Serverless na AWS

Gabriel Bella Martini
Arquiteto de Soluções AWS

© 2022, Amazon Web Services, Inc. or its affiliates.

Overview of Serverless

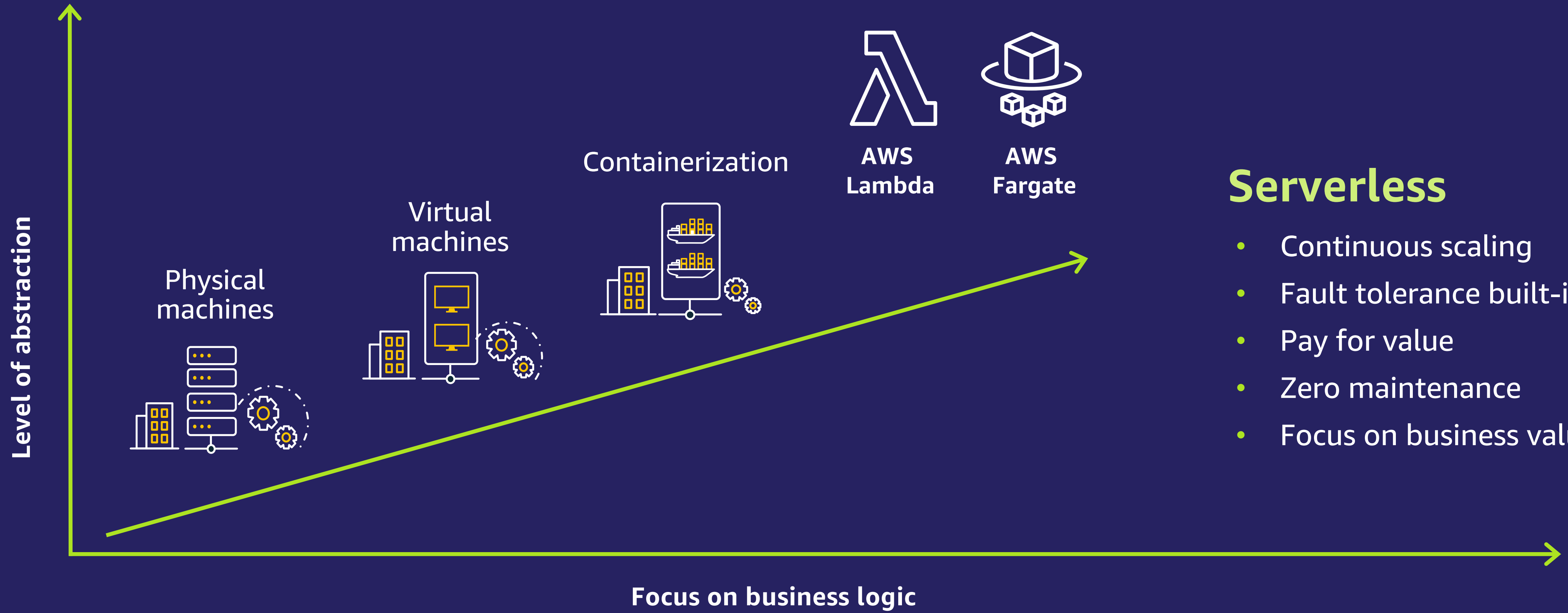
© 2022, Amazon Web Services, Inc. or its affiliates.

What does the future look like?

ALL THE CODE YOU EVER WRITE IS BUSINESS LOGIC

© 2022, Amazon Web Services, Inc. or its affiliates.

There’s a paradigm shift happening



Serverless

- Continuous scaling
- Fault tolerance built-in
- Pay for value
- Zero maintenance
- Focus on business value

© 2022, Amazon Web Services, Inc. or its affiliates.

What is serverless?

No infrastructure provisioning, no management

Automatic scaling

Pay for use

Highly available and secure

© 2022, Amazon Web Services, Inc. or its affiliates.

Serverless is more than compute

COMPUTE

AWS Lambda

AWS Fargate

DATA STORES

Amazon S3

Amazon Aurora Serverless

Amazon DynamoDB

INTEGRATION

Amazon EventBridge

Amazon API Gateway

Amazon SQS

Amazon SNS

Amazon MQ

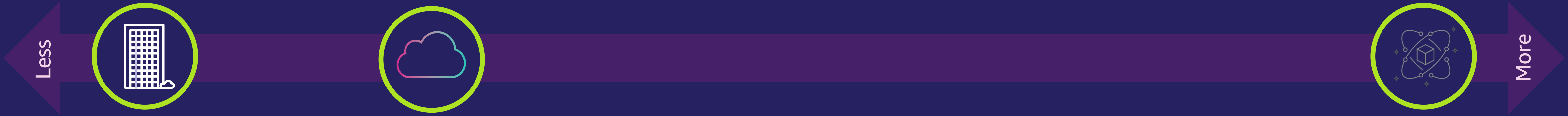
STREAMING

Amazon Kinesis

Amazon Managed Streaming for Apache Kafka (MSK)

© 2022, Amazon Web Services, Inc. or its affiliates.

AWS operational responsibility models



Compute	Virtual machine	Amazon EC2	AWS Elastic Beanstalk		Fargate	Lambda
Databases	MySQL	MySQL on Amazon EC2	Amazon RDS for MySQL	Amazon RDS	Amazon Aurora	DynamoDB
Storage	Storage					S3
Messaging	ESBs		Amazon MQ	Amazon Kinesis		EventBridge/SQS/SNS
Analytics	Hadoop	Hadoop on EC2	EMR	Amazon OpenSearch Service		Amazon Athena

© 2022, Amazon Web Services, Inc. or its affiliates.

