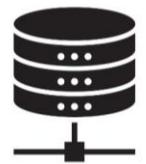




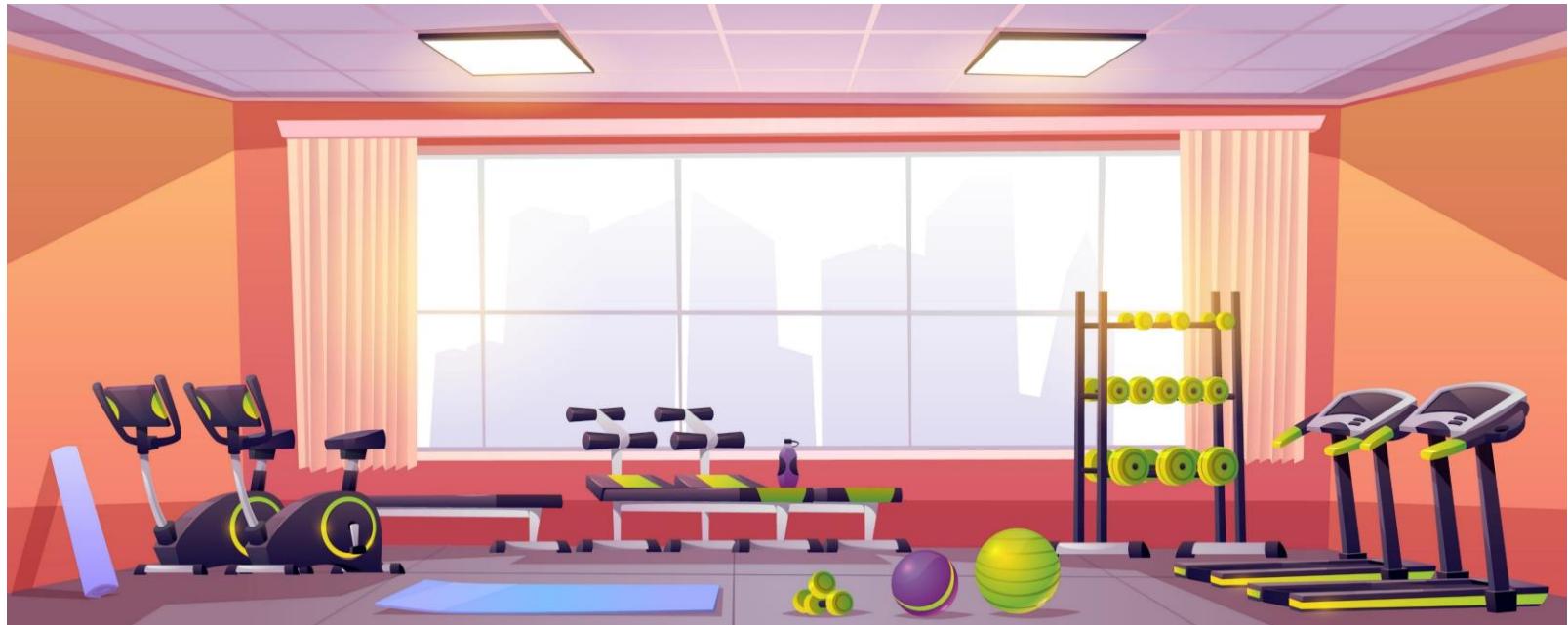
Serverless Architectures





What is serverless?

Some services in real life

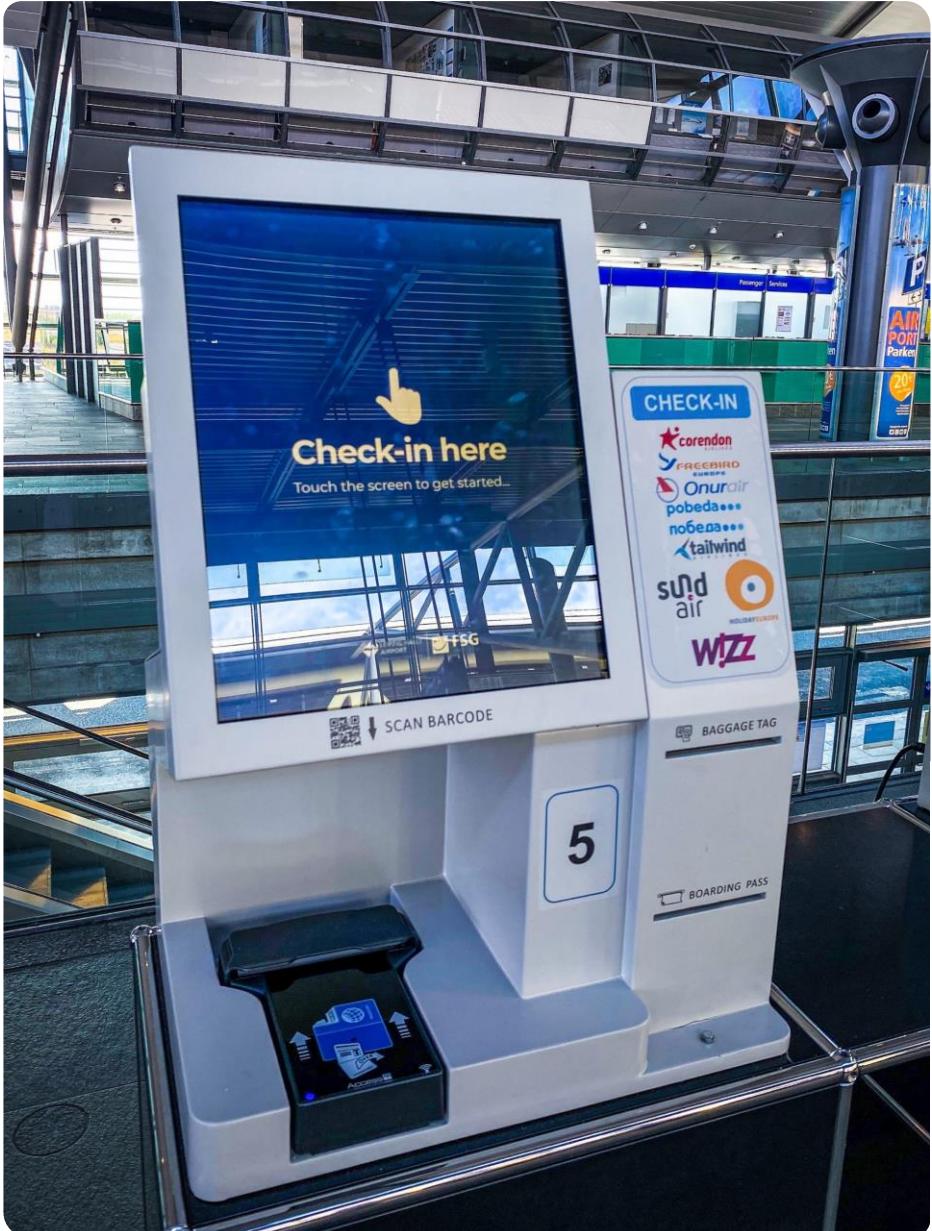


What is serverless?

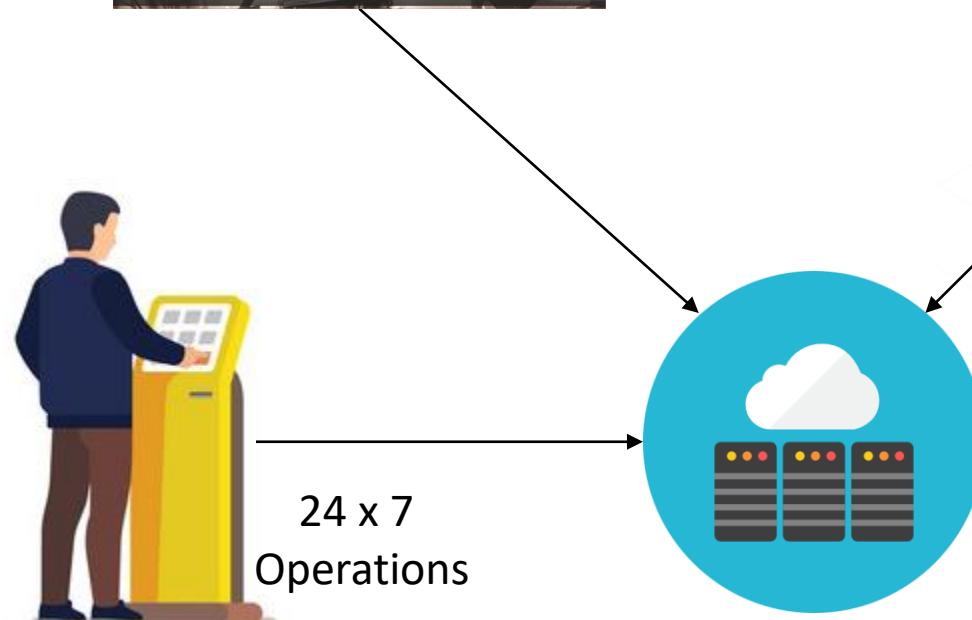
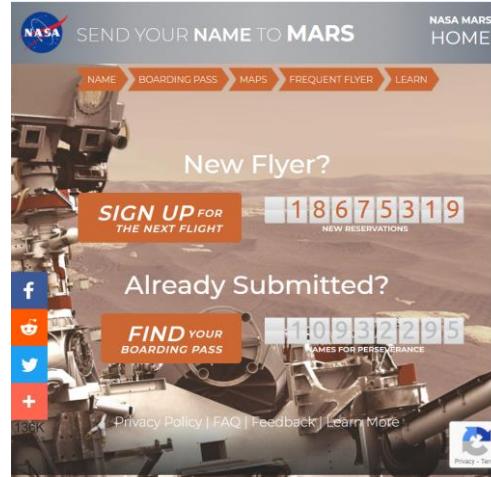
- Serverless is a way to build and run applications and services without having to manage infrastructure.
- Your application still runs on servers, but all the server management is done by AWS.
- You no longer have to provision, scale, and maintain servers to run your applications, databases, and storage systems.



Why serverless?



Web Applications



ChatBots



24 x 7 Running Backend
Delivers some service

- Mostly idle
- Costing Money
- Operational overhead

Why serverless?



No server
management



Flexible
scaling



Pay for
value



Automated high
availability

Some Serverless Services in AWS

Compute



AWS
Lambda



AWS
Fargate

API Proxy



Amazon
API Gateway

Database



Amazon
DynamoDB



Amazon
Aurora

Inter-process Messaging



Amazon
SNS



Amazon
SQS

Orchestration



AWS Step
Functions

Storage



Amazon
S3

Analytics



Amazon
Kinesis



Amazon
Athena

And many more...

