AWS re:Invent

SVS305

# How to secure your serverless applications

**Angela Wang**
R&D Engineer
Amazon Web Services

**Roberto Iturralde**
Solutions Architect
Amazon Web Services

AWS re:Invent



aws

# Agenda

Serverless security: Is it different?

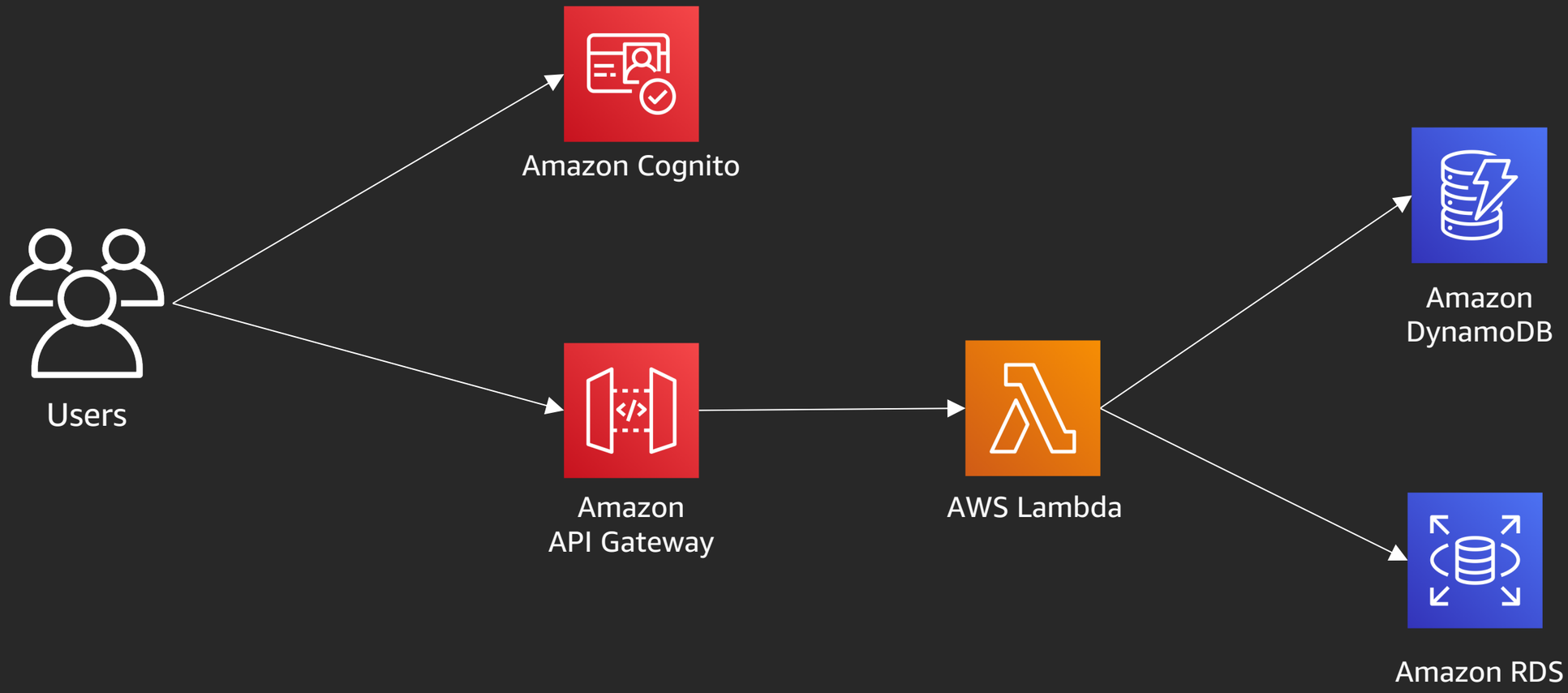
Security domains for serverless applications

