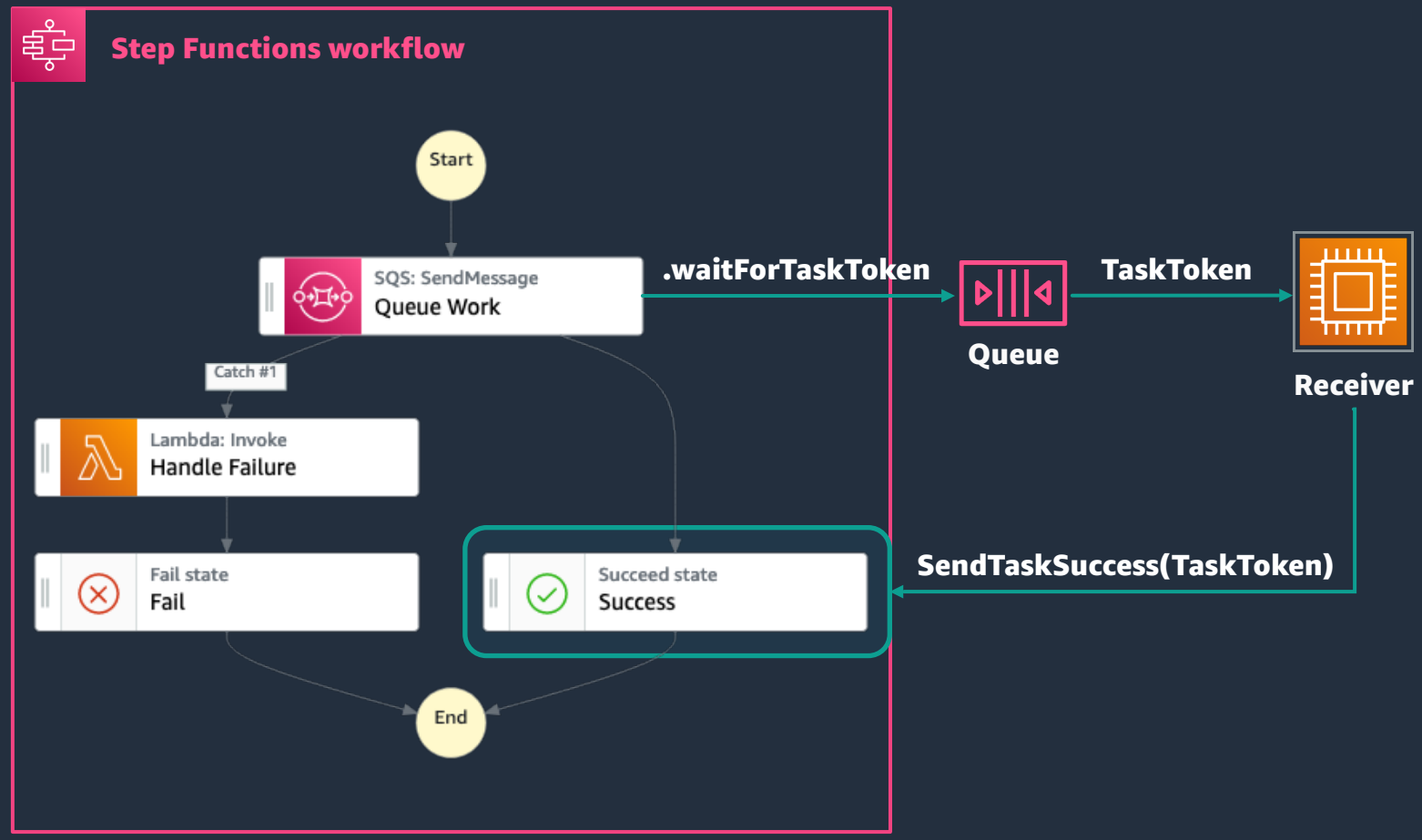
SERVICE INTEGRATION PATTERN

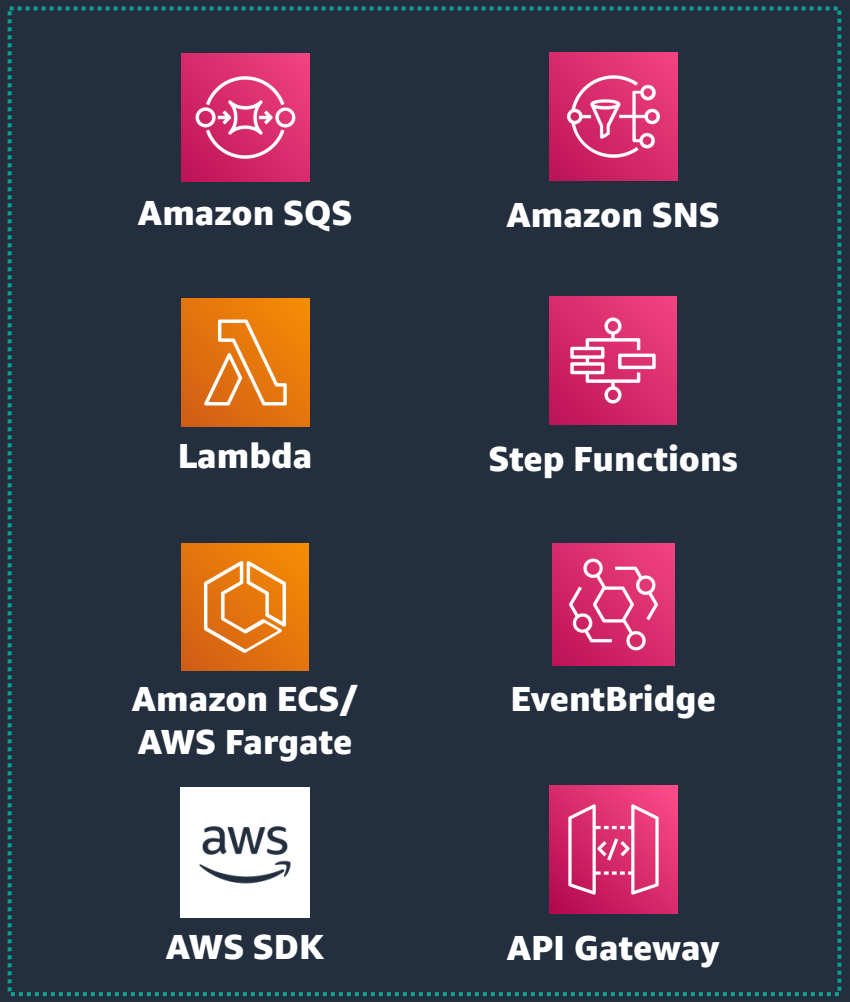




Wait for a Callback (.waitForTaskToken)

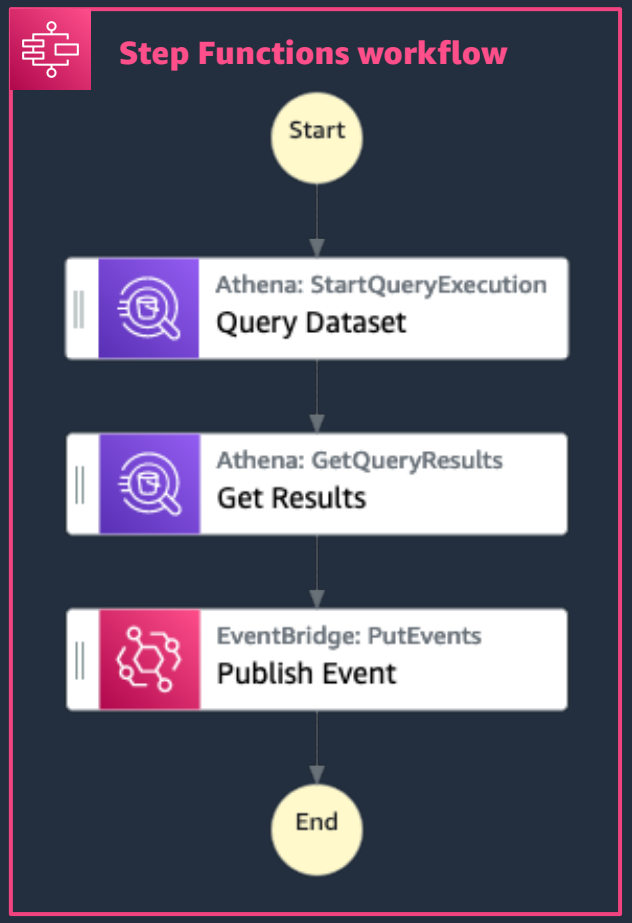
SERVICE INTEGRATION PATTERN





Run a Job (.sync)

SERVICE INTEGRATION PATTERN



Amazon EMR

AWS Glue

Amazon ECS/
AWS Fargate

Amazon Athena

Amazon EKS

Step Functions

SageMaker

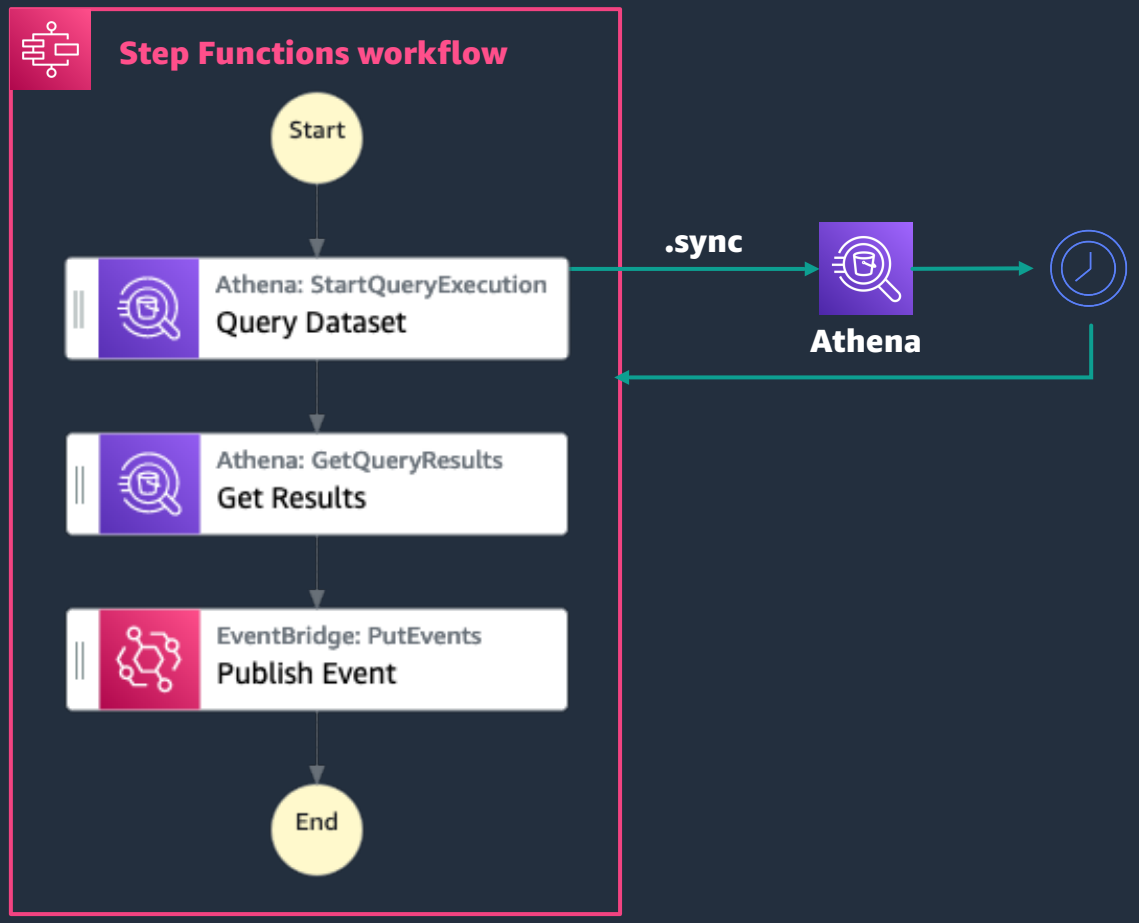
AWS Batch

Amazon DynamoDB

API Gateway

Run a Job (.sync)

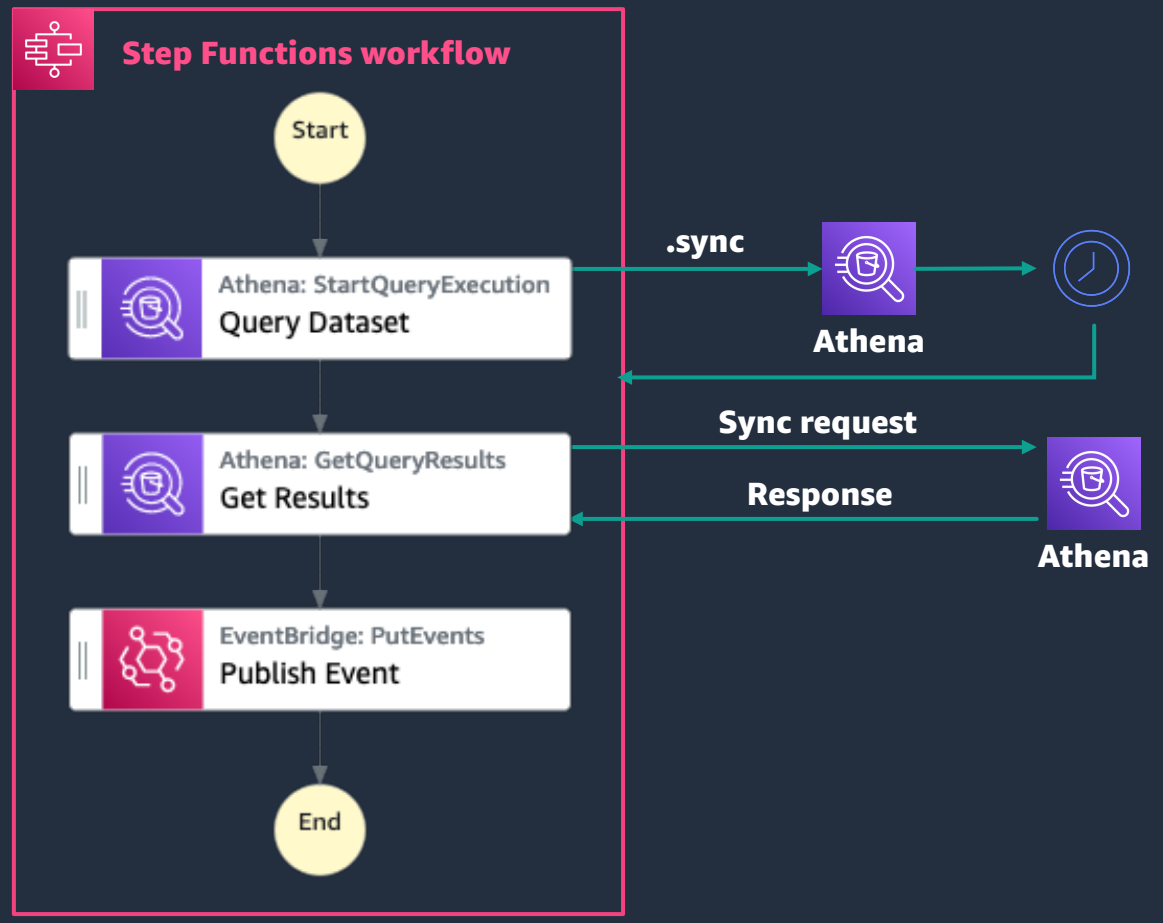
SERVICE INTEGRATION PATTERN



- Amazon EMR
- AWS Glue
- Amazon ECS/ AWS Fargate
- Amazon Athena
- Amazon EKS
- Step Functions
- SageMaker
- AWS Batch
- Amazon DynamoDB
- API Gateway