AWS Lambda

Event-driven function-as-a-service

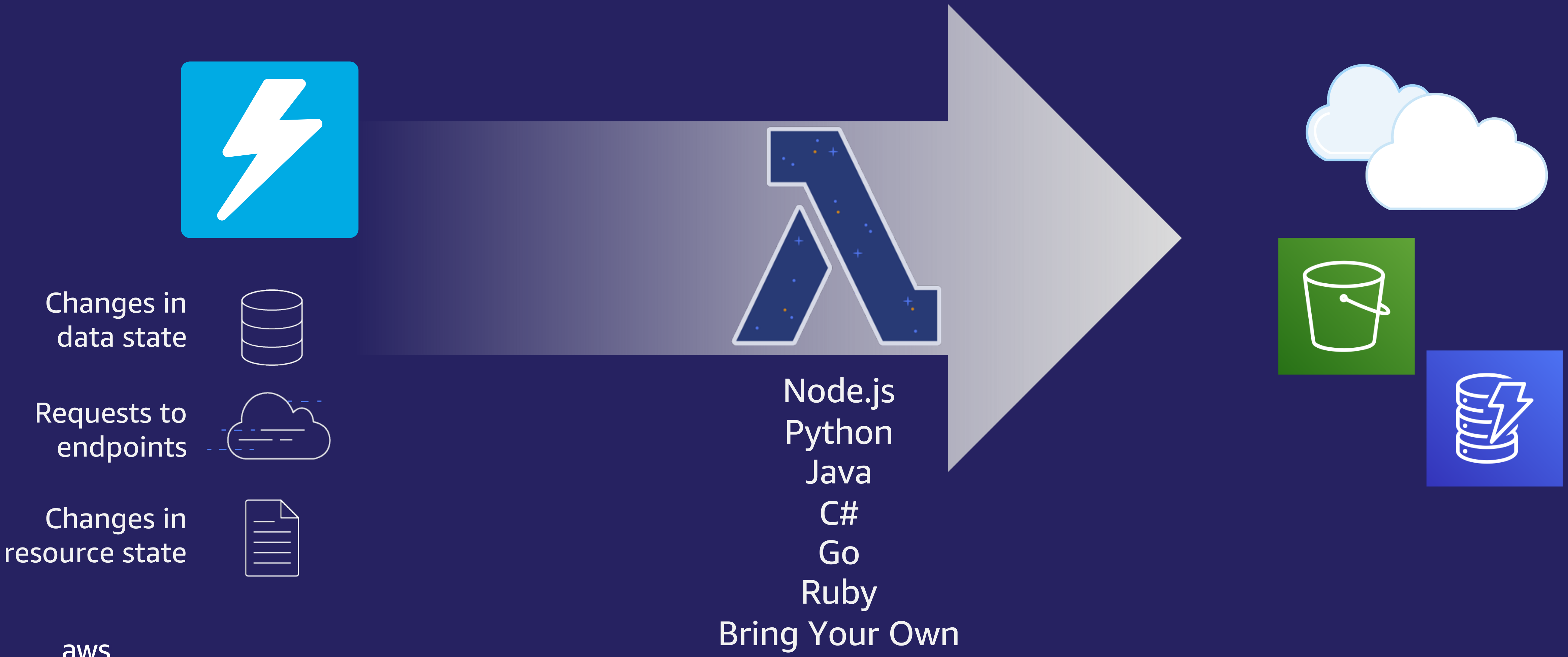
© 2022, Amazon Web Services, Inc. or its affiliates.