Workshop scenario

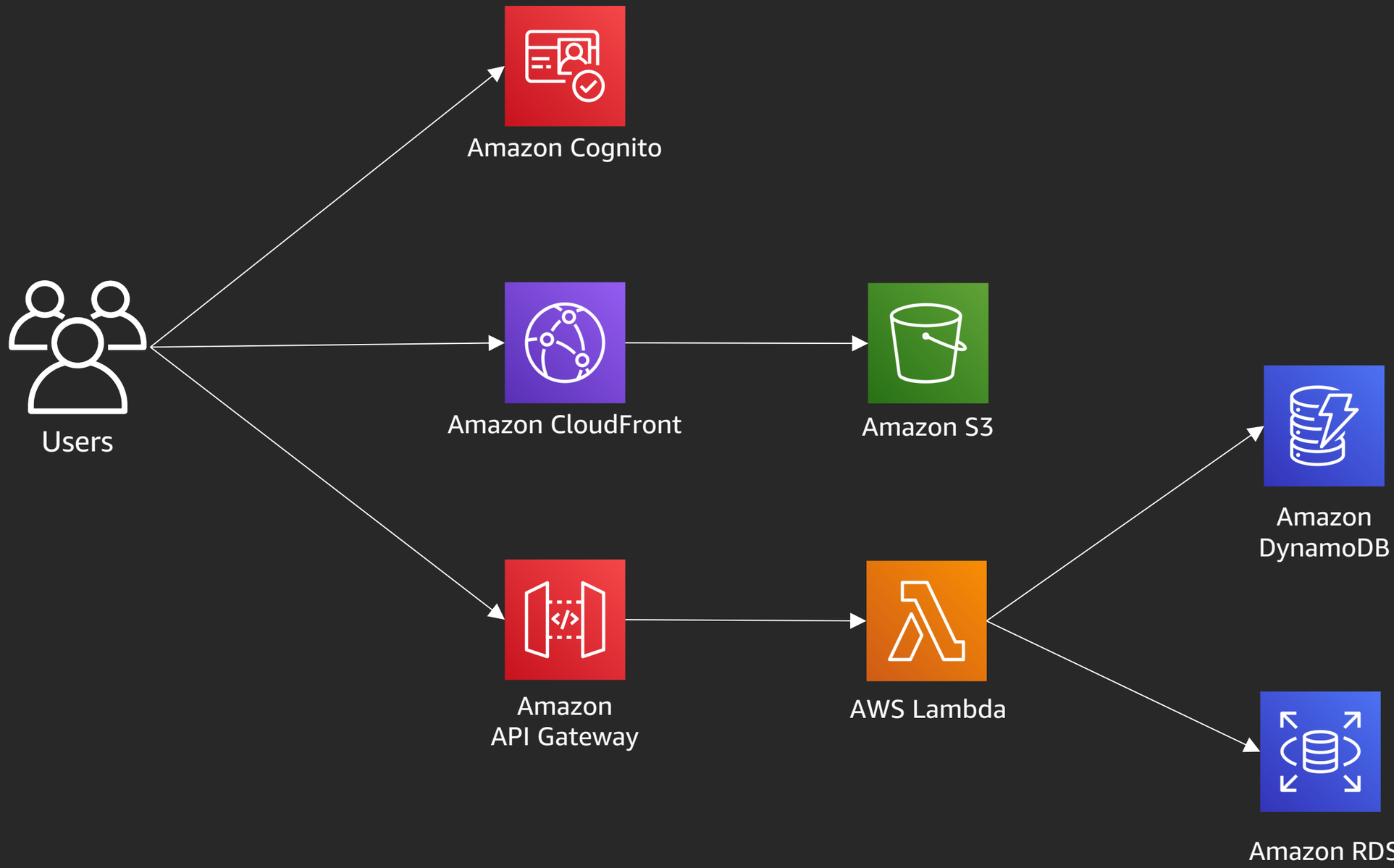
How to secure serverless applications

Hands-on practice

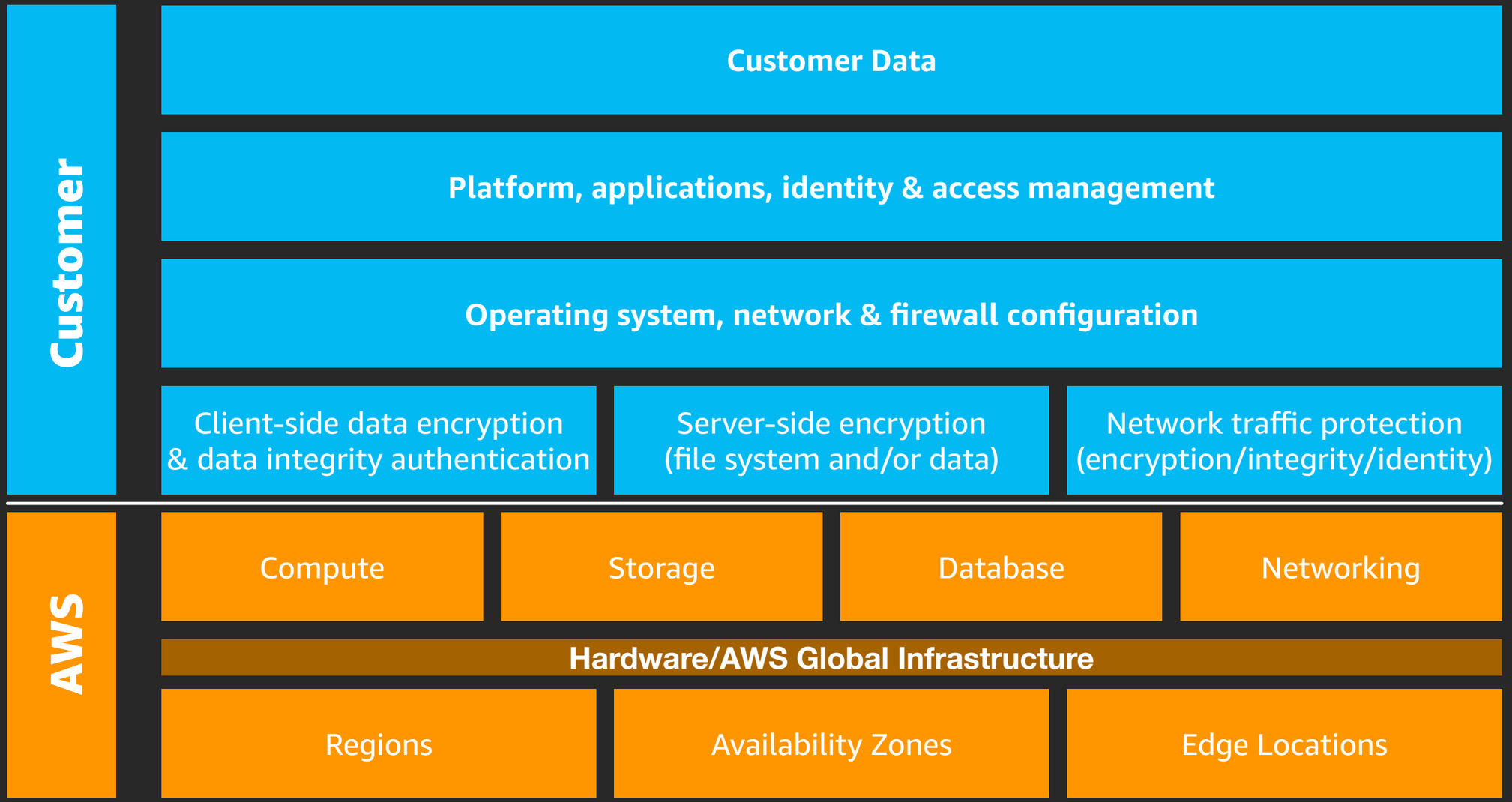
# Sample architecture for serverless API endpoint



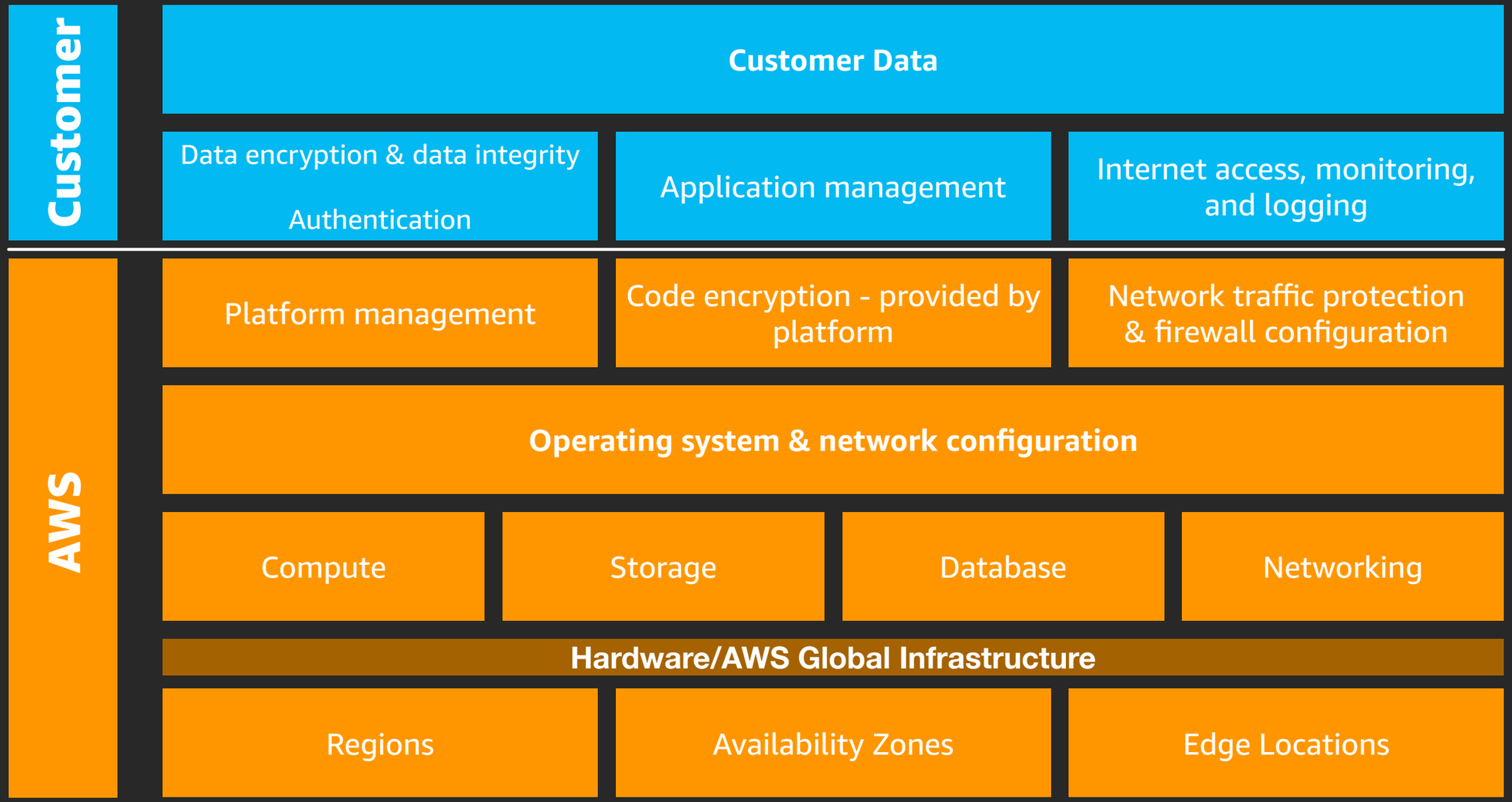
# Sample architecture for serverless web app