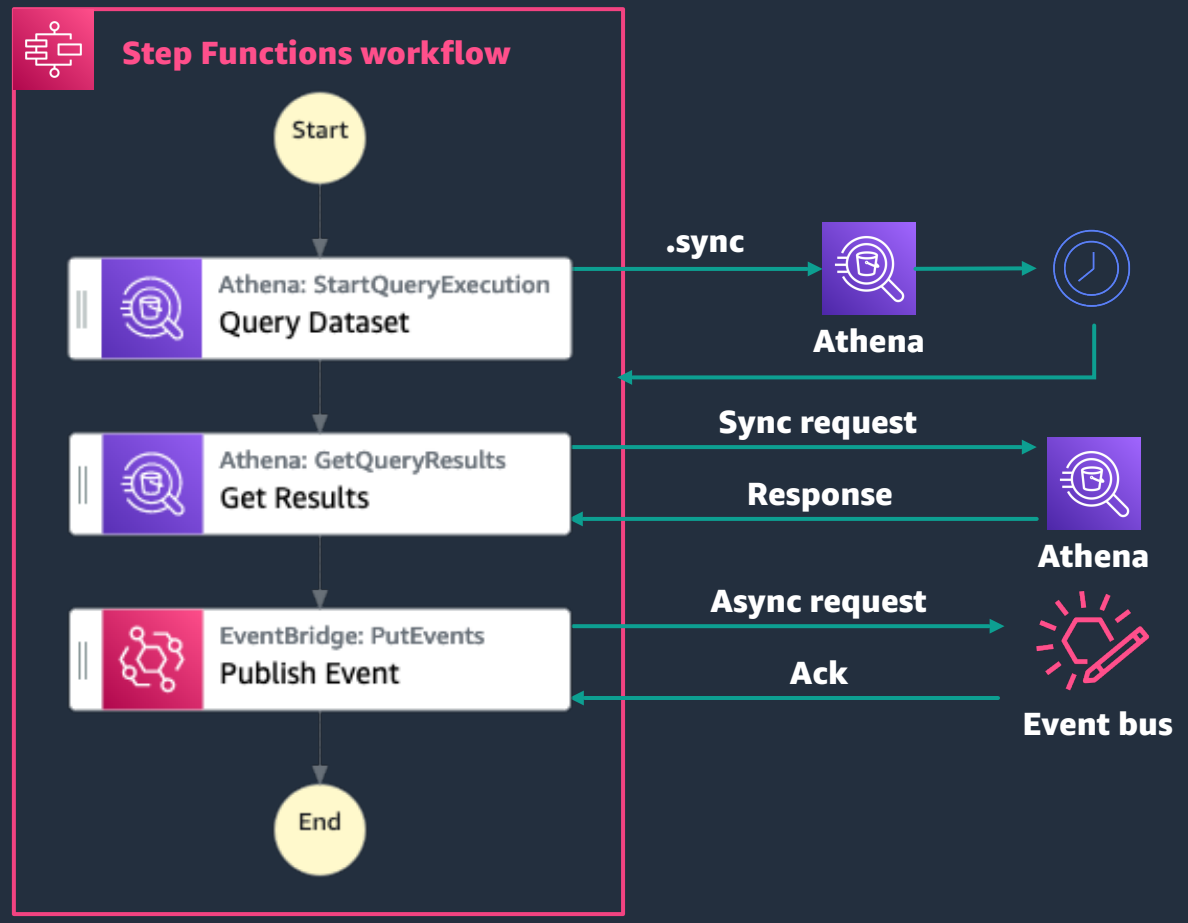
Run a Job (.sync)

SERVICE INTEGRATION PATTERN



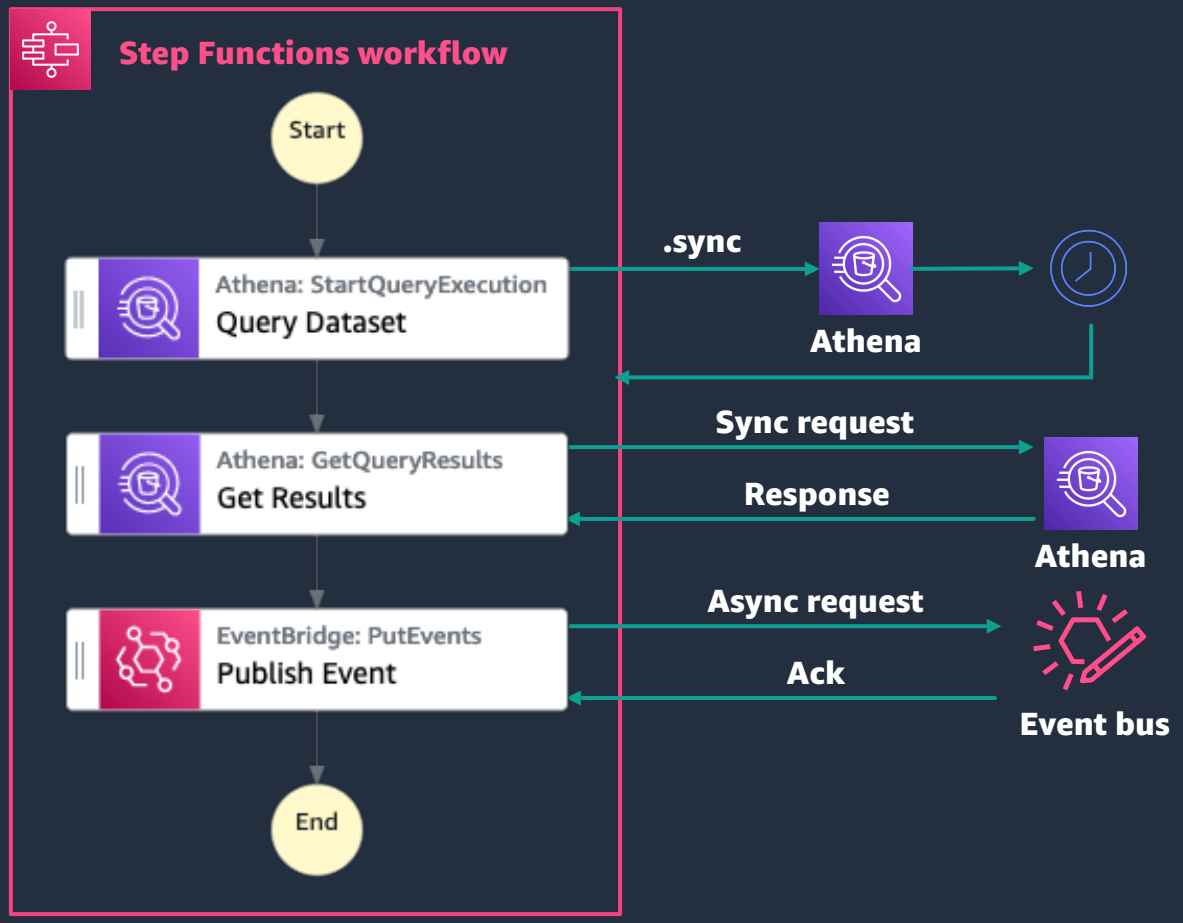
Run a Job (.sync)

SERVICE INTEGRATION PATTERN



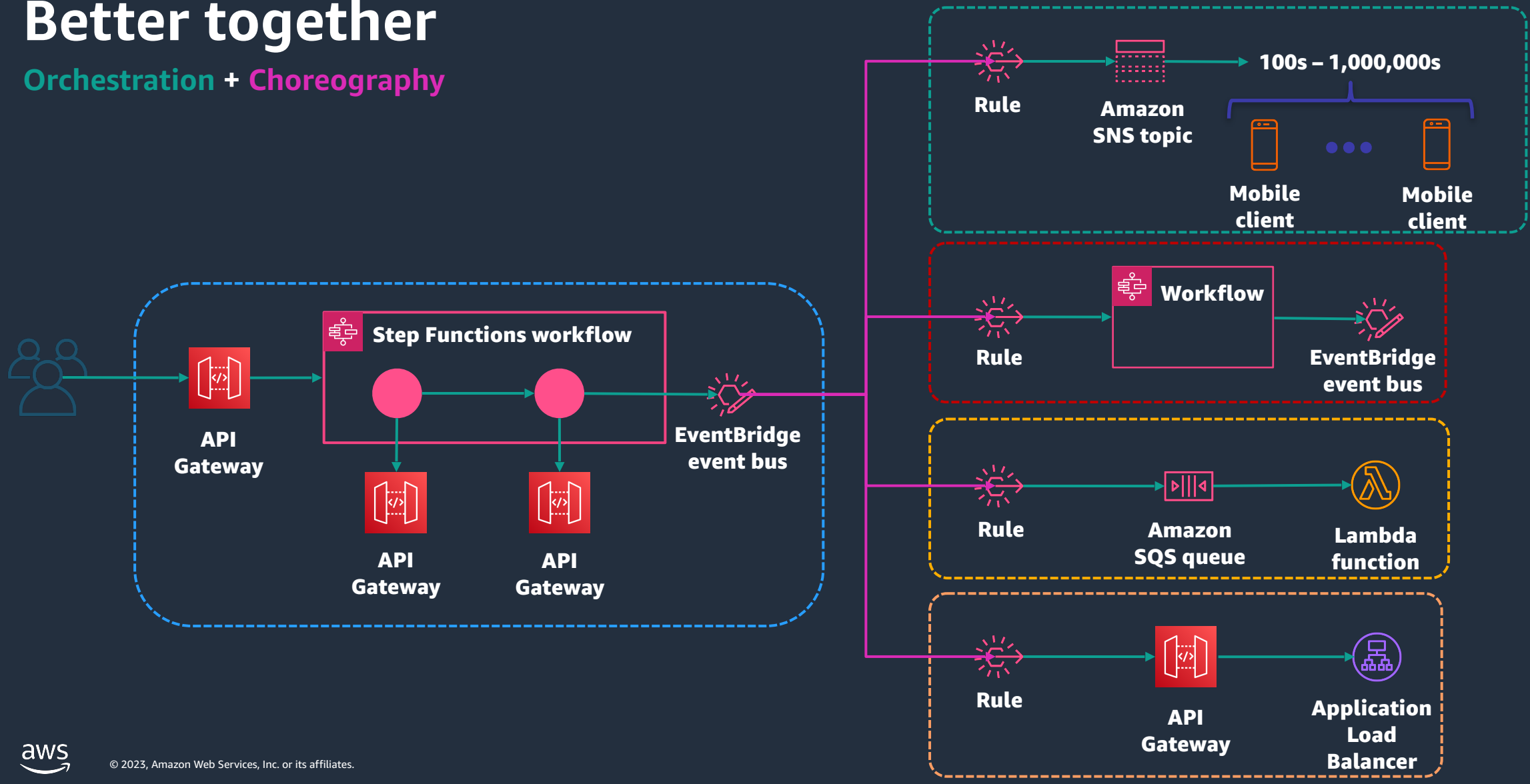
Run a Job (.sync)

SERVICE INTEGRATION PATTERN



Better together

Orchestration + Choreography



Maintaining idempotency

What is idempotence, idempotency, idempotent?

The mathematical definition

Operations that can be applied multiple times without changing the result.

$$f(x) = x + 0 = x \quad \textit{or} \quad f(x) = x * 1 = x$$

What is idempotence, idempotency, idempotent?

The architectural definition

"A message that has the same effect whether it is received once or multiple times."

– "Enterprise Integration Patterns" (Hohpe, Woolf)

What is idempotence, idempotency, idempotent?

The real world definition

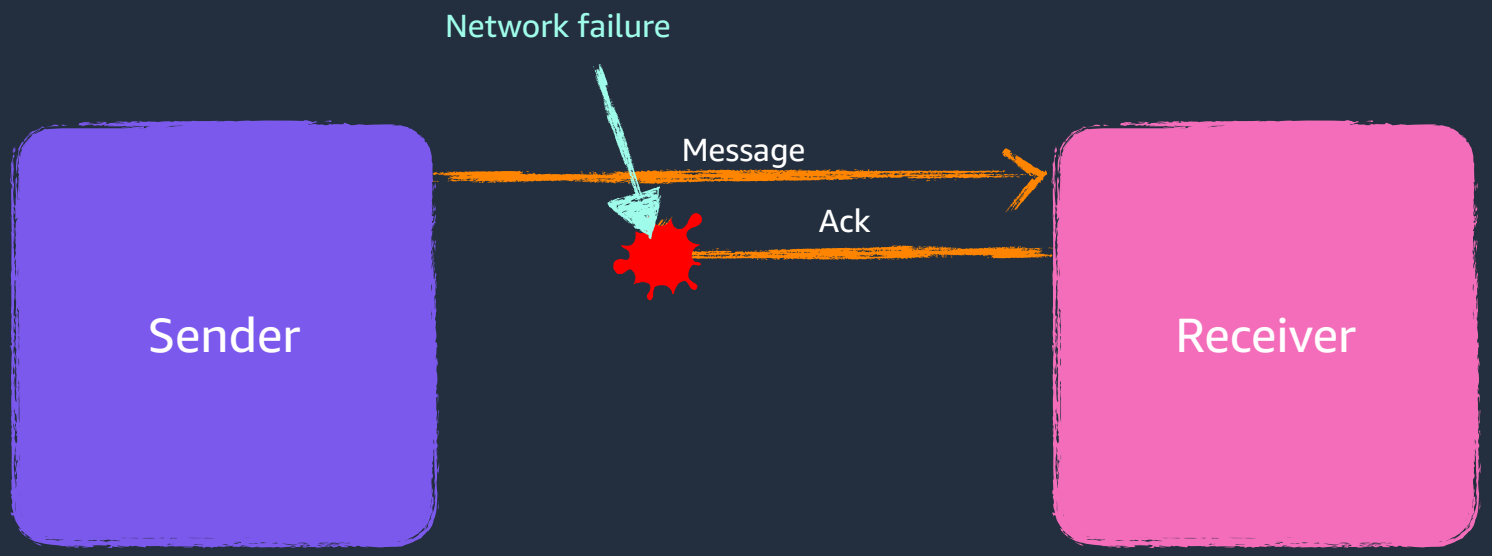
"Was my credit card charged twice?"