Service Summary Cards (SSC)

Reference:

[Deep Dive](#)

Category:

Serverless



Serverless Services
In Cloud

More SSCs:

[Click Here](#)

Complete Book

[Click Here](#)

Created by:

Ashish Prajapati



What?

- A serverless service is a building block to build and run applications without having to manage infrastructure. Your application still runs on servers, but all the server management is done by cloud services provider.
- Serverless services automatically scale from zero to peak demands, so you can adapt to customer needs faster.

Why?

- You can use serverless services to build nearly any type of application or backend service, and everything required to run and scale your application with high availability is handled for you.
- You can build your application quickly and iterate to get to market faster.

When?

- You want to focus on your core product instead of worrying about managing and operating servers or runtimes. This reduced overhead lets you reclaim time and energy that you can spend on developing great products which scale and that are reliable.

Where?

- Serverless applications are generally built using fully managed services as building blocks across the compute, data, messaging and integration, streaming, and user management and identity layers which operate in cloud provider's infrastructure. For testing purposes you can also run some serverless services on-prem.

Who?

- With serverless, the cloud provider manages layers of infrastructure, including operating systems and networking.
- Serverless provides built-in availability and fault tolerance.

How?

- Serverless applications are event-driven and loosely coupled via technology-agnostic APIs or messaging. Event-driven code is executed in response to an event, such as a change in state or an endpoint request. Event-driven architectures decouple code from state and integration between loosely coupled components is usually done asynchronously, with messaging.

How
much?

- With serverless you pay for consistent throughput or execution duration rather than by server unit.
- With a pay-for-value billing model, resource utilization is automatically optimized and you never pay for over-provisioning.



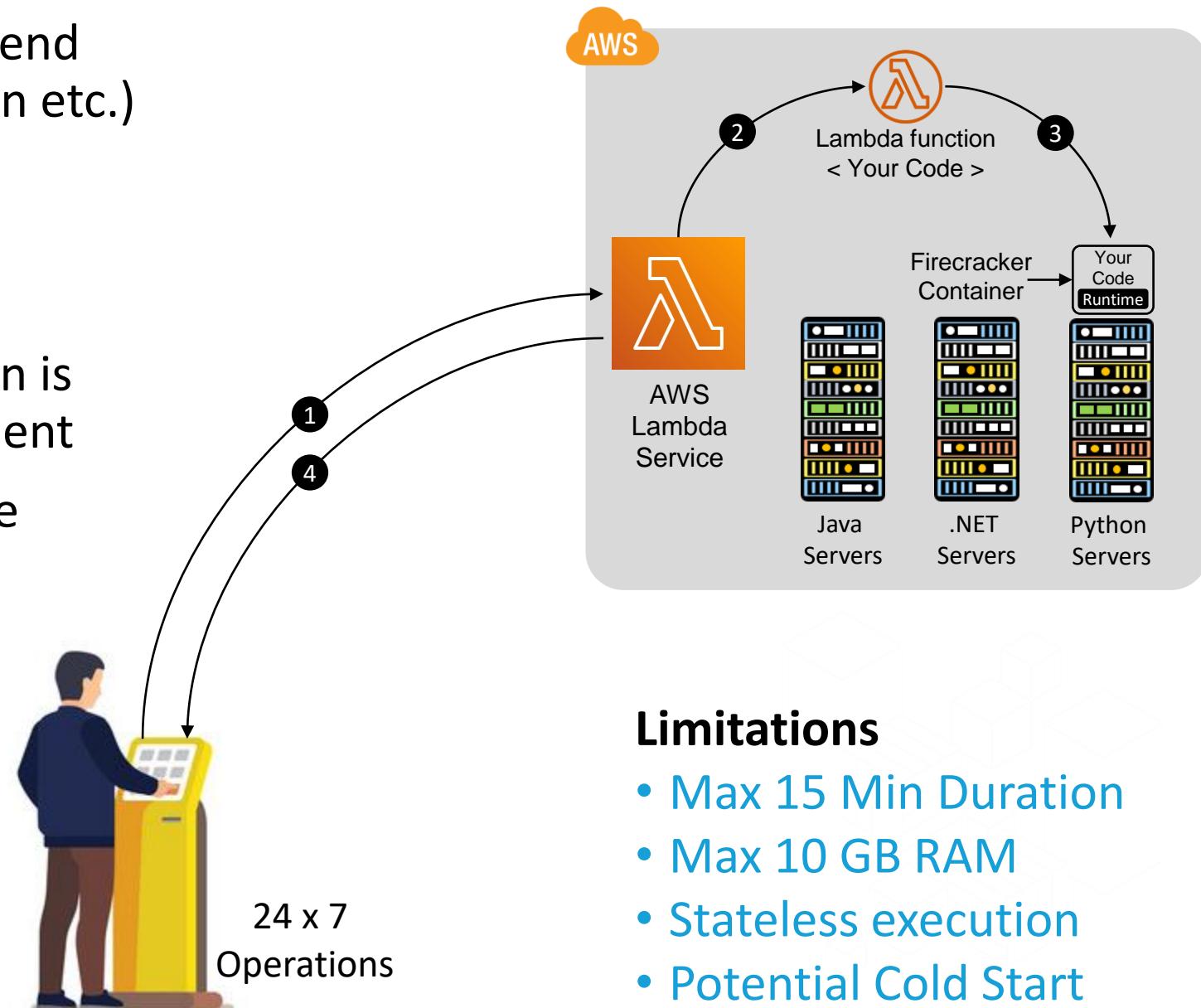
AWS Lambda

How compute evolved in AWS?

	<p>The diagram illustrates the evolution of AWS compute. It starts with 'Virtual Machines' represented by three boxes labeled 'VM' stacked vertically, with a 'Hypervisor' layer above them and a 'Server' layer at the bottom. An arrow points to the right, leading to 'Containers' represented by a black Docker ship icon. Another arrow points to the right, leading to 'Functions' represented by a cloud icon containing a computer monitor with code brackets (less than and greater than signs) and a network connection below it.</p> <p>Virtual Machines</p> <p>Containers</p> <p>Functions</p>		
AWS Services	Amazon EC2	Amazon ECS / Amazon EKS	AWS Lambda
Virtualization / Isolation	Hardware	Operating System	Runtime
Packaging	AMI	Container Image	Function
Scalability	Controlled by you	Controlled by you	Implicit scaling
Pricing	Infrastructure based	Infrastructure based	Execution duration

How AWS Lambda works?

1. An event is triggered from frontend
(Click / API Call / Fingerprint scan etc.)
2. AWS Lambda Service receives invocation call and triggers corresponding Lambda function
3. Code inside the Lambda function is executed in a suitable environment
4. Results are delivered back to the frontend (optional)



Advantages

- No servers to manage
- Millisecond billing
- Automatic scaling
- Rich ecosystem

Limitations

- Max 15 Min Duration
- Max 10 GB RAM
- Stateless execution
- Potential Cold Start

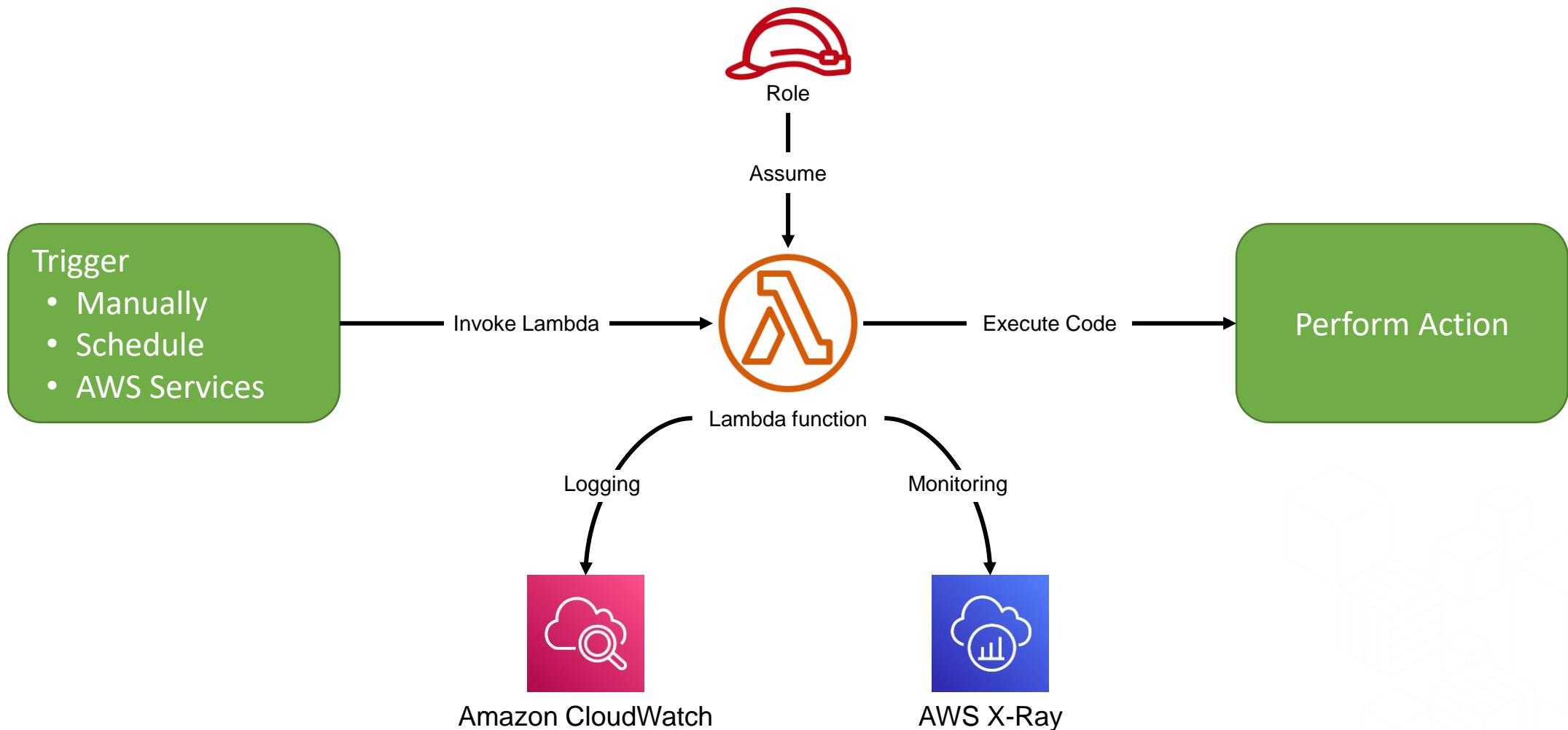
Will Lambda suit all types of work load?