# Shared responsibility model: “Serverful"



# Shared responsibility model: Serverless



# What else is different

Ephemerality

Diffused perimeter

Fine-grained control

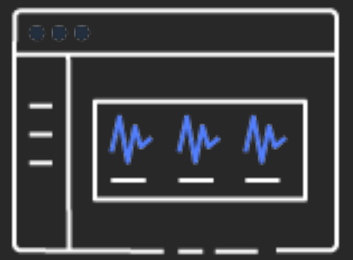
Tooling

# What stays the same

Securing data

Quality code

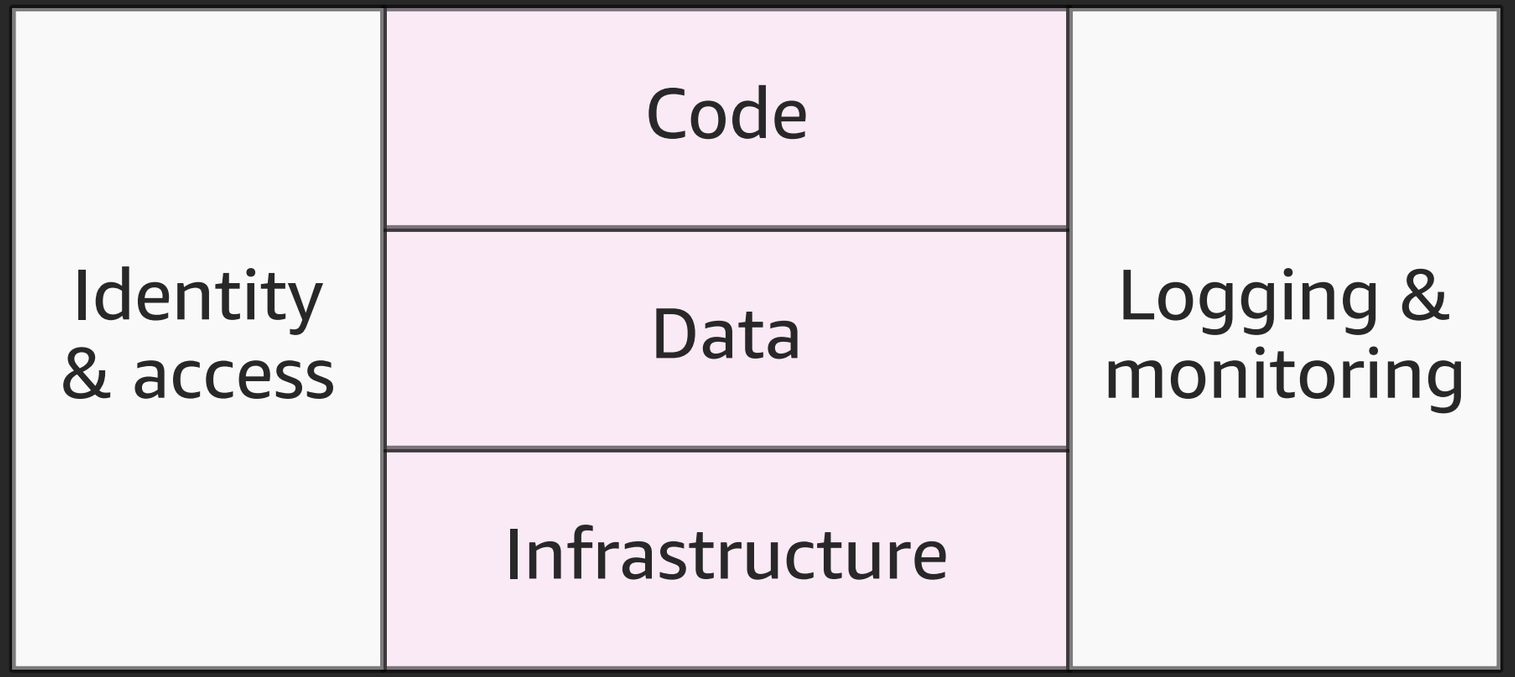
Least privilege



Monitoring

# Security domains for serverless applications

# Domains of security for (serverless) applications



# OWASP 2017: Top 10 web application security risks

- **Exploitability**
- **Prevalence**
- **Detectability**
- **Technical impact**

| Rank | Security risks |
|---|---|
| 1 | Injection |
| 2 | Broken authentication |
| 3 | Sensitive data exposure |
| 4 | XML external entities (XXE) |
| 5 | Broken access control |
| 6 | Security misconfiguration |
| 7 | Cross-site scripting (XSS) |
| 8 | Insecure deserialization |
| 9 | Using components with known vulnerabilities |
| 10 | Insufficient logging & monitoring |

https://www.owasp.org

# OWASP: Top 10 mapped to security domains

## Identity & access

- Broken authentication(#2)
- Broken access control (#5)

## Code

- Injection (#1)
- XXE (#4)
- XSS (#7)
- Insecure deserialization (#8)
- Using components with known vulnerabilities (#9)

## Data

- Sensitive data exposure (#3)

## Infrastructure

- Using components with known Vulnerabilities (#9)

## Logging & monitoring

- Security misconfiguration (#6)
- Insufficient logging & Monitoring (#10)

# Workshop scenario

# Scenario: Wild Rydes (www.wildrydes.com)

# Third-party functionality: Unicorn customization


Sock image credit: Freepik from www.flaticon.com

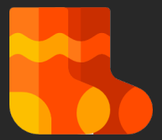
# Third-party API: Unicorn customization

List customization options and prices:

GET /capes 

GET /glasses 

GET /horns 

GET /socks 

# Third-party API: Unicorn customization

Create and manage customizations

POST /customizations

GET /customizations

GET /customizations/{id}

DELETE /customizations/{id}

# Admin API: Register third-party partners

Register new partners

POST /partners

# Workshop architecture: Starting point

Deployed using
AWS Serverless Application Model (AWS SAM)


Not secure!

