



# aws INNOVATE

MODERN APPLICATIONS EDITION

27 & 28 October 2021

# Modern applications design patterns: Implementing microservice architectures

Chris Modica  
Principal Solutions Architect  
Amazon Web Services

# About me

chris\_modica\_



## Background

- Application architecture and software development (CTO, VPoE ...)
- Enterprises and scale-ups (ISV, DNBs)
- From Sydney, Australia

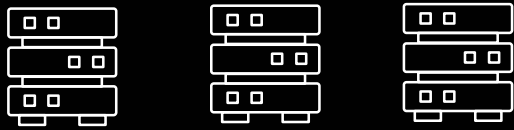
# Modern application architectures

# Traditional three-tier application architecture



Web servers

Presentation layers



Application servers

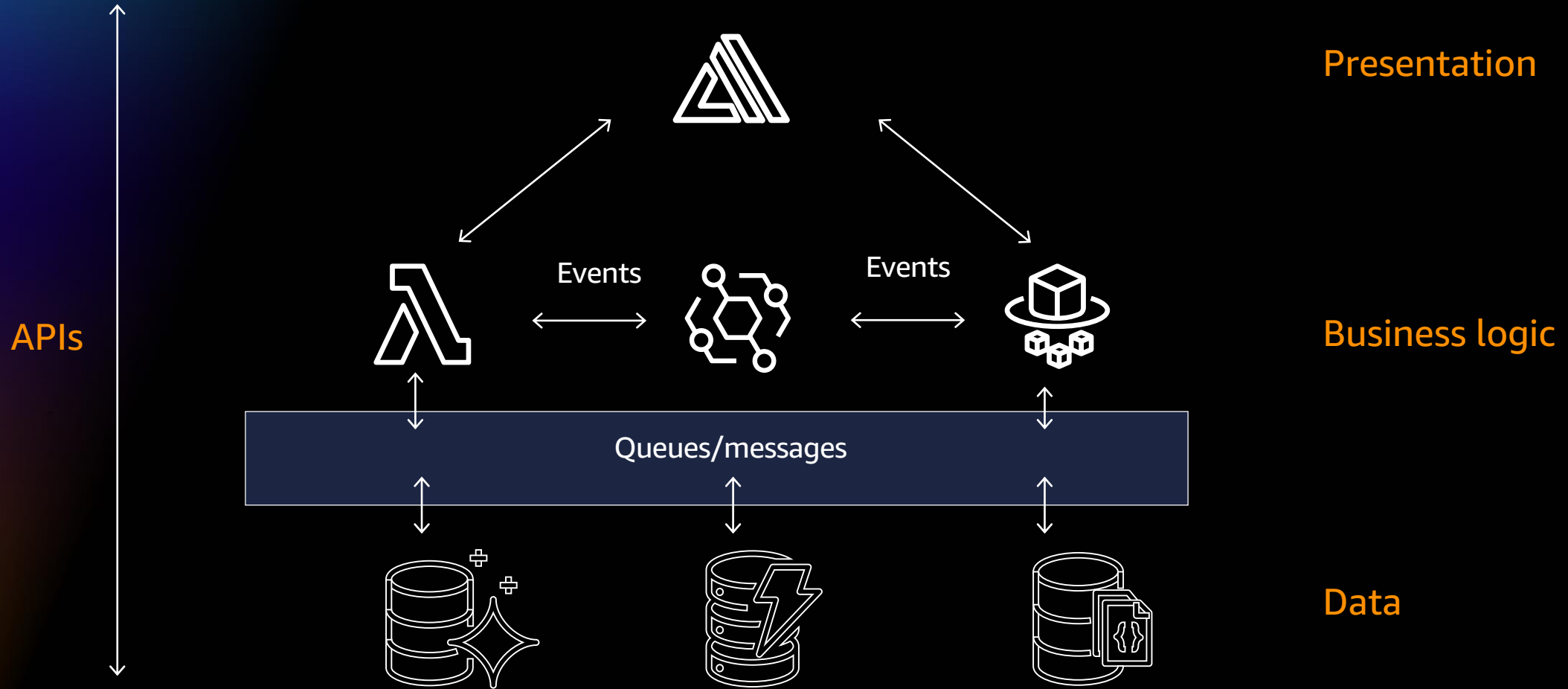
Business logic



Database servers

Data layer

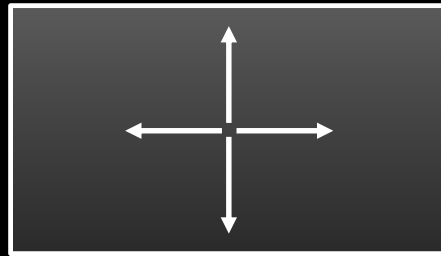
# Modern application architecture



# Why building cloud native matters



**Speed**

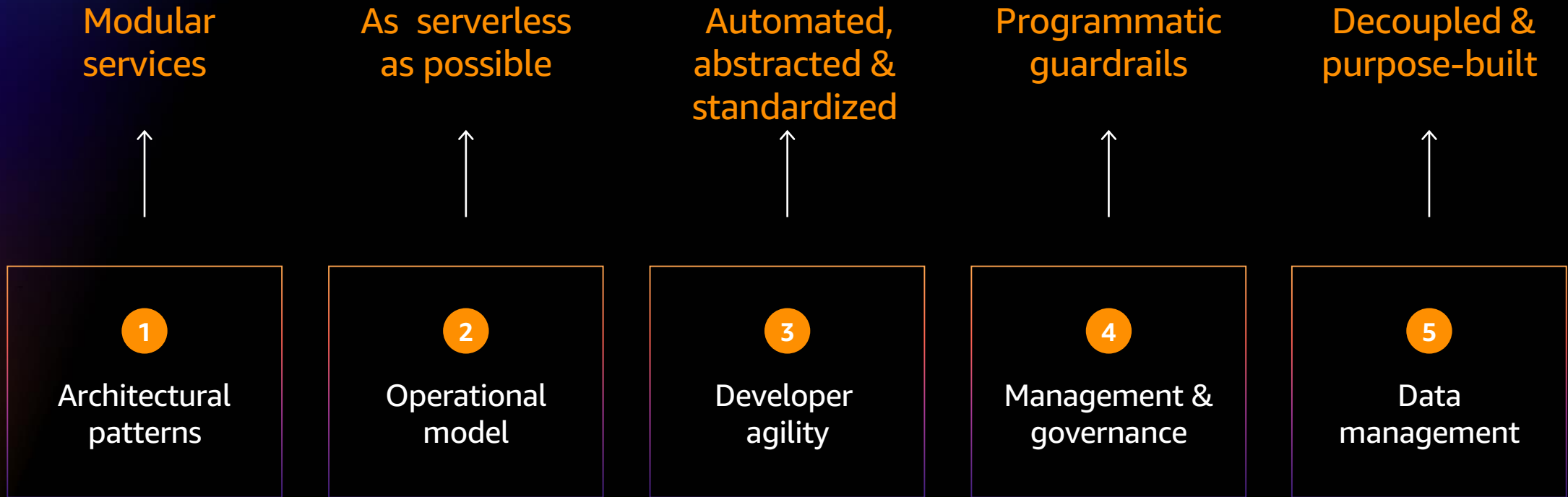


**Scale**



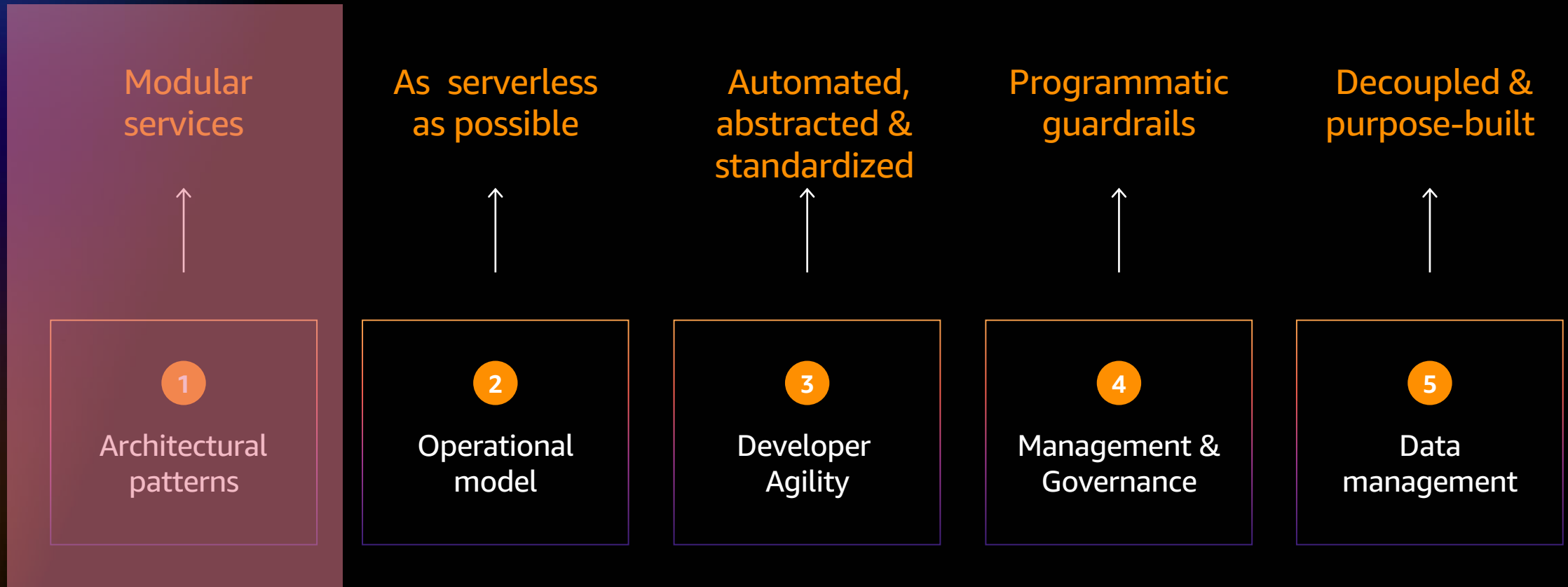
**Resiliency**

# What is the best way to build a modern application?





# What is the best way to build a modern application?



# Architectural principles

# Divide and conquer

## *Breaking up the monolith*

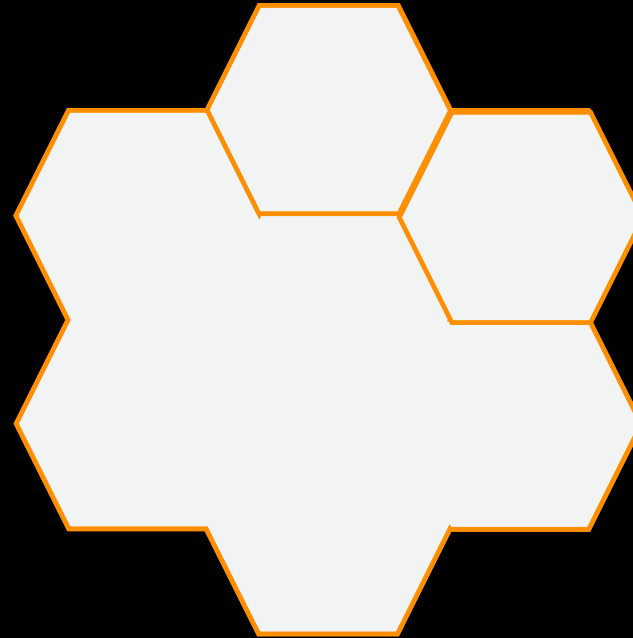
**Use AWS cloud native services with legacy applications**

Decouple functionality for quicker and safer development

Gradually carve out one component or task at a time to refactor

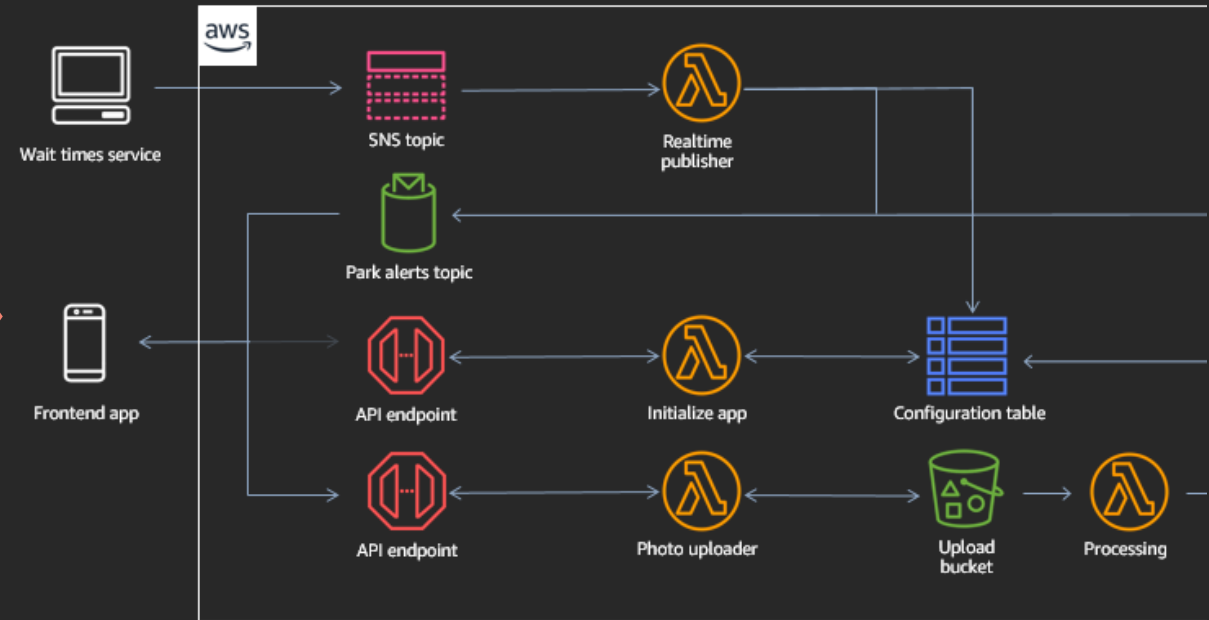
New functionality cloud native, microservice first