– Mom

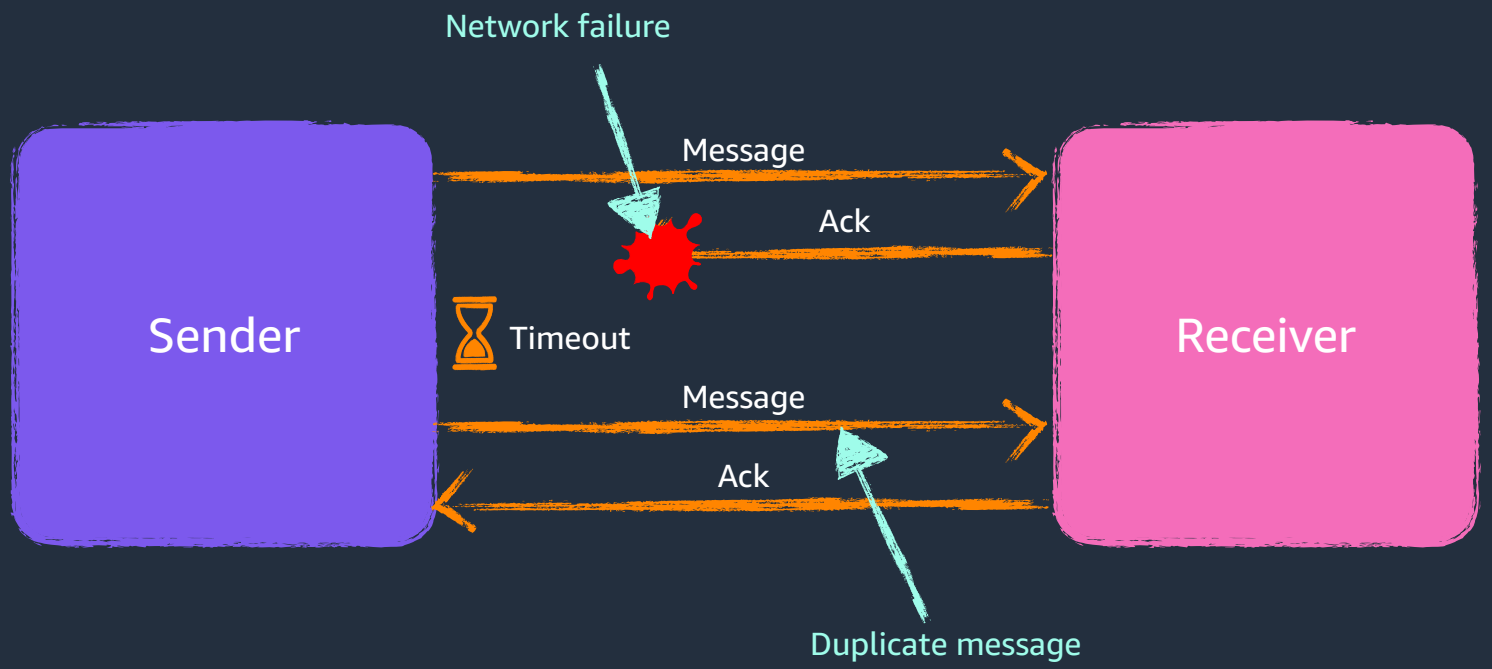
Where can duplicates occur?