AWS Lambda



AWS Lambda Function Execution



The Swiss Army Knife of Serverless – AWS Lambda

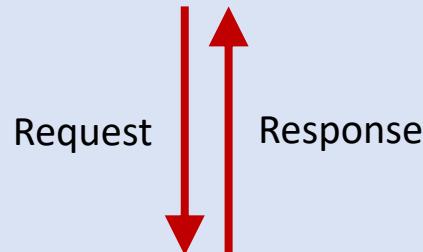


The AWS Lambda invocation model

Synchronous Request/Response

-  Application Load Balancer
-  Amazon API Gateway
-  AWS Lambda function URL

and more...

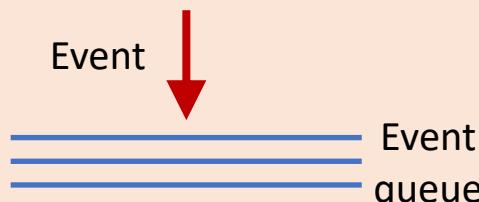


AWS Lambda function

Asynchronous Event

-  Amazon EventBridge
-  Amazon S3
-  Amazon SNS

and more...



AWS Lambda function

Asynchronous Polling

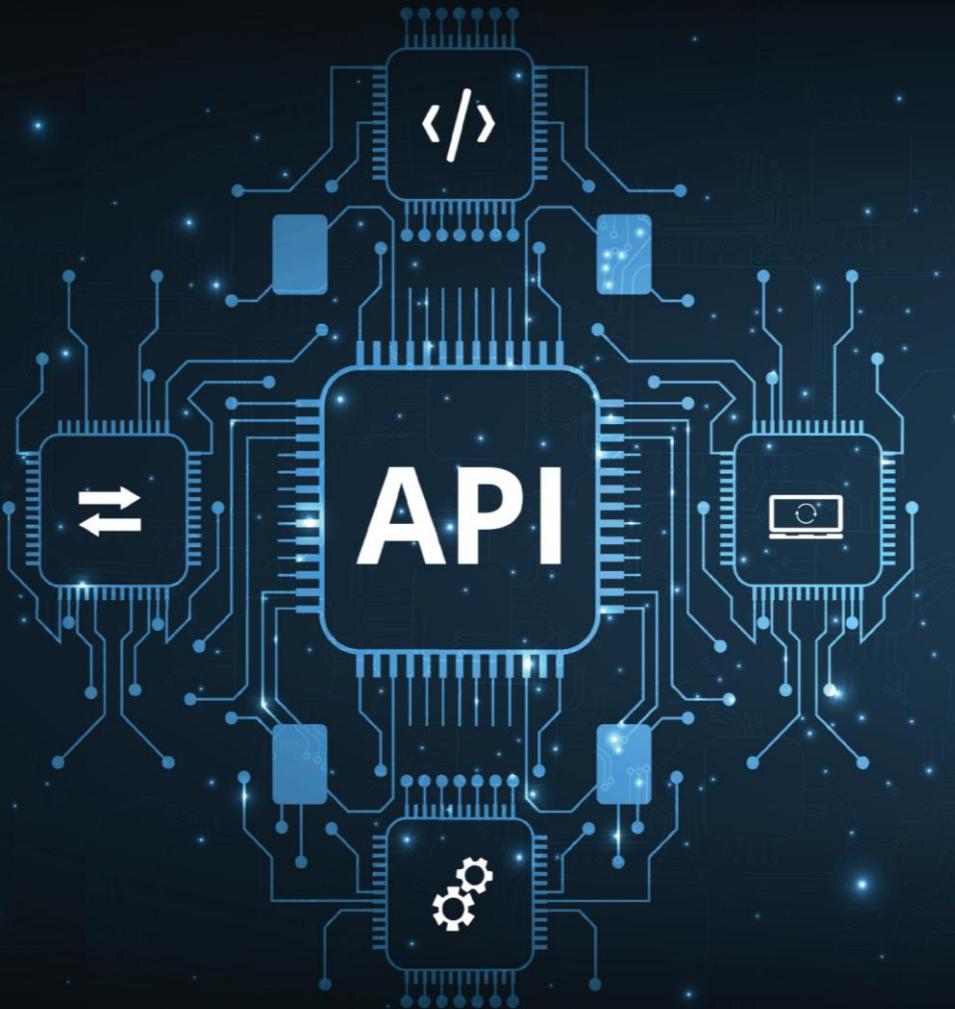
- Amazon Kinesis 
 - Amazon SQS 
 - Amazon MQ 
- DynamoDB Streams 
 - Amazon MSK 
 - Apache Kafka 

and more...

Event source mapping



AWS Lambda function

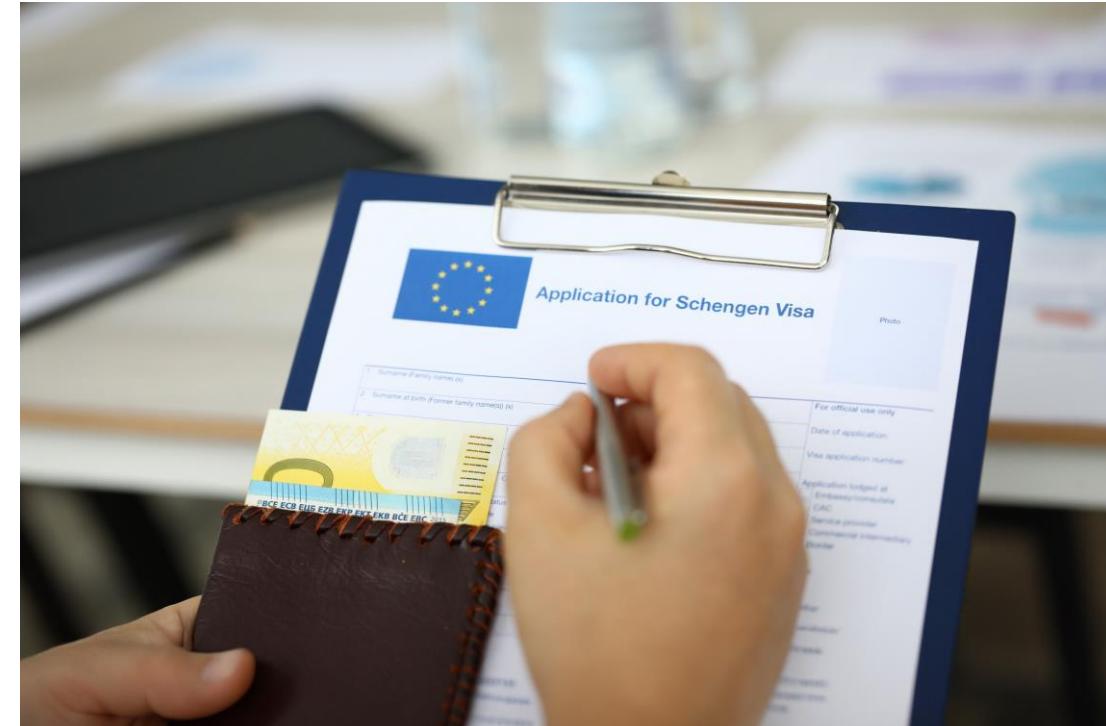


What is an API?

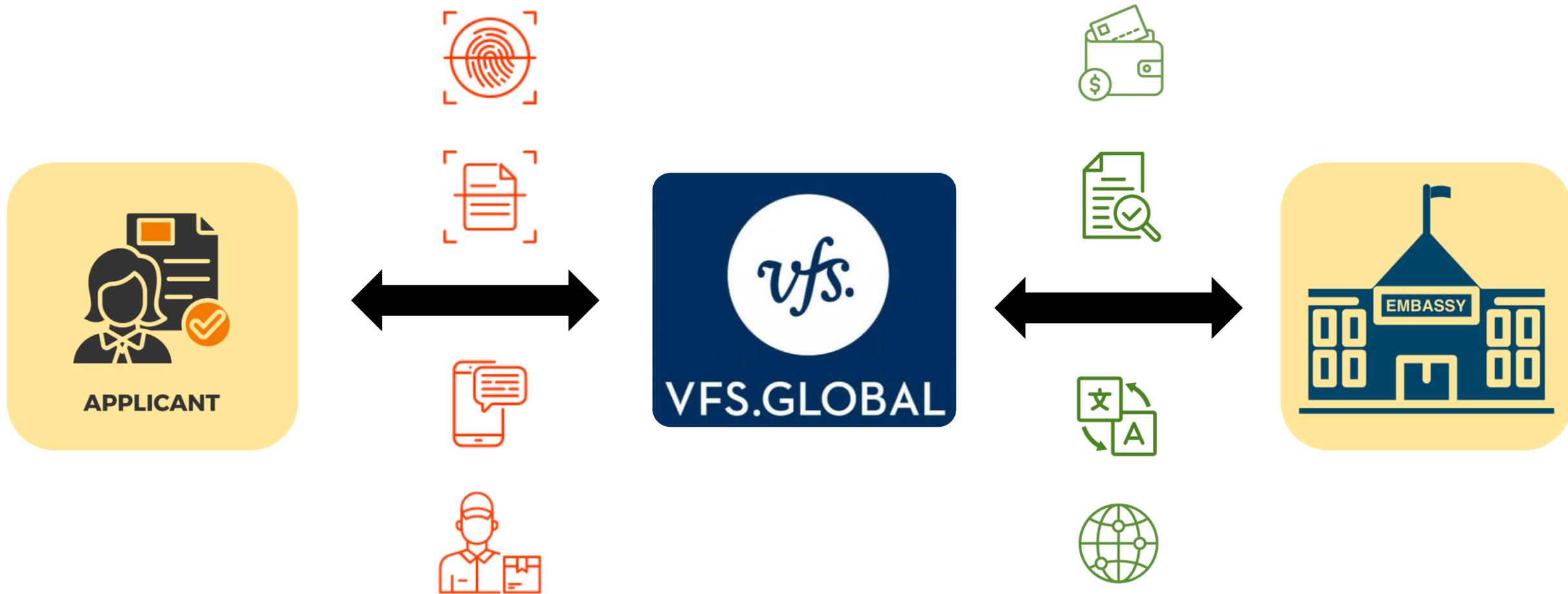


An Analogy - What is an API?

- Have you ever applied for a Visa?