Design patterns works with cloud or hybrid applications



# Small pieces, loosely joined

## *Loose coupling*

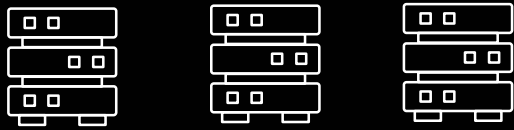


# Traditional three-tier application architecture



Web servers

Presentation layers



Application servers

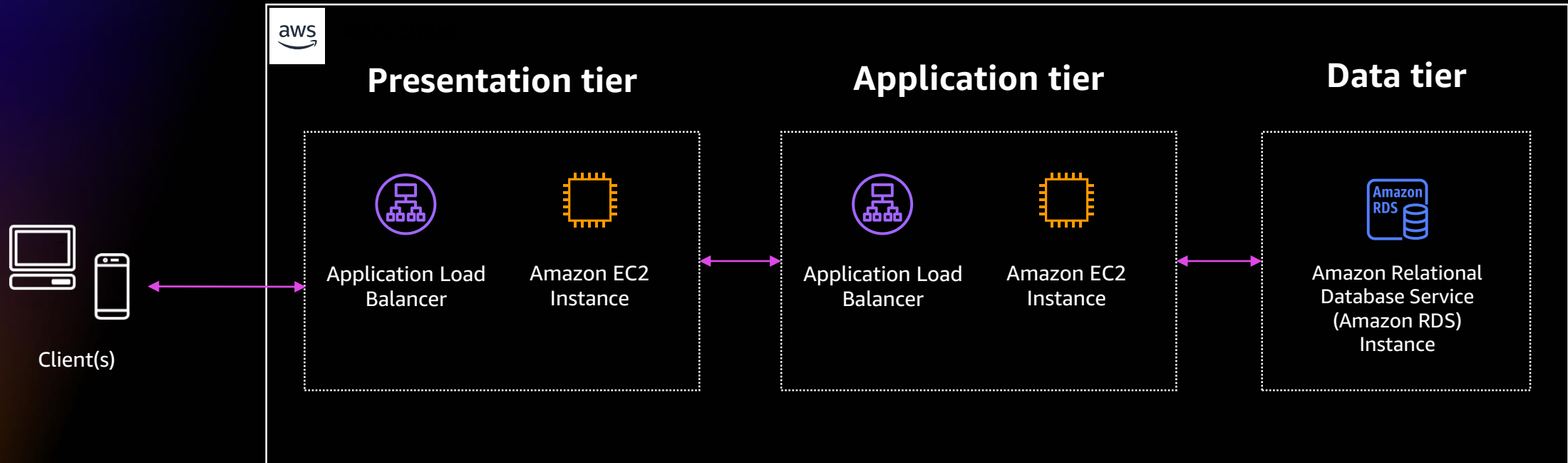
Business logic



Database servers

Data layer

# Three-tier architecture



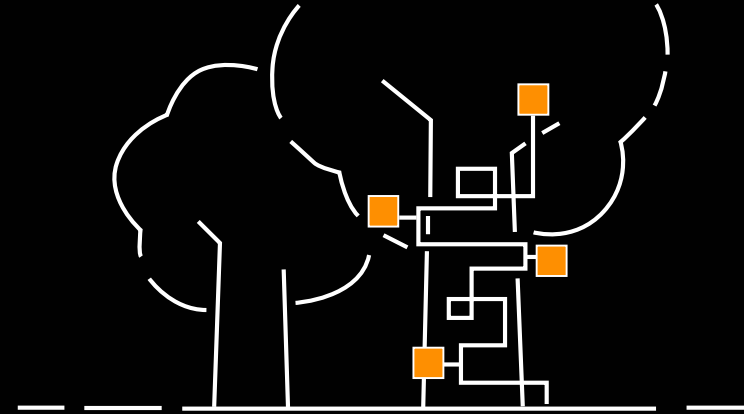
# Architectural patterns

# Strangler pattern



# Typically starts with breaking down the monolith

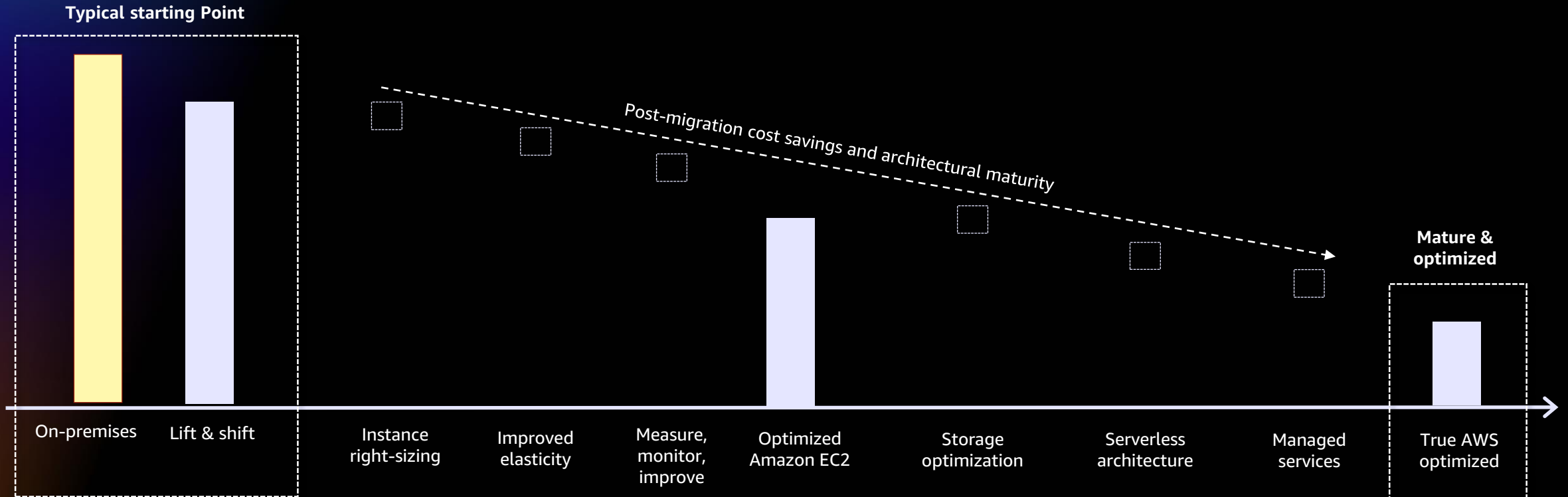
- Moving **monolithic** applications to **microservices** by **gradually** creating events and **APIs** for various components on of the legacy application



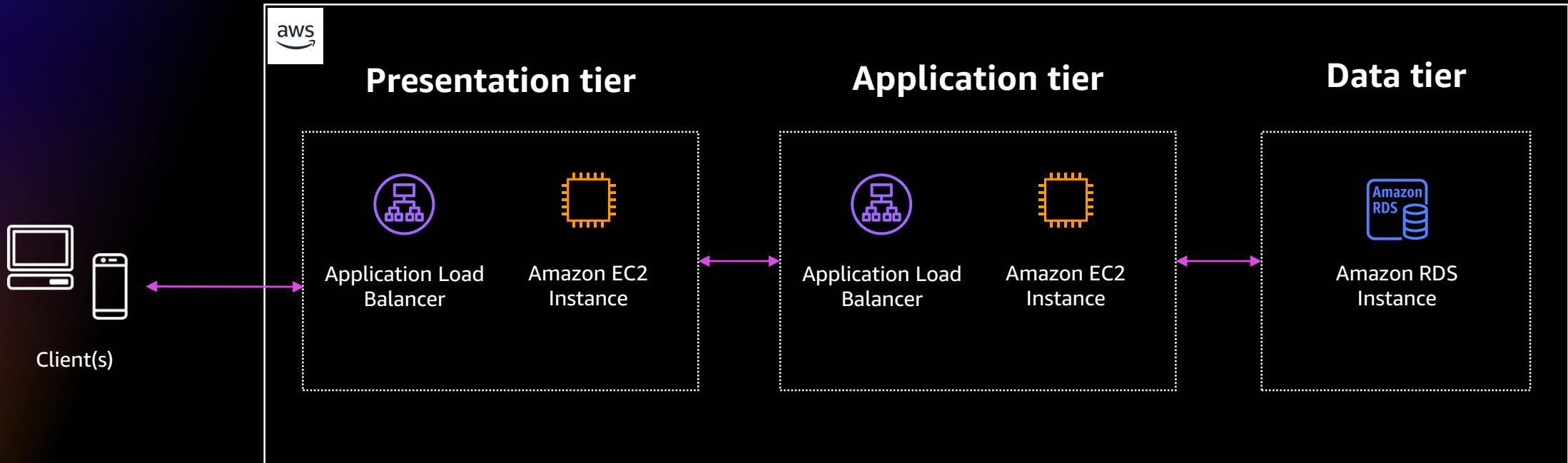
THE STRANGLER PATTERN

<https://martinfowler.com/bliki/StranglerFigApplication.html>

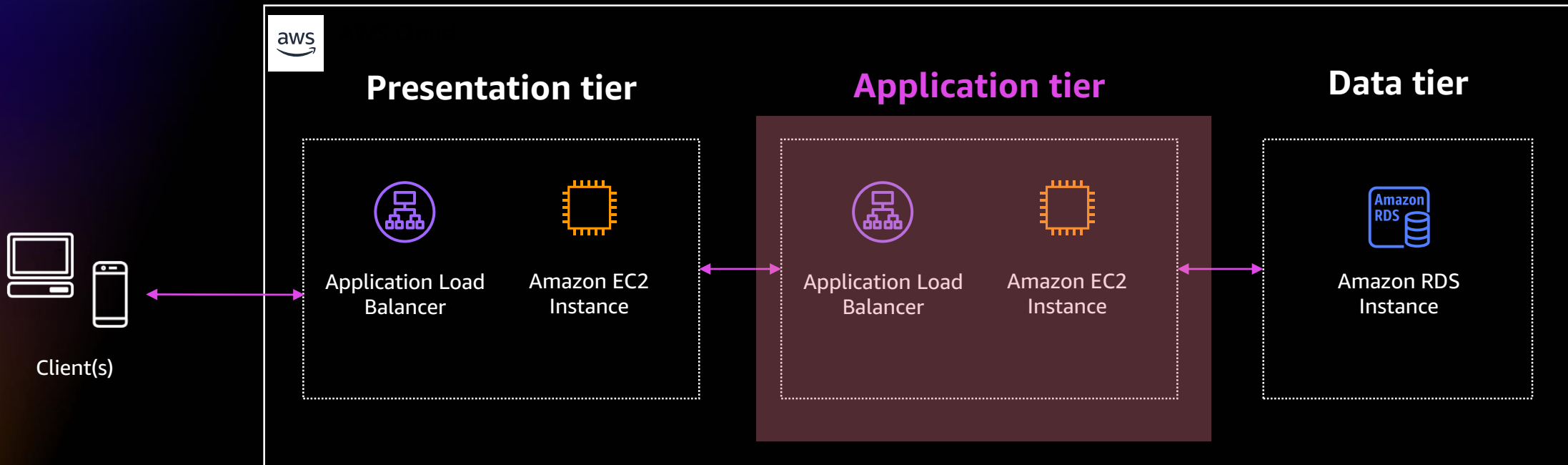
# Making improvements along the way



# Three-tier architecture



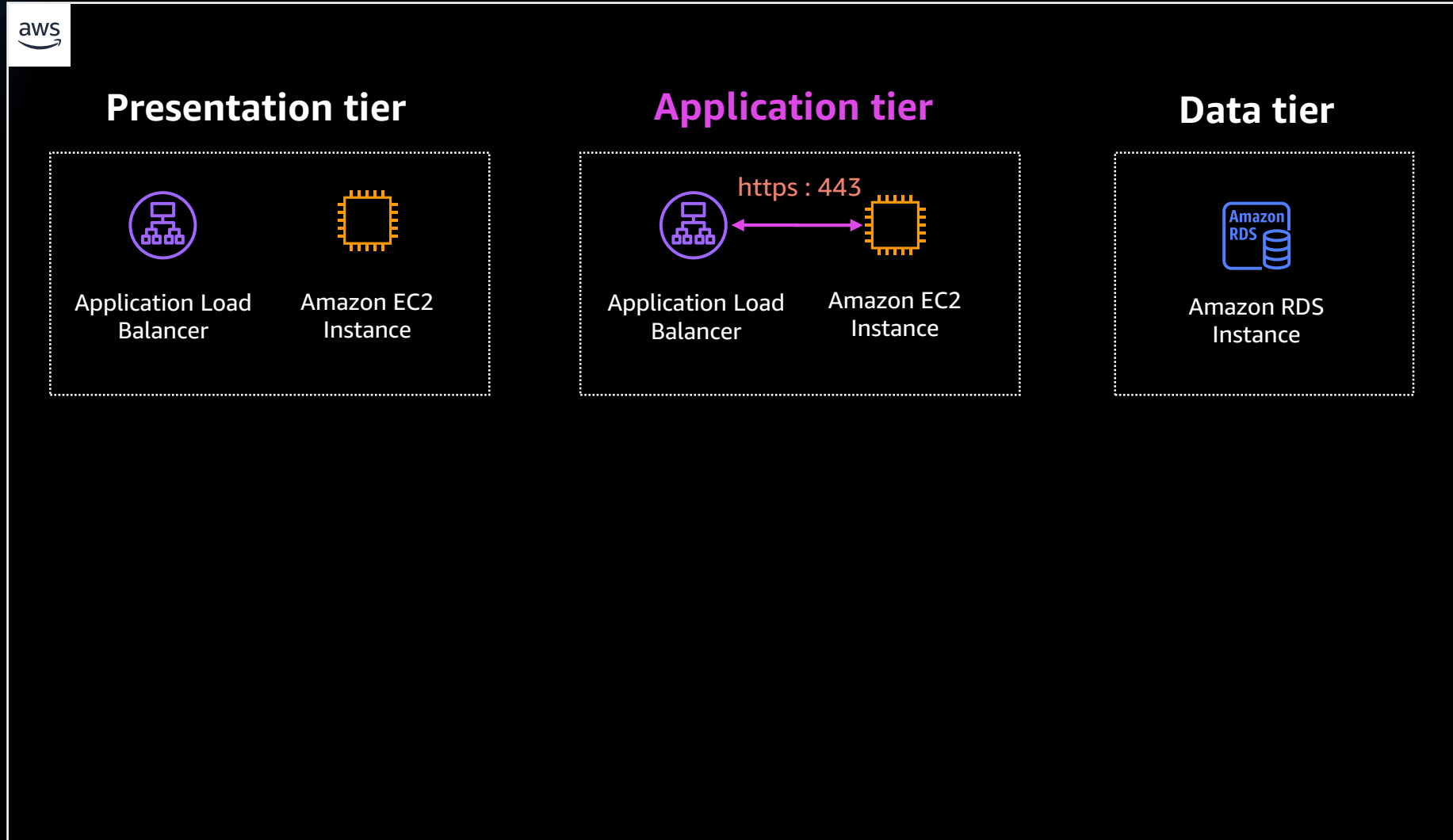
# Three-tier architecture : Strangler pattern



# Strangler pattern : Application tier



Client(s)



# AWS Lambda functions as targets – Application Load Balancers

EC2 > **Target groups** > Create target group

Step 1  
**Specify group details**

Step 2  
Register targets

## Specify group details

Your load balancer routes requests to the targets in a target group and performs health checks on the targets.

### Basic configuration

Settings in this section cannot be changed after the target group is created.

Choose a target type

☐ Instances

- Supports load balancing to instances within a specific VPC.

☐ IP addresses

- Supports load balancing to VPC and on-premises resources.
- Facilitates routing to multiple IP addresses and network interfaces on the same instance.
- Offers flexibility with microservice based architectures, simplifying inter-application communication.

☒ **Lambda function**

- Facilitates routing to a single Lambda function.
- Accessible to Application Load Balancers only.

Target group name

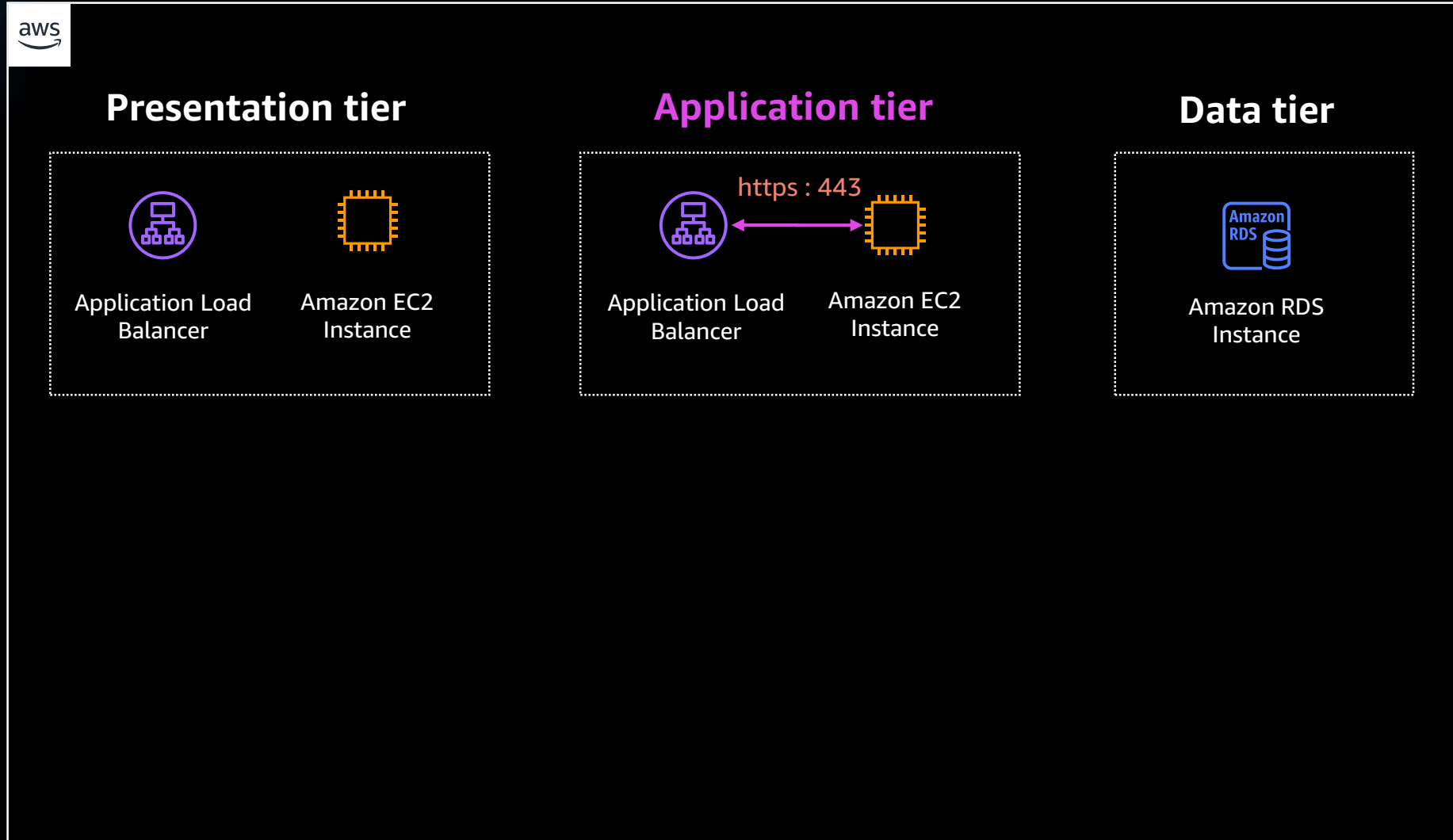
A maximum of 32 alphanumeric characters including hyphens are allowed, but the name must not begin or end with a hyphen.