Duplication caused by transmission issues

Duplication caused by transmission issues

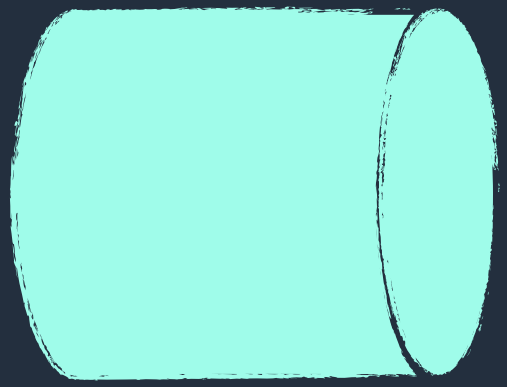


Duplication caused by transmission issues



Duplication caused by receiver issues

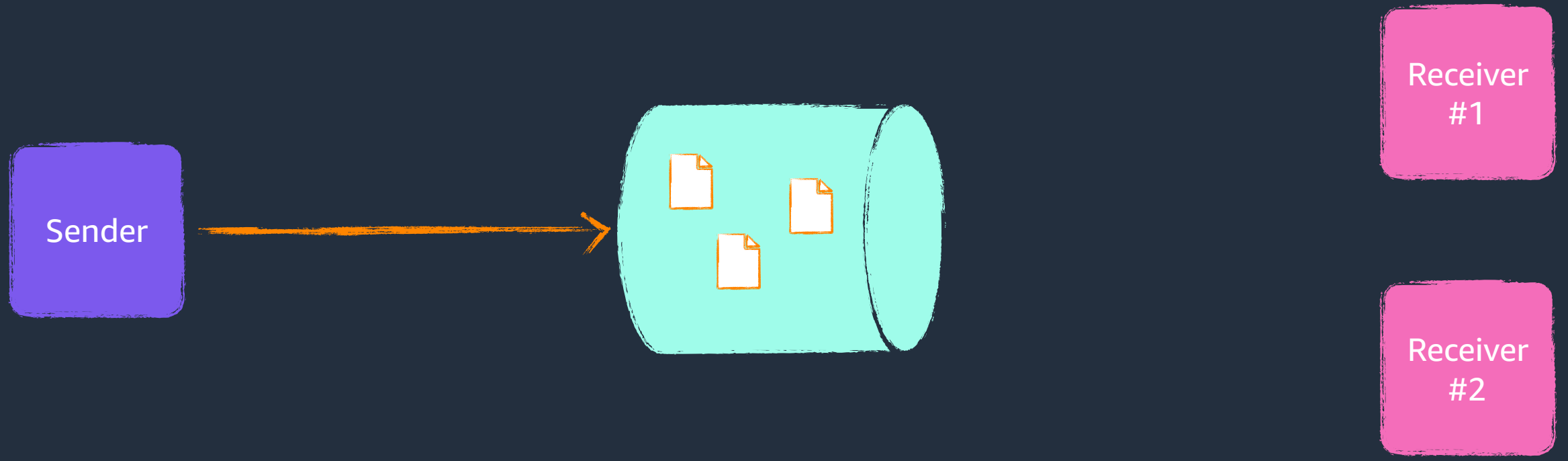
Sender



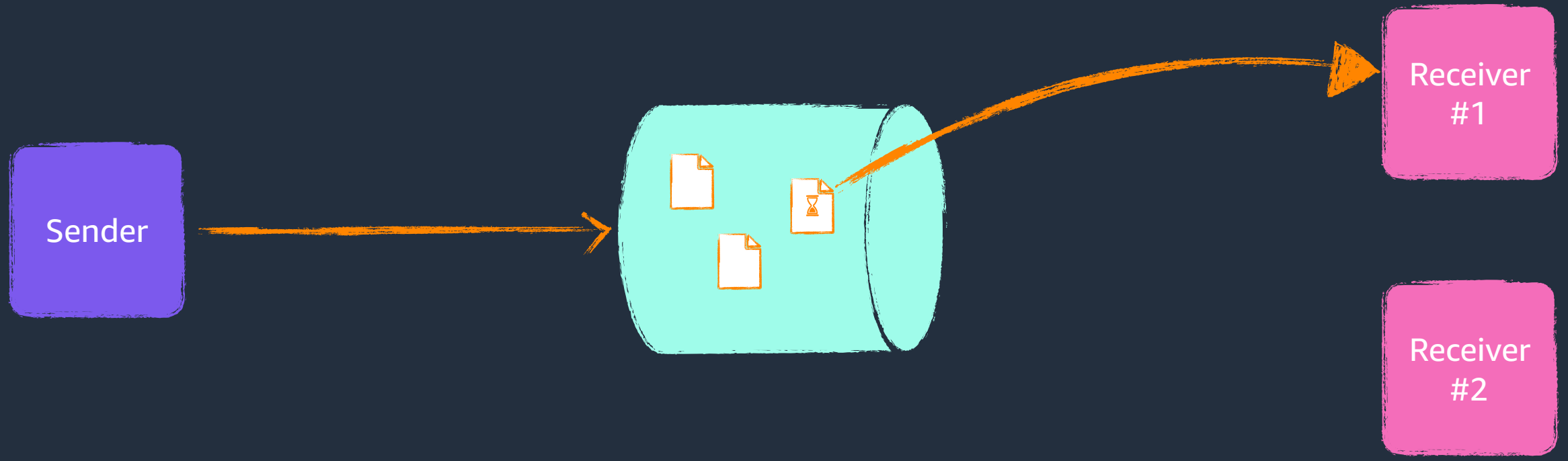
Receiver #1

Receiver #2

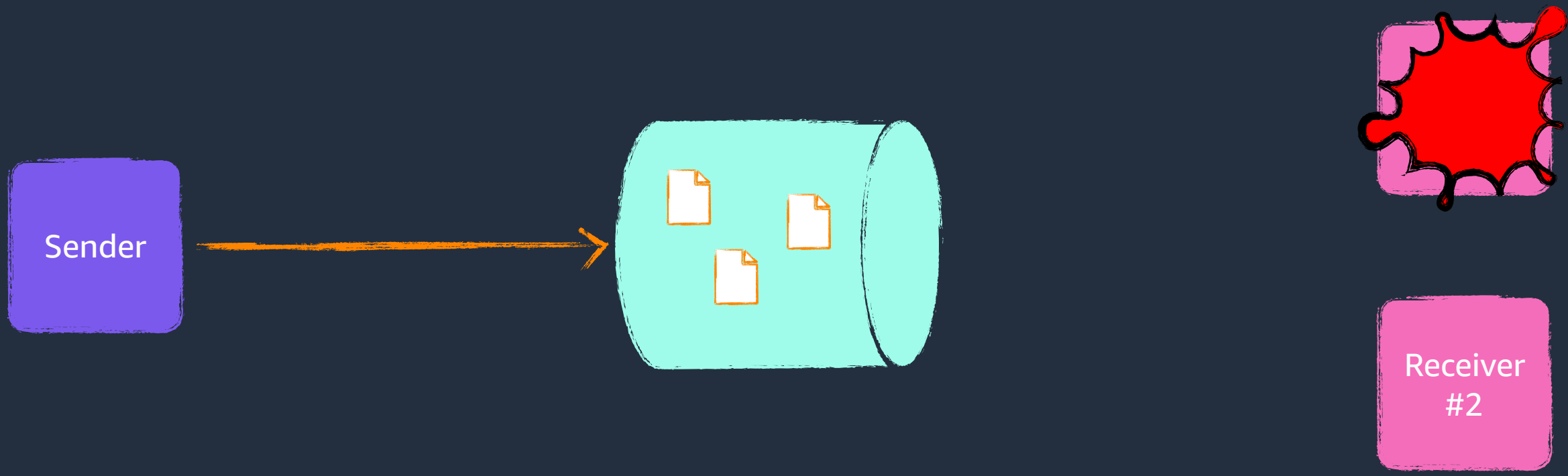
Duplication caused by receiver issues



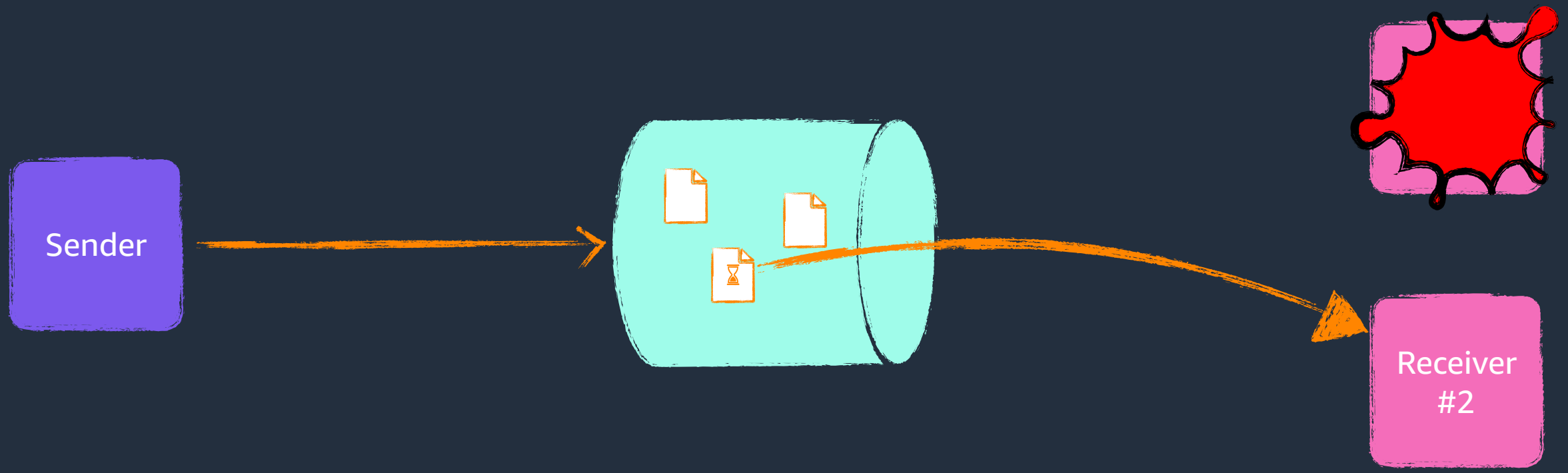
Duplication caused by receiver issues



Duplication caused by receiver issues

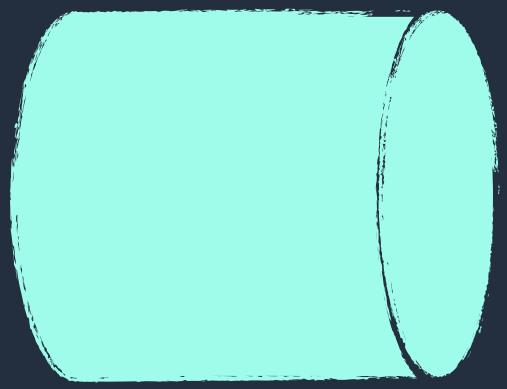


Duplication caused by receiver issues



Duplication caused by service issues

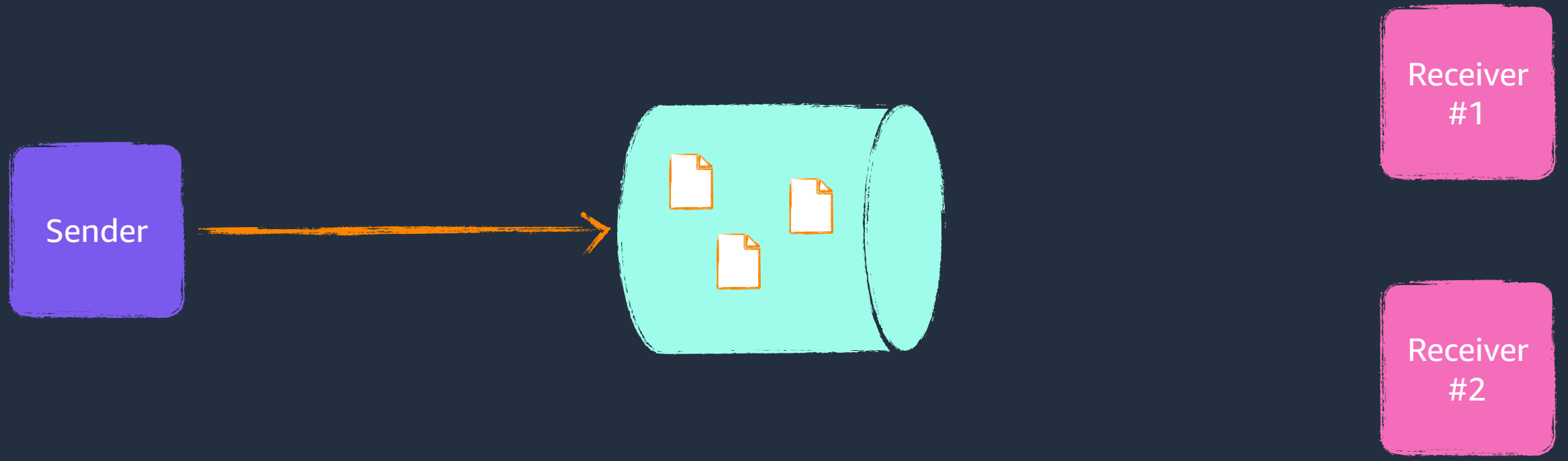
Sender



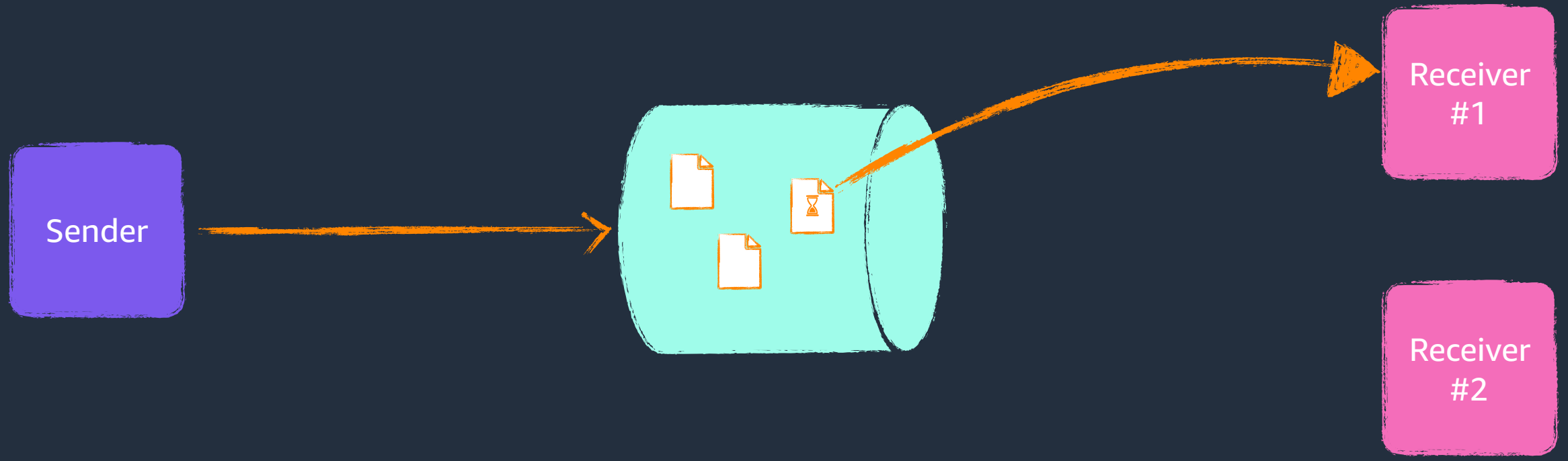
Receiver #1

Receiver #2

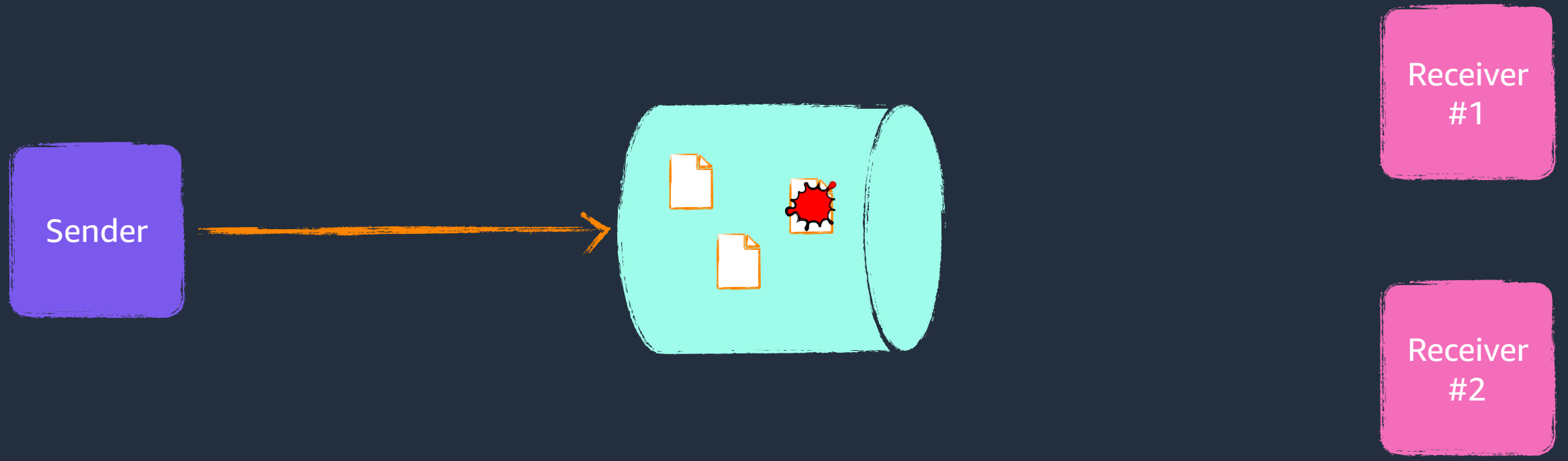
Duplication caused by service issues



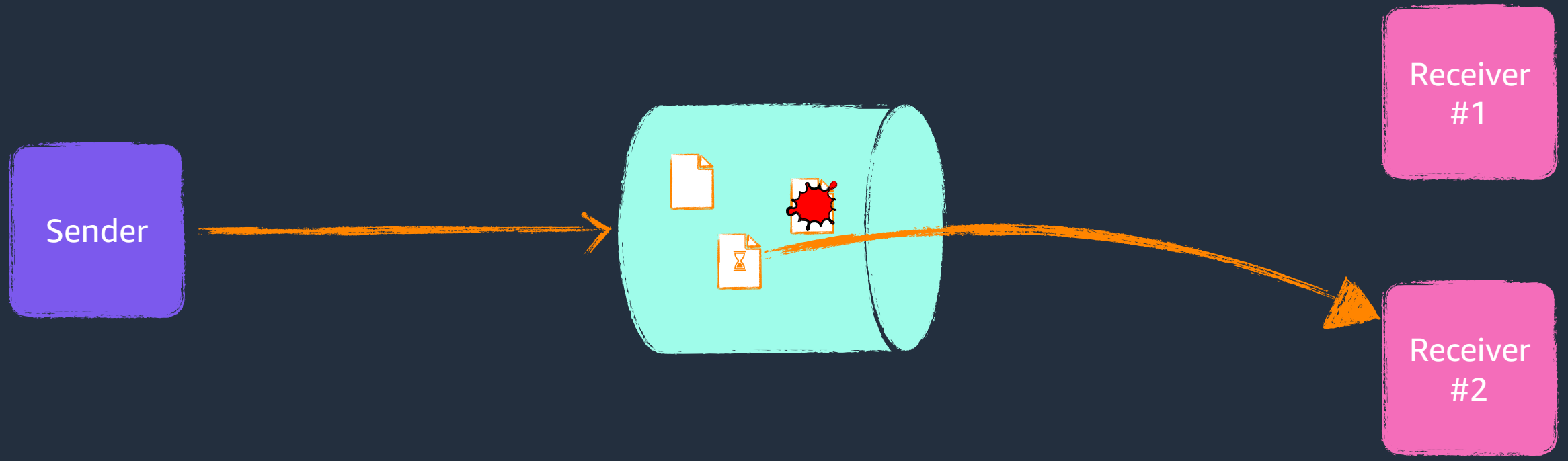
Duplication caused by service issues



Duplication caused by service issues



Duplication caused by service issues



Building idempotent code

Idempotency tokens

Idempotency tokens uniquely identify messages to enable receivers to avoid duplicate side effects

Well-behaved idempotency tokens:

Idempotency tokens

Idempotency tokens uniquely identify messages to enable receivers to avoid duplicate side effects

Well-behaved idempotency tokens:

- Are **generated by the client**

Idempotency tokens

Idempotency tokens uniquely identify messages to enable receivers to avoid duplicate side effects

Well-behaved idempotency tokens:

- Are **generated by the client**
- Are ***regenerated*** **by the client** on retry

Idempotency tokens

Idempotency tokens uniquely identify messages to enable receivers to avoid duplicate side effects

Well-behaved idempotency tokens:

- Are **generated by the client**
- Are ***regenerated* by the client** on retry
- Are unique per message (**UUID** is common)

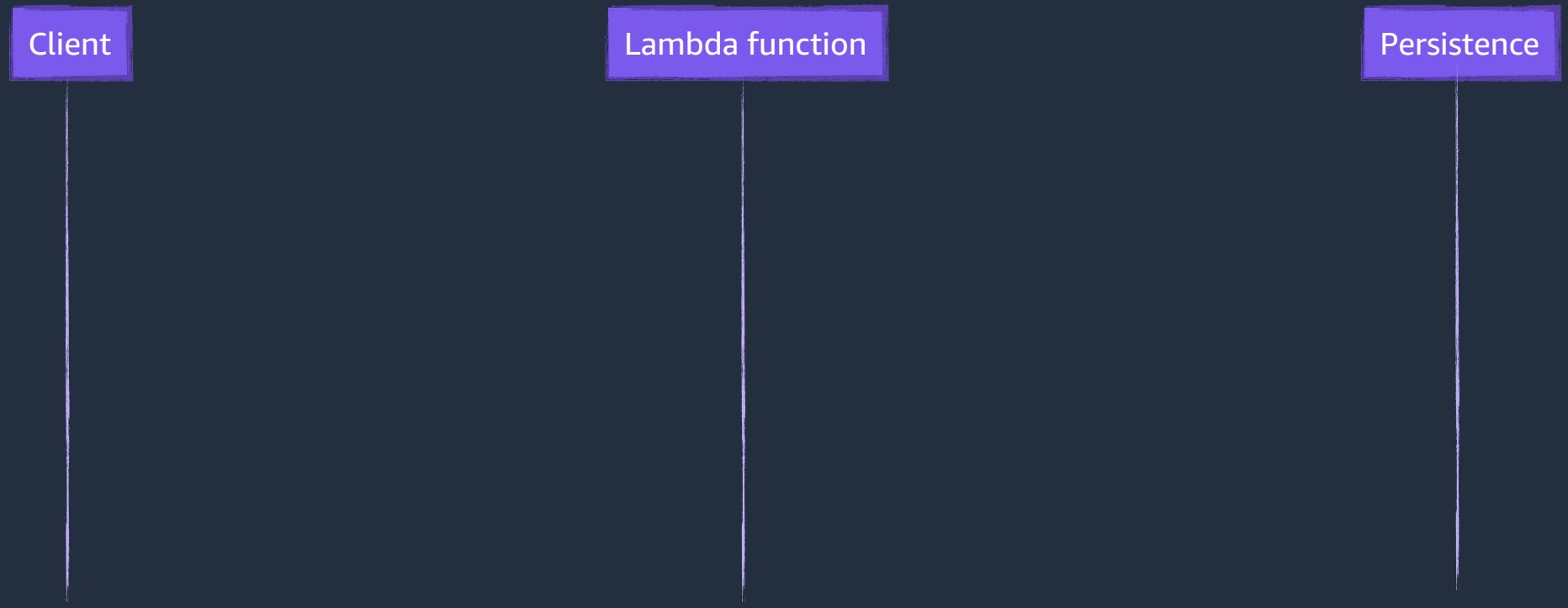
Idempotency tokens

Idempotency tokens uniquely identify messages to enable receivers to avoid duplicate side effects