Serverless Architecture

Event Source

Function

Services / Other



Changes in data state

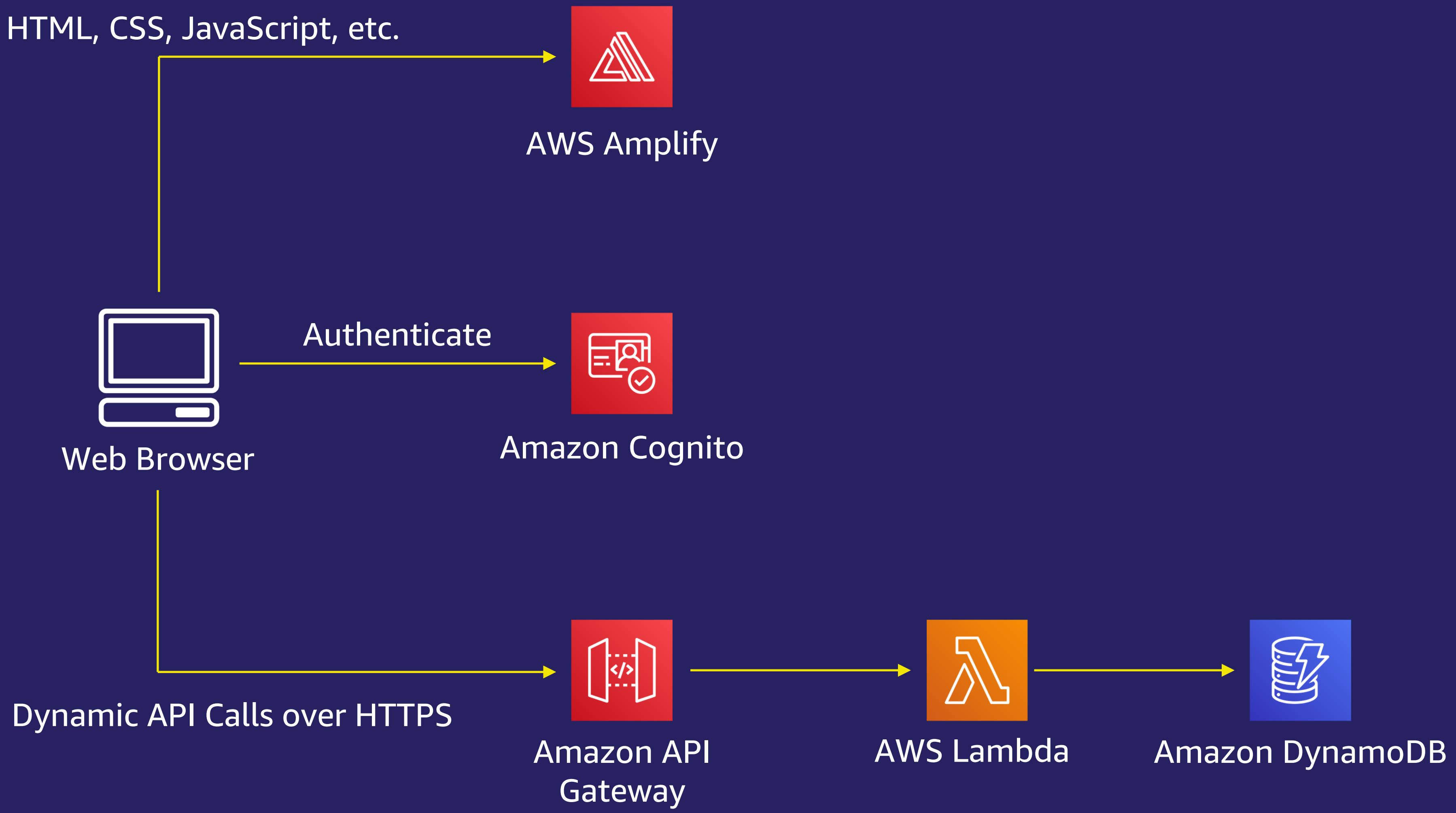
Requests to endpoints

Changes in resource state

Node.js
Python
Java
C#
Go
Ruby
Bring Your Own

© 2022, Amazon Web Services, Inc. or its affiliates.

Serverless Web Architecture



© 2022, Amazon Web Services, Inc. or its affiliates.

Anatomy of a Lambda Function

Handler function

- Function executed on invocation
- Processes incoming event

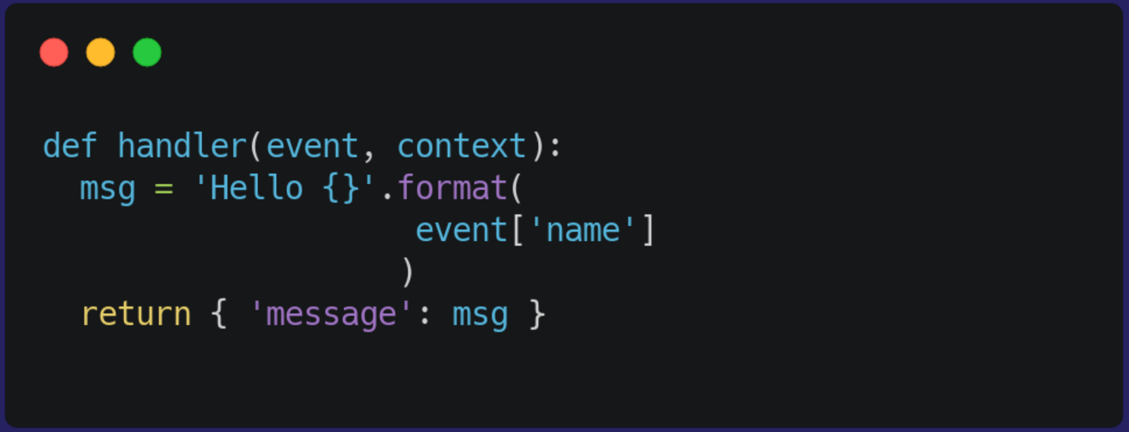
Event

- Invocation data sent to function
- Shape differs by event source

Context

- Additional information from Lambda service
- Examples: request ID, time remaining

app.py



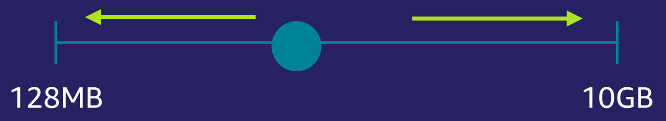
```
def handler(event, context):
  msg = 'Hello {}'.format(
                    event['name']
                   )
  return { 'message': msg }
```

© 2022, Amazon Web Services, Inc. or its affiliates.

Lambda Function Configuration

Power Rating

- Select between 128MB and 10GB
- CPU and network allocated proportionally
- Power tune to balance cost and speed



Permissions Model

- Execution Role grants function access to resources via IAM
- Function Policy controls invocation

© 2022, Amazon Web Services, Inc. or its affiliates.

Lambda Function Configuration

Timeout

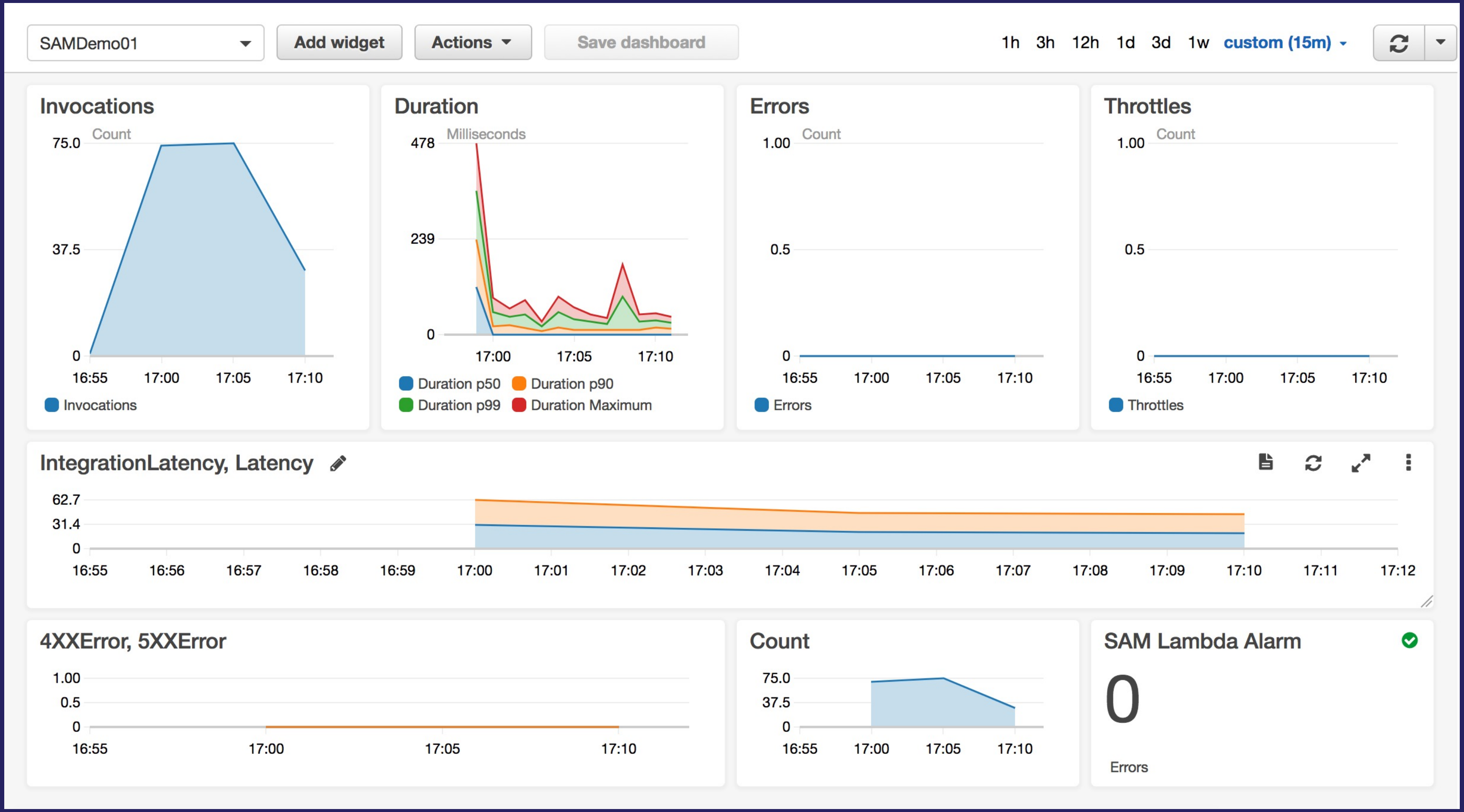
- Up to 15 minutes
- Synchronous vs Asynchronous
- API Gateway timeout = 30 sec