<https://docs.aws.amazon.com/elasticloadbalancing/latest/application/lambda-functions.html>

# Strangler pattern : Application tier



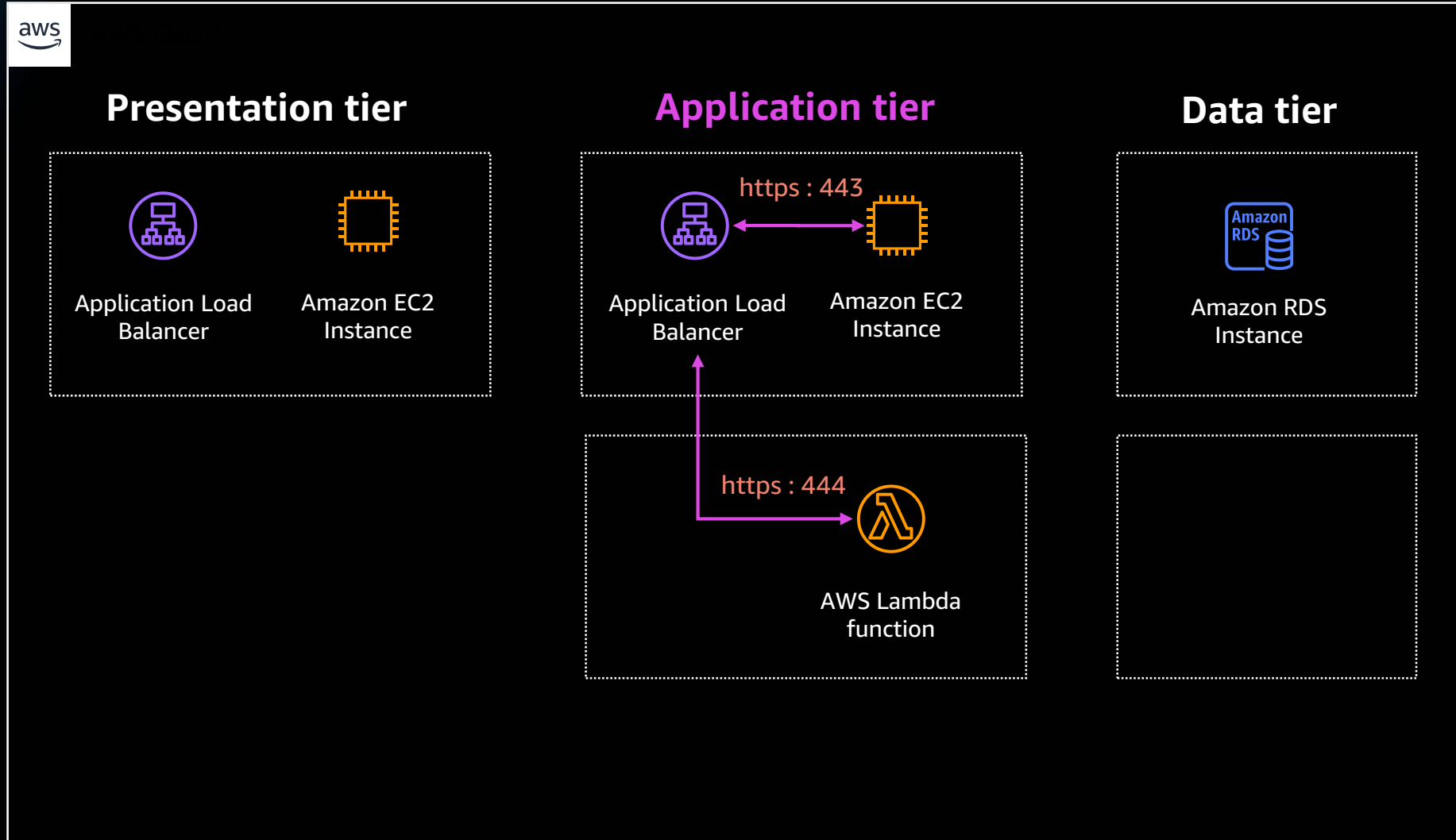
Client(s)



# Strangler pattern : Application tier



Client(s)

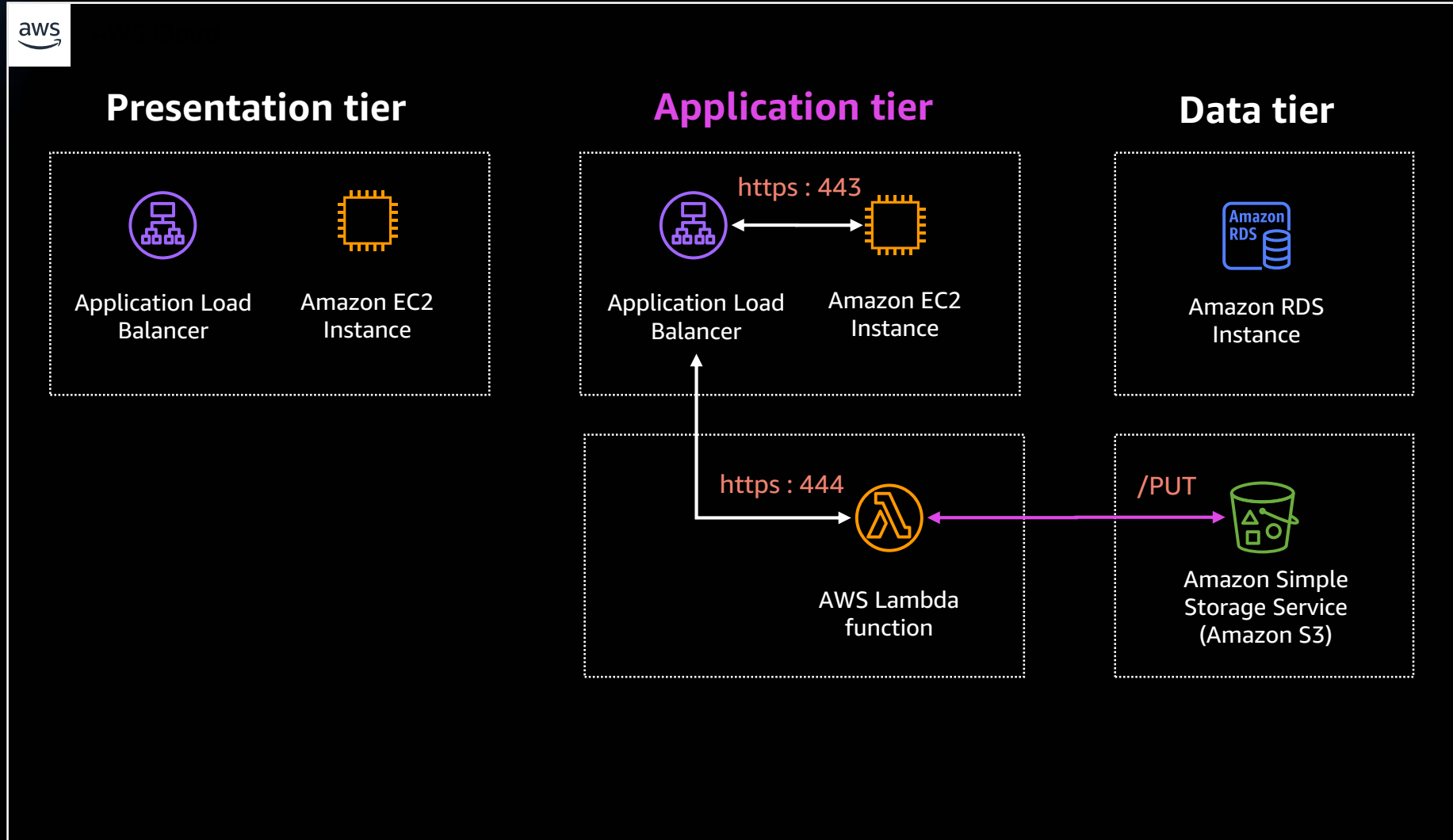




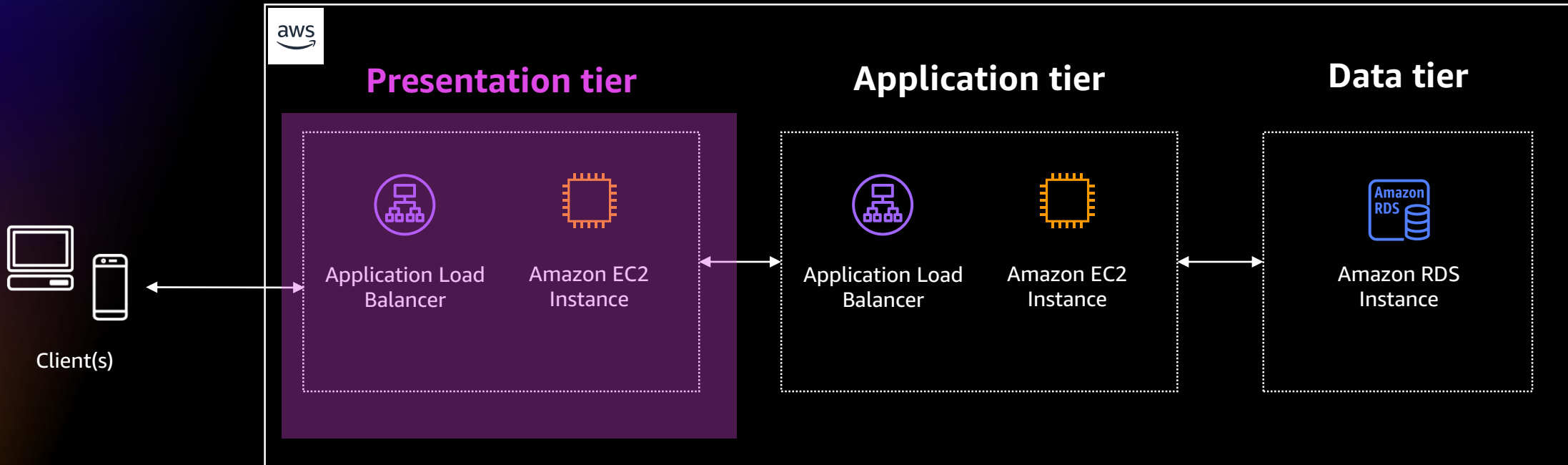
# Strangler pattern : Application tier



Client(s)