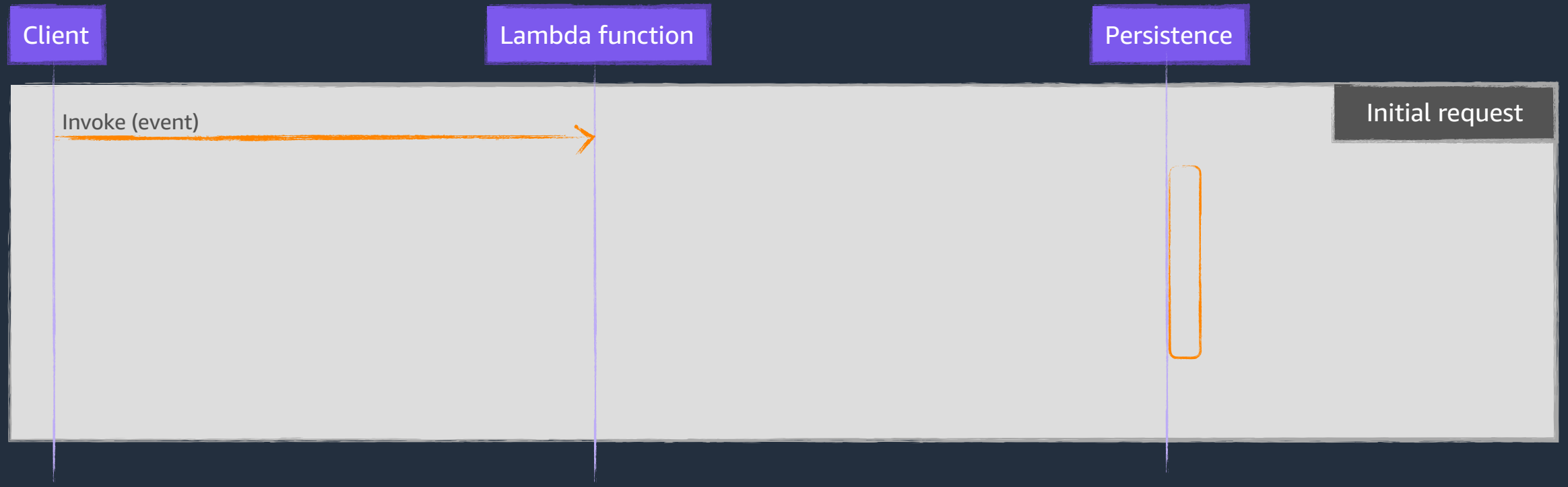
Well-behaved idempotency tokens:

- Are **generated by the client**
- Are ***regenerated* by the client** on retry
- Are unique per message (**UUID** is common)
- Use a **dedicated field**, separate from message content