# Your task: Secure the application against attackers

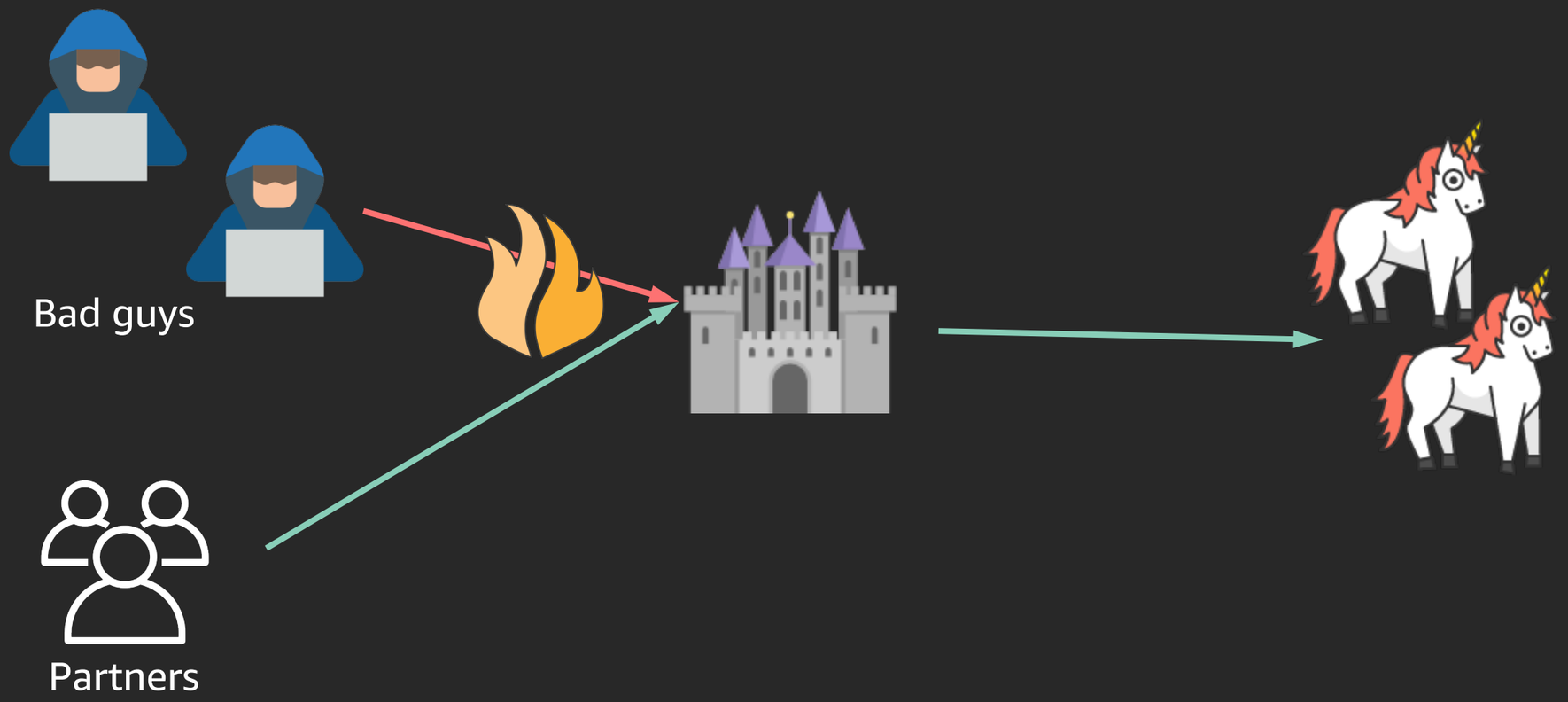



Image Credit:
pongsakornred, Freepik from www.flaticon.com

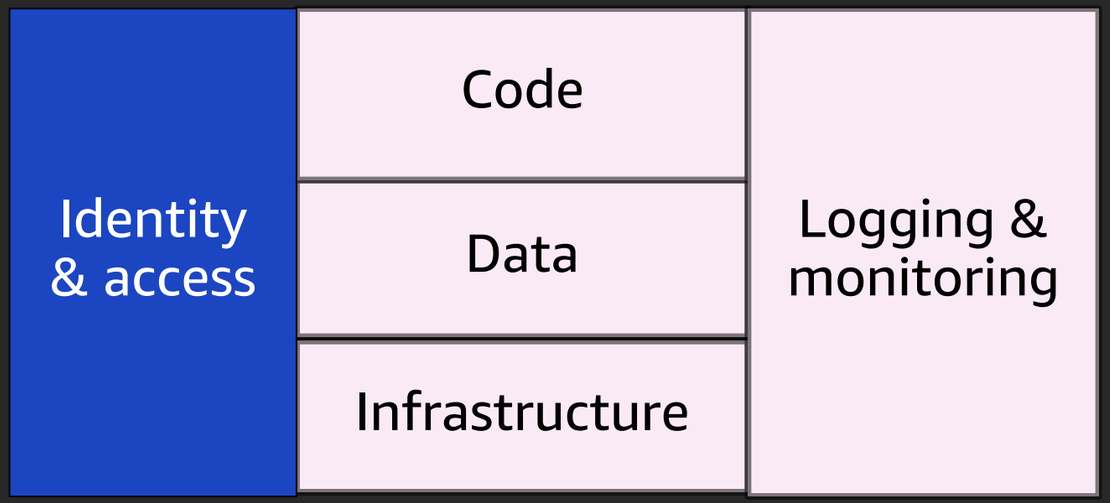
# How to secure serverless applications

AWS re:Invent



aws

# Identity and access management for serverless applications




- Authenticate and authorize users or clients
- Access between backend services (e.g., AWS Lambda to Amazon DynamoDB tables)

# Identity and access management for serverless applications

## Authenticate & authorize users/clients

API Gateway

3 ways for AuthN & AuthZ:

- AWS IAM
- Lambda custom authorizer
- Cognito user pool authorizer

Cognito

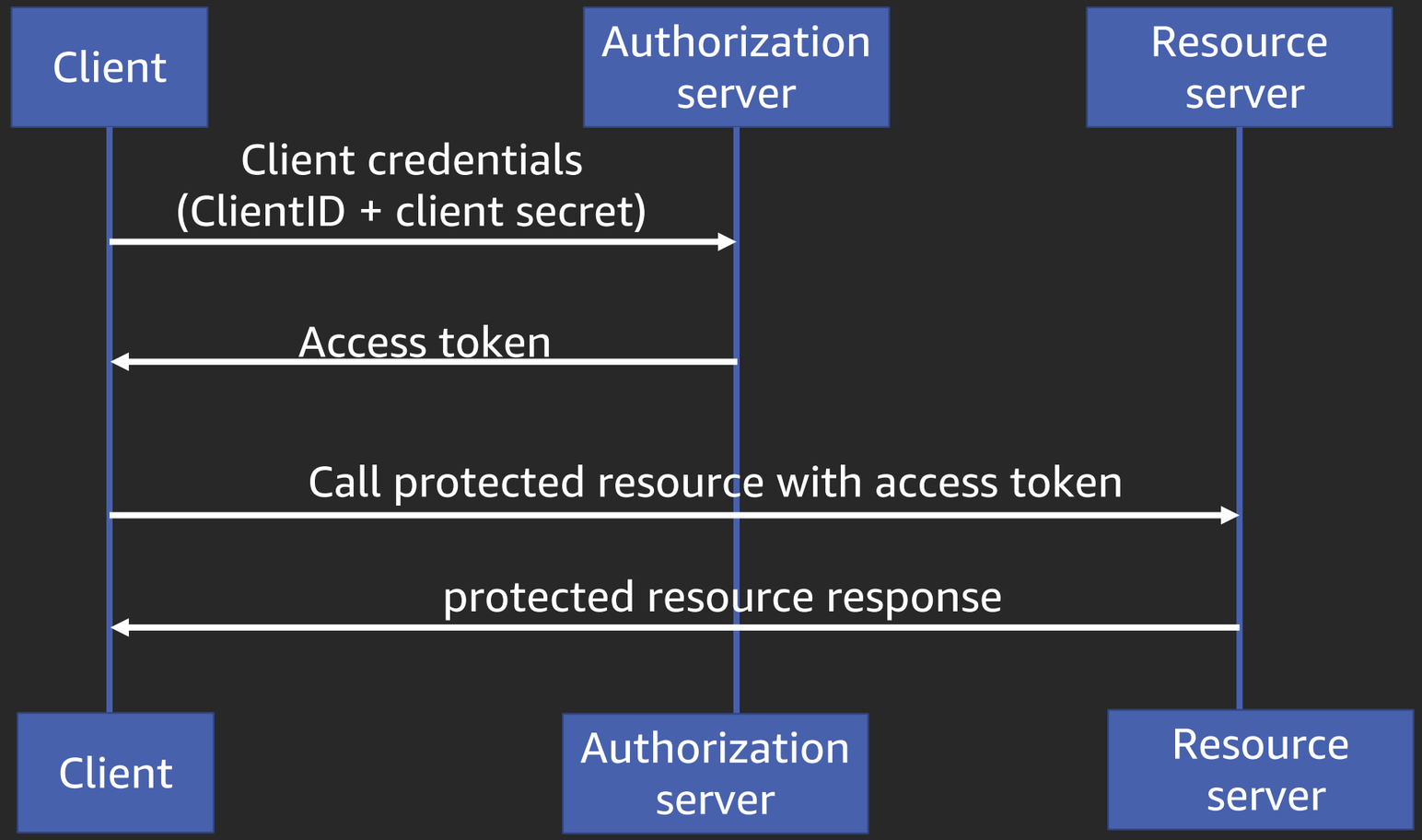
- Managed user directory or federation with other Idps
- Standard JWT tokens or AWS credentials

## Access control between services

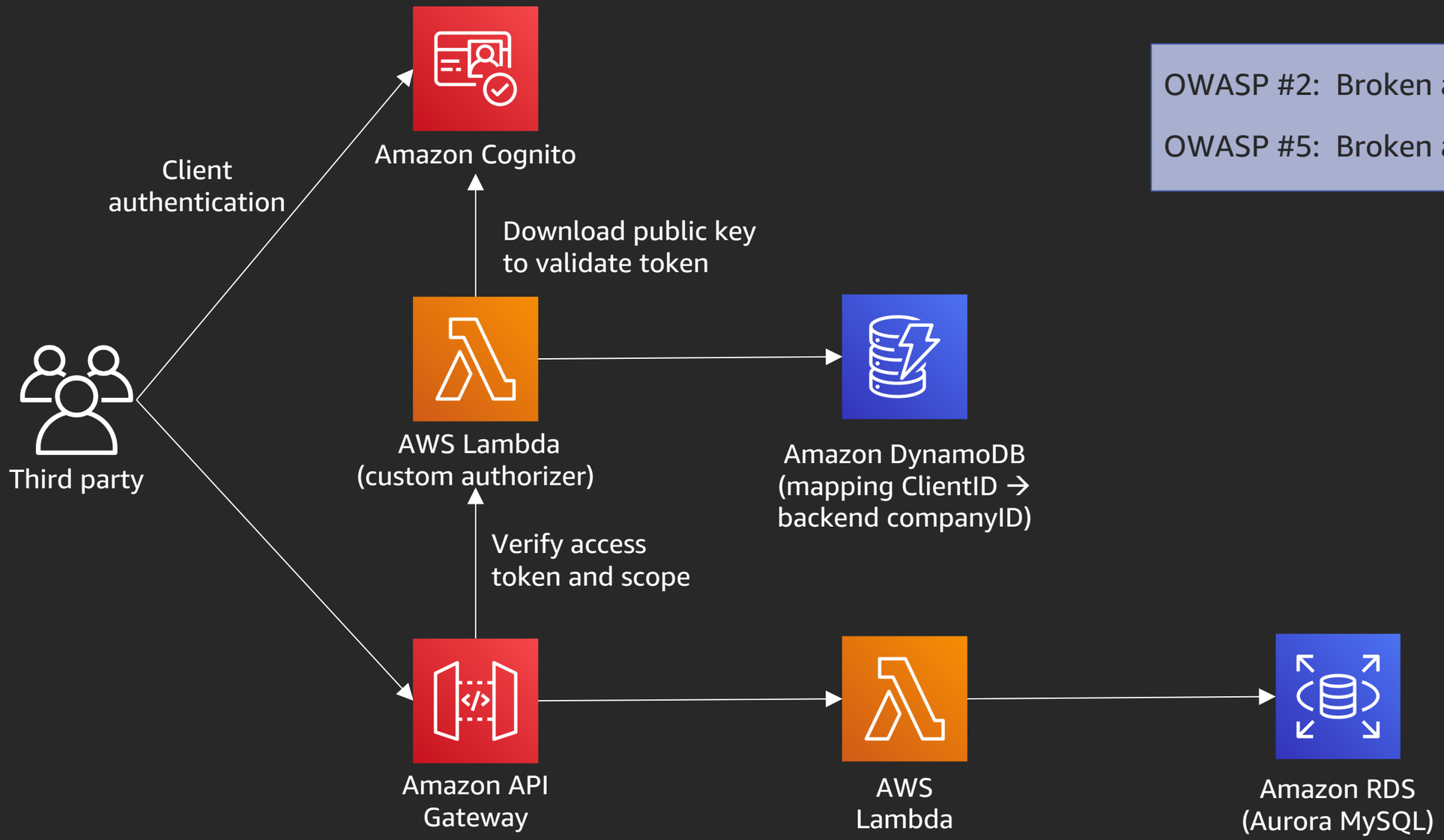
AWS Lambda:

- Invocation permissions
- Execution permissions

# Workshop module 1: OAuth client credentials flow



# Workshop module 1: Add authentication




OWASP #2: Broken authentication

OWASP #5: Broken access control

# Workshop module 1: Add authentication

Amazon Cognito

Admin app client:
- Client ID: ZZZ
- Client Secret

Company foo app client:
- Client ID: XXX
- Client Secret

Company bar app client:
- Client ID: YYY
- Client Secret

...