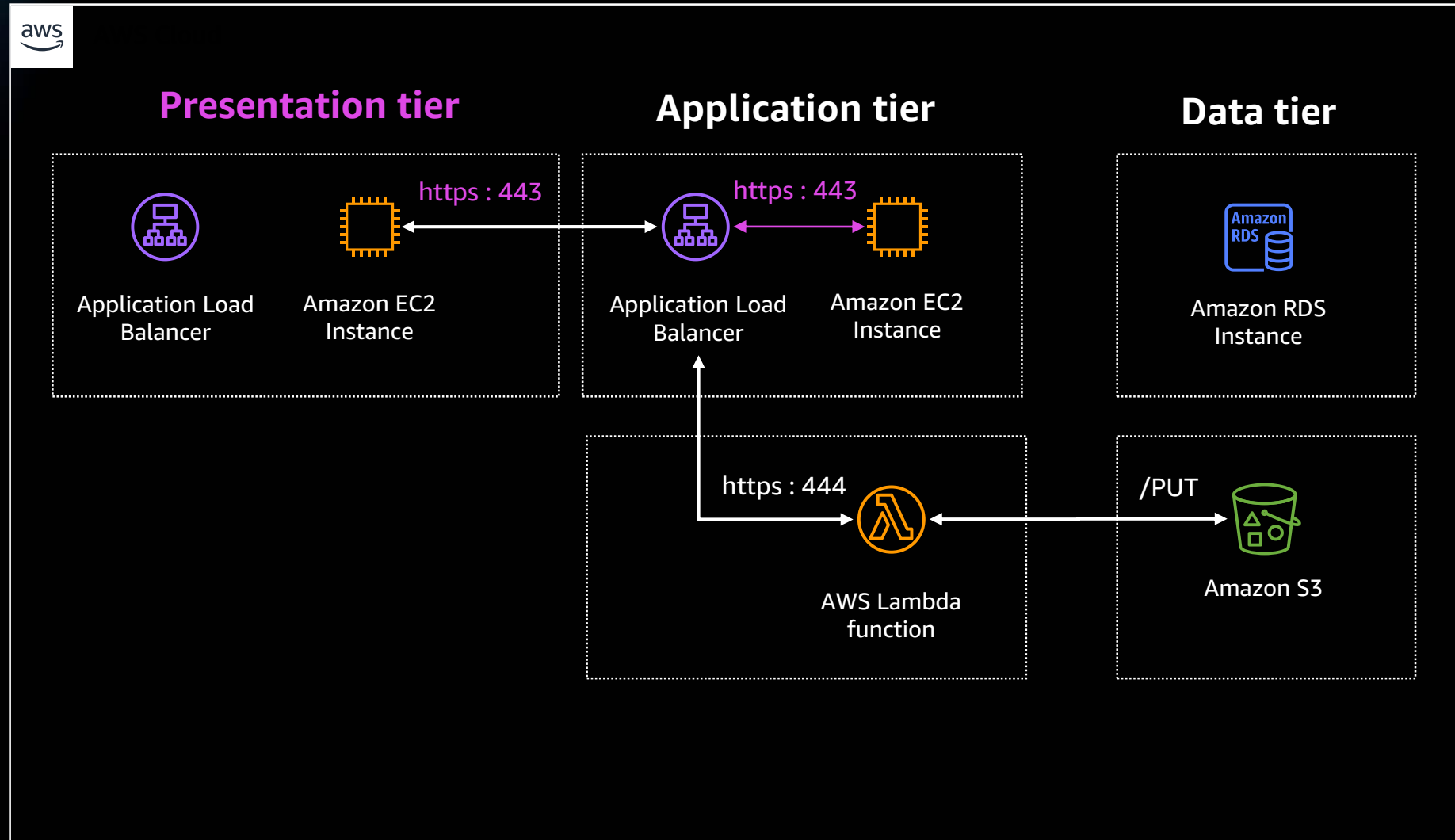
# Strangler pattern : Presentation tier



# Three-tier architecture



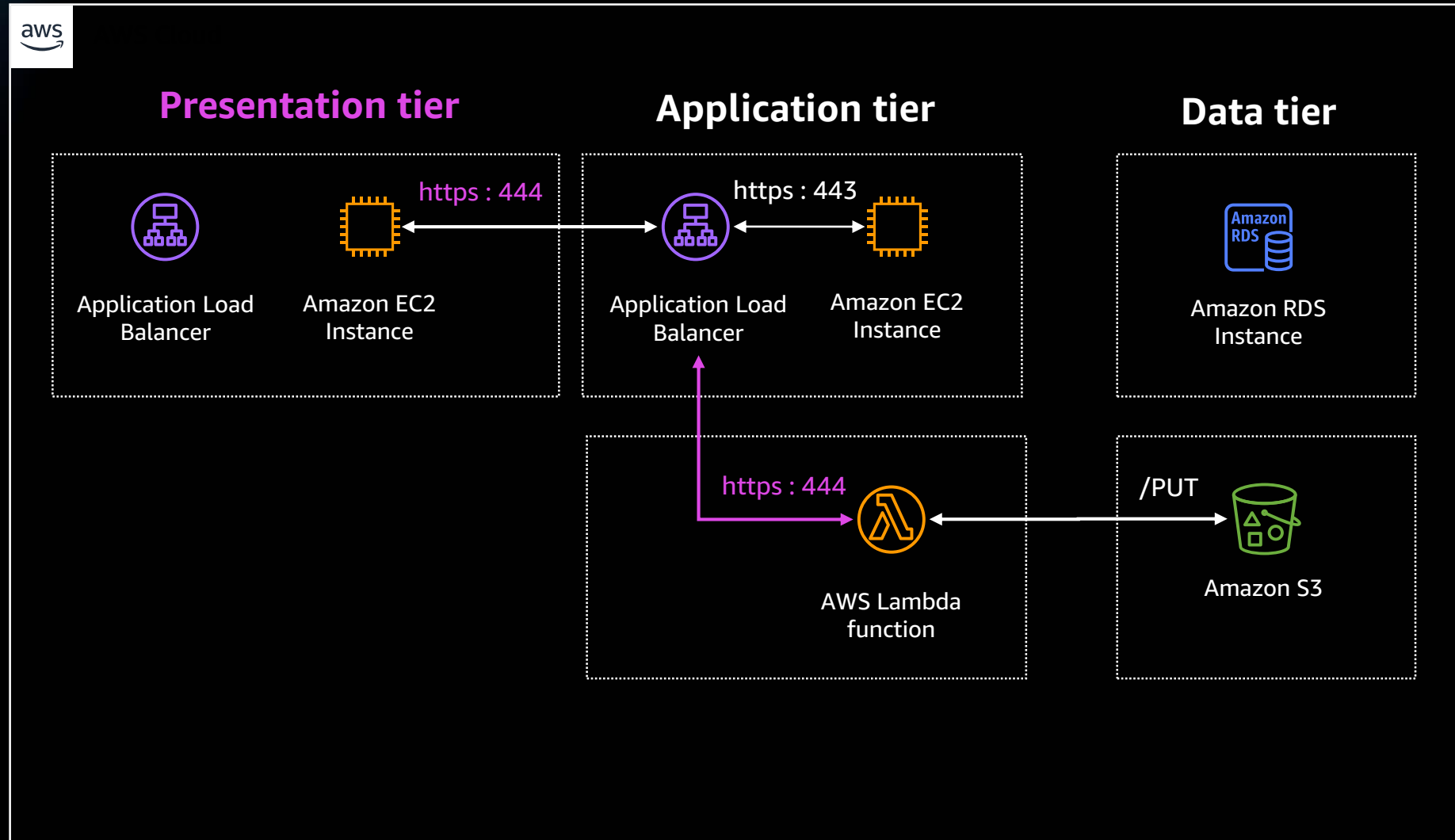
Client(s)



# Three-tier architecture

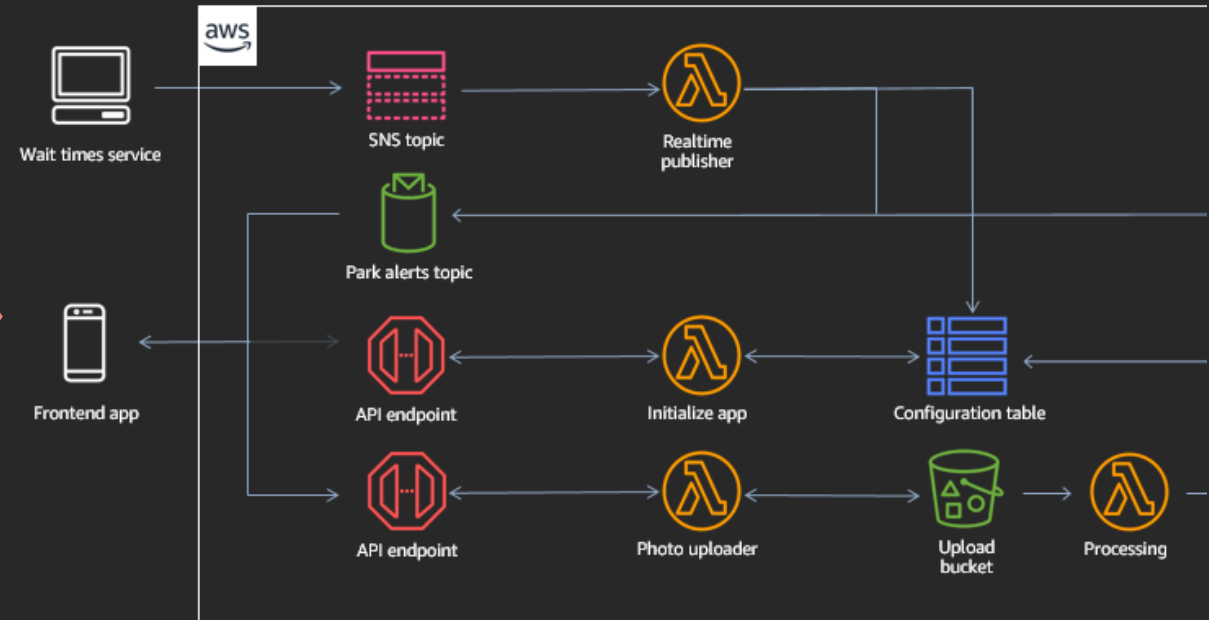


Client(s)