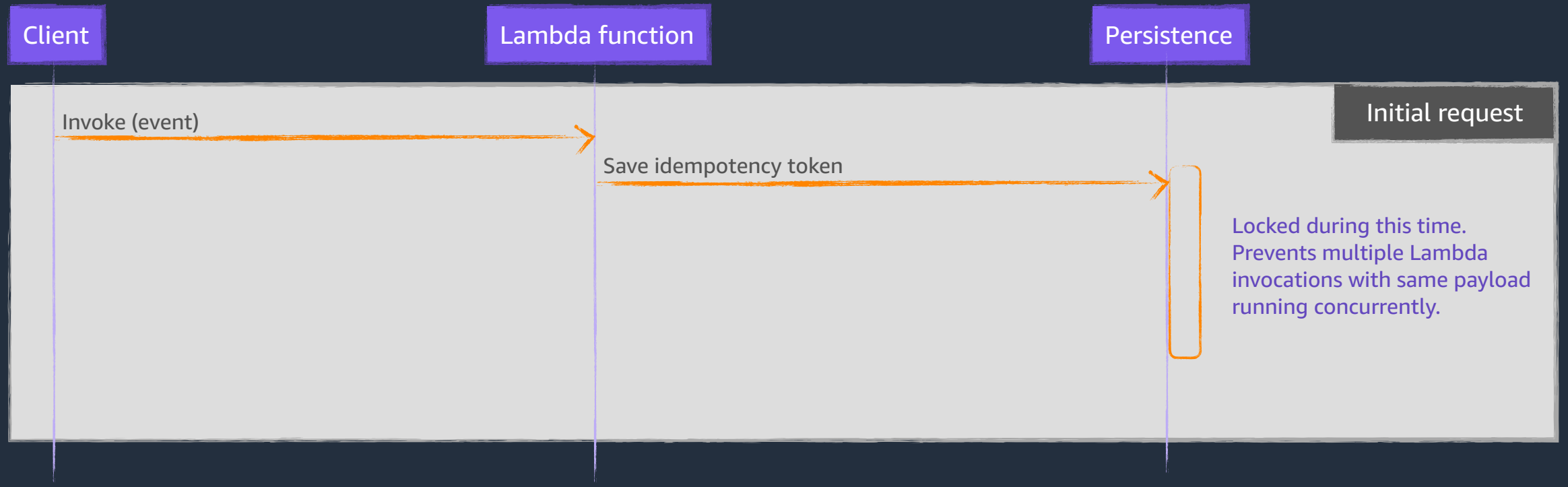
Idempotent request sequence



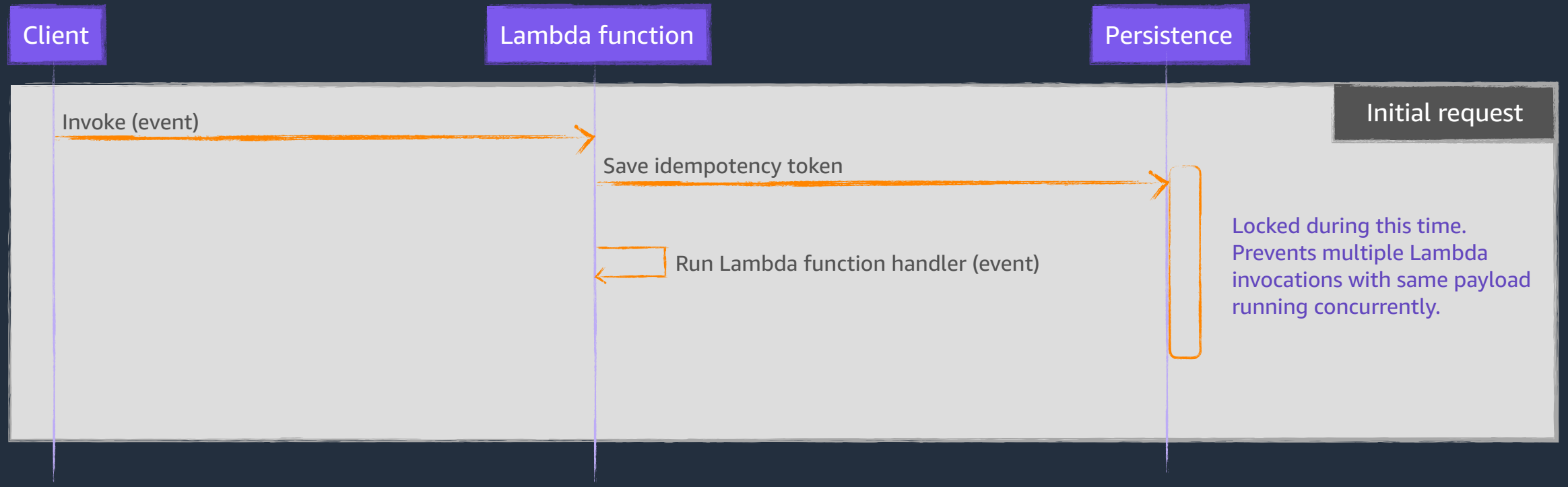
Idempotent request sequence



Idempotent request sequence



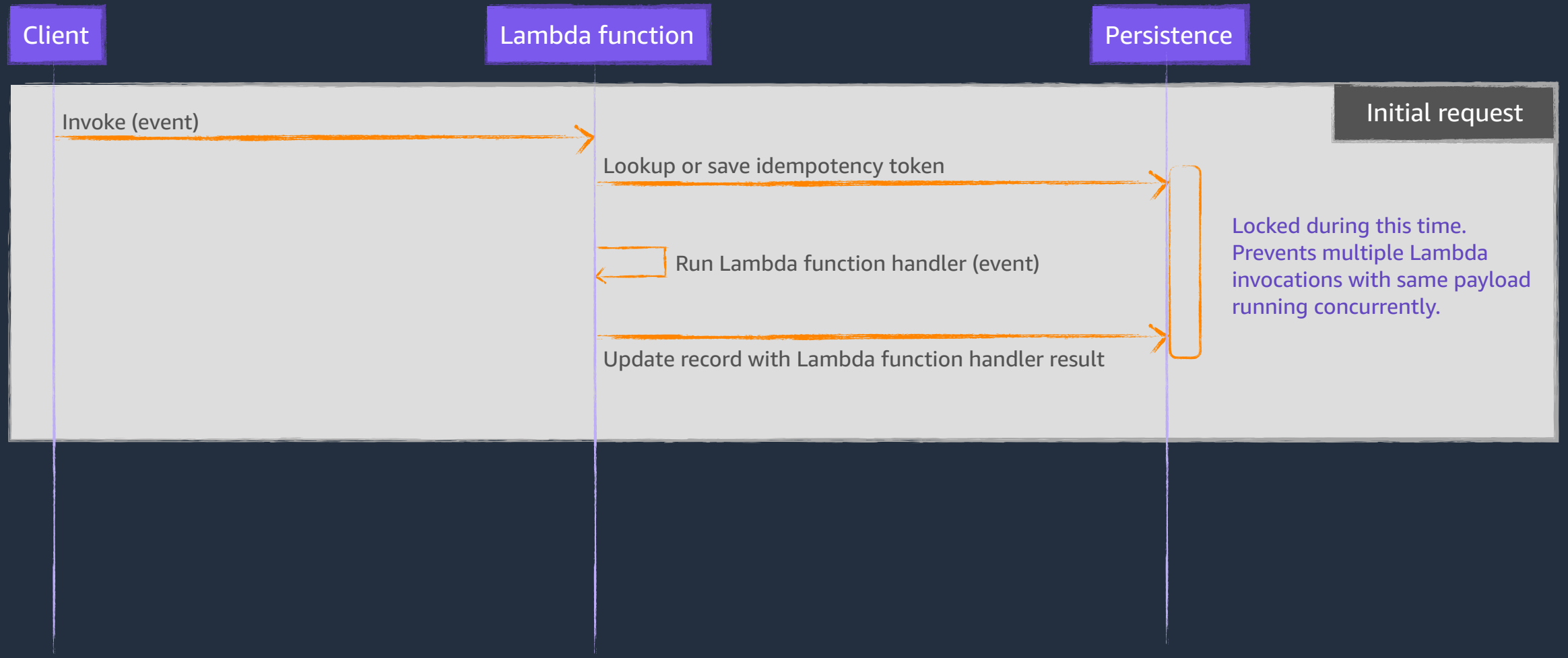
Idempotent request sequence



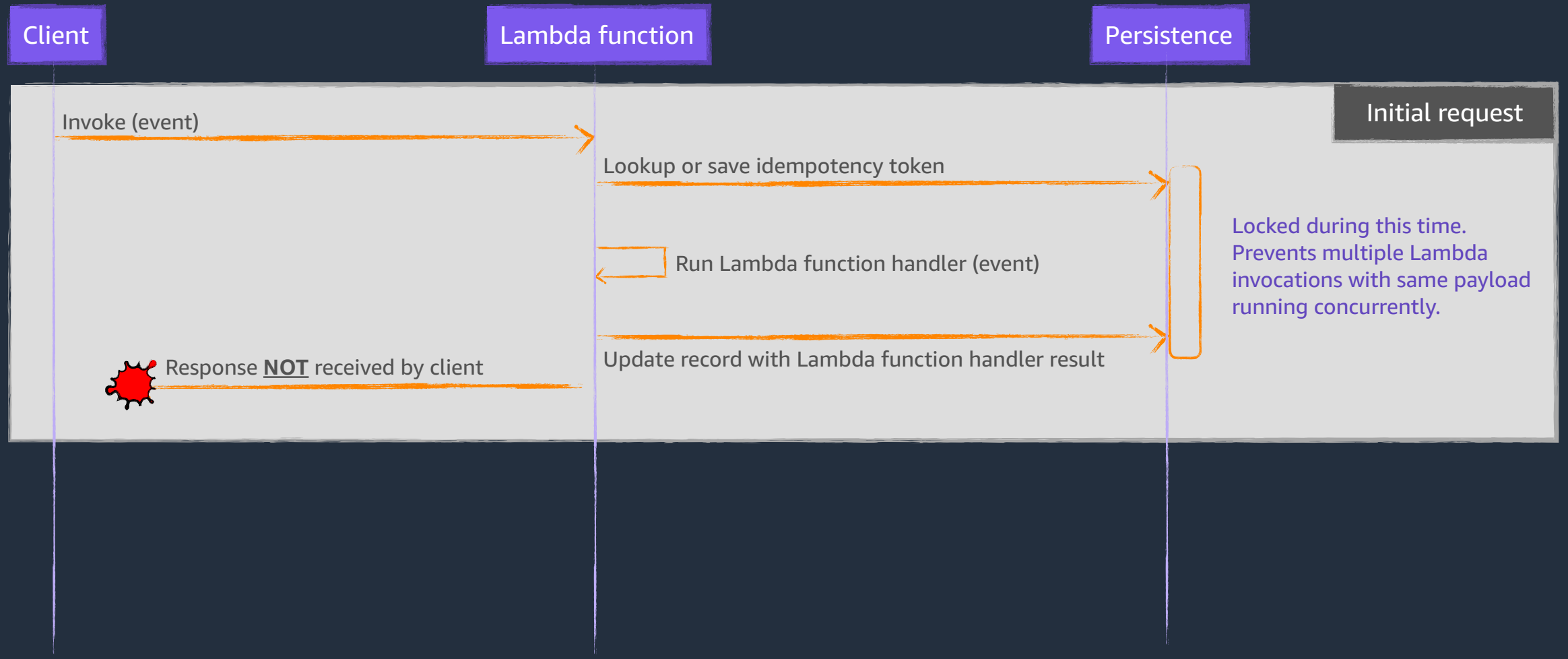
Idempotent request sequence

Idempotent request sequence

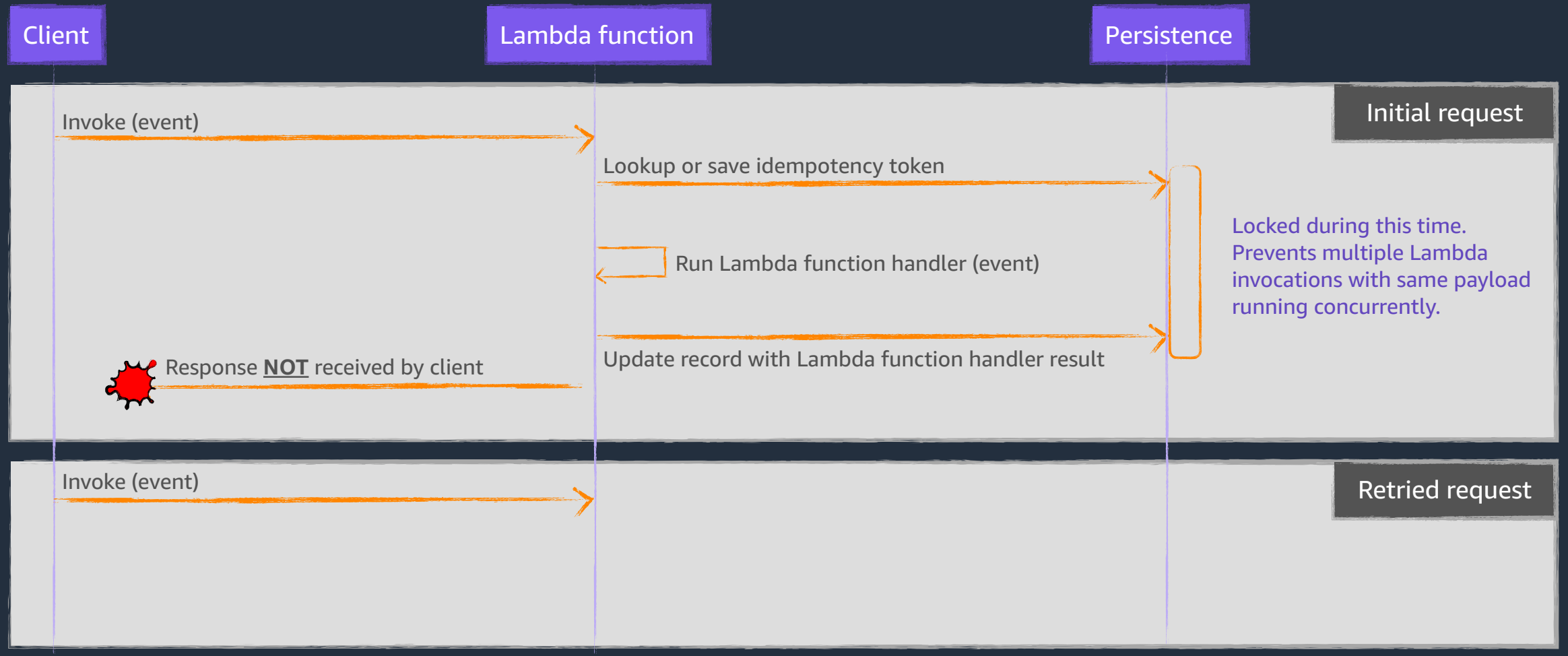
Retried idempotent request sequence



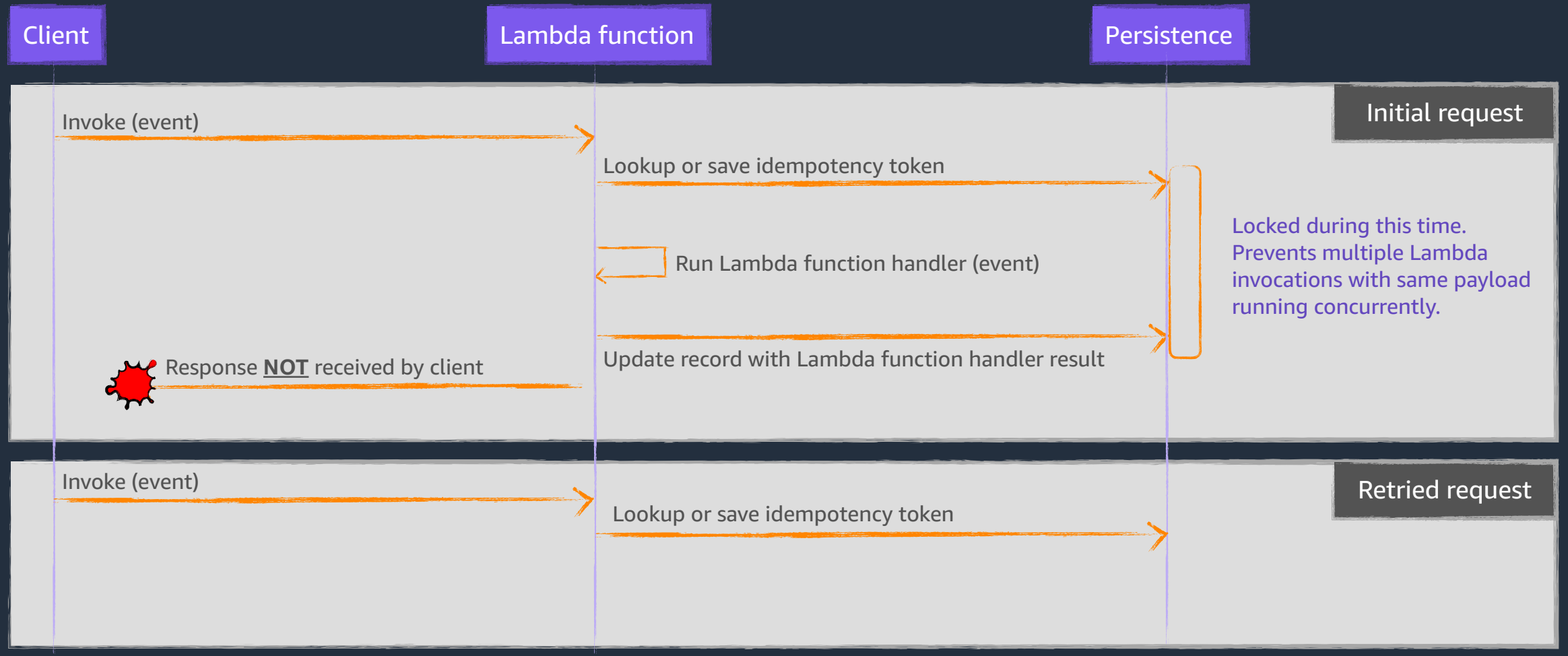
Retried idempotent request sequence



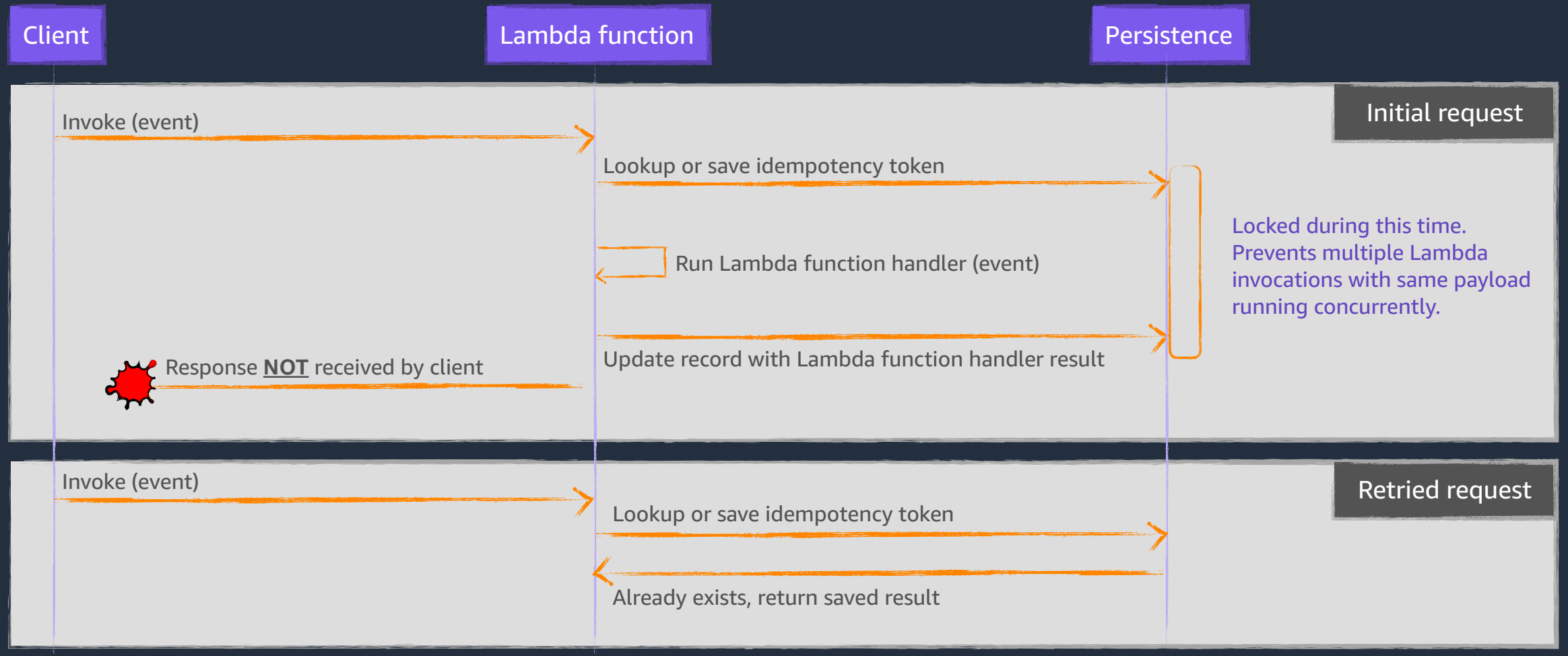
Retried idempotent request sequence



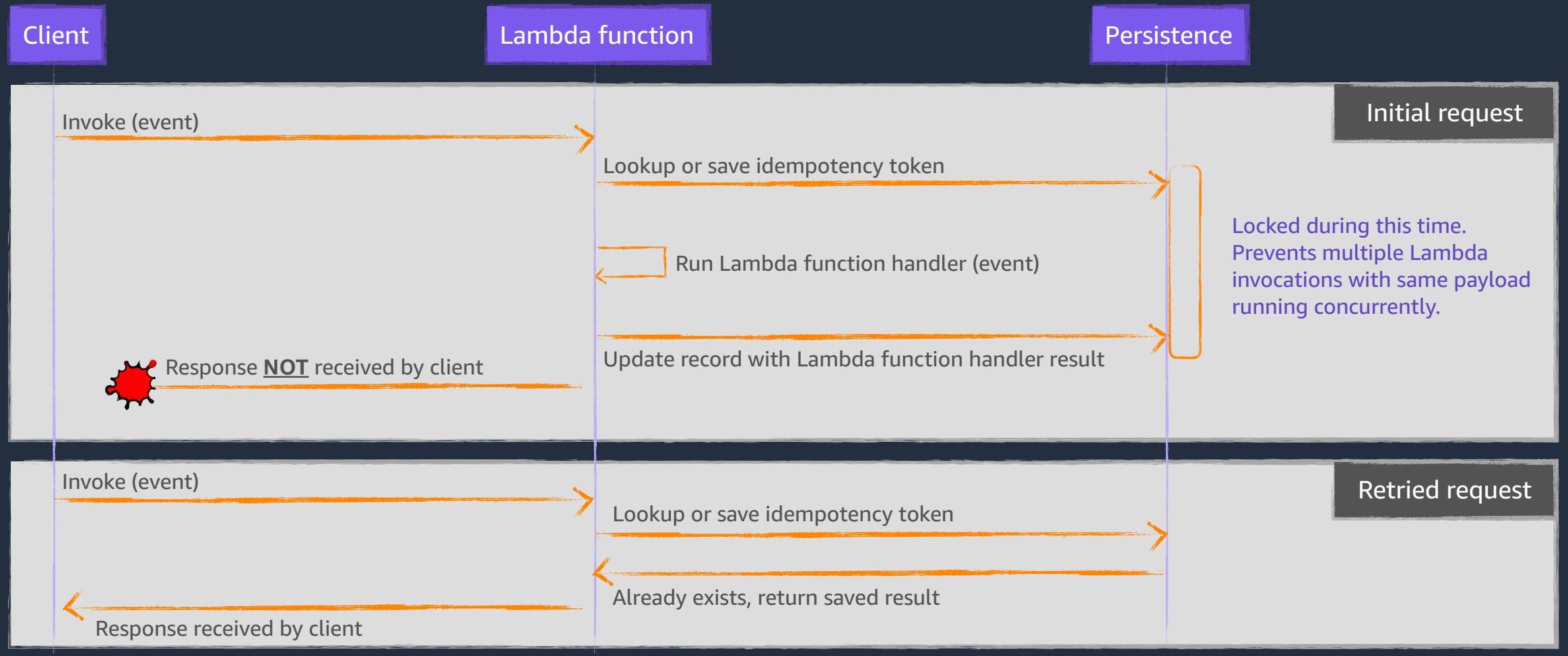
Retried idempotent request sequence



Retried idempotent request sequence



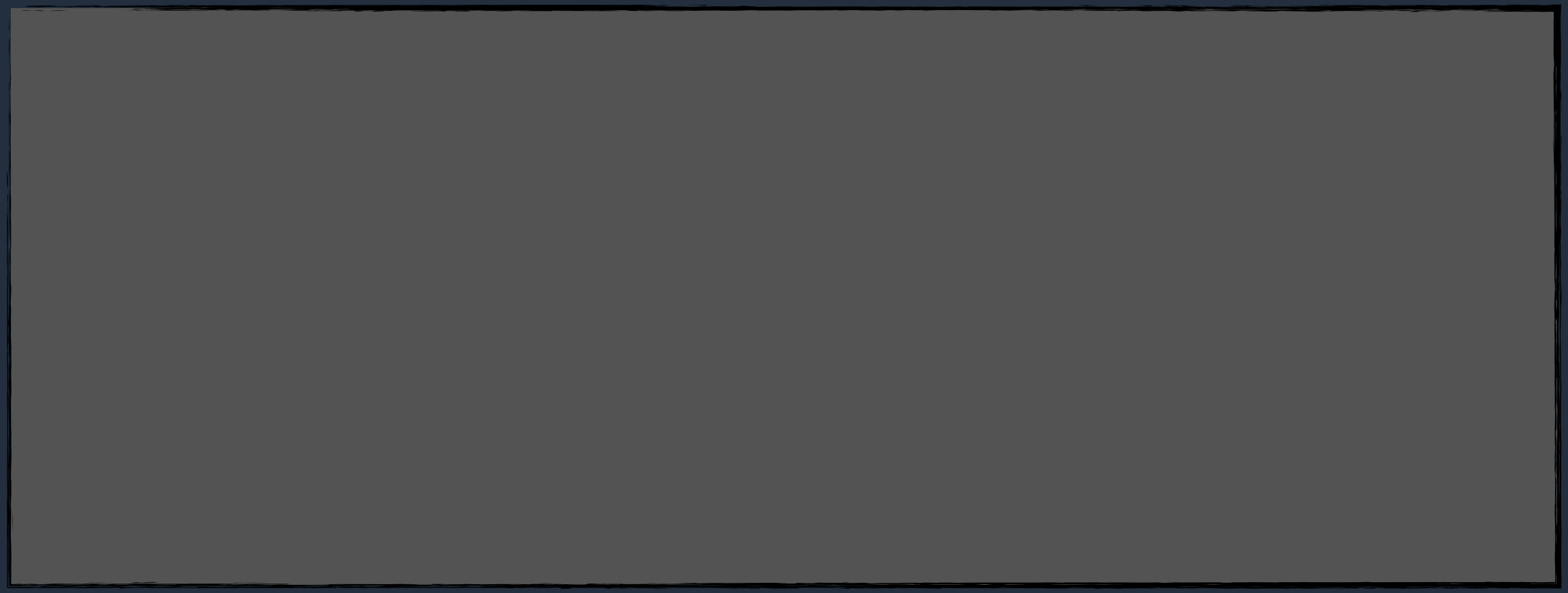
Retried idempotent request sequence



Using services that are idempotent

Amazon SQS

FIFO SendMessage API



Amazon SQS

FIFO SendMessage API

```
~ TOKEN=$(uuidgen) && echo $TOKEN
2DAA31B6-3ED6-4a9a-A4eE-825c88AA8a67
```

Amazon SQS
FIFO SendMessage API

```
~ TOKEN=$(uuidgen) && echo $TOKEN

2DAA31B6-3ED6-4a9a-A4eE-825c88AA8a67

~ aws sqs send-message --queue-url “https://sqs.us-east-1.amazonaws.com/98764738236/idempotence.fifo”
--message-deduplication-id “$TOKEN” --message-body “Hello, world” --message-group-id foo

{
   “MD5ofMessageBody”: “9238e7ruyt23uwe78r9f09oqiwe”,
   “MessageId”: “9ce79aca-2bbd-4a29-a751-0cc2dc3Cd1B9”,
   “SequenceNumber”: ”436782374689038”
}
```

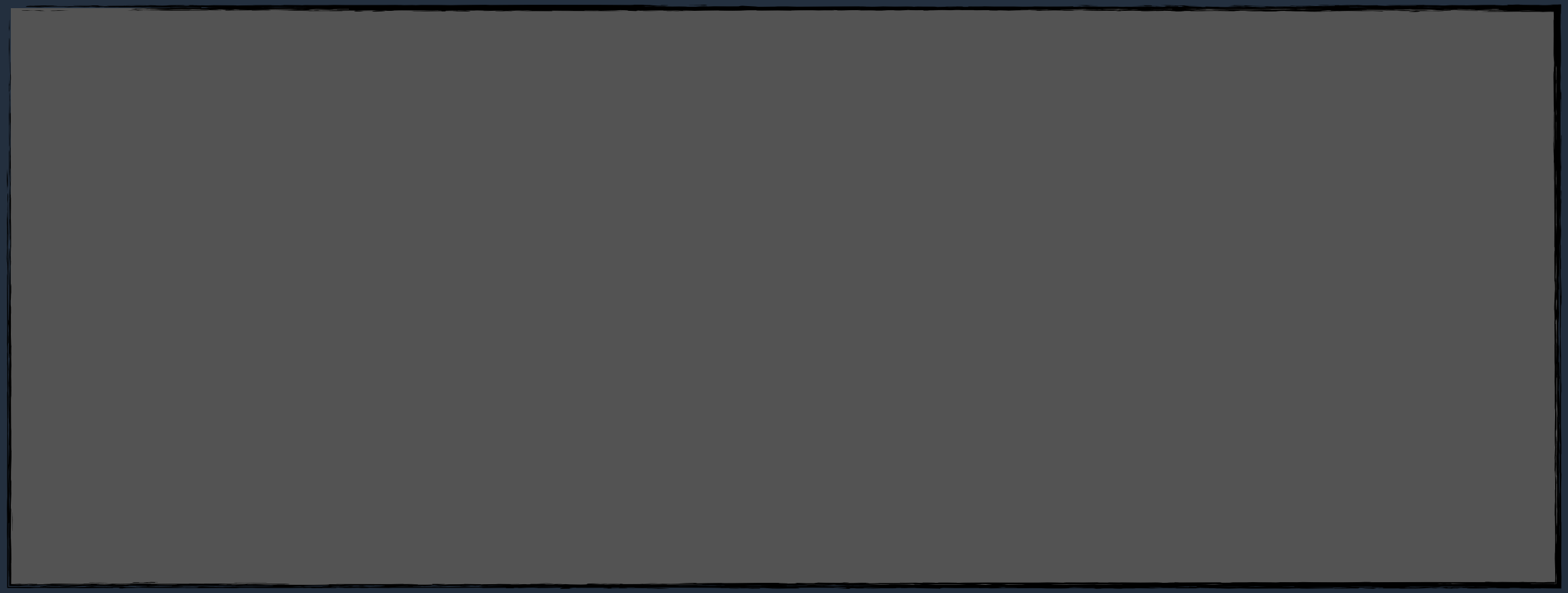
Amazon SQS

FIFO SendMessage API

```
~ TOKEN=$(uuidgen) && echo $TOKEN

2DAA31B6-3ED6-4a9a-A4eE-825c88AA8a67

~ aws sqs send-message --queue-url “https://sqs.us-east-1.amazonaws.com/98764738236/idempotence.fifo”
--message-deduplication-id “$TOKEN” --message-body “Hello, world” --message-group-id foo

{
   “MD5ofMessageBody”: “9238e7ruyt23uwe78r9f09oqiwe”,
   “MessageId”: “9ce79aca-2bbd-4a29-a751-0cc2dc3Cd1B9”,
   “SequenceNumber”: ”436782374689038”
}

~ aws sqs send-message --queue-url “https://sqs.us-east-1.amazonaws.com/98764738236/idempotence.fifo”
--message-deduplication-id “$TOKEN” --message-body “Hello, world” --message-group-id foo

{
   “MD5ofMessageBody”: “9238e7ruyt23uwe78r9f09oqiwe”,
   “MessageId”: “9ce79aca-2bbd-4a29-a751-0cc2dc3Cd1B9”,
   “SequenceNumber”: ”436782374689038”
}
```

AWS Step Functions

StartExecution API



AWS Step Functions

StartExecution API

```
~ TOKEN=$(uuidgen) && echo $TOKEN
2DAA31B6-3ED6-4a9a-A4eE-825c88AA8a67
```

AWS Step Functions
StartExecution API

```
~ TOKEN=$(uuidgen) && echo $TOKEN
2DAA31B6-3ED6-4a9a-A4eE-825c88AA8a67
~ aws step functions start-execution --state-machine-arn arn:aws:states:us-east-1:98764738236:statemachine:IdempotentStateMachine
--name "$TOKEN"
{
   "executionArn": "arn:aws:states:us-east-1:98764738236:statemachine:IdempotentStateMachine:2DAA31B6-3ED6-4a9a-A4eE-825c88AA8a67",