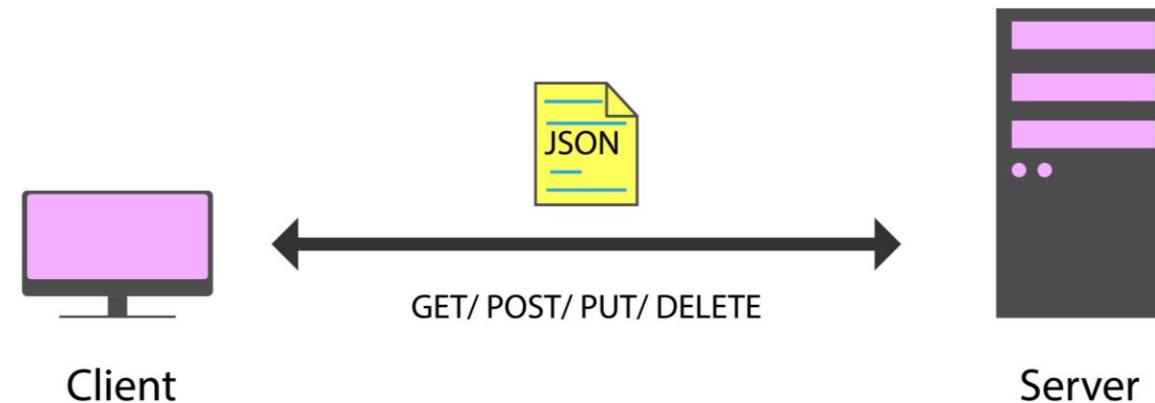


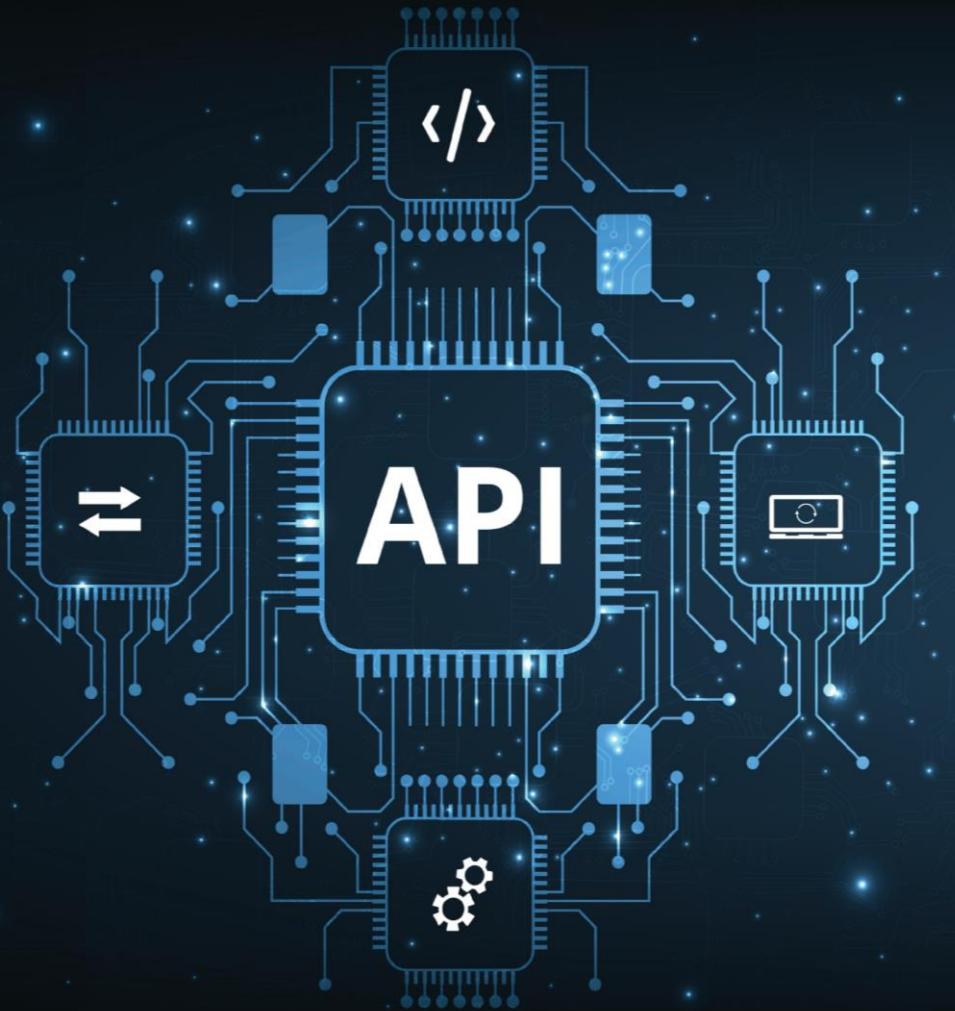
An Analogy - What is an API?



What is an API (Application Programming Interface)?

- API is the middleman – It creates a handshake between the software you use and the data you need.
- An API is designed to be directly consumed by another piece of software running somewhere on a network.
- API interfaces are documented, consistent and predictable.

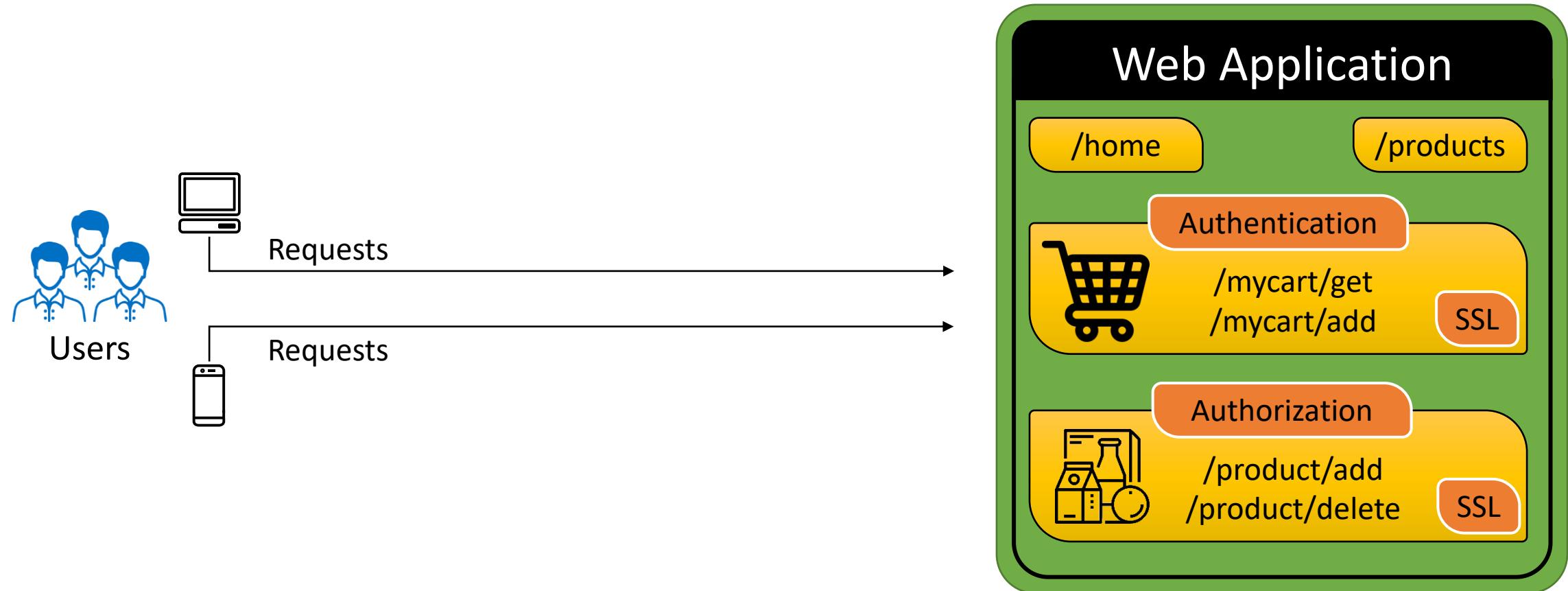




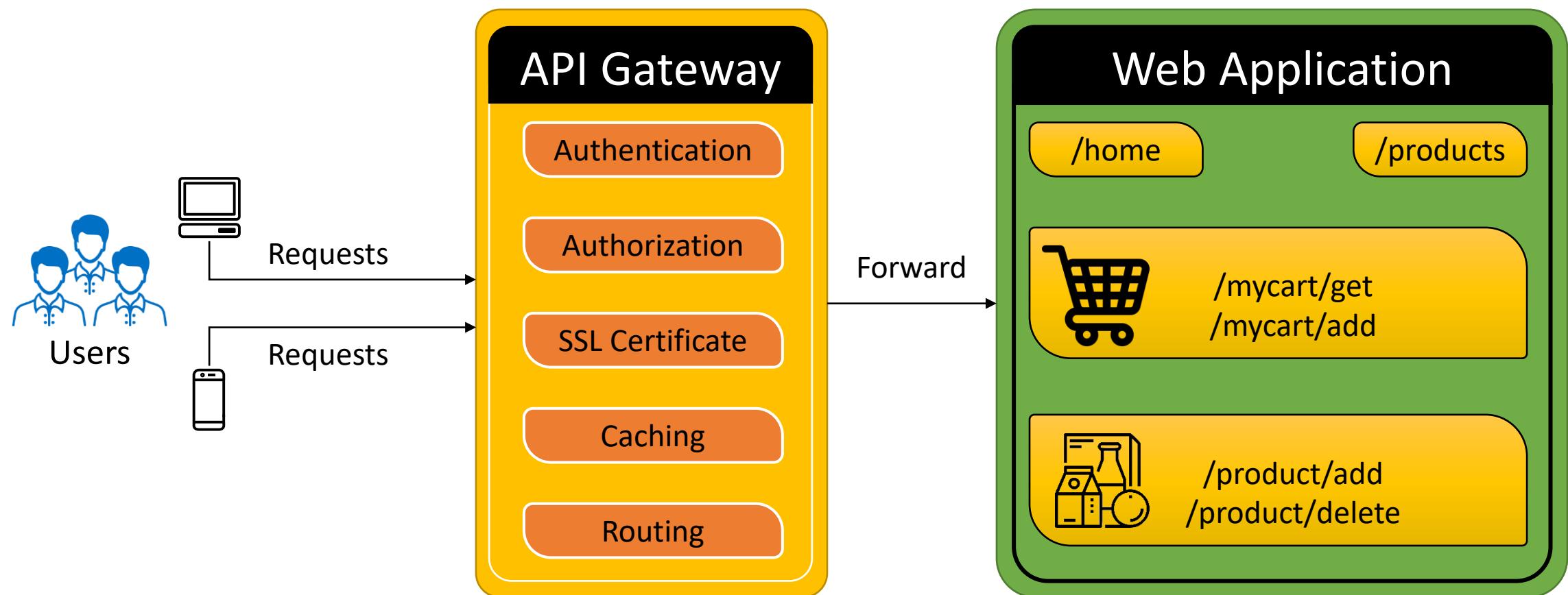
Why we need API Gateway?