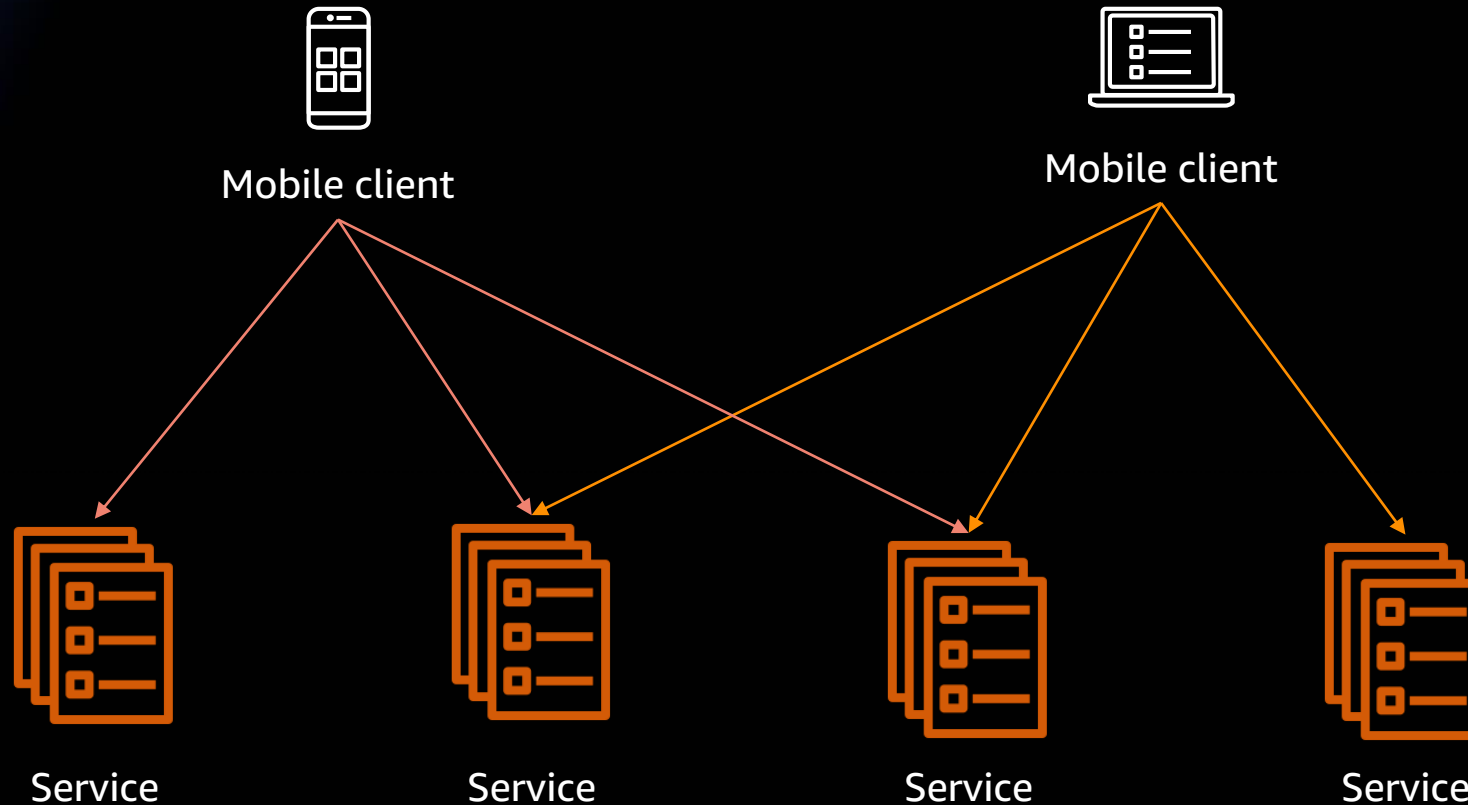
# Small pieces, loosely joined

## *Loose coupling*

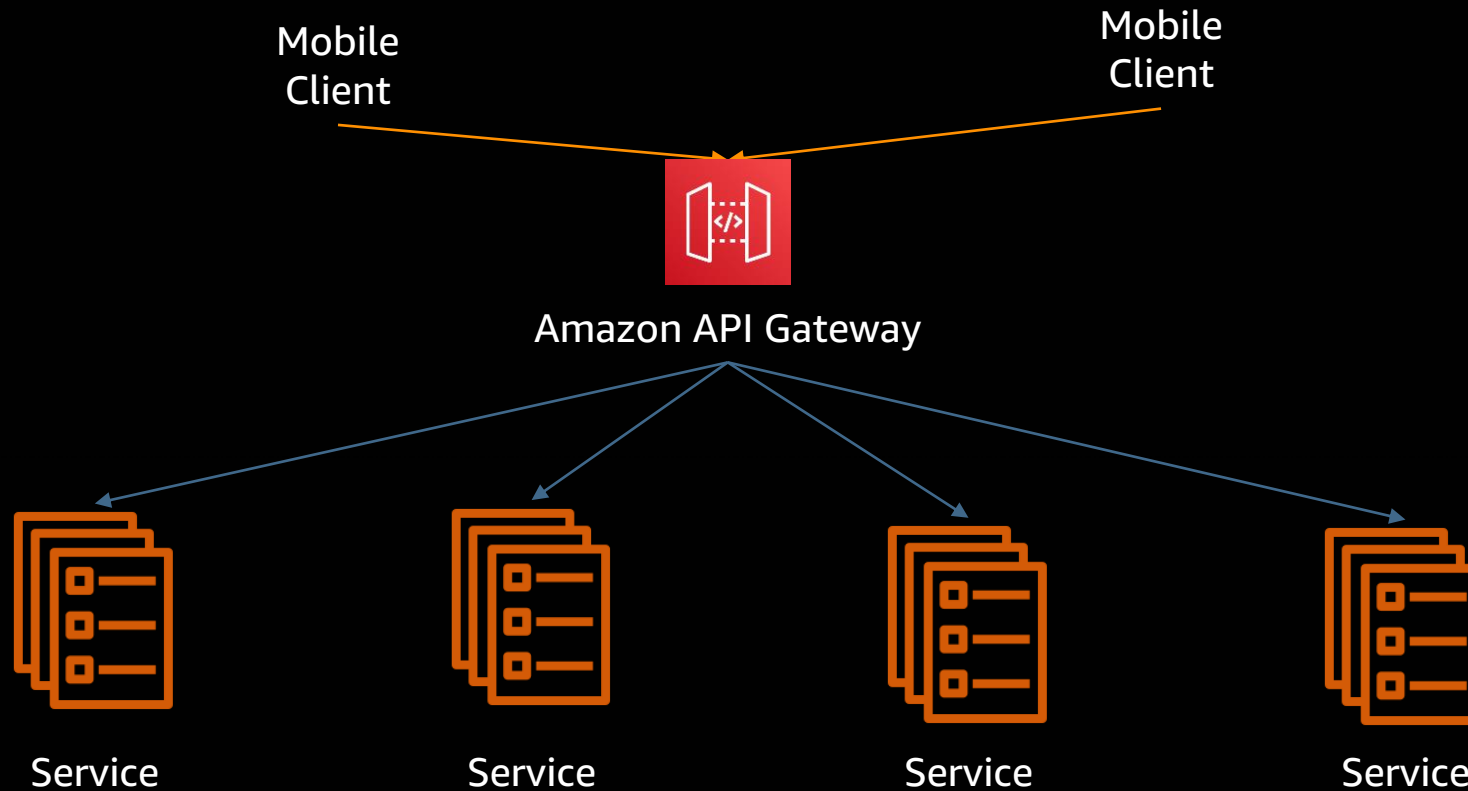


# API Gateway pattern

# Example: Communication between devices and services (without an API Gateway)



# Example: Communication between devices and services (with an API Gateway)

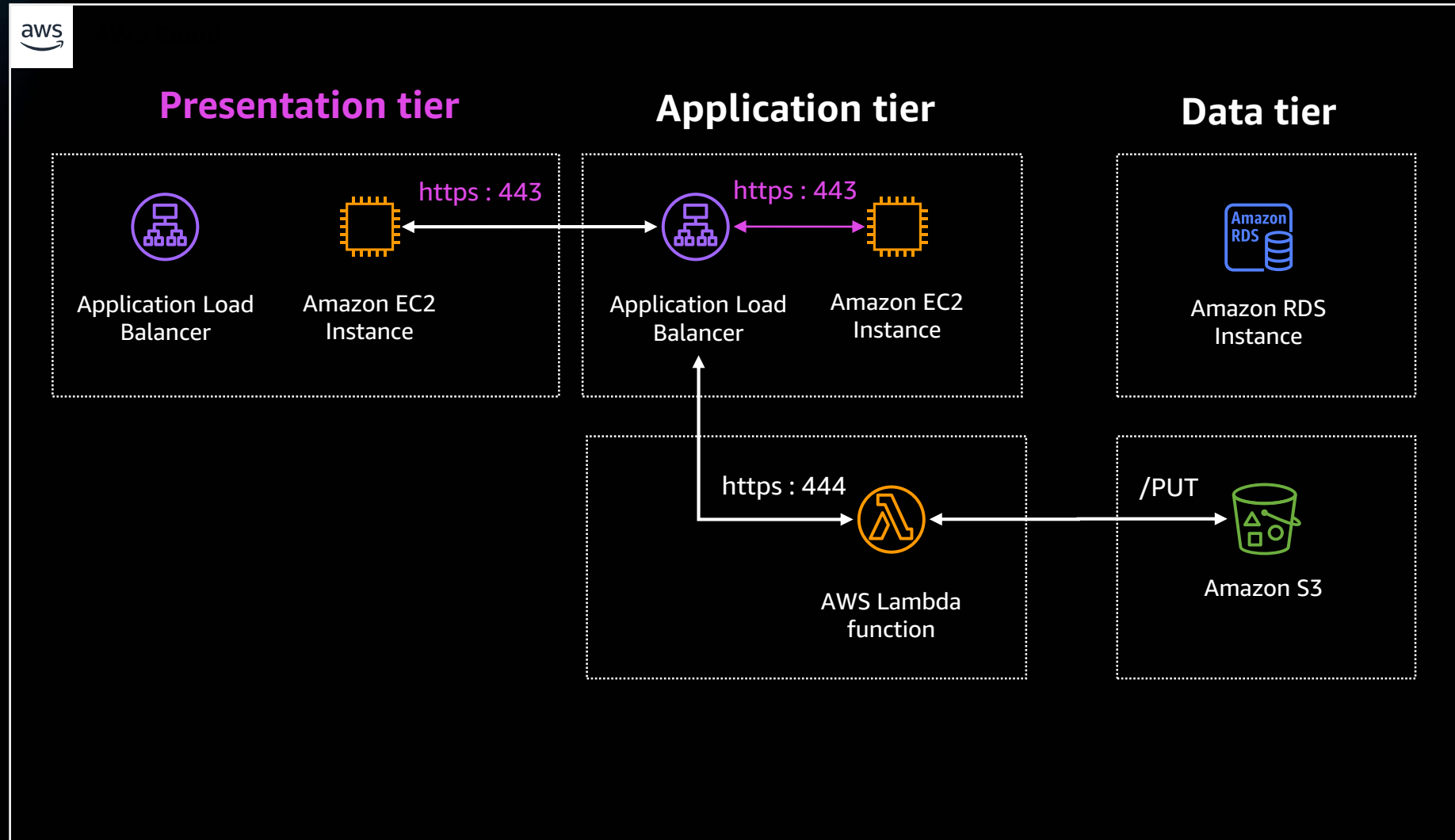




# Three-tier architecture



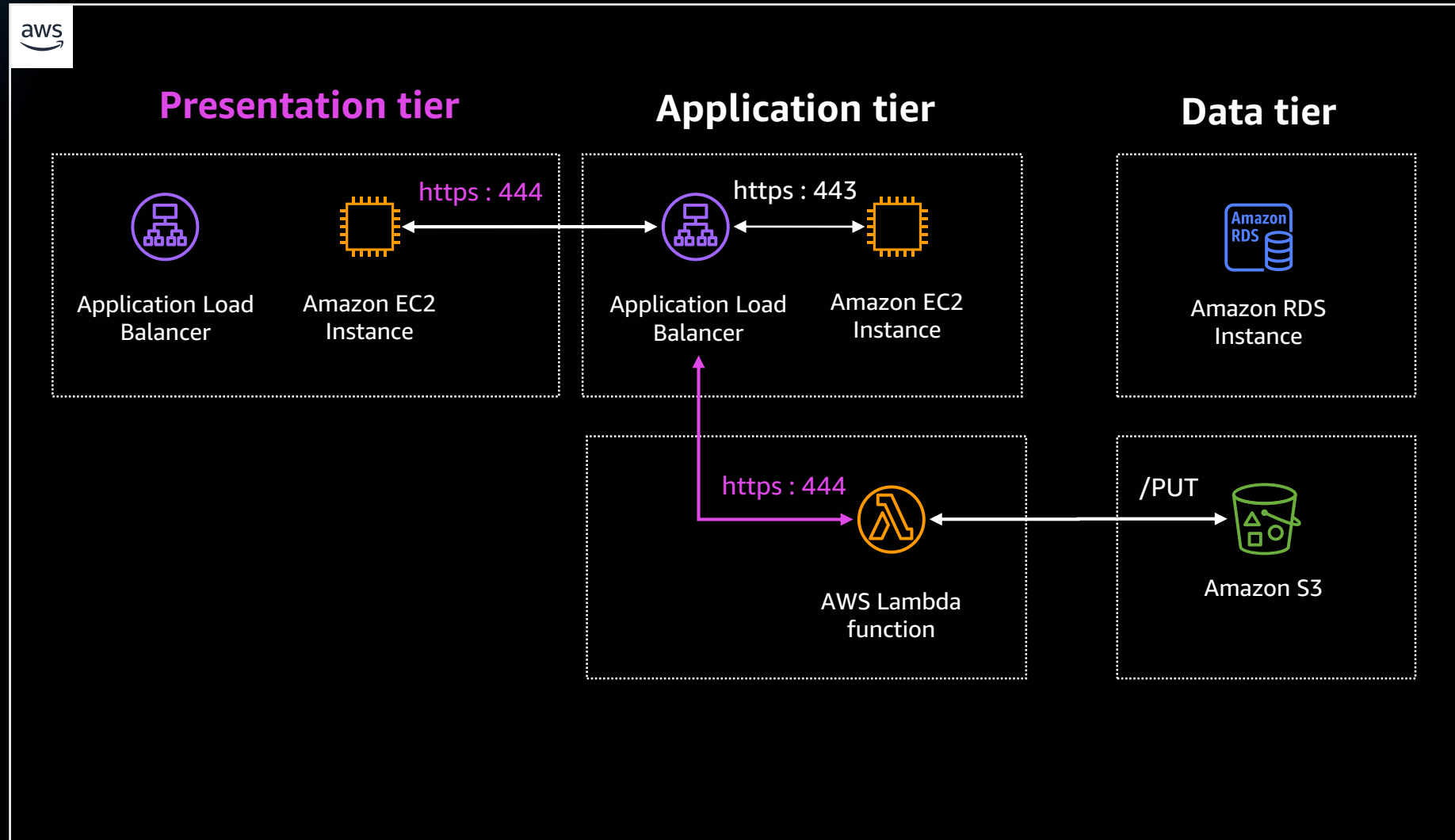
Client(s)



# Three-tier architecture



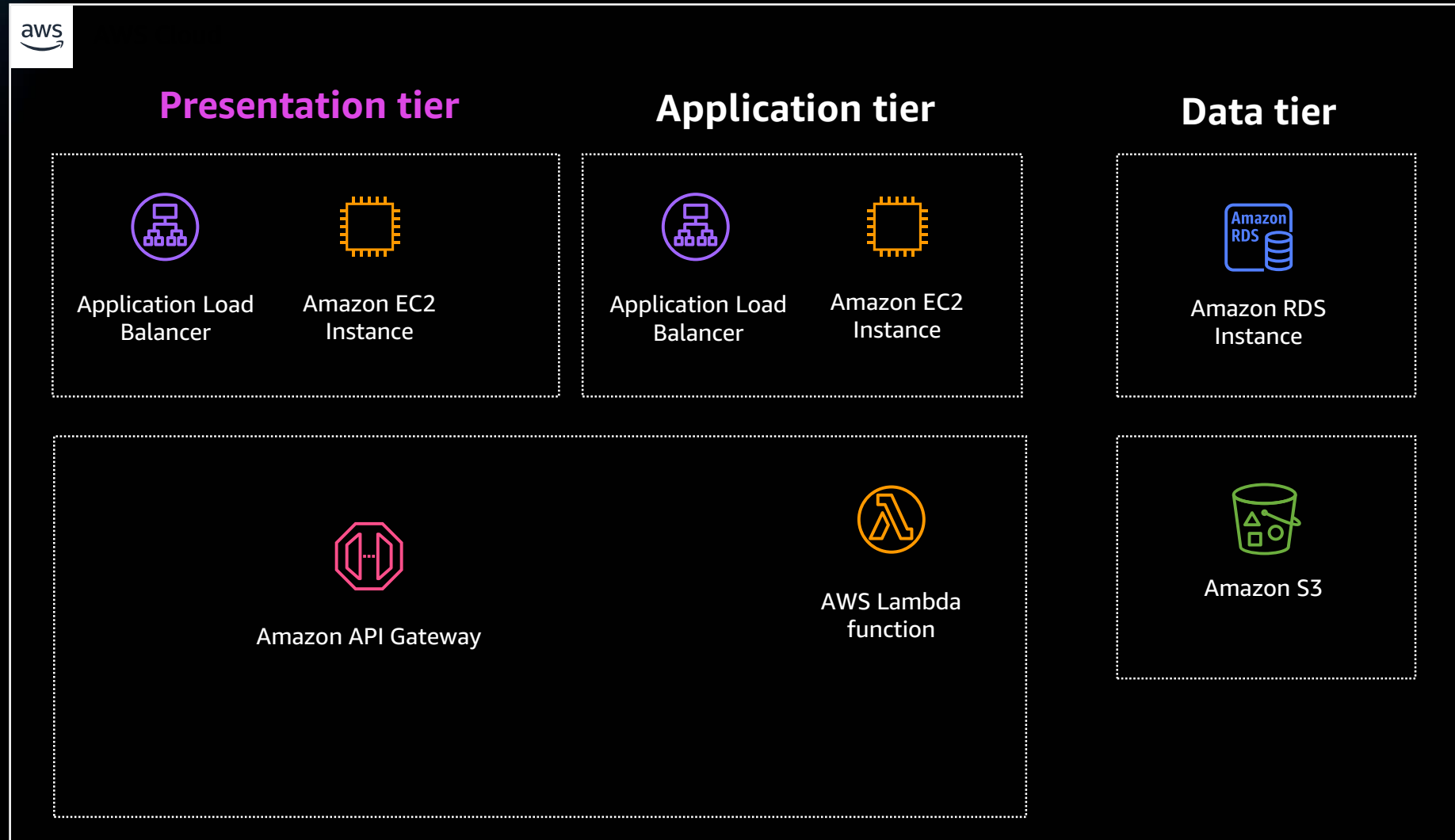
Client(s)



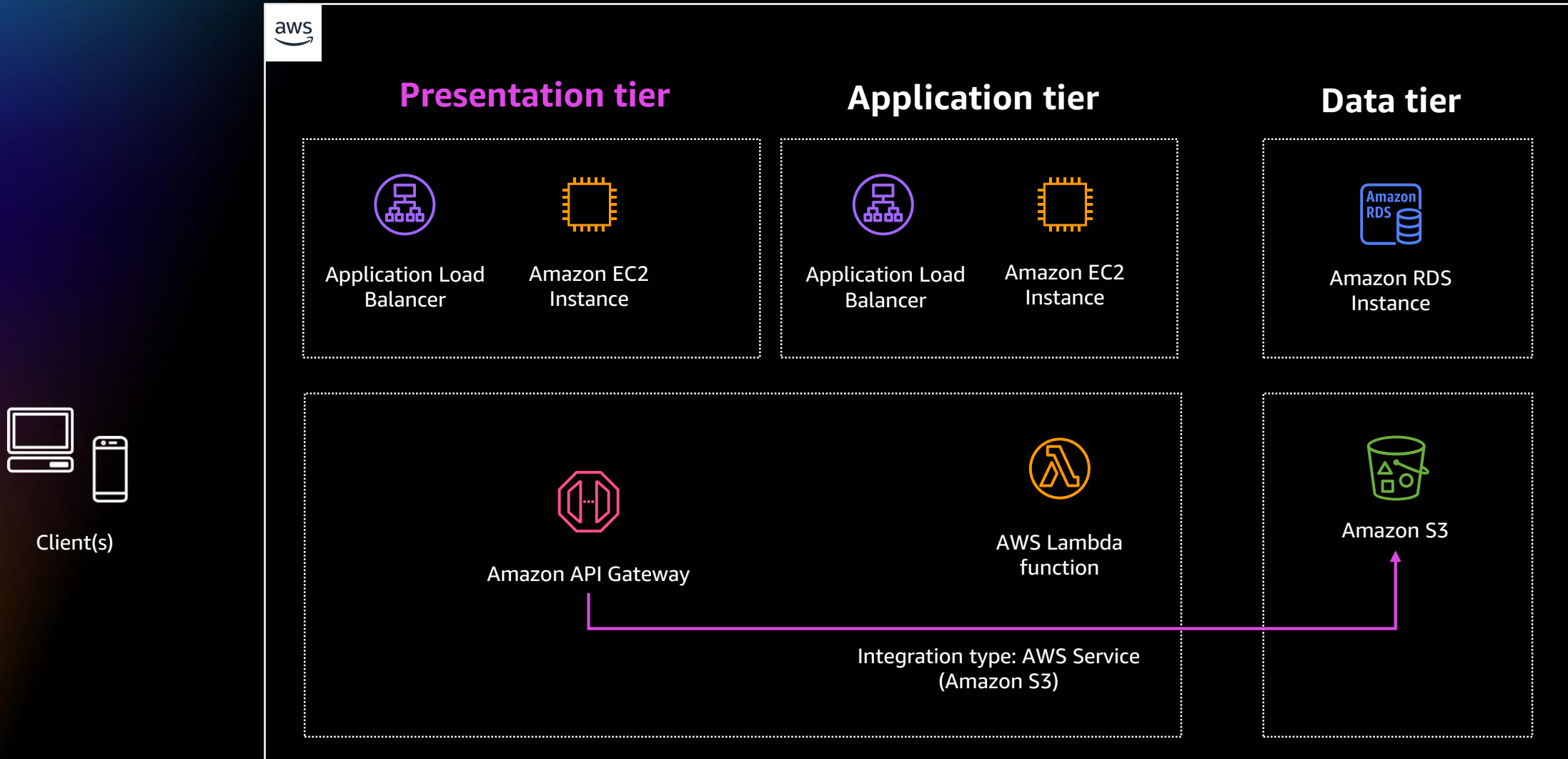
# API Gateway pattern



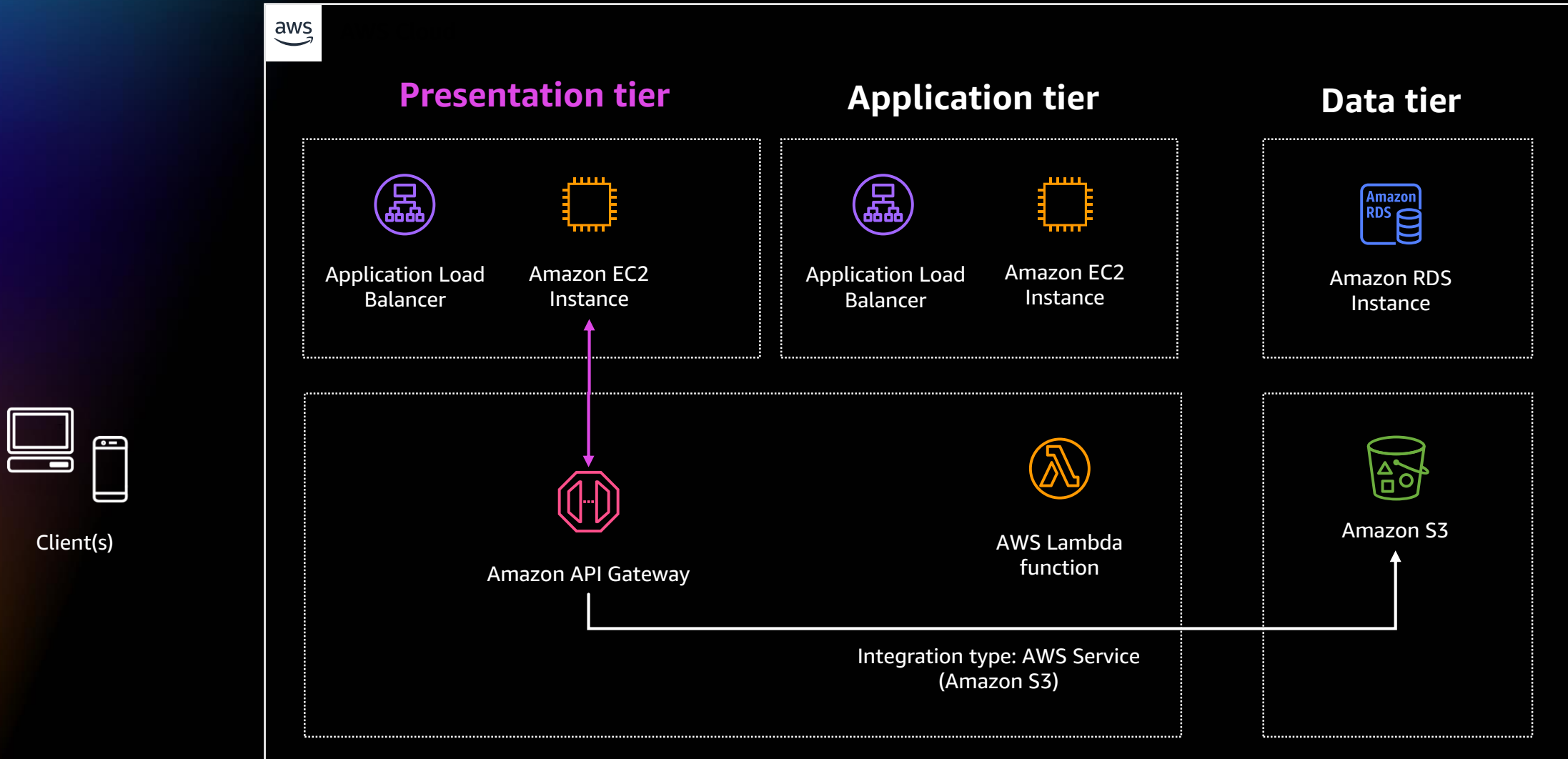
Client(s)