Network Access

- Configure access to VPC
- Security Group rules apply
- VPC does not enhance security of function

© 2022, Amazon Web Services, Inc. or its affiliates.

Built in monitoring



© 2022, Amazon Web Services, Inc. or its affiliates.

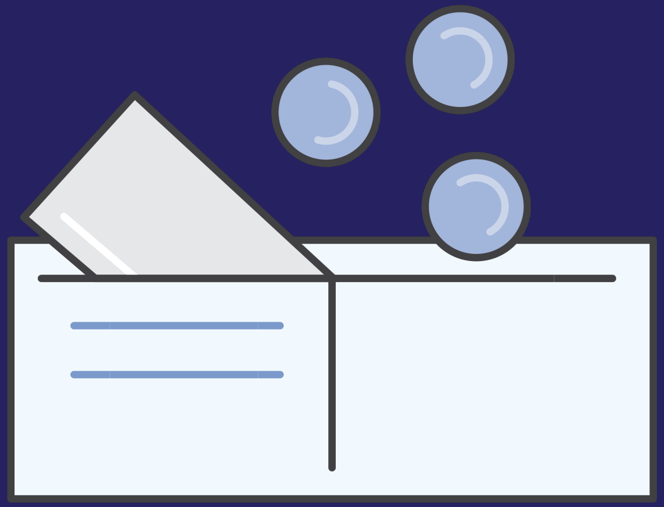
A few items to keep in mind...

- Functions are stateless, no affinity to underlying infrastructure
- Event triggers an invocation
- Lambda can handle a wide variety of event sources
    - Depending on event source, payload differs
    - Some event sources are batched (e.g. S3, SQS)
- Lambda service manages scaling, invocation
- Lambda service team manages platform security

Build something!

© 2022, Amazon Web Services, Inc. or its affiliates.

Fine-grained pricing



Free Tier

1M requests and 400,000 GBs of compute.

Every month, every customer.

- Pay for value
  - Priced by power rating
  - Charged in **1ms** increments
  - Low per-request charge
- No minimum
- Never pay for idle

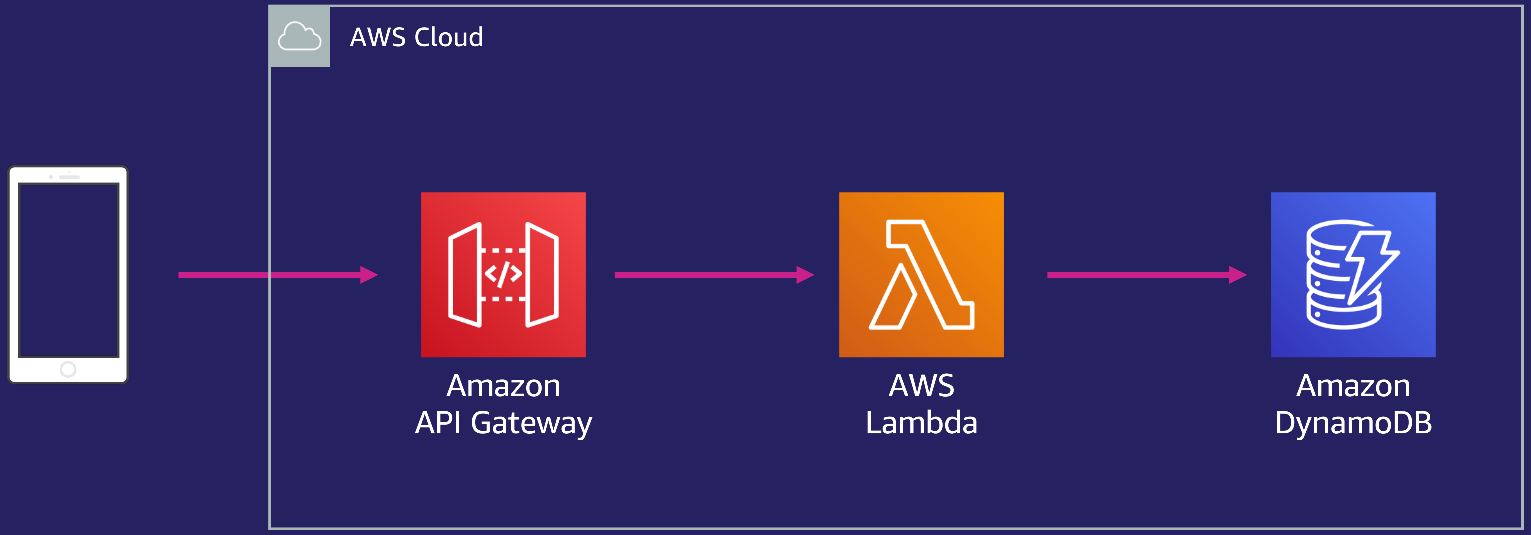
© 2022, Amazon Web Services, Inc. or its affiliates.