Amazon DynamoDB

**Mapping table**

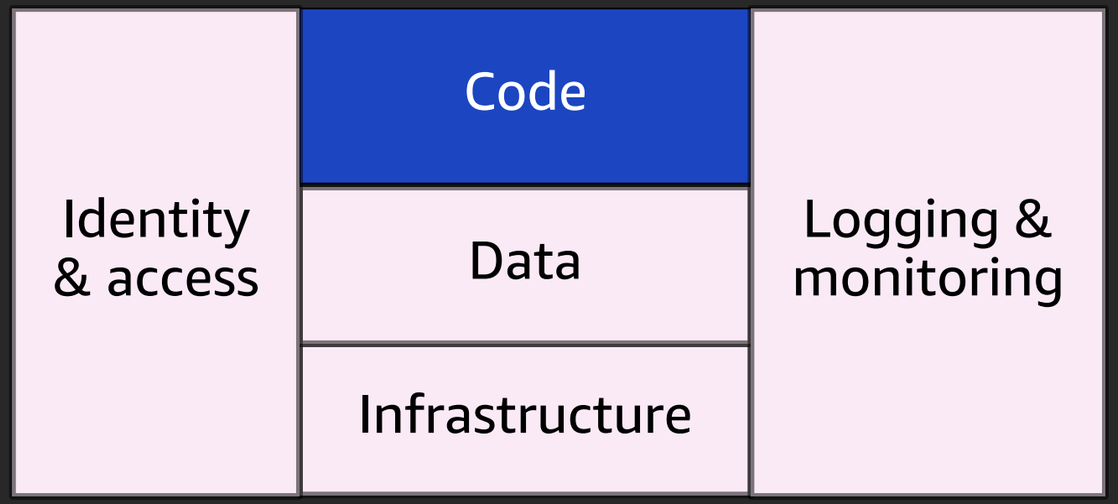
| ClientID | BackendID |
|---|---|
| XXX | 1 |
| YYY | 2 |

Amazon Aurora

**Company table**

| ID | Name |
|---|---|
| 1 | Foo |
| 2 | Bar |

# Securing code for serverless applications




- Input validation
- Dependency vulnerabilities
- Secrets in source code

# Securing code for serverless applications

## Input validation

AWS WAF:

- XSS rules
- SQL injection rules

Amazon API Gateway:

- Request validation

AWS Lambda:

- Sanitize input in code

## Dependency vulnerabilities

AWS Lambda:

- Minimize dependencies
- Vulnerability dependency check tools:
  - OWASP
  - Snyk
  - Twistlock
  - …