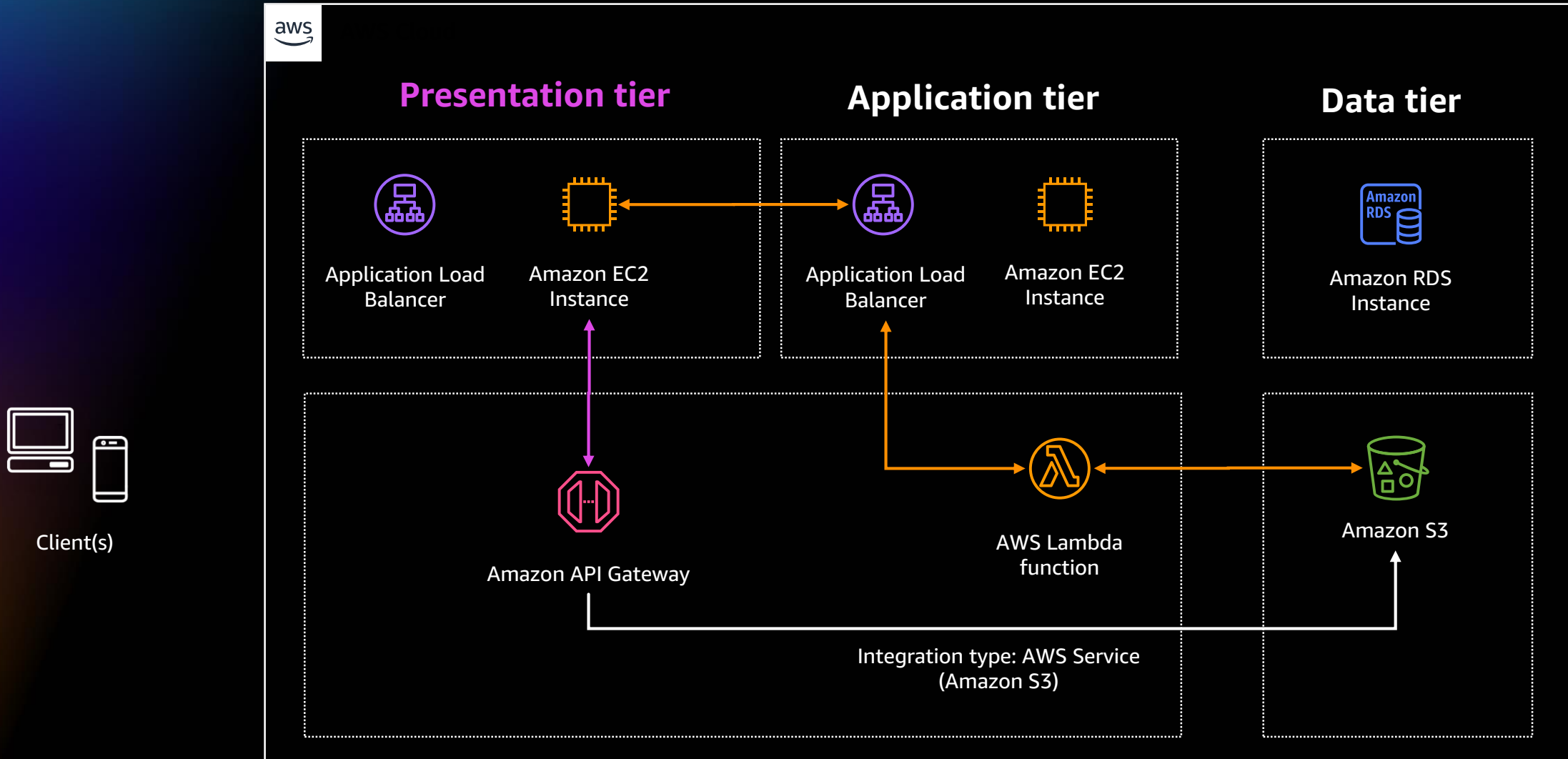
# Three-tier architecture



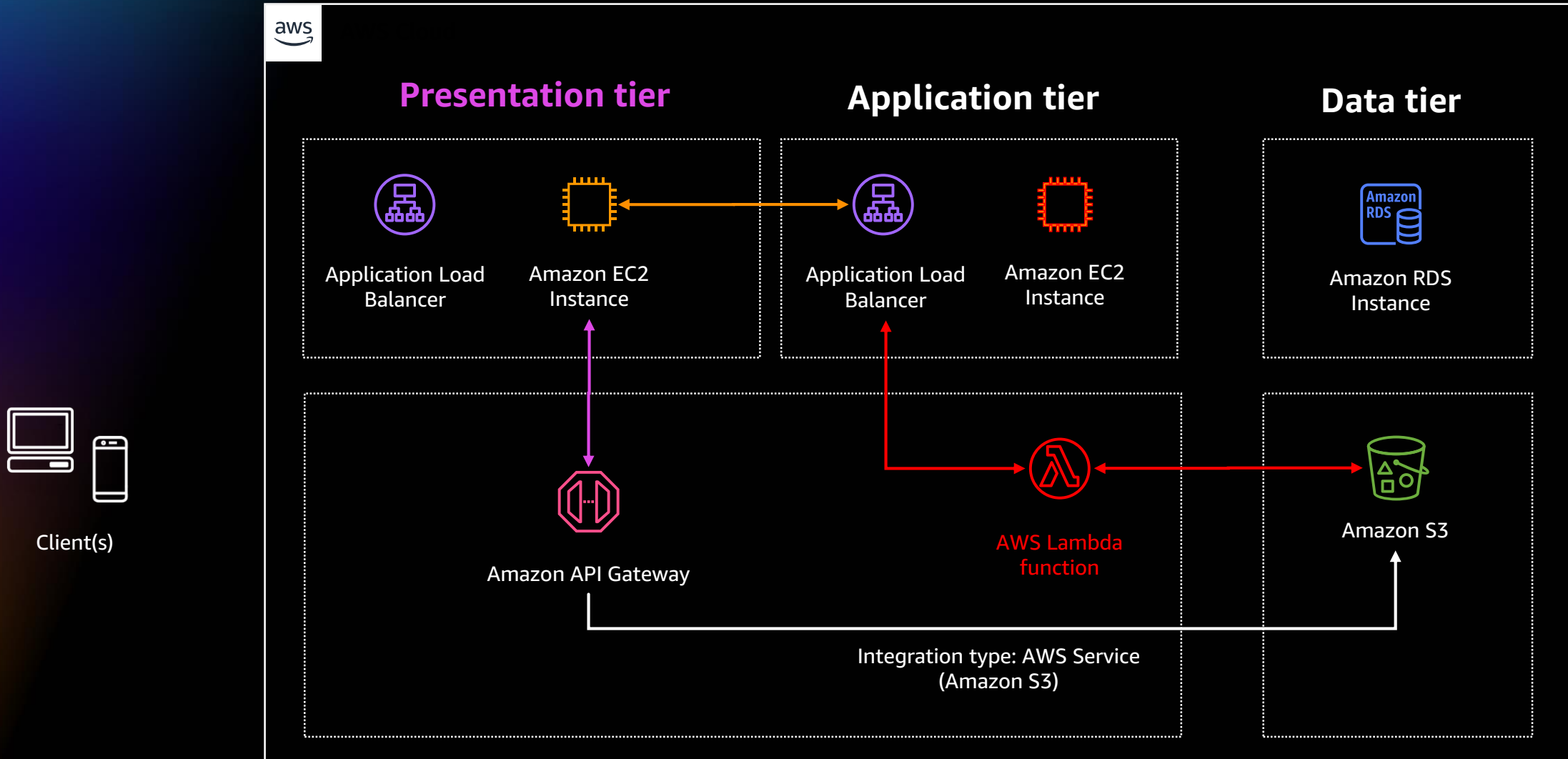
# API Gateway pattern



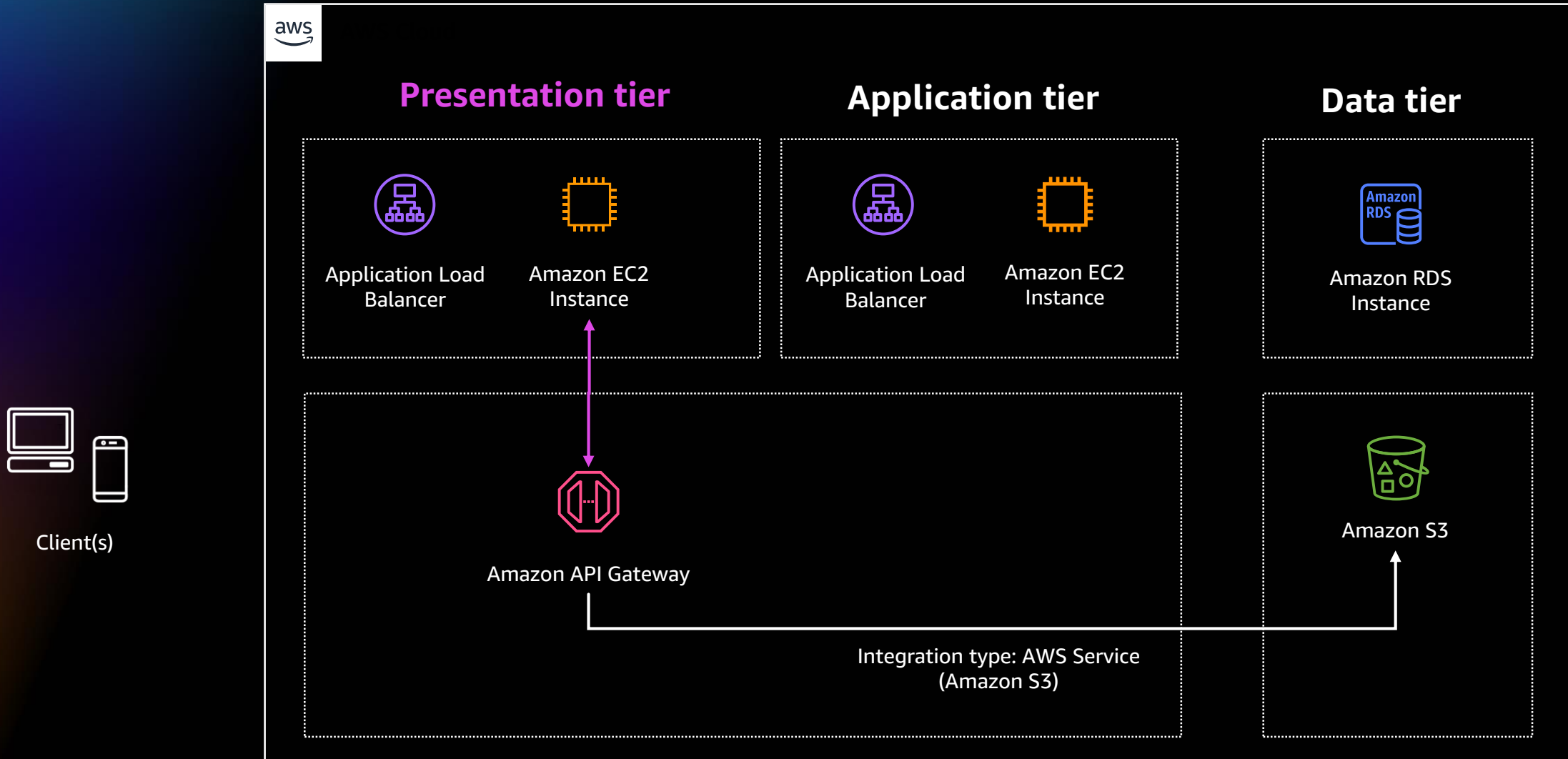
# API Gateway pattern



# API Gateway pattern

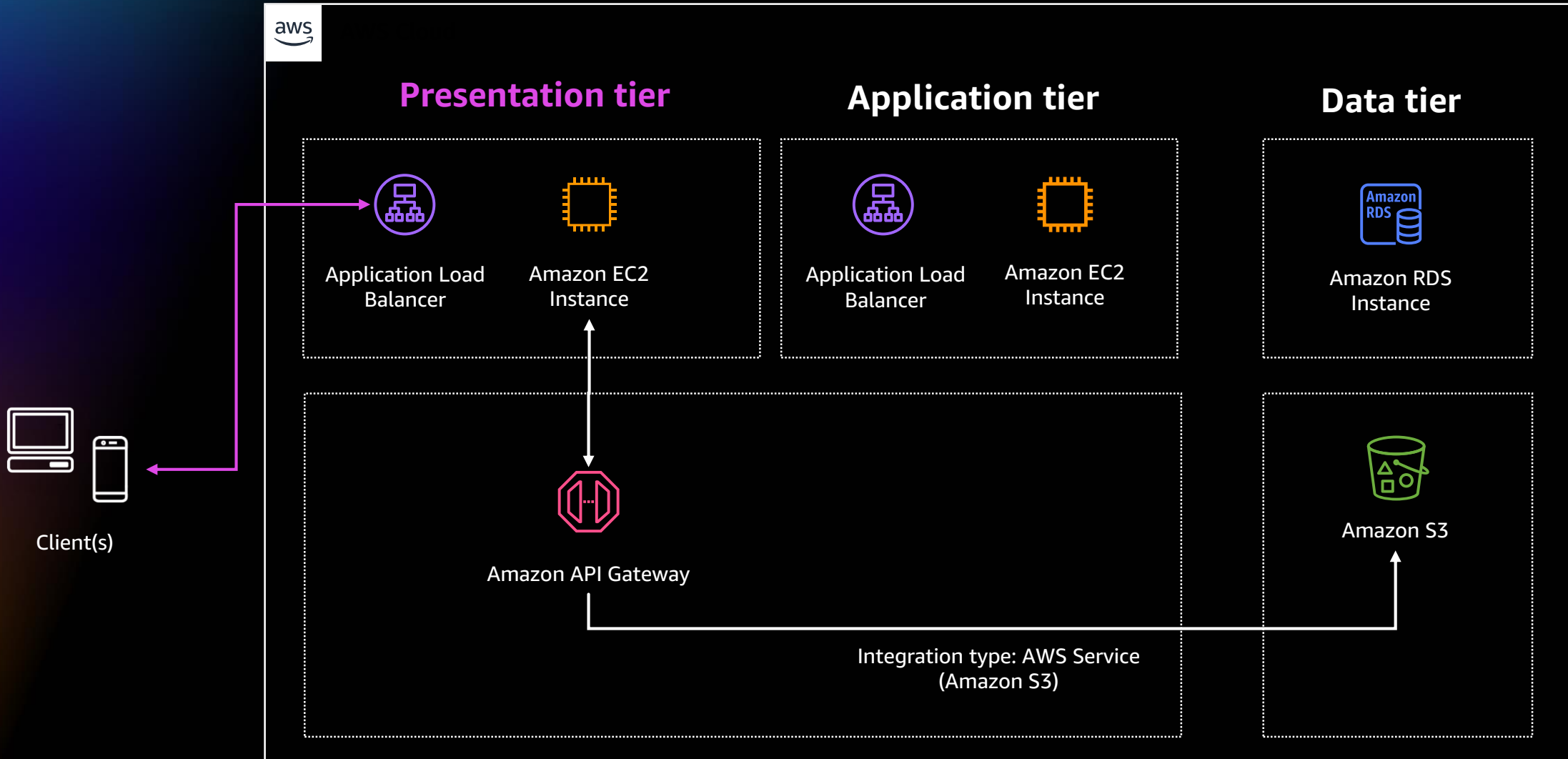