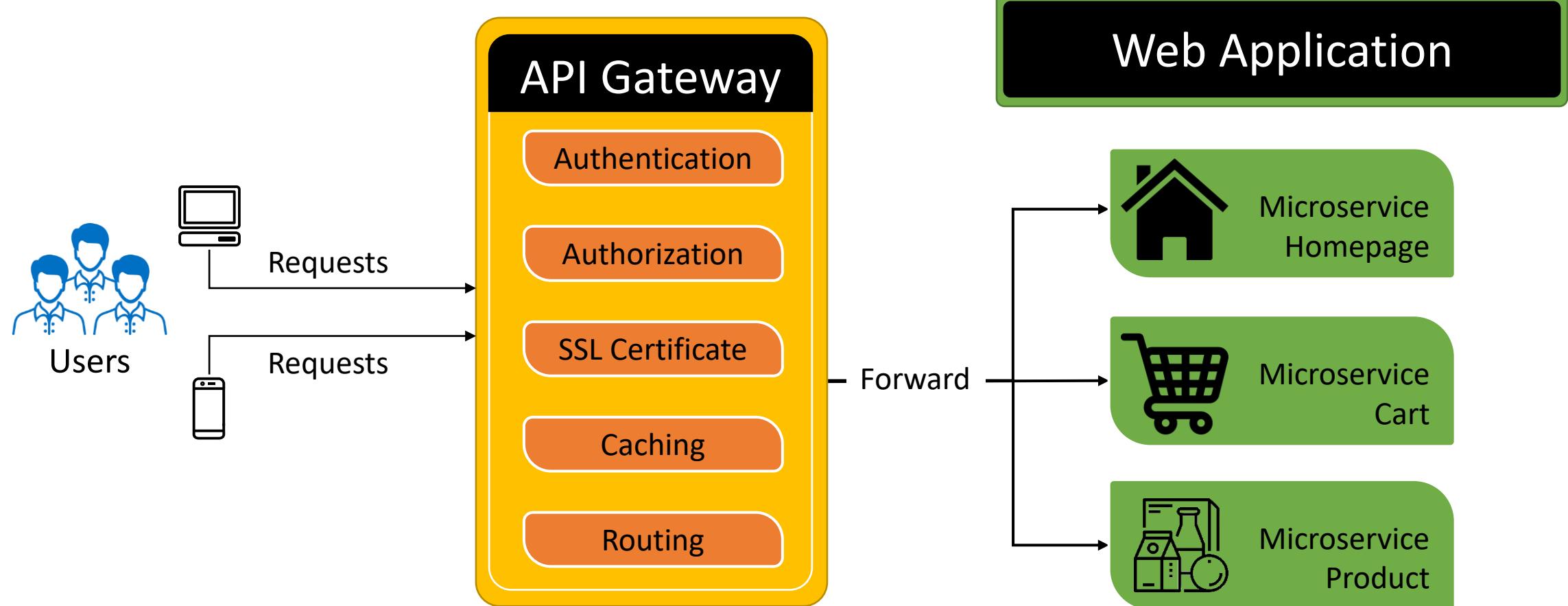
E-Commerce Application

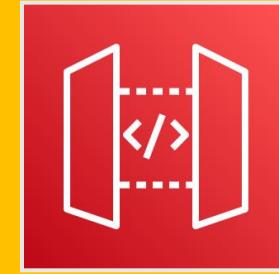
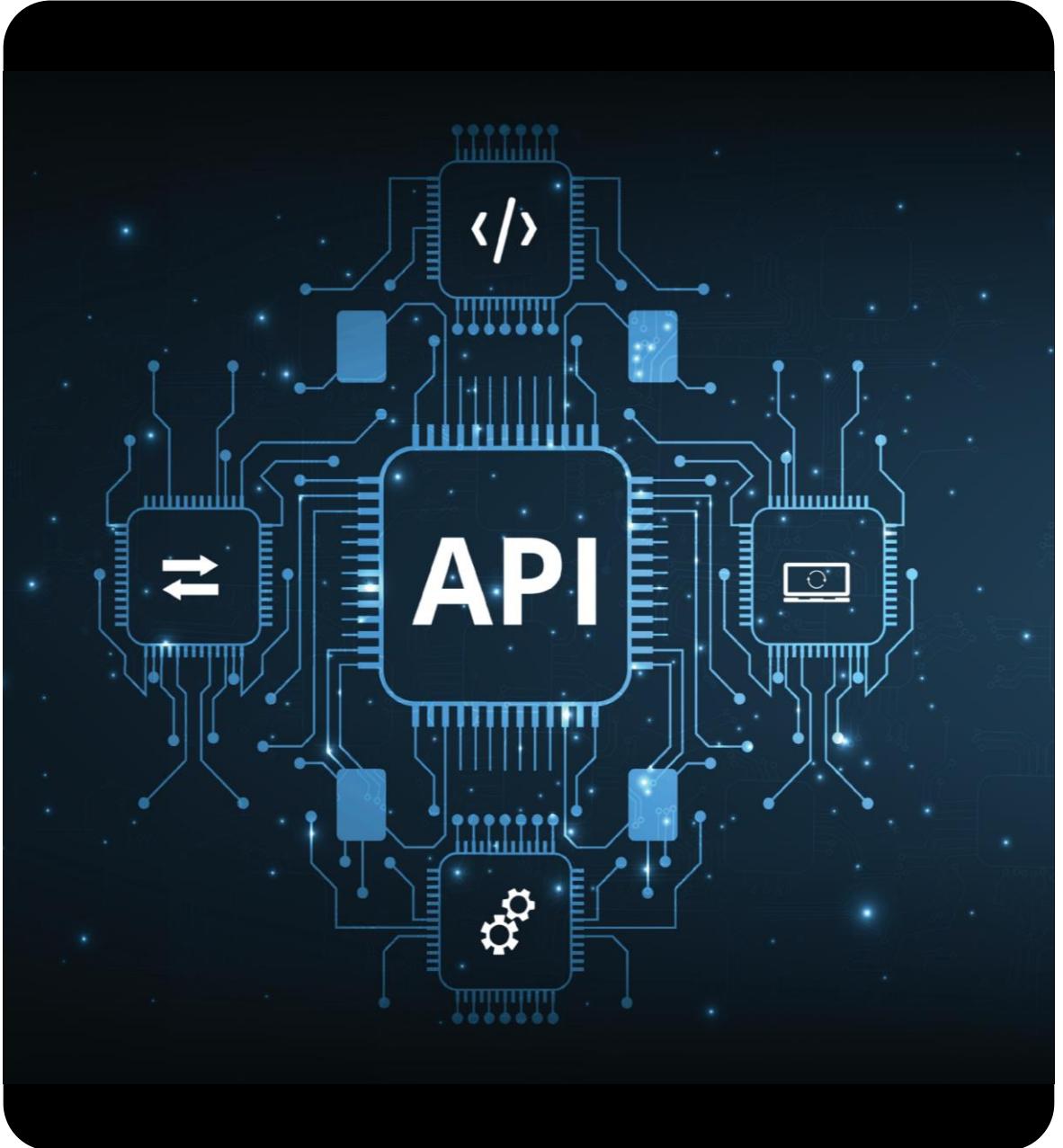


E-Commerce Application with API Gateway



E-Commerce Application with API Gateway using Microservices





Amazon API Gateway

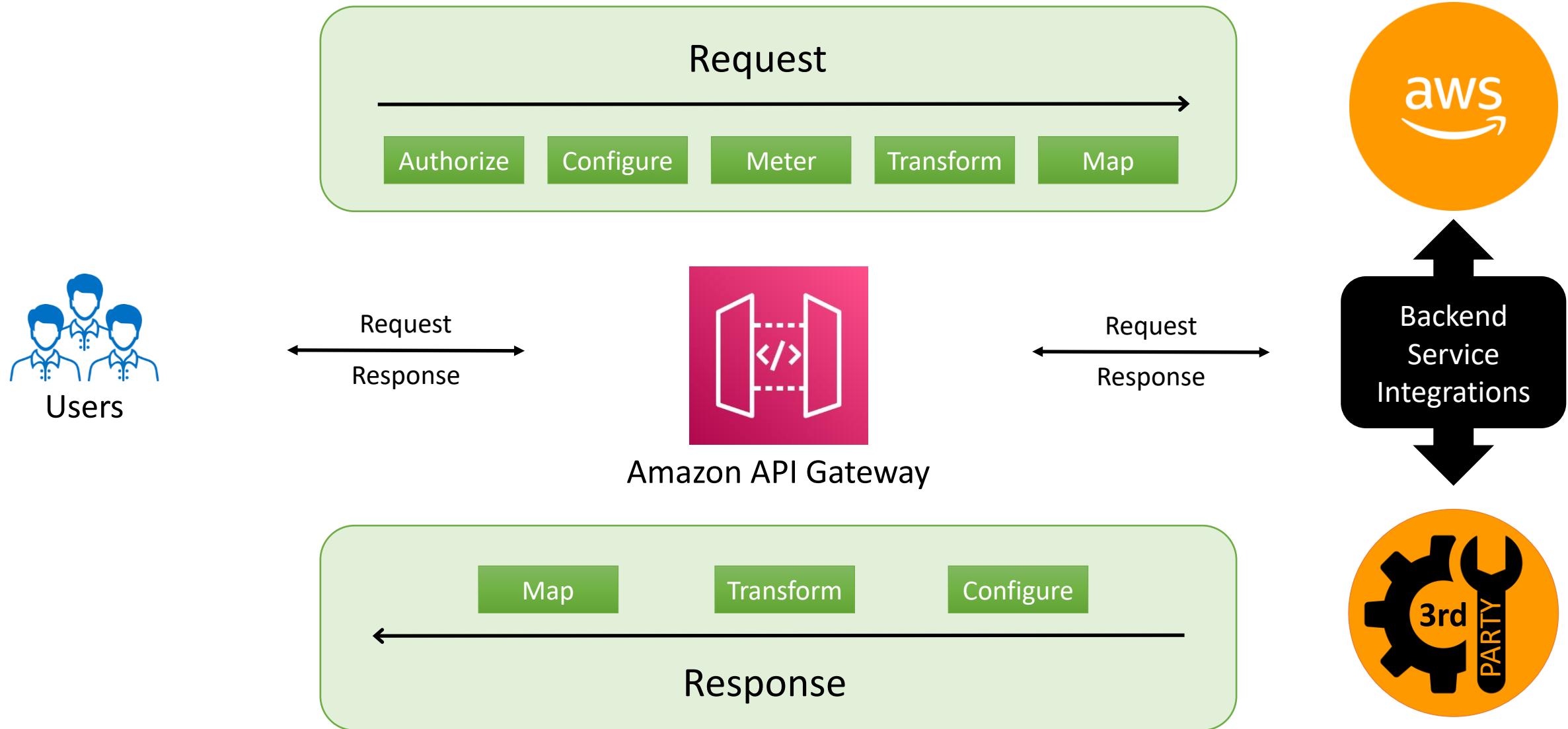


Reference:

FAQs

- 
- Amazon API Gateway
- What?
 - Amazon API Gateway is a fully managed service that makes it easy for developers to create, publish, maintain, monitor, and secure APIs at any scale. API Gateway supports containerized and serverless workloads, as well as web applications.
 - It allows you to create RESTful APIs and WebSocket APIs that enable real-time two-way communication applications.
 - Why?
 - Amazon API Gateway provides developers with a simple, flexible, fully managed, pay-as-you-go service that handles all aspects of creating and operating robust APIs for application back ends. With API Gateway, you can launch new services faster and with reduced investment so you can focus on building your core business services.
 - When?
 - You want to use a managed service to save undifferentiated heavy lifting involved in securely and reliably running APIs (REST, HTTP, and WebSocket APIs).
 - To save effort on API development and API management and generate client SDKs for a number of languages.
 - Where?
 - Amazon API Gateway is a regional service.
 - An API endpoint refers to the hostname of the API. For the REST APIs the API endpoint type can be edge-optimized, regional, or private, depending on where the majority of your API traffic originates from.
 - Who?
 - Amazon API Gateway handles all of the tasks involved in accepting and processing concurrent API calls, including traffic management, authorization and access control, monitoring, and API version management.
 - As an API developer, you can create and manage an API by using the API Gateway console, or by calling the API references.
 - How?
 - API Gateway acts as a "front door" for applications to access data, business logic, or functionality from your backend services, such as workloads running on Amazon EC2, code running on AWS Lambda, any web application, or real-time communication applications.
 - How much?
 - For HTTP APIs and REST APIs, you pay only for the API calls you receive and the amount of data transferred out.
 - For WebSocket APIs, you pay for messages sent and received and for the time a user/device is connected to the WebSocket API.
 - API Gateway also provides optional data caching charged at an hourly rate that varies based on the cache size you select.

Amazon API Gateway



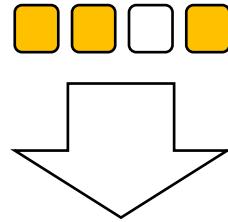
A Simple Demo



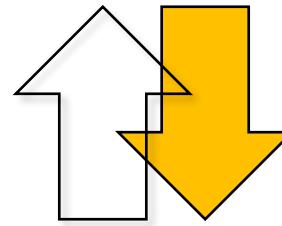
Amazon API Gateway tutorials and workshops

- <https://docs.aws.amazon.com/apigateway/latest/developerguide/api-gateway-tutorials.html>

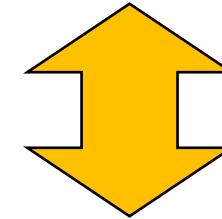
Protocols for APIs



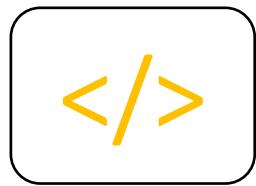
GraphQL



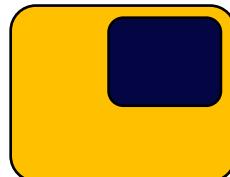
REST



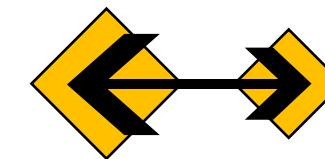
WebSockets



SOAP

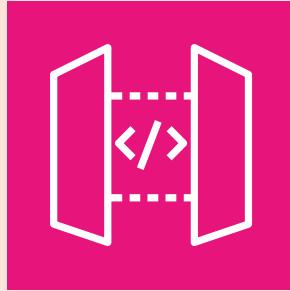


RPC

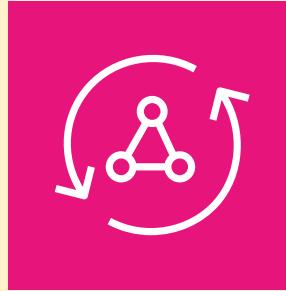


gRPC

AWS Services supporting these protocols



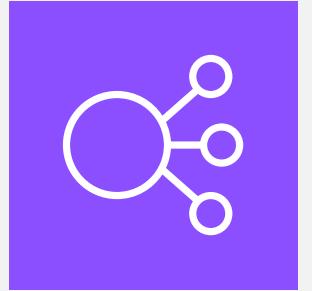
Amazon API Gateway



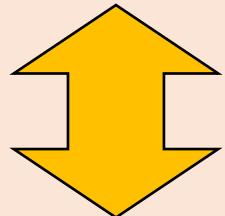
AWS AppSync



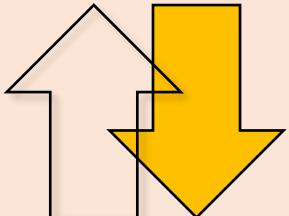
AWS IoT Core



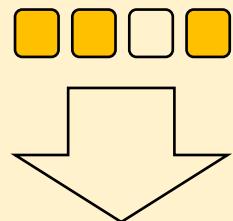
Elastic Load Balancing (ELB)



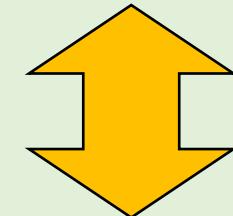
WebSockets



REST



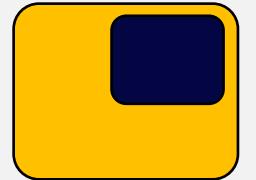
GraphQL



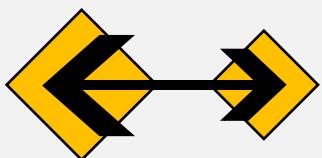
WebSockets



SOAP



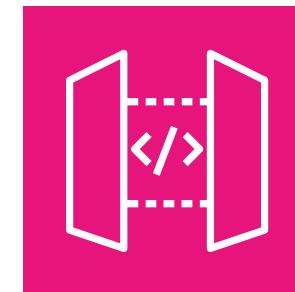
RPC



gRPC

Two types of Amazon API Gateway APIs

Both are HTTP-based
API providers



Amazon API Gateway

REST API

- Protocols
 - REST
 - WebSocket
- Full featured

HTTP API

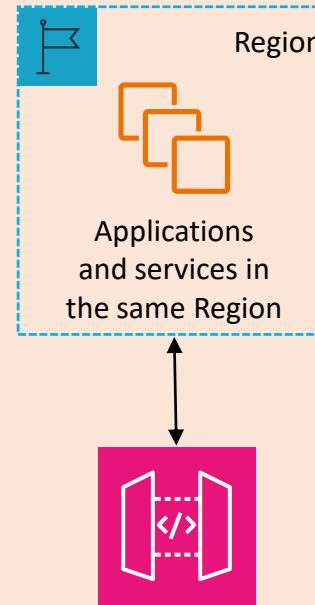
- REST only
- Simple interface

API endpoint type

Decide based on where the majority of your API traffic originates from.

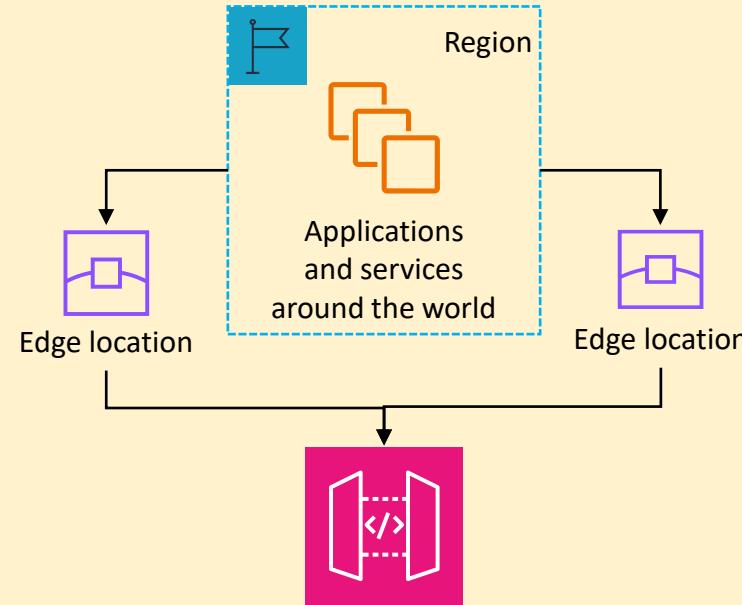
Regional

Deployed in the same AWS Region where you applications are



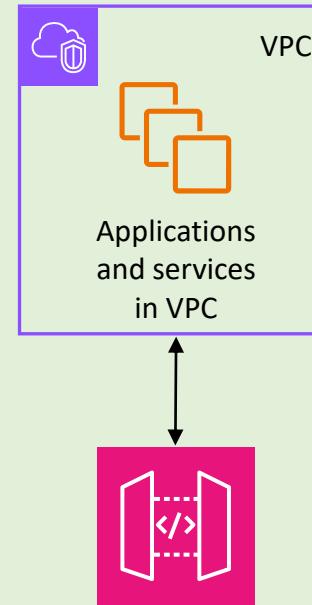
Edge-optimized

Routes requests through the nearest CloudFront Point of Presence



Private

Only accessible from VPCs



When you have a small number of clients with high demands, a regional API reduces connection overhead.

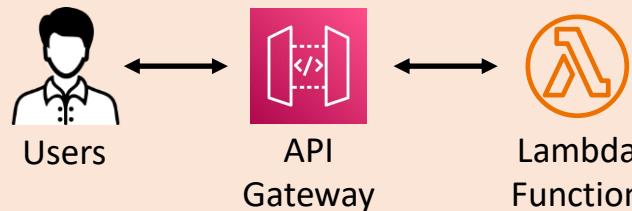
When your clients are geographically distributed.

When you want to access the APIs only from within your VPCs and keep it isolated from the public internet.