## Storing secrets

AWS Secrets Manager

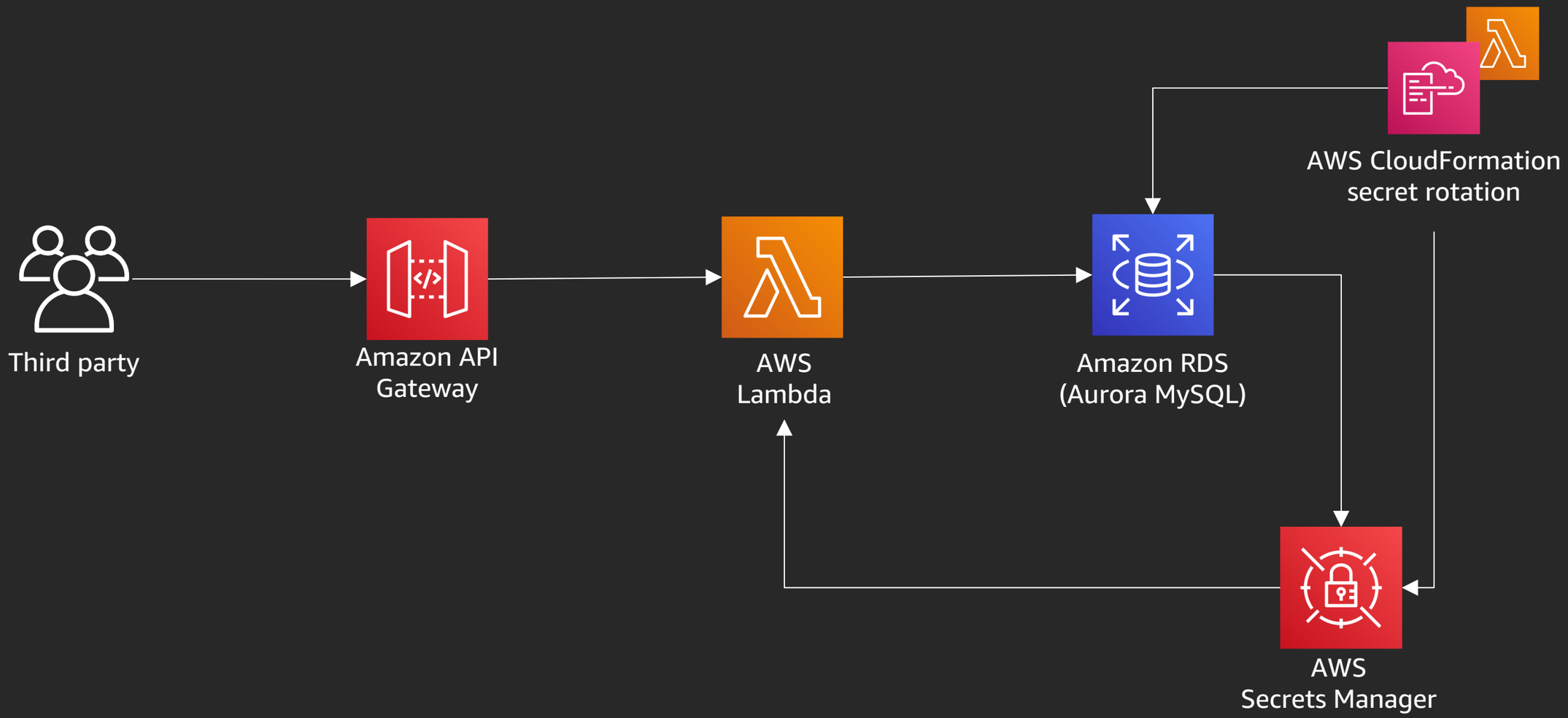
AWS Systems Manager Parameter Store

AWS Lambda encrypted environment variables

Go passwordless with IAM auth

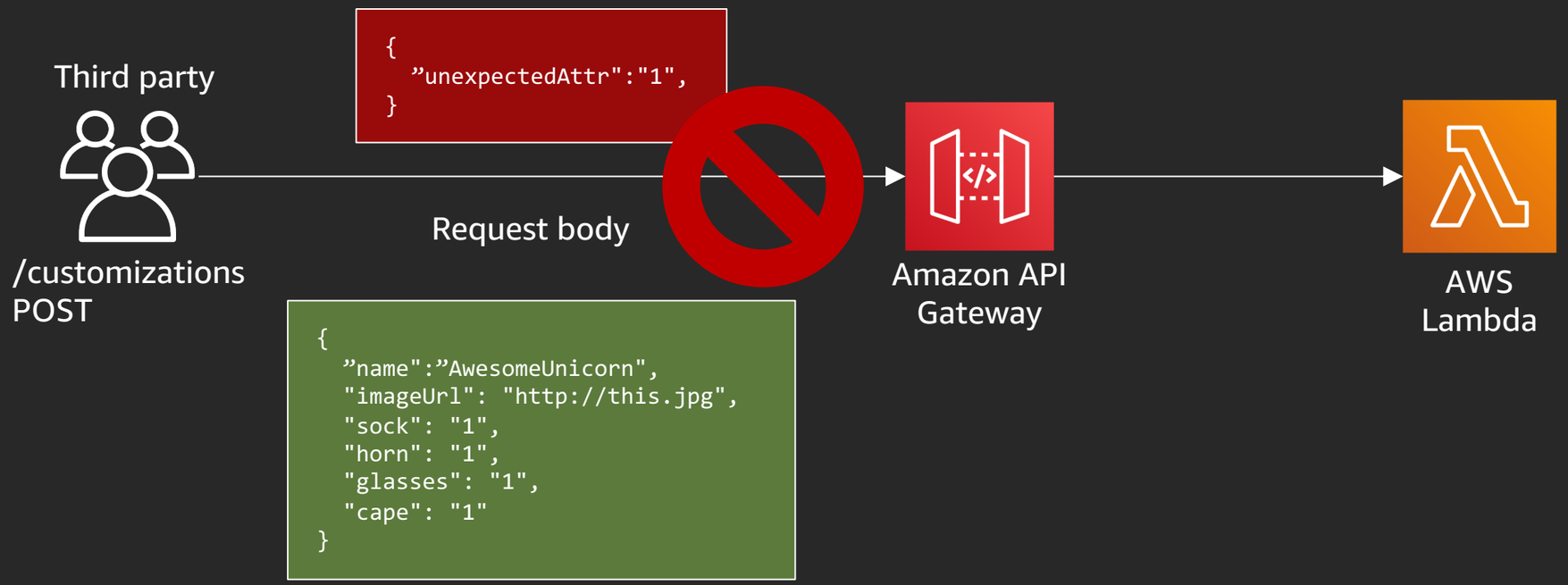
# Module 2: Secret manager

OWASP #3: Sensitive data exposure



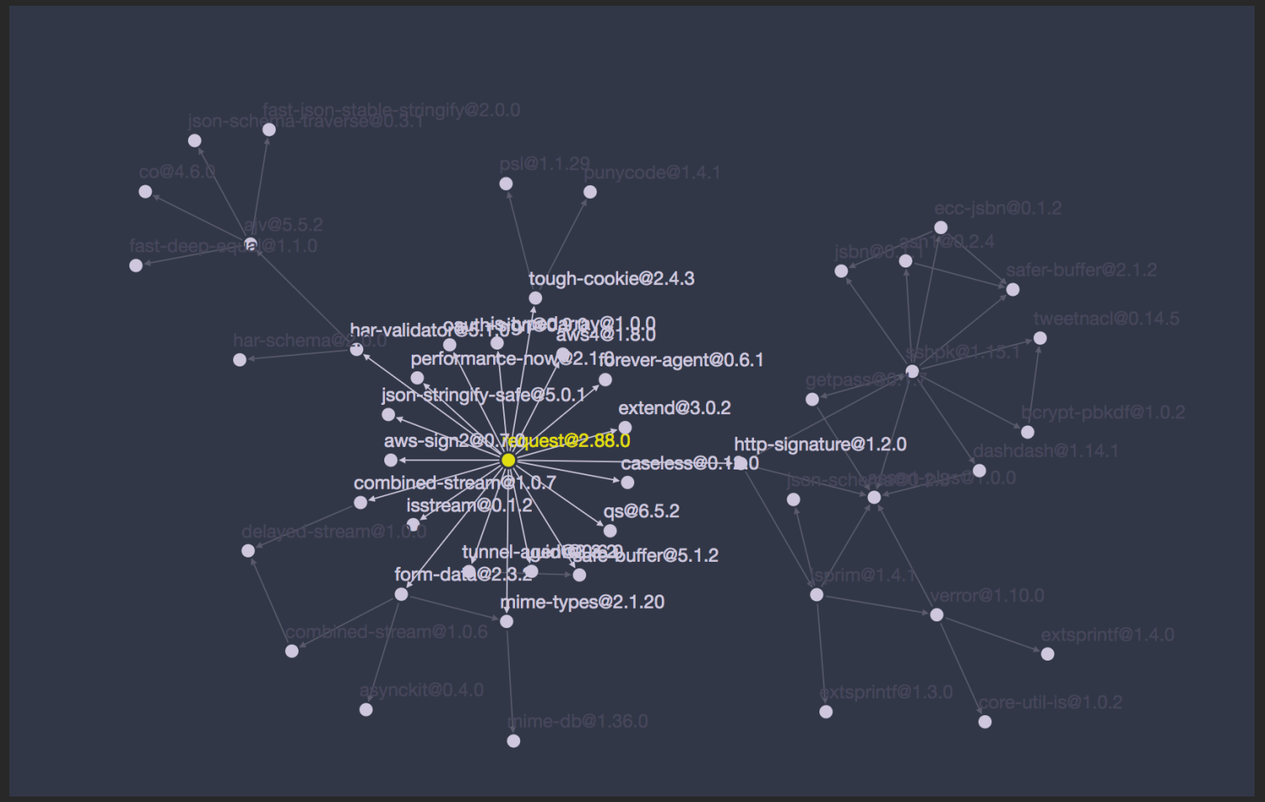
# Module 3: Input validation

- OWASP #1: Injection
- OWASP #8: Insecure deserialization



# Module 7: Dependency vulnerability

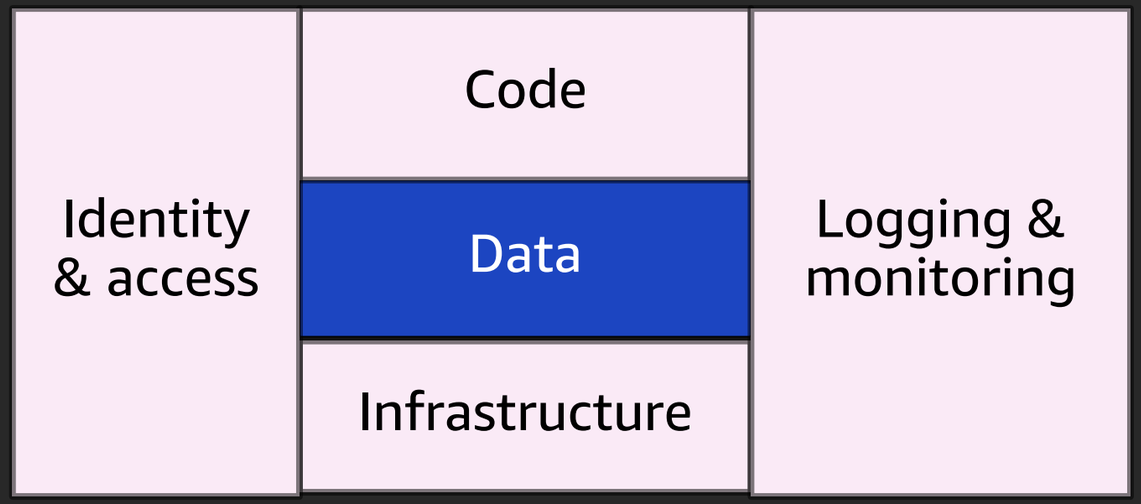
- OWASP #9: Using components with known vulnerabilities




http://npm.anvaka.com/#/view/2d/request

- Check for vulnerabilities on our dependencies
  - OWASP Dependency Check: https://www.owasp.org/index.php/OWASP_Dependency_Check
  - Third-party tools
- Remove unused dependencies
  - depcheck: https://www.npmjs.com/package/depcheck

# Securing data for serverless applications




Your responsibility:

- Data classification and data flow
- Tokenization
- Encryption at rest
- Encryption in transit
- Data backup, replication & recovery

AWS takes care of:

- Automatic replication of data across Availability Zones for high durability
- Managed backups & encryption

# Securing data for serverless applications

## Data classification

Amazon Macie

## Data flow

AWS X-Ray

## Data tokenization

- DIY
- AWS Marketplace

## Data encryption at rest

AWS KMS:

- Server-side encryption
  - Amazon S3, Amazon DynamoDB, Amazon RDS
  - …
- Client-side encryption

## Data encryption in transit

API Gateway: HTTPS only

Amazon Certificate Manager:

- Manage SSL certs for custom domains

## Data backup & replication

Amazon S3

- Versioning
- MFA delete
- Cross-region replication

Amazon DynamoDB

- On-demand backup
- Point-in-time restore
- Change streams

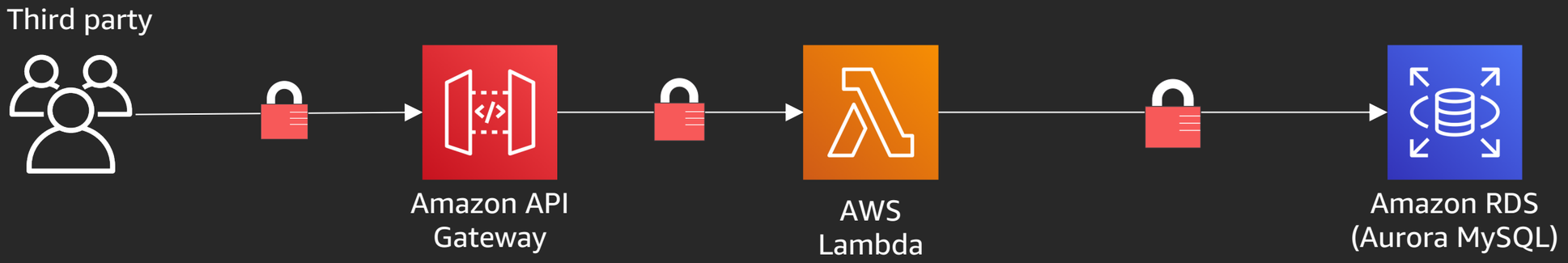
Amazon RDS

- Automated backups

# Module 4: Encryption in transit

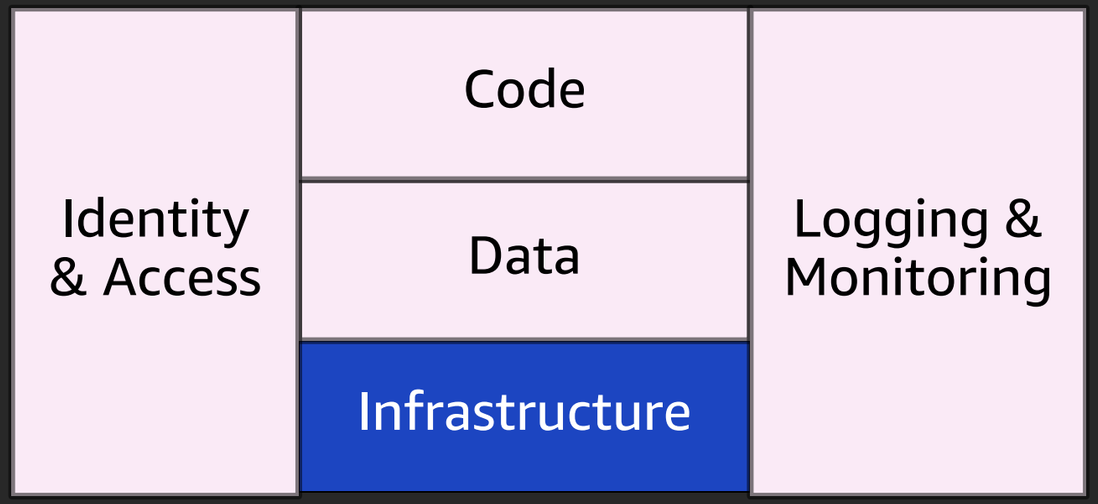
OWASP #3: Sensitive data exposure

```
{
      host: ”database.host.com",
      user: "admin",
      password: ”xxxxxxxxx",
      database: "unicorn_customization"
}
```




```
{
      host: ”database.host.com",
      user: "admin",
      password: ”xxxxxxx",
      database: "unicorn_customization",
      ssl: "Amazon RDS"
}
```

# Securing infrastructure for serverless applications




Your responsibility:

- DDoS protection
- Throttling/ Rate limiting
- Network boundaries

Serverless platform takes care of:

- Physical security
- Virtualization
- OS security & patching
- Scaling & HA

# Securing infrastructure for serverless applications

## DDoS protection + Throttling or rate limiting + Network boundaries


AWS Shield Standard


AWS Shield Advanced


AWS WAF:

- Geoblocking
- IP reputation lists
- Rate-based rules
- Size constraint
- …


API Gateway:

- Account level throttling
- API Stage level throttling
- Usage plan
  - Method level throttling
  - Metered by API key
  - Request rate and quota limits


AWS Lambda:

- Concurrency limits

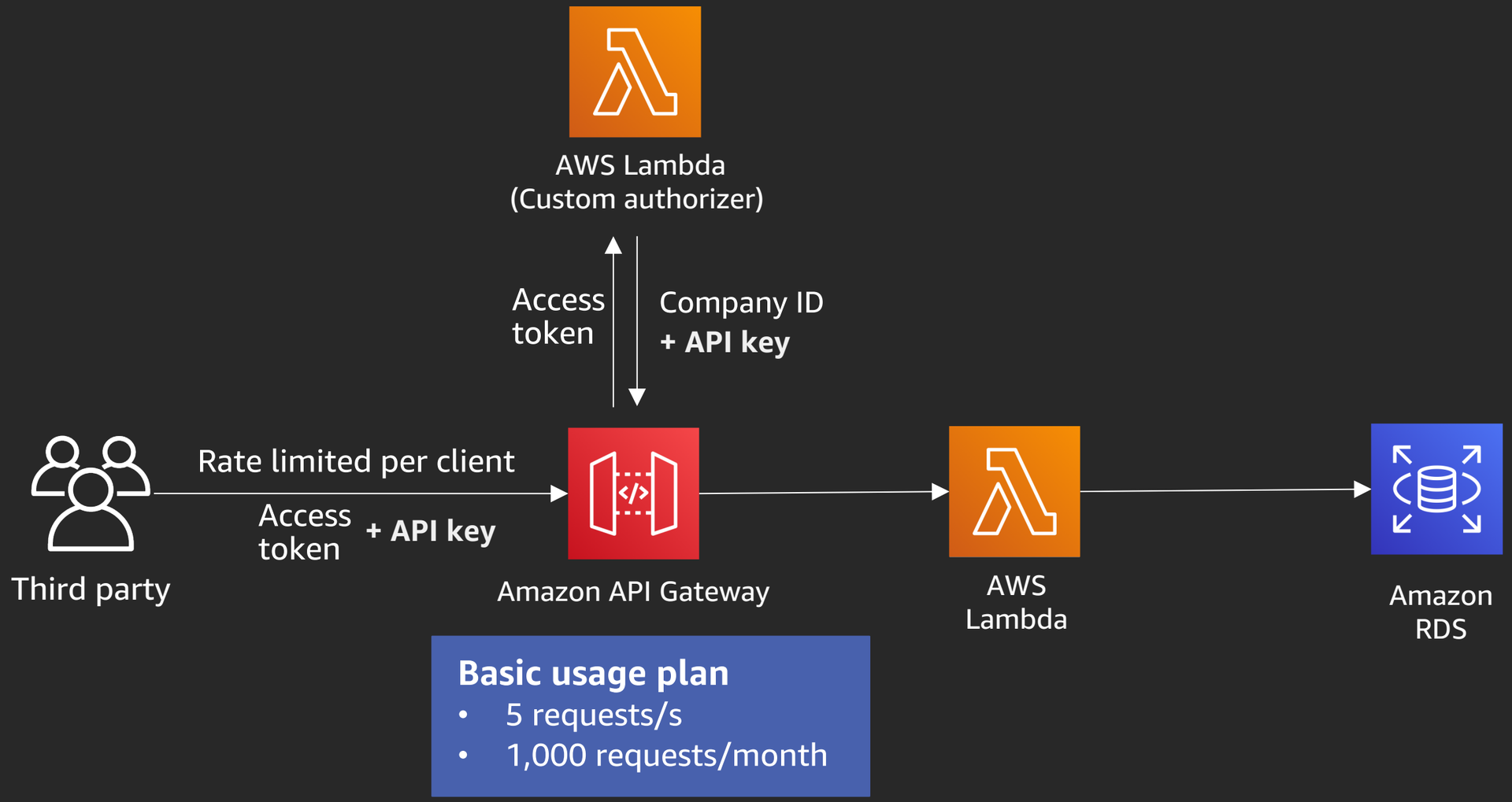

API Gateway:

- Private VPC endpoints
- Resource policy


AWS Lambda:

- Access resources in VPC
- Security groups
- NACLs
- Proxy-based egress filtering