# API Gateway pattern

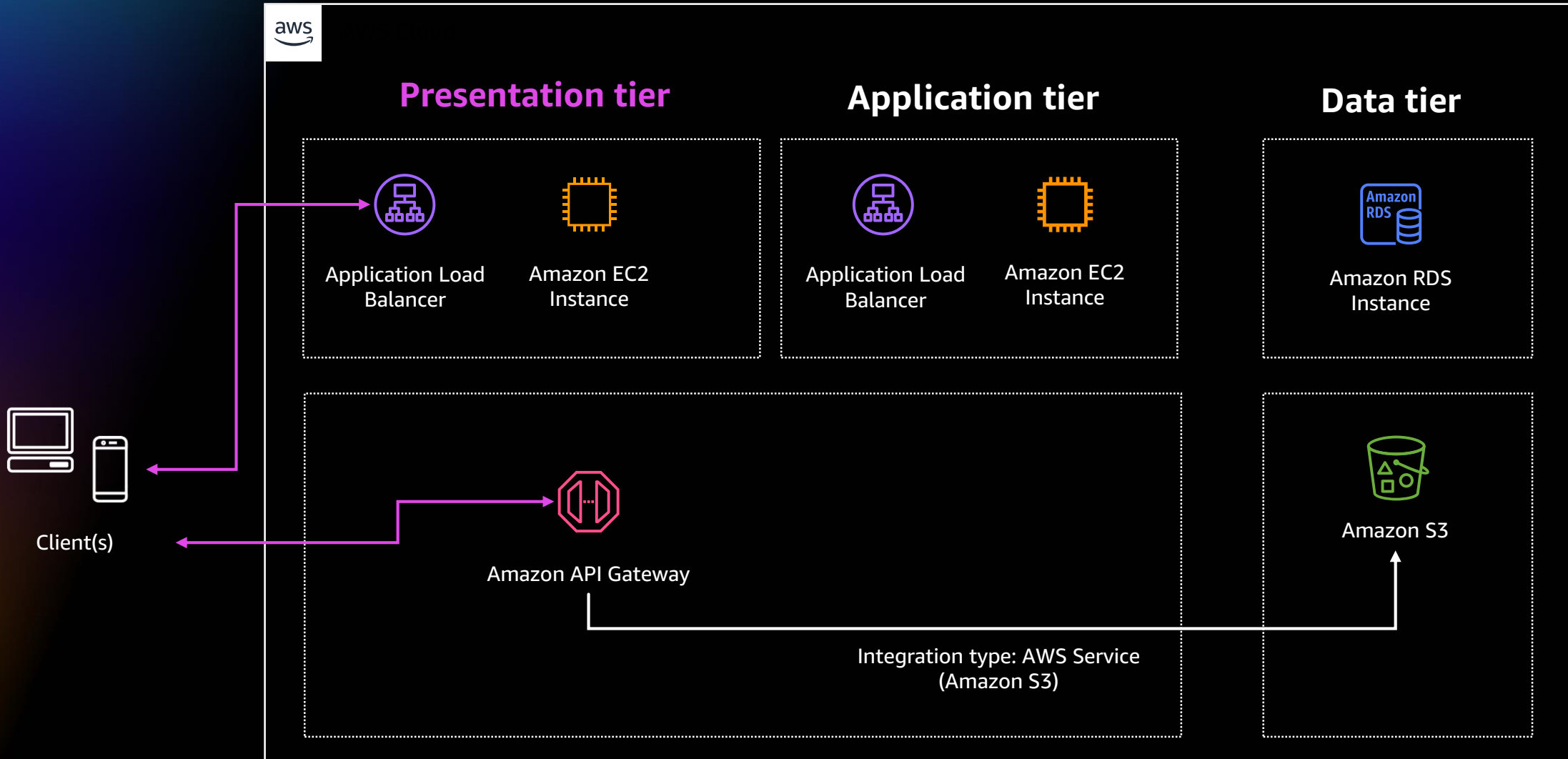




# API Gateway pattern



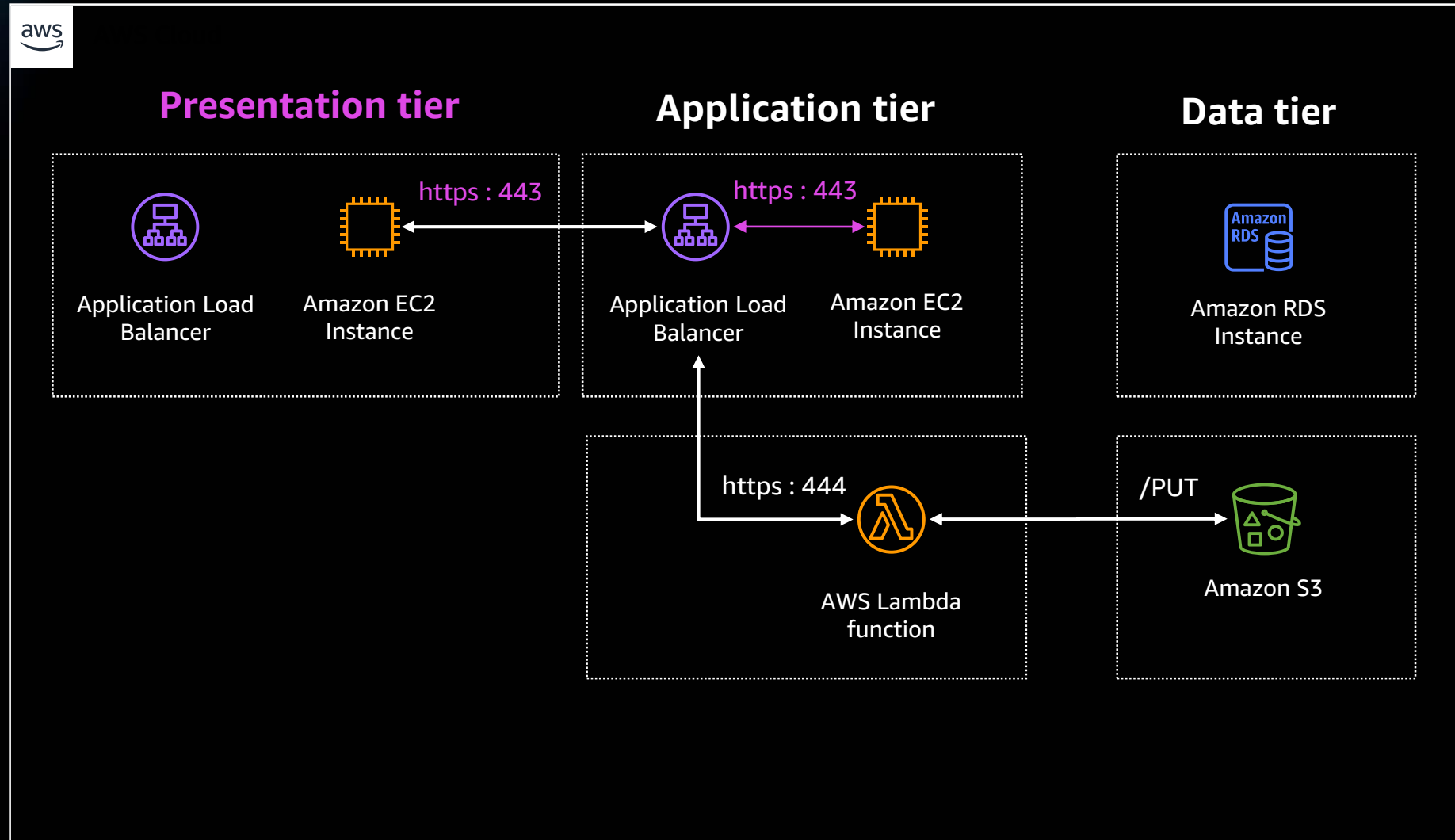
# API Gateway pattern



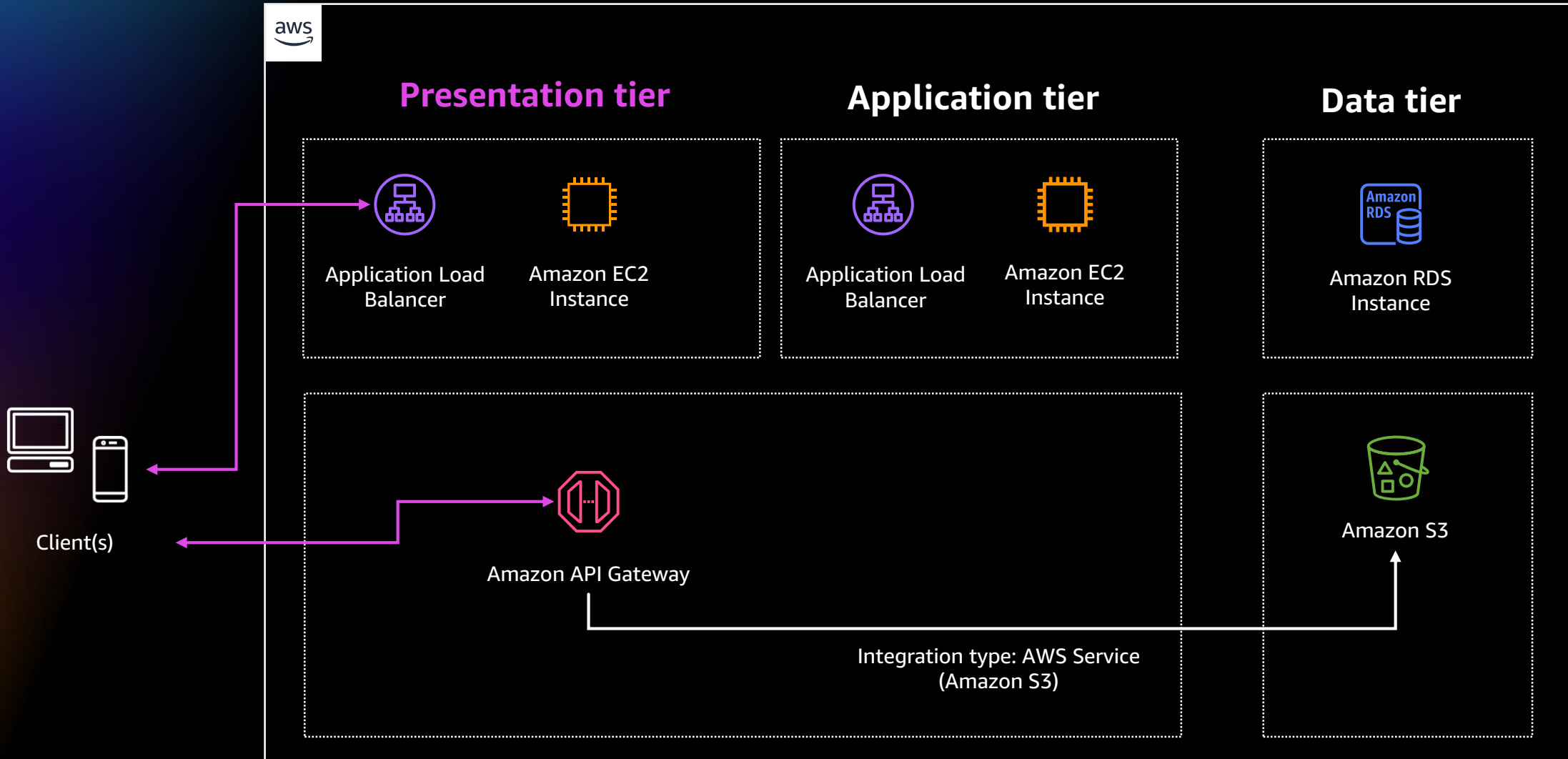
# Coupling concern



Client(s)