Amazon API Gateway - Integration types

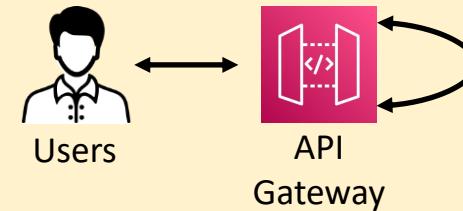
Lambda function

Connect to Lambda function via proxy or direct integration



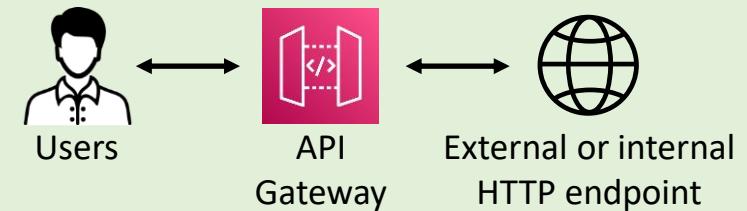
Mock

Respond to requests without a backend service



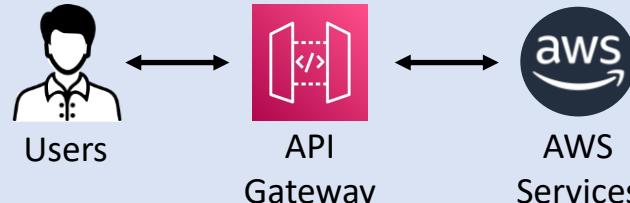
HTTP

Connect to any HTTP(S) endpoint inside



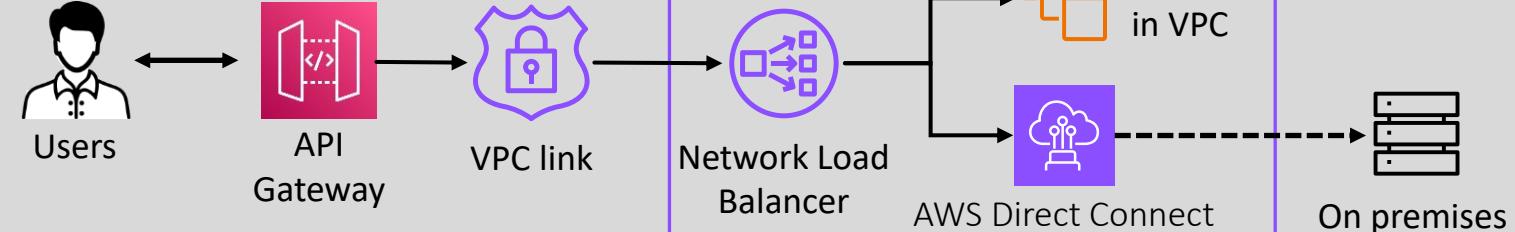
AWS service

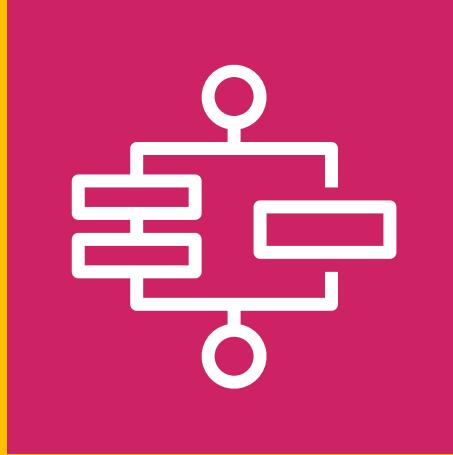
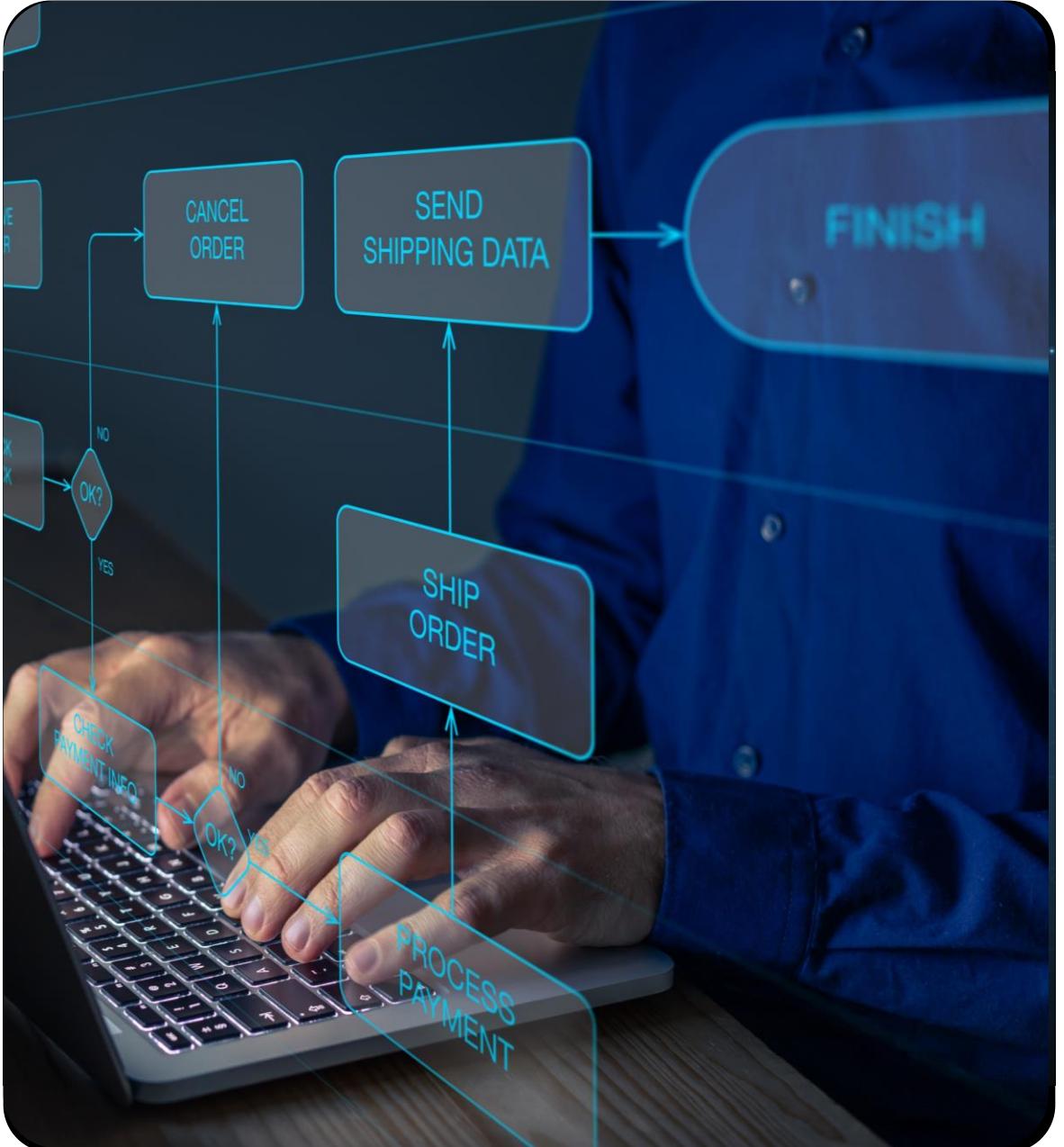
Connect to 100+ service endpoints inside of AWS



VPC Link

Connect to a VPC link that allows access to resources in a VPC via a NLB

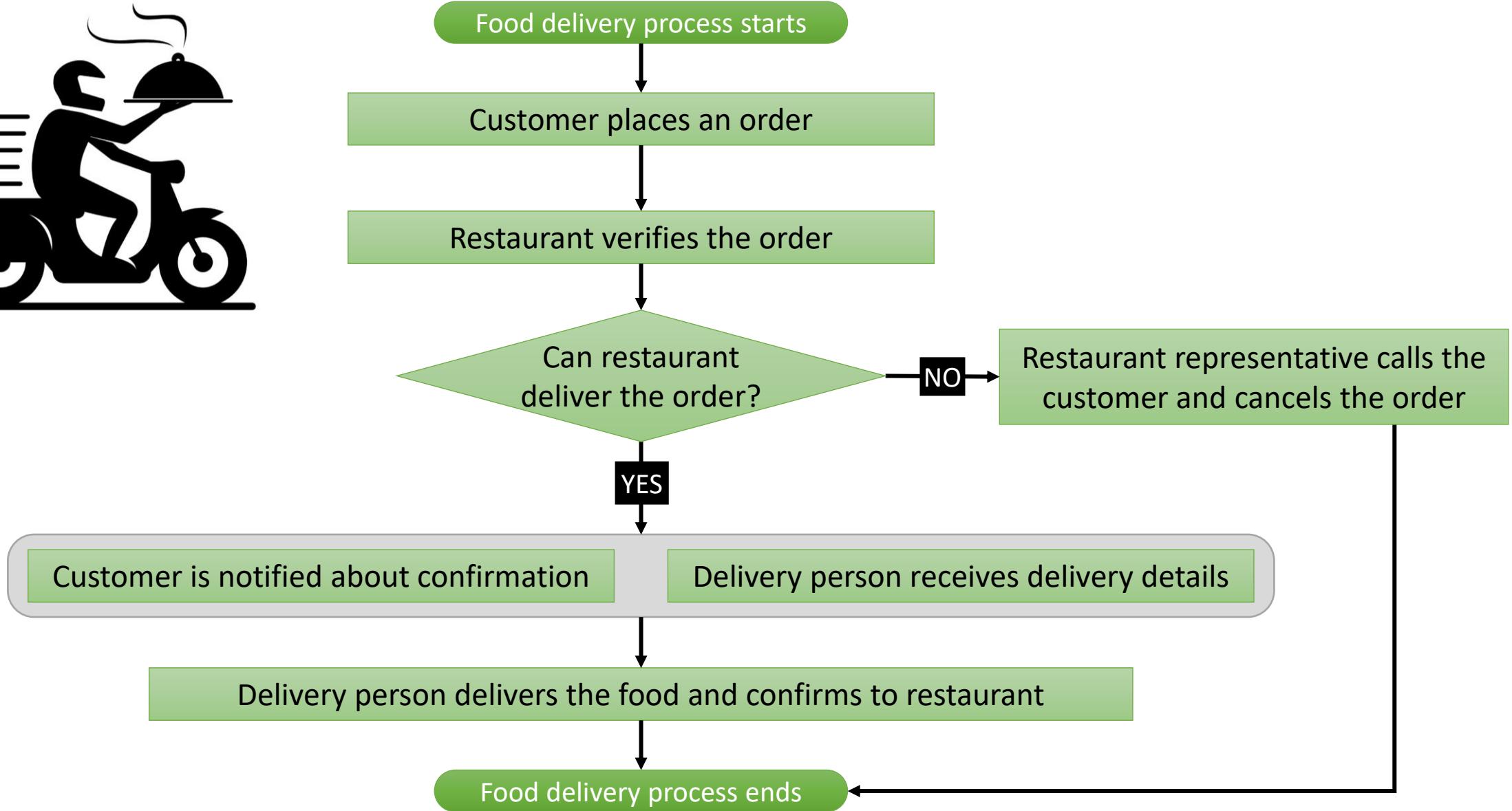




AWS Step Functions



A typical food delivery workflow



Processing a new bank account application

Account Applications



Accept new application

Consolidate data checks

Human review?