   "startDate": "2023-03-01T17:50:48.073000-04:00"
}
```

AWS Step Functions
StartExecution API

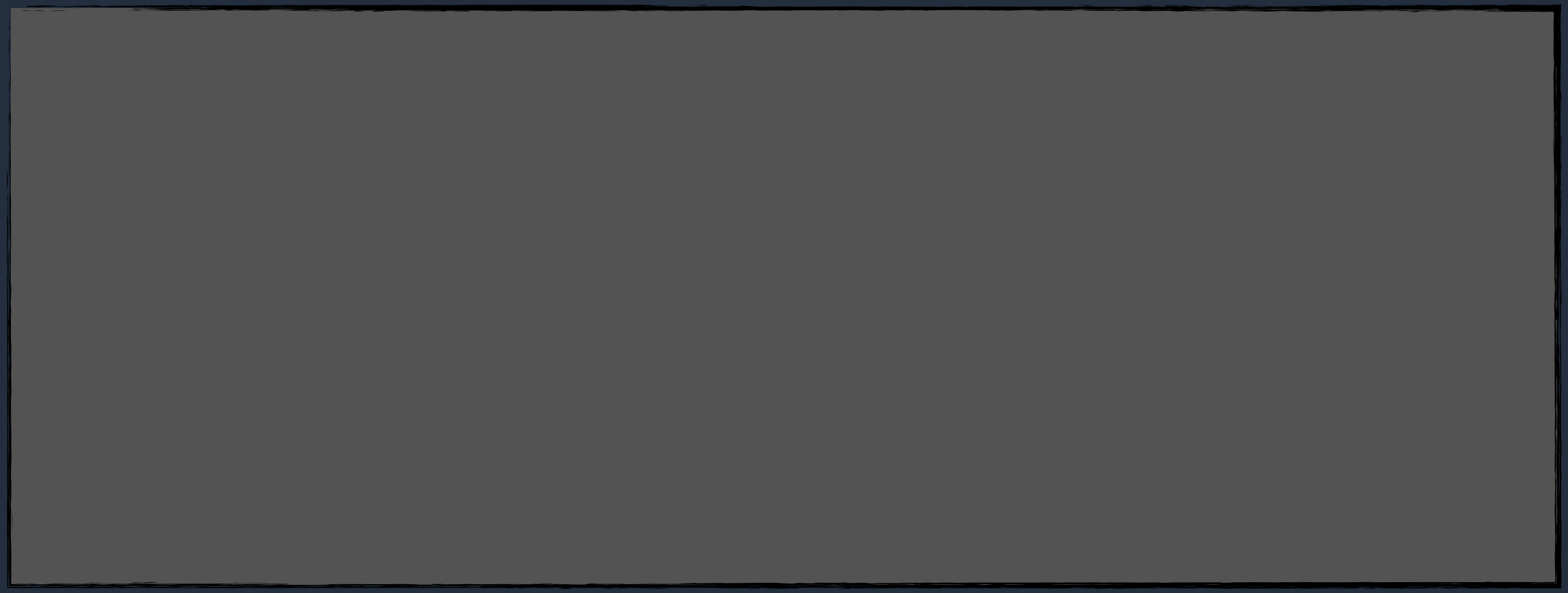
```
~ TOKEN=$(uuidgen) && echo $TOKEN
2DAA31B6-3ED6-4a9a-A4eE-825c88AA8a67
~ aws step functions start-execution --state-machine-arn arn:aws:states:us-east-1:98764738236:statemachine:IdempotentStateMachine
--name "$TOKEN"
{
    "executionArn": "arn:aws:states:us-east-1:98764738236:statemachine:IdempotentStateMachine:2DAA31B6-3ED6-4a9a-A4eE-825c88AA8a67",
    "startDate": "2023-03-01T17:50:48.073000-04:00"
}
~ aws step functions start-execution --state-machine-arn arn:aws:states:us-east-1:98764738236:statemachine:IdempotentStateMachine
--name "$TOKEN"
{
    "executionArn": "arn:aws:states:us-east-1:98764738236:statemachine:IdempotentStateMachine:2DAA31B6-3ED6-4a9a-A4eE-825c88AA8a67",
    "startDate": "2023-03-01T17:50:48.073000-04:00"
}
```

AWS Step Functions
StartExecution API

```
~ TOKEN=$(uuidgen) && echo $TOKEN
2DAA31B6-3ED6-4a9a-A4eE-825c88AA8a67
~ aws step functions start-execution --state-machine-arn arn:aws:states:us-east-1:98764738236:statemachine:IdempotentStateMachine
--name "$TOKEN"
{
    "executionArn": "arn:aws:states:us-east-1:98764738236:statemachine:IdempotentStateMachine:2DAA31B6-3ED6-4a9a-A4eE-825c88AA8a67",
    "startDate": "2023-03-01T17:50:48.073000-04:00"
}
~ aws step functions start-execution --state-machine-arn arn:aws:states:us-east-1:98764738236:statemachine:IdempotentStateMachine
--name "$TOKEN"
{
    "executionArn": "arn:aws:states:us-east-1:98764738236:statemachine:IdempotentStateMachine:2DAA31B6-3ED6-4a9a-A4eE-825c88AA8a67",
    "startDate": "2023-03-01T17:50:48.073000-04:00"
}
```

Amazon EventBridge

PutEvents API



Amazon EventBridge

PutEvents API

```
~ aws events put-event --entries '[{"Source": "Foo", "DetailType": "bar", "Detail": "{}"}]'
{
    "Entries": [
        {
            "EventId": "24a2939E-8d31-67Ff-dR99e4e0eAfd"
        }
    ]
}
```

Amazon EventBridge

PutEvents API

```
~ aws events put-event --entries '[{"Source": "Foo", "DetailType": "bar", "Detail": "{}"}]'
{
    "Entries": [
        {
            "EventId": "24a2939E-8d31-67Ff-dR99e4e0eAfd"
        }
    ]
}
~ aws events put-event --entries '[{"Source": "Foo", "DetailType": "bar", "Detail": "{}"}]'
{
    "Entries": [
        {
            "EventId": "61a15878-F74r-6ye6-5Tu99dw982Ks"
        }
    ]
}
```

Amazon EventBridge

PutEvents API

```
~ aws events put-event --entries '[{"Source": "Foo", "DetailType": "bar", "Detail": "{}"}]'
{
    "Entries": [
        {
            "EventId": "24a2939E-8d31-67Ff-dR99e4e0eAfd"
        }
    ]
}
~ aws events put-event --entries '[{"Source": "Foo", "DetailType": "bar", "Detail": "{}"}]'
{
    "Entries": [
        {
            "EventId": "61a15878-F74r-6ye6-5Tu99dw982Ks"
        }
    ]
}
```