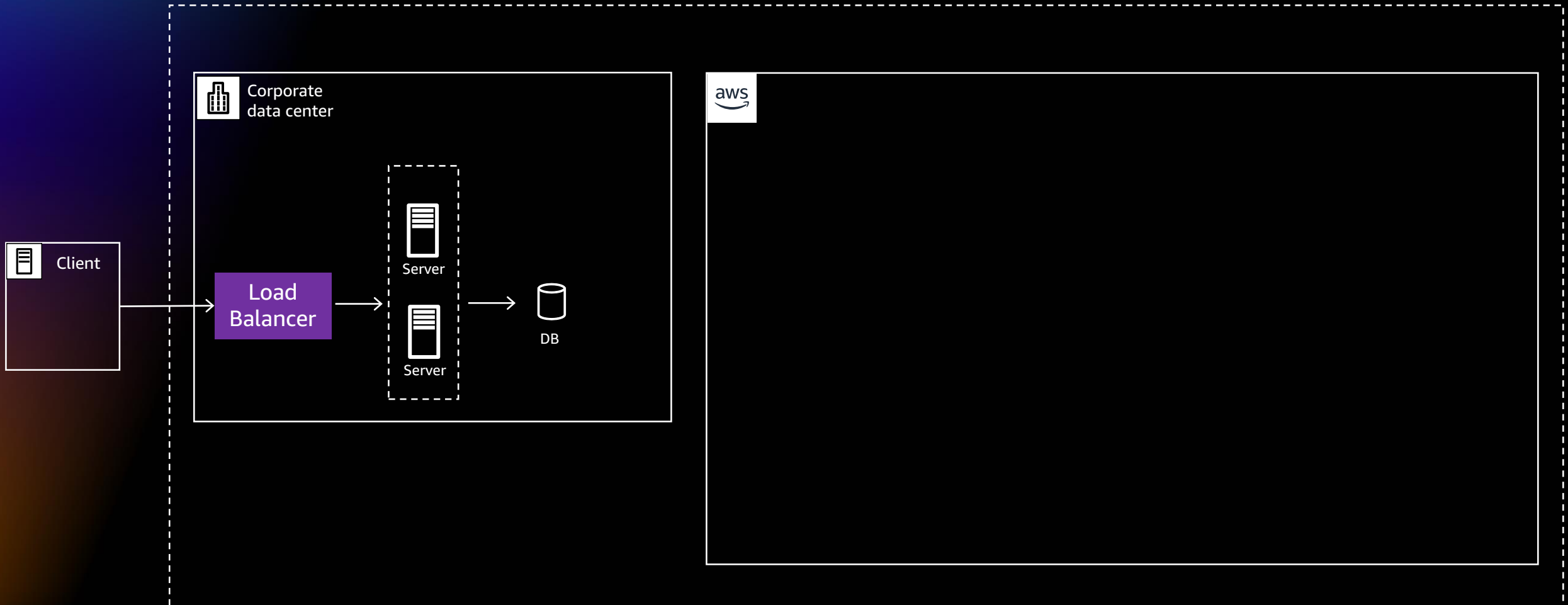


# API Gateway pattern

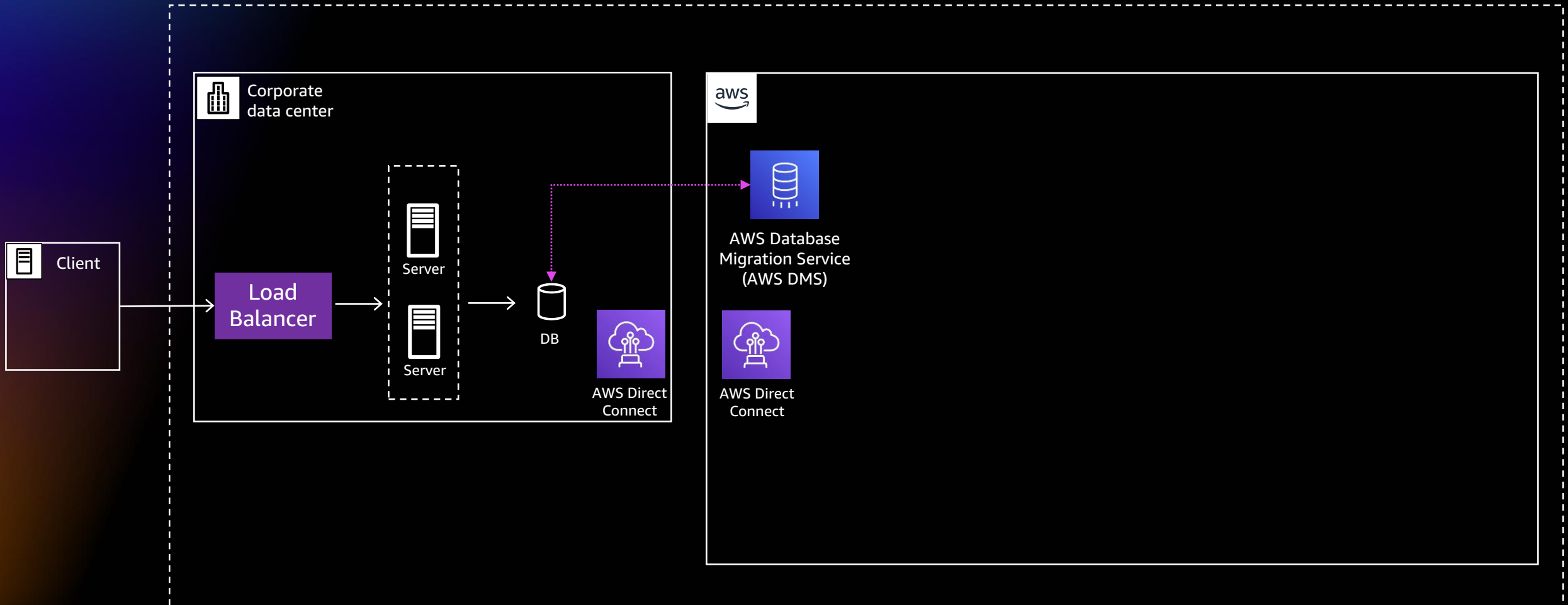


# Event Sourcing pattern

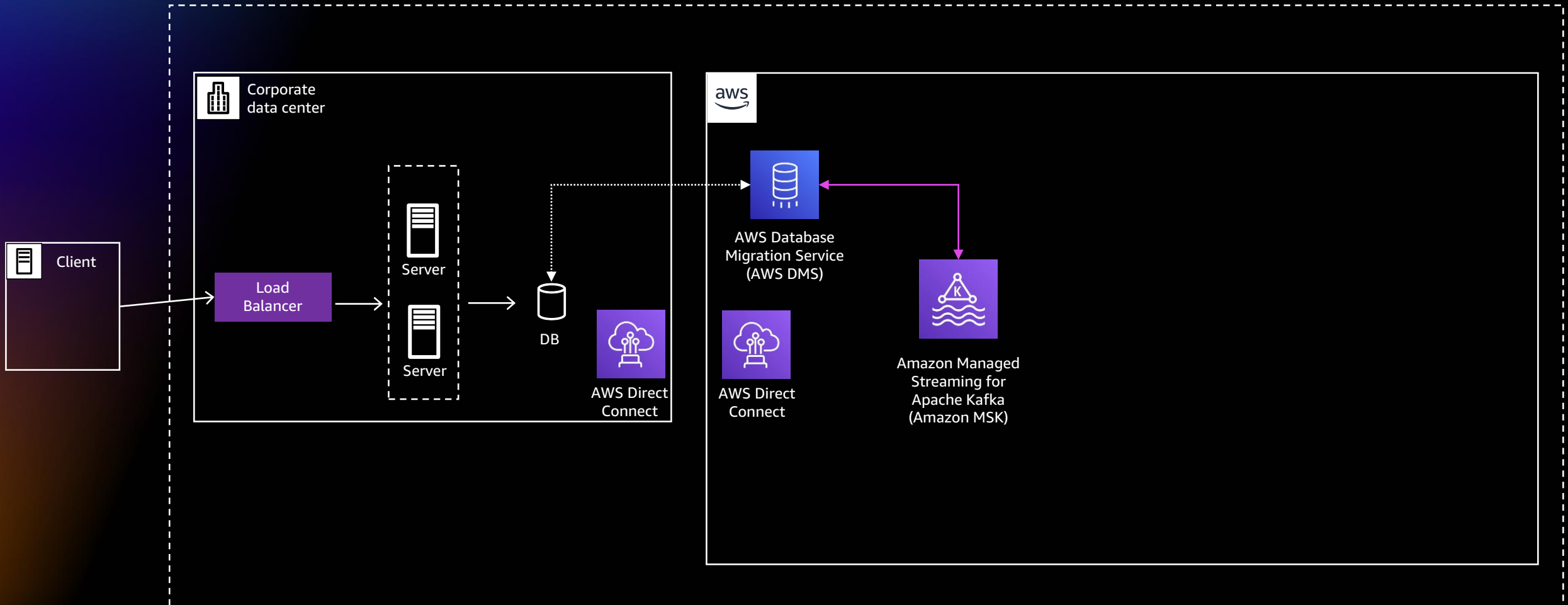
# Event Sourcing pattern



# Event Sourcing pattern

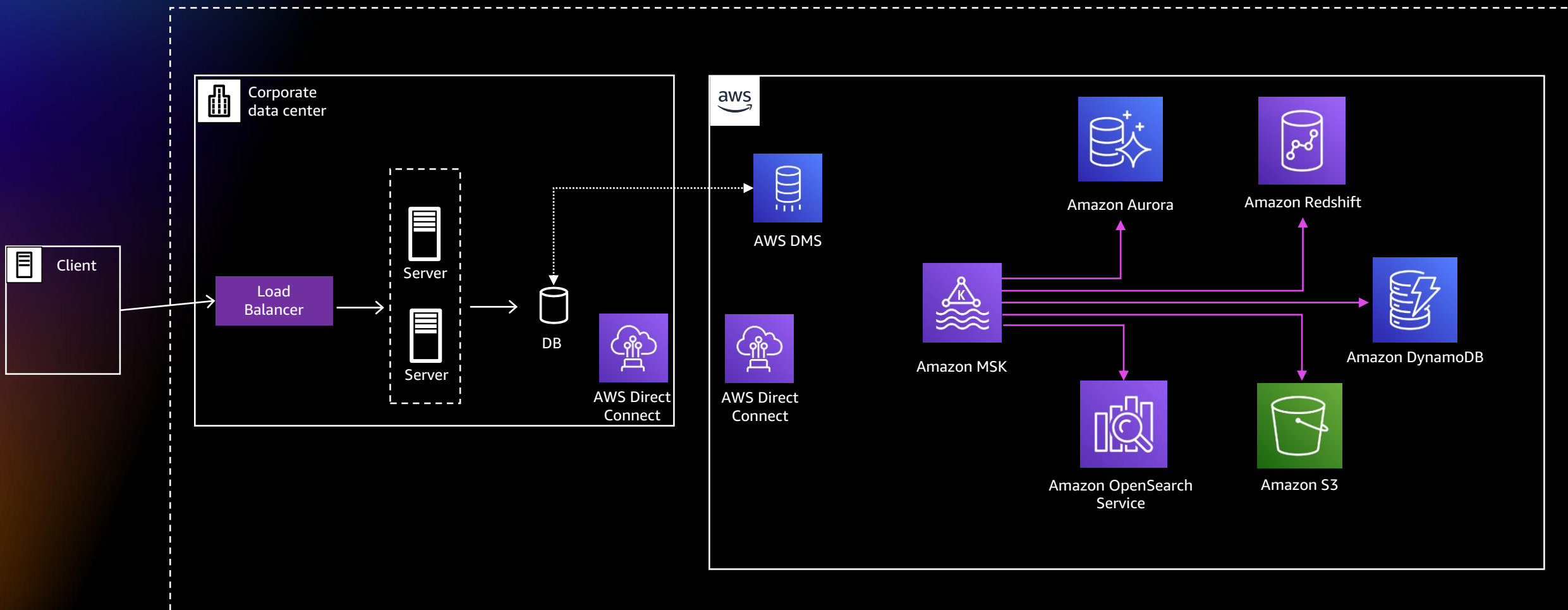


# Event Sourcing pattern

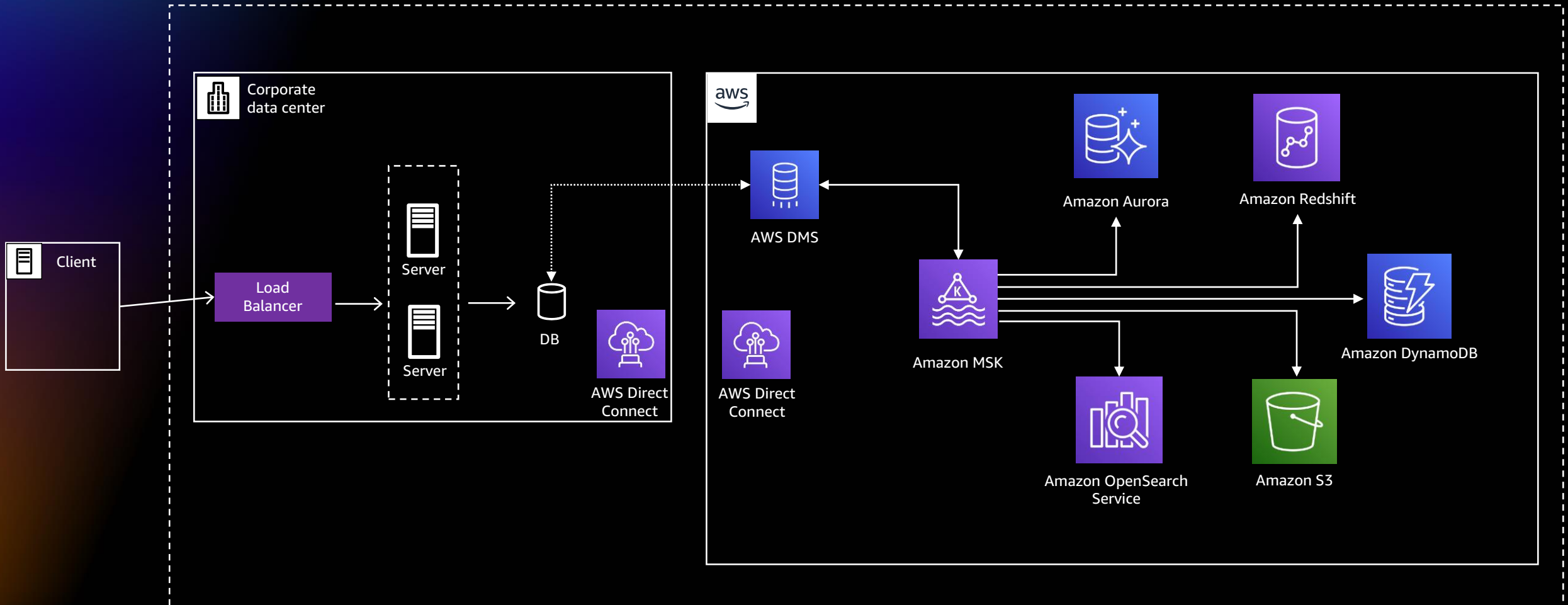




# Fan-Out pattern



# Event Sourcing and Fan Out patterns



# Other resources – Messaging and patterns

[Modern Application Development on AWS : Cloud-native Modern Development and Design Patterns on AWS whitepaper](#)

[AWS Blog - messaging](#)

[AWS re:Invent 2020: Application integration patterns for microservices](#)

[AWS Asynchronous messaging workshop](#)

# Other resources – AWS Solutions Constructs

[AWS Solutions Constructs patterns](#)

[Getting started with AWS Solutions Constructs](#)

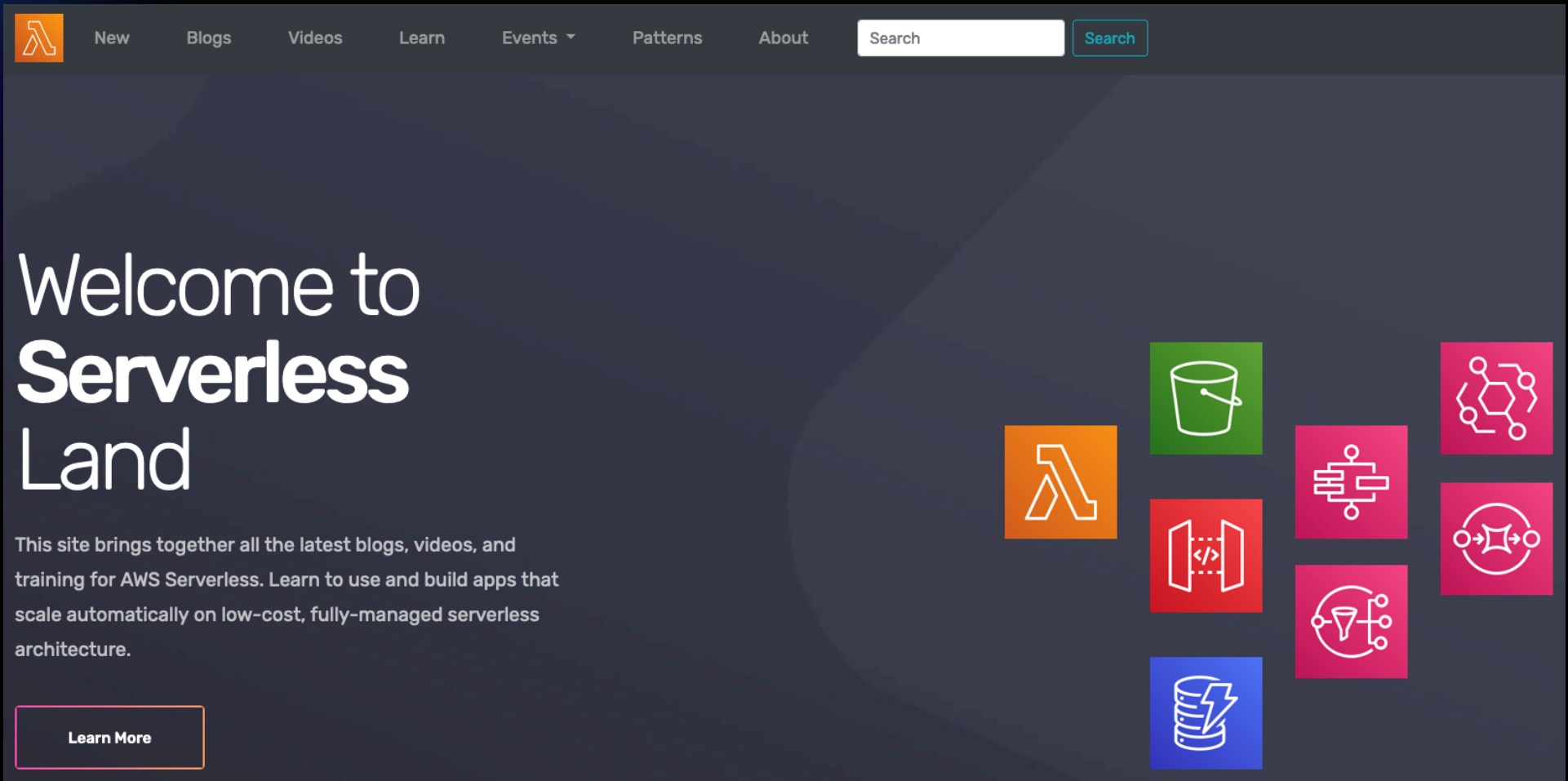
[AWS labs – AWS Solution Constructs](#)

[AWS Construct Hub \(\*Developer Preview\*\)](#)

[Beyond prototypes: Real world applications with AWS Solutions Constructs](#)

# Other resources : AWS Serverless Land

<https://serverlessland.com/>



# Visit the Modern Applications Resource Hub for more resources

Dive deeper with these resources to help you develop an effective plan for your modernization journey.

- Build modern applications on AWS e-book
- Build mobile and web apps faster e-book
- Modernize today with containers on AWS e-book
- Adopting a modern Dev+Ops model e-book
- Modern apps need modern ops e-book
- Determining the total cost of ownership: Comparing Serverless and Server-based technologies paper
- Continuous learning, continuous modernization e-book
- ... and more!



<https://bit.ly/3yfOvbK>

**Visit resource hub »**

# AWS Training and Certification

Accelerate modernization with continuous learning



Free digital courses, including:  
[Architecting serverless solutions](#)  
[Getting started with DevOps on AWS](#)



Earn an industry-recognized credential:  
[AWS Certified Developer – Associate](#)  
[AWS Certified DevOps – Professional](#)



Hands-on classroom training  
(available virtually) including:  
[Running containers on Amazon Elastic  
Kubernetes Service \(Amazon EKS\)](#)  
[Advanced developing on AWS](#)



Create a self-paced learning roadmap  
[AWS ramp-up guide - Developer](#)  
[AWS ramp-up guide - DevOps](#)



Take [Developer](#)  
[and DevOps training](#)  
today



Learn more about  
[Modernization training](#) for you  
and your team

# Thank you for attending AWS Innovate Modern Applications Edition

We hope you found it interesting! A kind reminder to **complete the survey**.  
Let us know what you thought of today's event and how we can improve the event  
experience for you in the future.



[aws-apj-marketing@amazon.com](mailto:aws-apj-marketing@amazon.com)



[twitter.com/AWSCloud](https://twitter.com/AWSCloud)



[facebook.com/AmazonWebServices](https://facebook.com/AmazonWebServices)



[youtube.com/user/AmazonWebServices](https://youtube.com/user/AmazonWebServices)



[slideshare.net/AmazonWebServices](https://slideshare.net/AmazonWebServices)



[twitch.tv/aws](https://twitch.tv/aws)



# Thank you!