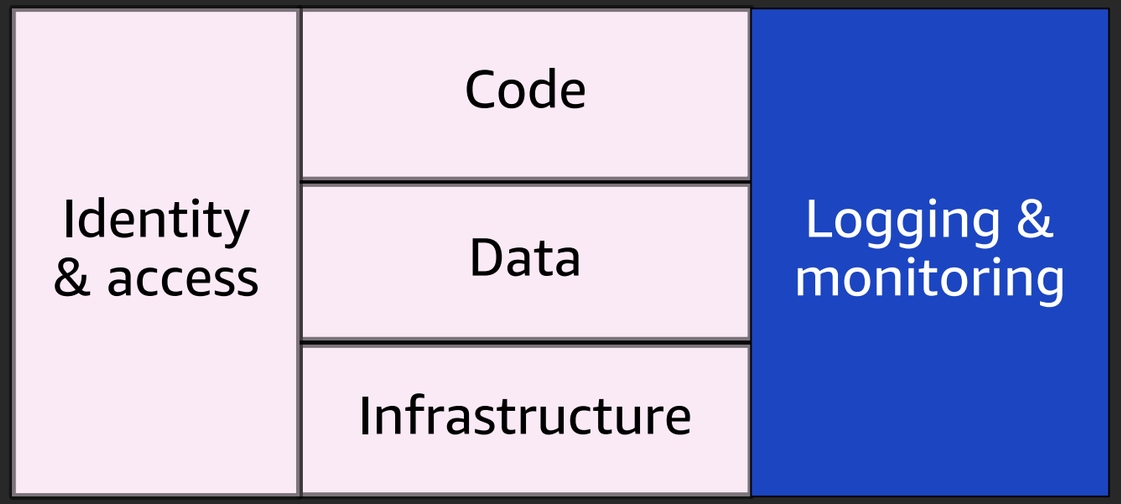
# Module 5: Usage plans



# Module 6: AWS WAF

- IP reputation lists
- Size restrictions
- SQL injection
- XSS
- …

# Logging & monitoring for serverless applications




- Application logs
- Access logs
- Control plane audit logs
- Metrics
- Alarms
- Compliance validation

# Logging & monitoring for serverless applications

## Logging and tracing

API Gateway:

- Access logs
- Execution logs

AWS Lambda:

- CloudWatch Logs

X-Ray

## Metrics

API Gateway:

- Built-in CloudWatch metrics
- Detailed CloudWatch metrics

AWS Lambda :

- Built-in CloudWatch metrics
- Custom CloudWatch metrics
- Metrics from CloudWatch Logs
- Third-party tools:
  - IOPipe, Datadog, …

## Compliance validation

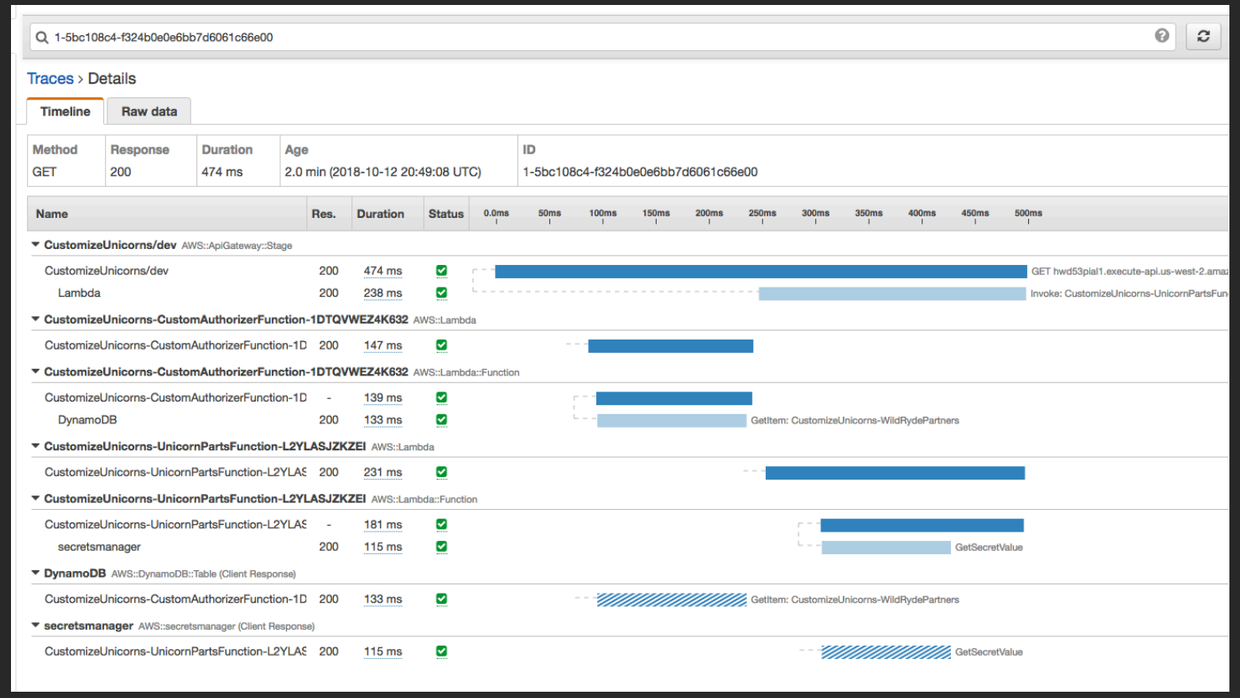
AWS Config

CloudWatch Events

AWS Budgets

# Module 8: AWS X-Ray

OWASP #10: Insufficient logging & monitoring

# Serverless security partners



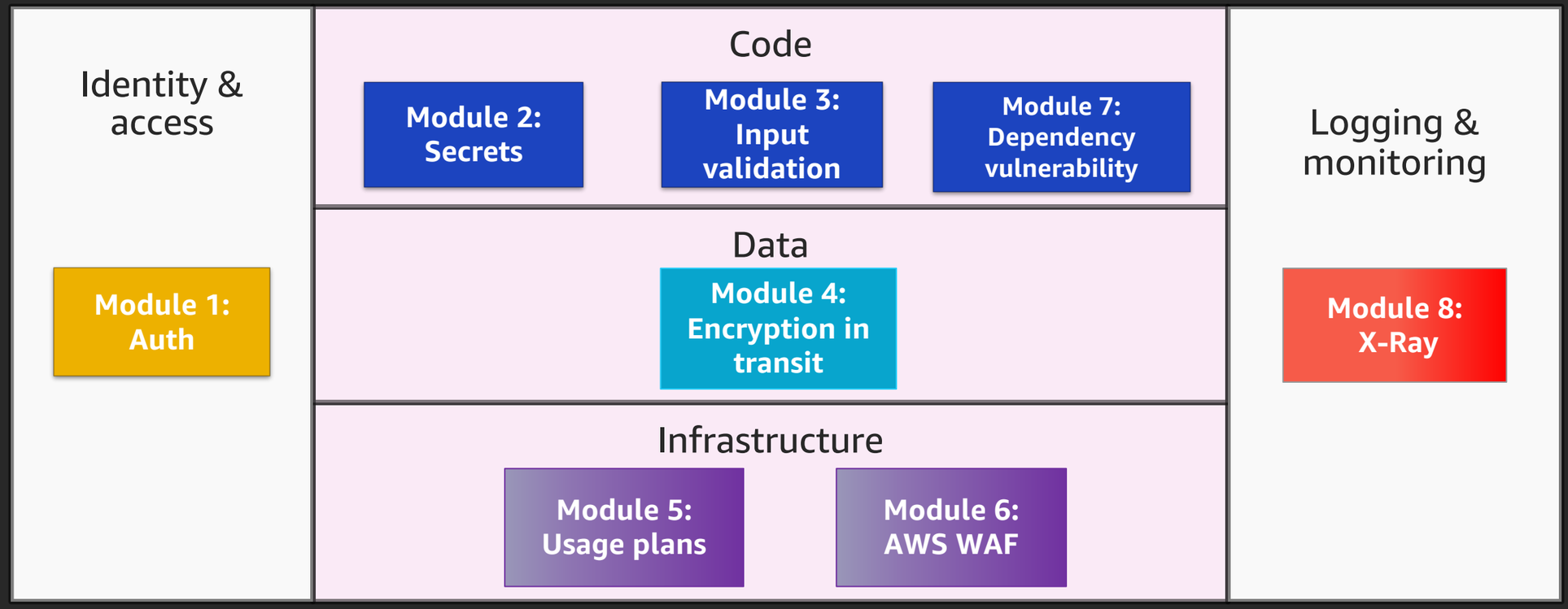


# Workshop

Link to the workshop: http://bit.ly/secure-serverless

Module 0 mandatory

Module 1–8: Pick your own adventure



# Thank you!

AWS re:Invent



aws