Amazon EventBridge

Idempotency identifier


```
{
  "version": "0",
  "id": "61a15356-f8d3-4b6e-7da9-5bfccde8016d",
  "detail-type": "OrderCreated",
  "source": "com.orders",
  "account": "068896461592",
  "time": "2022-05-01T22:15:20Z",
  "region": "us-east-1",
  "detail": {
    "metadata": {
      "idempotency-key": "AF8074B2-3C23-415B-B465-71A849C63452"
    },
    "data": {
      "order-id": "1073459984"
    }
}
```

Amazon EventBridge

Idempotency identifier

!! Event.id is not an idempotency ID !!


```
{
  "version": "0",
  "id": "61a15356-f8d3-4b6e-7da9-5bfccde8016d",
  "detail-type": "OrderCreated",
  "source": "com.orders",
  "account": "068896461592",
  "time": "2022-05-01T22:15:20Z",
  "region": "us-east-1",
  "detail": {
    "metadata": {
      "idempotency-key": "AF8074B2-3C23-415B-B465-71A849C63452"
    },
    "data": {
      "order-id": "1073459984"
    }
}
```

Amazon EventBridge

Idempotency identifier

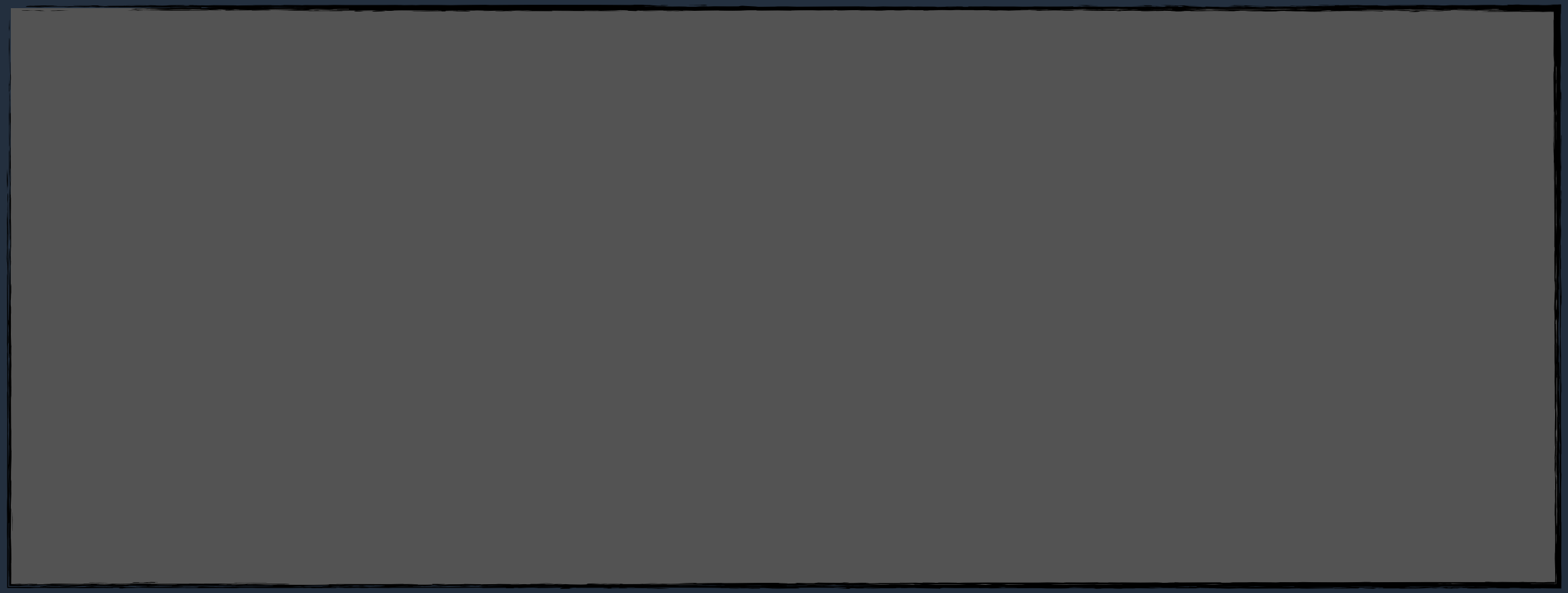
!! Event.id is not an idempotency ID !!

```
{
  "version": "0",
  "id": "61a15356-f8d3-4b6e-7da9-5bfccde8016d",
  "detail-type": "OrderCreated",
  "source": "com.orders",
  "account": "068896461592",
  "time": "2022-05-01T22:15:20Z",
  "region": "us-east-1",
  "detail": {
    "metadata": {
      "idempotency-key": "AF8074B2-3C23-415B-B465-71A849C63452"
    },
    "data": {
      "order-id": "1073459984"
    }
}
```

Client provided idempotency ID

Amazon EventBridge

PutEvents API



Amazon EventBridge

PutEvents API

```
~ TOKEN=$(uuidgen) && echo $TOKEN
2DAA31B6-3ED6-4a9a-A4eE-825c88AA8a67
```

Amazon EventBridge

PutEvents API

```
~ TOKEN=$(uuidgen) && echo $TOKEN
2DAA31B6-3ED6-4a9a-A4eE-825c88AA8a67
~ aws events put-event --entries '[{"Source":"Foo", "DetailType":"bar", "Detail": "{\"metadata\":{\"idempotency-key\":\"$TOKEN\"}}"}]'
{
    "Entries": [
        {
            "EventId": "24a2939E-8d31-67Ff-dR99e4e0eAfd"
        }
    ]
}
```

Idempotency duration

Provider	Idempotency duration
Lambda Powertools idempotency utilities (default)	1 hour
Amazon SQS (FIFO) SendMessage	5 minutes
Amazon SNS (FIFO) Publish	5 minutes
Step Functions StartExecution	90 days

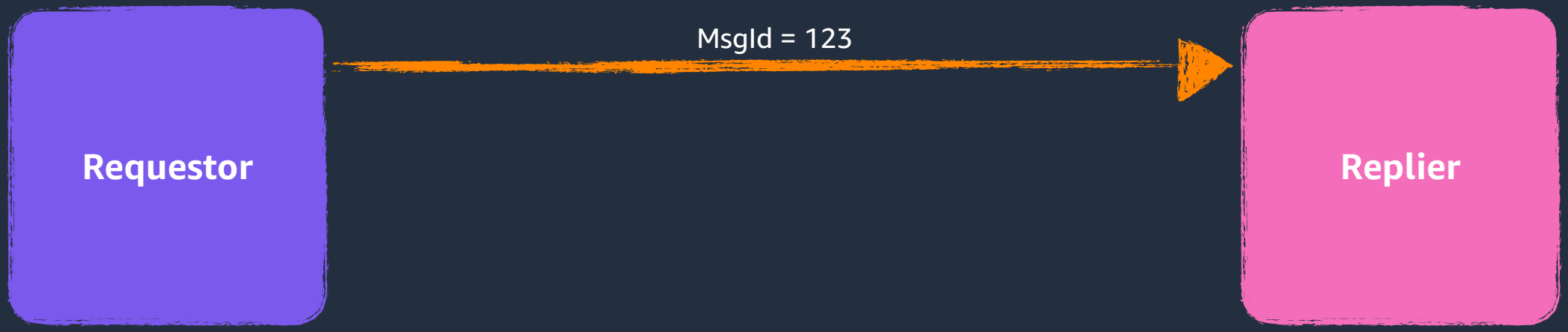
Idempotency by any other name would smell as sweet

Provider	Idempotency identifier
Step Functions StartExecution	name
Amazon SQS (FIFO) SendMessage	message-deduplication-id
Amazon EC2 RunInstances	client-token
Stripe	Idempotency-Key
PayPal	PayPal-Request-Id
Square	idempotency_key
IETF	Idempotency-Key
Enterprise Integration Patterns	Message ID

Idempotency tokens flow through systems

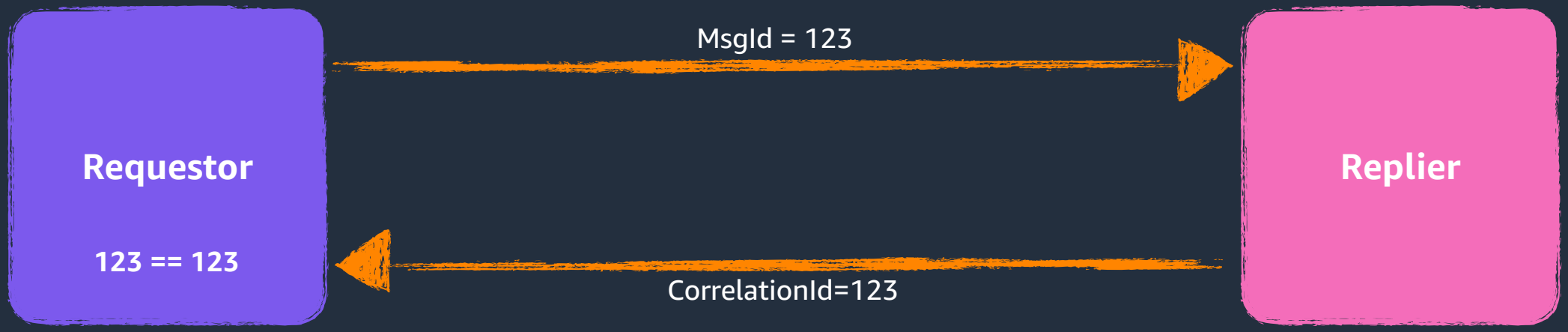
Source: Enterprise Integration Patterns, Correlation and Conversations (Hohpe)

Idempotency tokens flow through systems



Source: Enterprise Integration Patterns, Correlation and Conversations (Hohpe)

Idempotency tokens flow through systems



Source: Enterprise Integration Patterns, Correlation and Conversations (Hohpe)

Idempotency tokens flow through intermediaries

Source: Enterprise Integration Patterns, Correlation and Conversations (Hohpe)

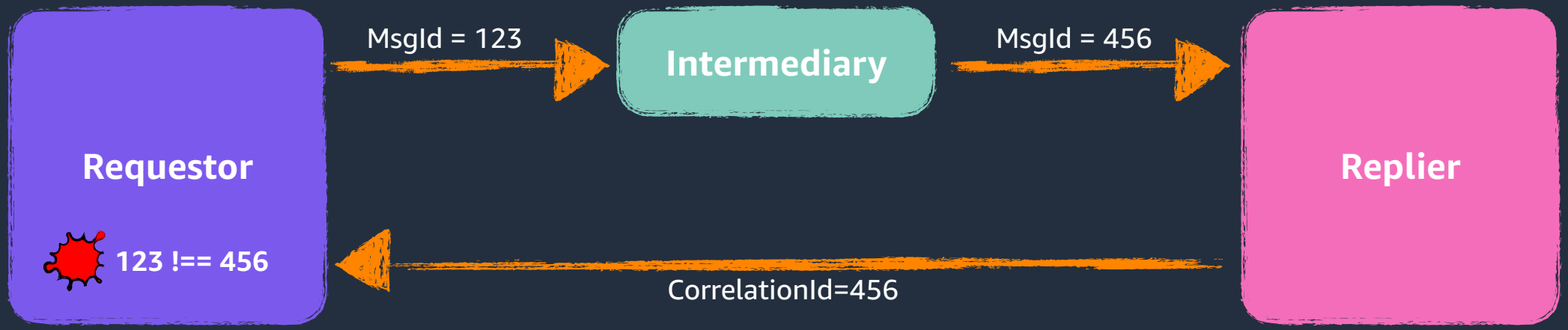
Idempotency tokens flow through intermediaries

Source: Enterprise Integration Patterns, Correlation and Conversations (Hohpe)

Idempotency tokens flow through intermediaries

Source: Enterprise Integration Patterns, Correlation and Conversations (Hohpe)

Idempotency tokens flow through intermediaries



Source: Enterprise Integration Patterns, Correlation and Conversations (Hohpe)

https://s12d.com/idempotent

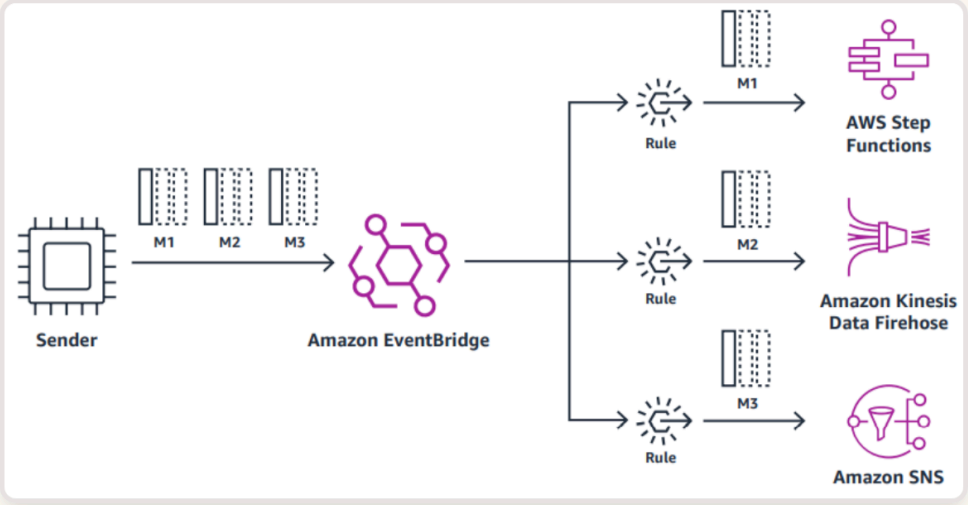
Welcome to *Serverless* Land

This site brings together the latest information, blogs, videos, code, and learning resources for AWS Serverless. Learn to use and build apps that scale automatically on low-cost, fully-managed serverless architecture.

Building **Event Driven** Architectures

Event-driven architectures are an architecture style that can help you boost agility and build reliable, scalable applications. Serverless services like EventBridge, Step Functions, SQS, SNS, and Lambda have a natural affinity with event-driven architectures - they are invoked by events, emit events, and have built-in capabilities for building with events.

Learn more about event-driven architectures, including key concepts, best practices, AWS services, and getting started



Thank you!

Eric Johnson
@edjgeek

goto;

Don't forget to
vote for this session
in the **GOTO Guide app**