Amazon API Gateway

Build and manage application interfaces

© 2022, Amazon Web Services, Inc. or its affiliates.

API Gateway is a front door...



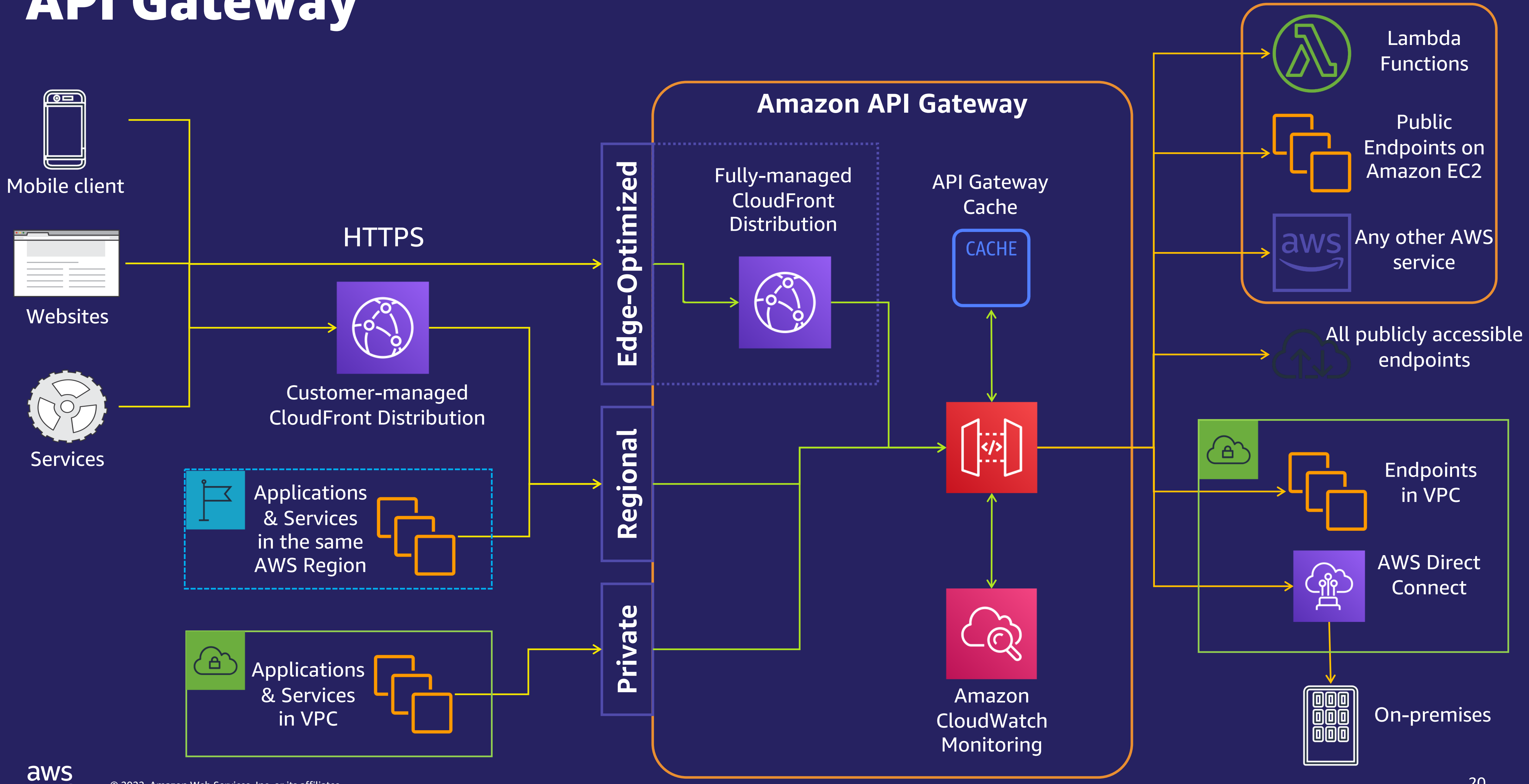
© 2022, Amazon Web Services, Inc. or its affiliates.

…and alleviates common concerns so developers can focus on business logic

- Throttling
- Caching
- Authorization
- API Keys
- Usage Plans
- Request/Response Mapping

© 2022, Amazon Web Services, Inc. or its affiliates.