Approve or reject

Data Verification



Verify identity documents

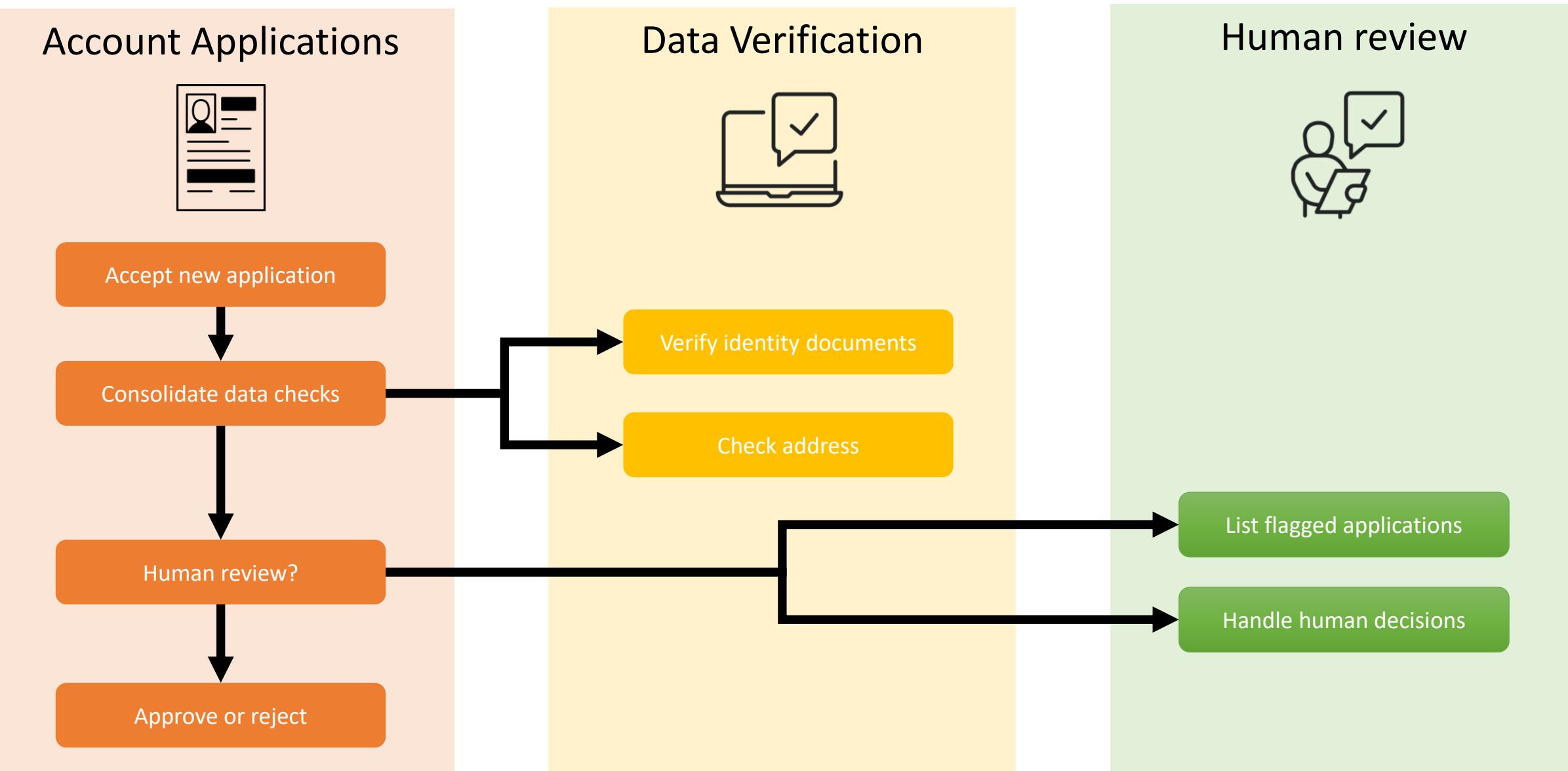
Check address

Human review

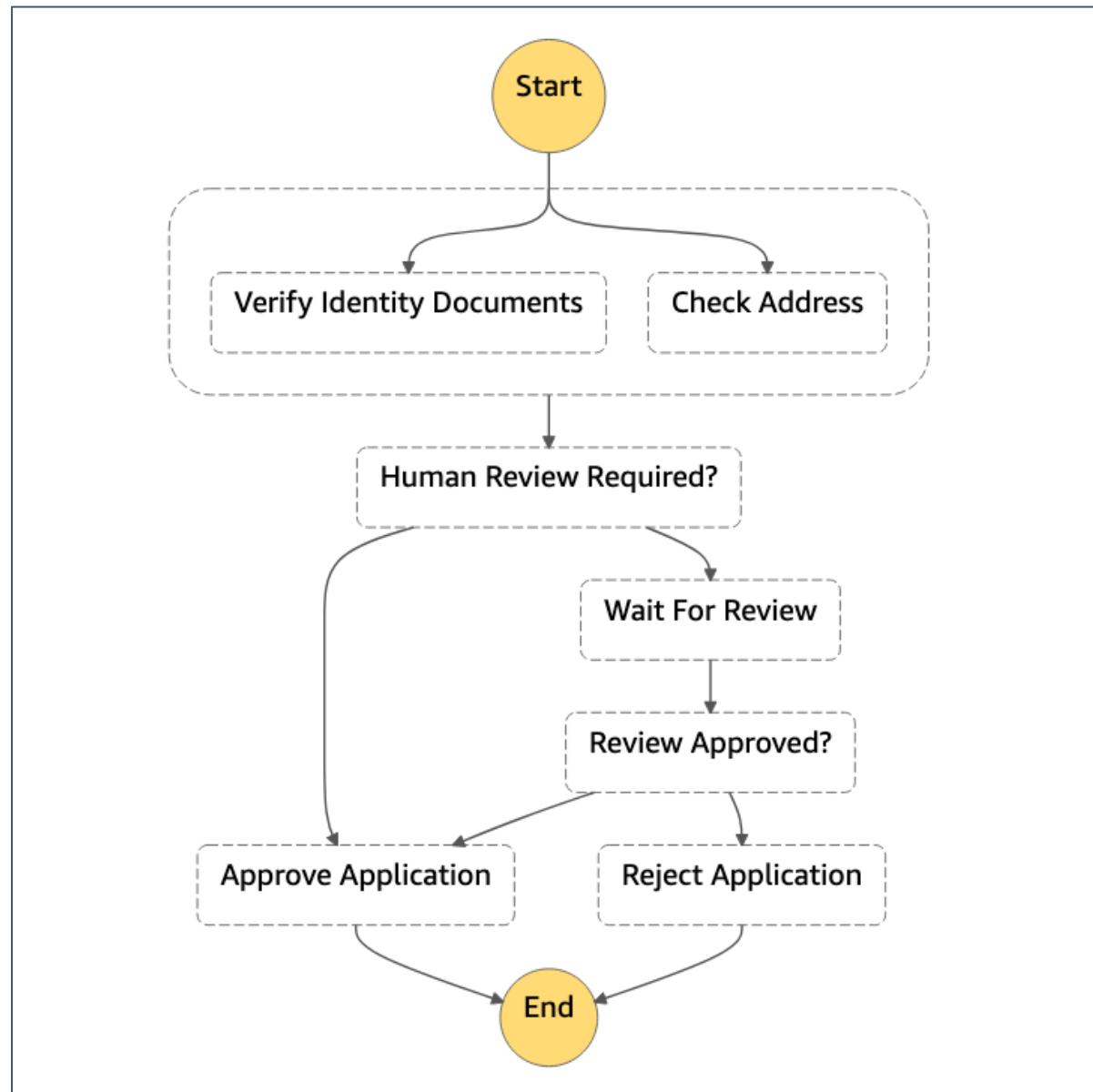


List flagged applications

Handle human decisions

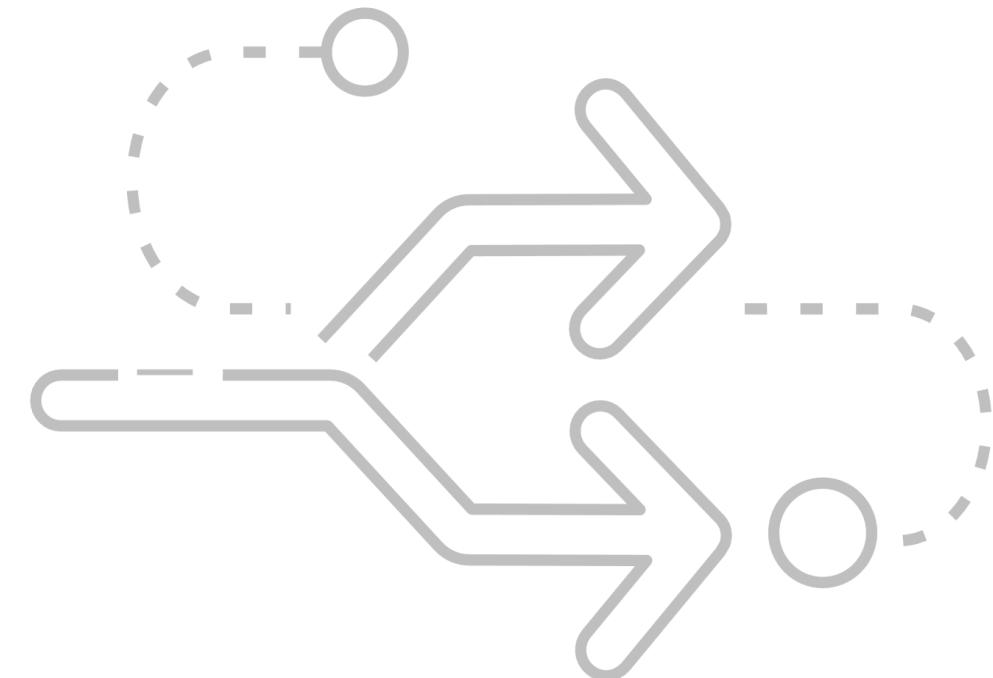


A state machine



A state machine

- Describes a collection of computational steps split into discrete states
- Has one starting state and always one active state (while executing)
- The active state receives input, takes some action, and generates output
- Transitions between states are based on state outputs and rules that we define



AWS Step Functions - Fully managed state machines on AWS

- Resilient workflow automation
- Built-in error handling
- Powerful AWS service integration
- Integrating with your own services
- Visual monitoring
- Auditable execution history