API Gateway

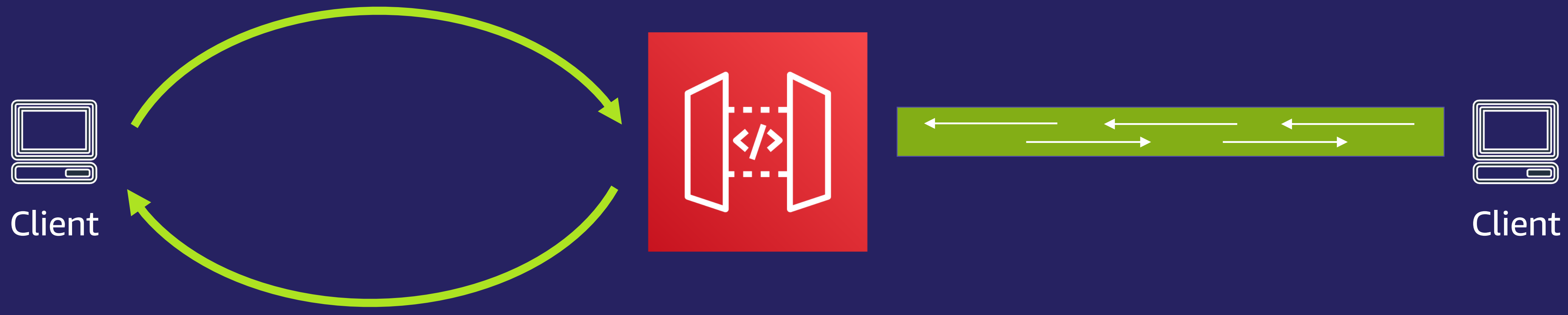

Mobile client
Websites
Services
HTTPS
Customer-managed CloudFront Distribution
Applications & Services in the same AWS Region
Applications & Services in VPC
Edge-Optimized
Regional
Private
Amazon API Gateway
Fully-managed CloudFront Distribution
API Gateway Cache
CACHE
Amazon CloudWatch Monitoring
Lambda Functions
Public Endpoints on Amazon EC2
aws
Any other AWS service
All publicly accessible endpoints
Endpoints in VPC
AWS Direct Connect
On-premises

© 2022, Amazon Web Services, Inc. or its affiliates.

Support for multiple API types

RESTful: HTTP APIs & REST APIs

WebSocket APIs



- Request / Response
- HTTP Methods like GET, POST, etc.
- Short-lived communication
- Stateless

- Serverless WebSocket
- Two-way communication channel
- Long-lived communication
- Stateful

© 2022, Amazon Web Services, Inc. or its affiliates.

REST versus WebSocket APIs

REST

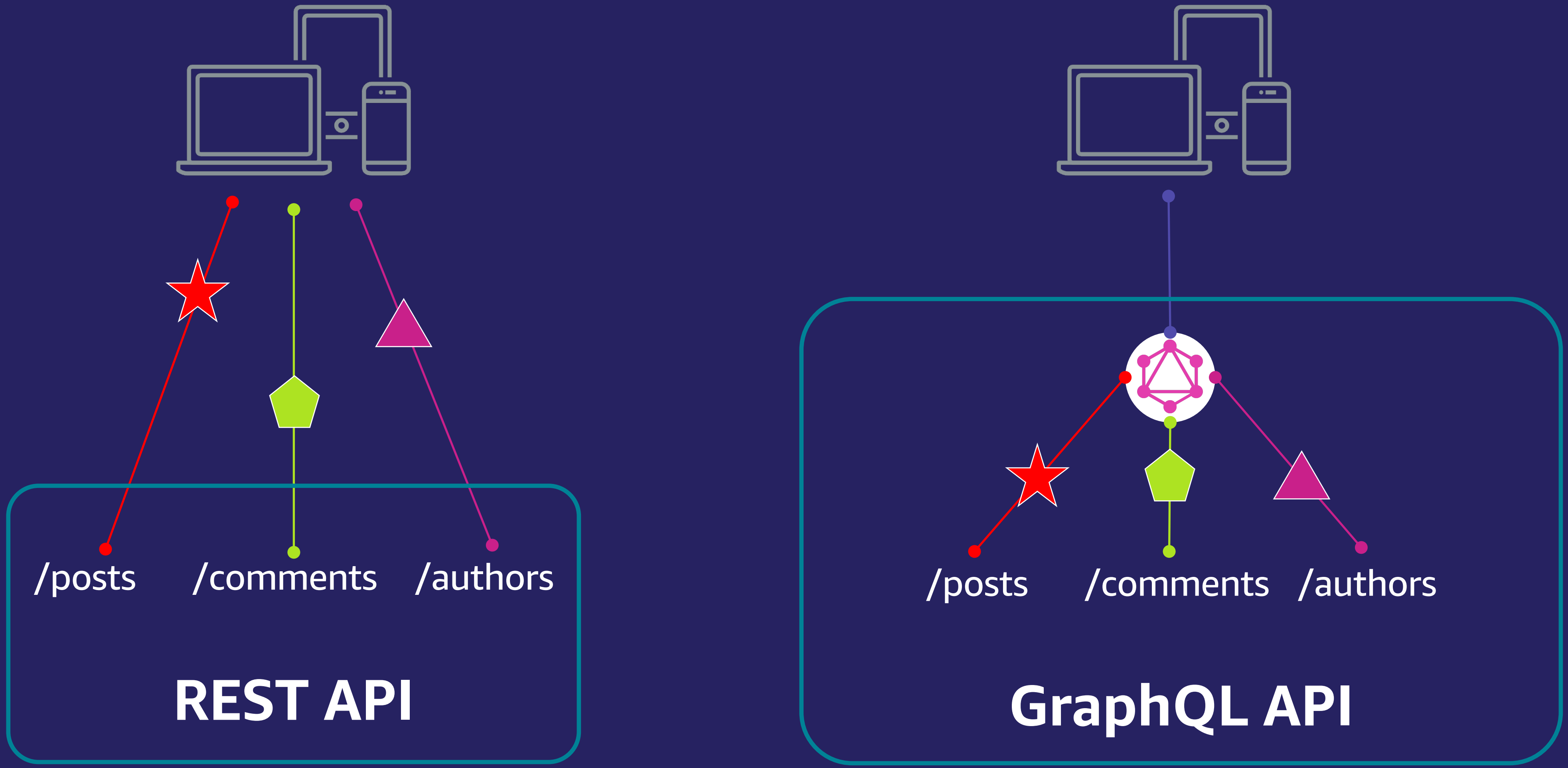
- Web services over HTTP
- Flexible
- Stateless
- Two flavors:
  - HTTP API (faster, cheaper)
  - REST API (more features)

WebSocket

- Two-way communication between application and clients
- Persistent connection, stateful
- Useful for:
  - Chat
  - Gaming
  - Data streaming
  - Real-time updates

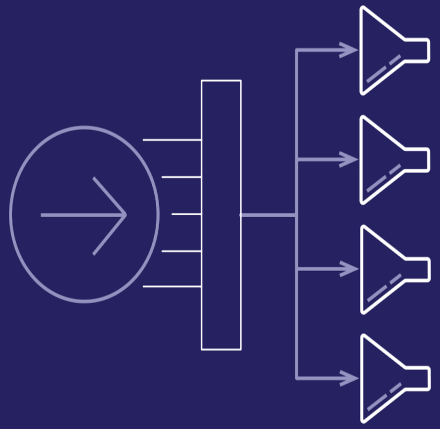
© 2022, Amazon Web Services, Inc. or its affiliates.

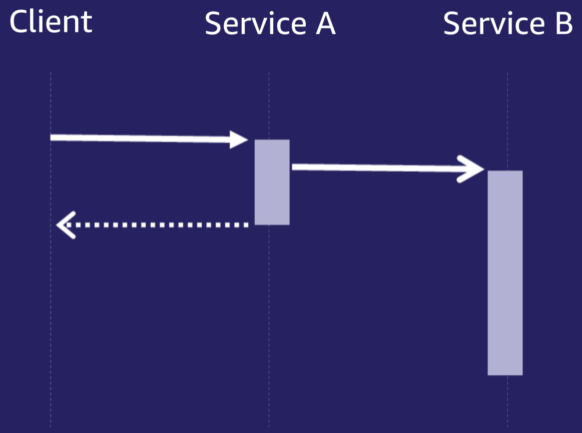
Or consider GraphQL for data-heavy applications

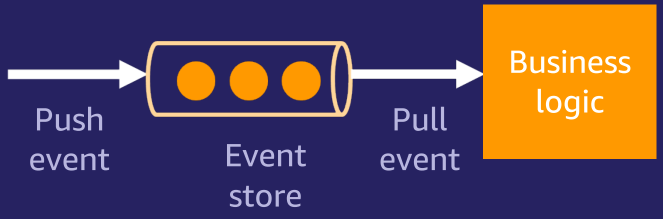


© 2022, Amazon Web Services, Inc. or its affiliates.

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

© 2022, Amazon Web Services, Inc. or its affiliates.

Events enable interaction between services

MANAGED SERVICES PROVIDE ROUTING, STORAGE, AND DISTRIBUTION OF EVENTS

Amazon SQS

Messaging

Durable and scalable
Fully managed
Comprehensive security

Amazon EventBridge

Eventing

Event filtering
Managed and scalable
SaaS integration

Amazon SNS

Eventing

Performance at scale
Fully managed
Enterprise-ready

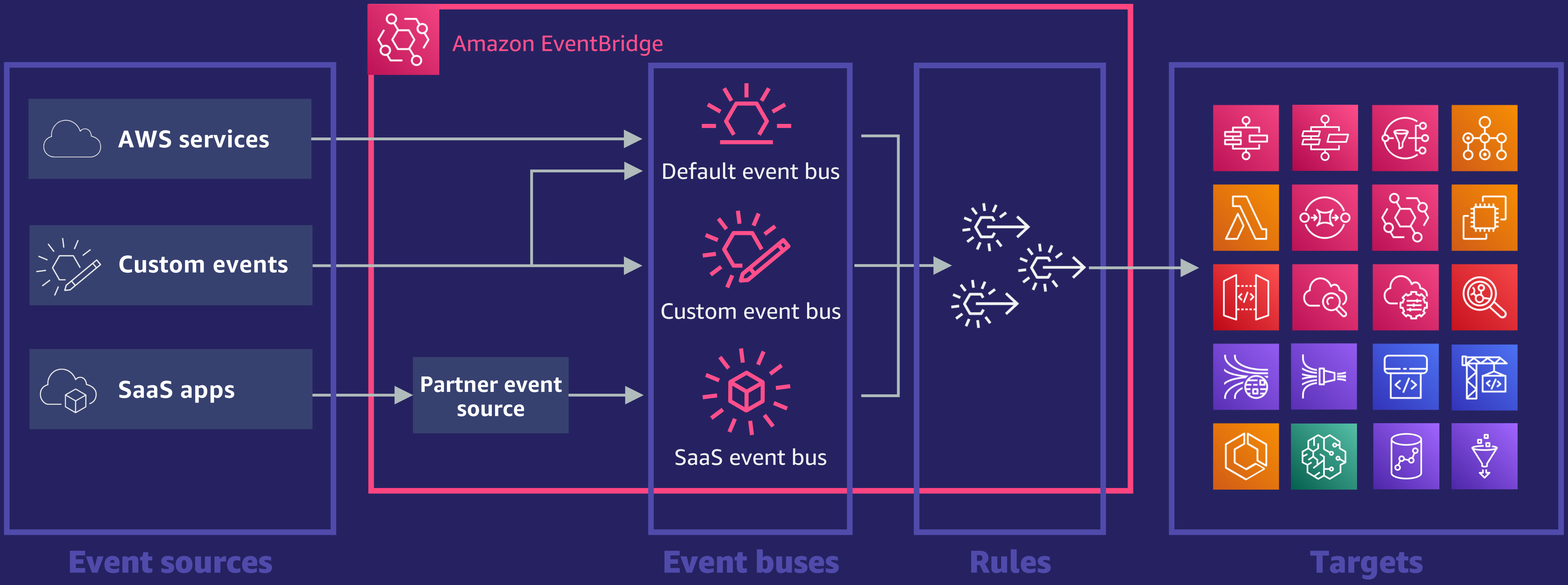
AWS Step Functions

Orchestration

Sequencing
Parallel execution
State management

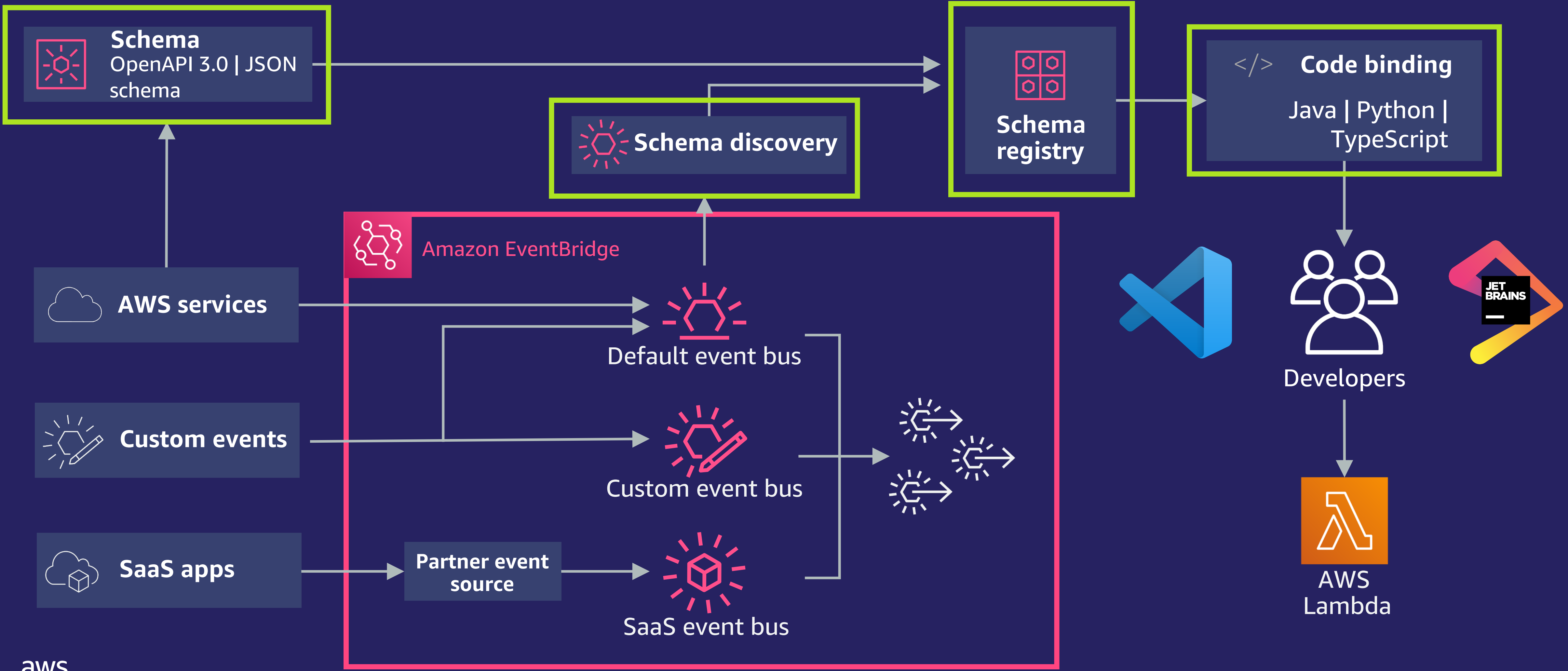
© 2022, Amazon Web Services, Inc. or its affiliates.

Amazon EventBridge architecture



© 2022, Amazon Web Services, Inc. or its affiliates.

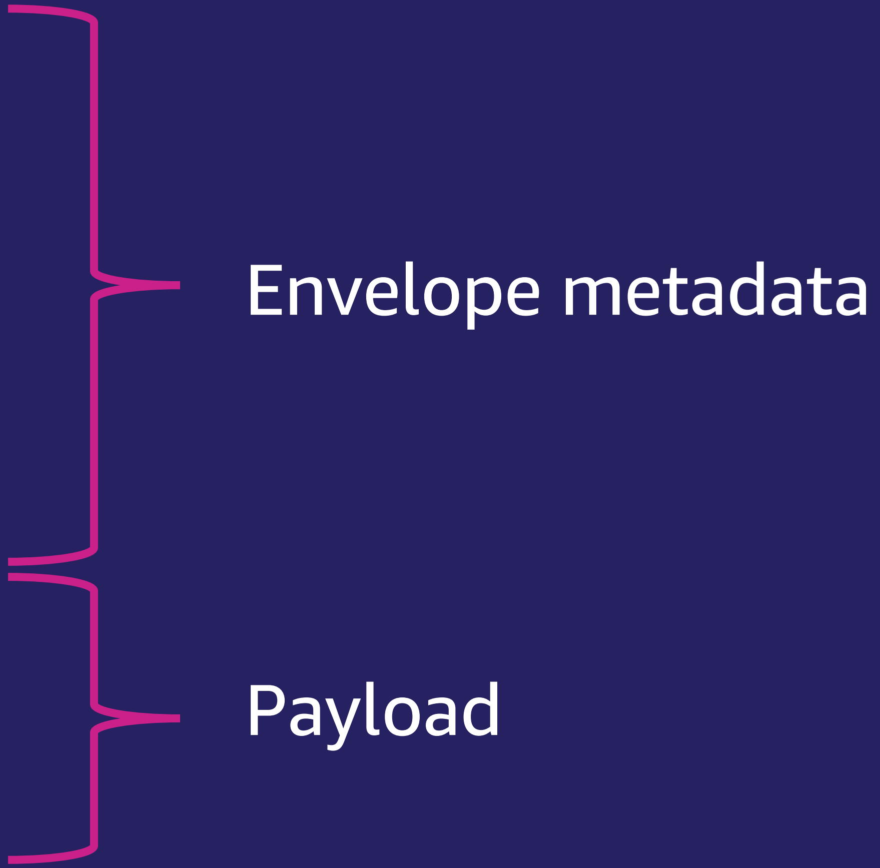
Amazon EventBridge Schema Registry



© 2022, Amazon Web Services, Inc. or its affiliates.

Anatomy of an EventBridge 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

© 2022, Amazon Web Services, Inc. or its affiliates.

When should I use serverless?

An age old question

© 2022, Amazon Web Services, Inc. or its affiliates.

Serverless fits numerous use cases

Typical early serverless uses include:

- IT Automation
- Microservices
- Data processing

If the workload is event-driven, stateless, and can be performed in under 15 minutes … it may be a good fit for serverless

© 2022, Amazon Web Services, Inc. or its affiliates.

Keep learning!

- https://explore.skillbuilder.aws/learn
- https://workshops.aws/
- https://catalog.us-east-1.prod.workshops.aws/workshops/63320e83-6abc-493d-83d8-f822584fb3cb/en-US/getting-started

© 2022, Amazon Web Services, Inc. or its affiliates.

Thank you!

https://aws.amazon.com/serverless/

© 2022, Amazon Web Services, Inc. or its affiliates.