

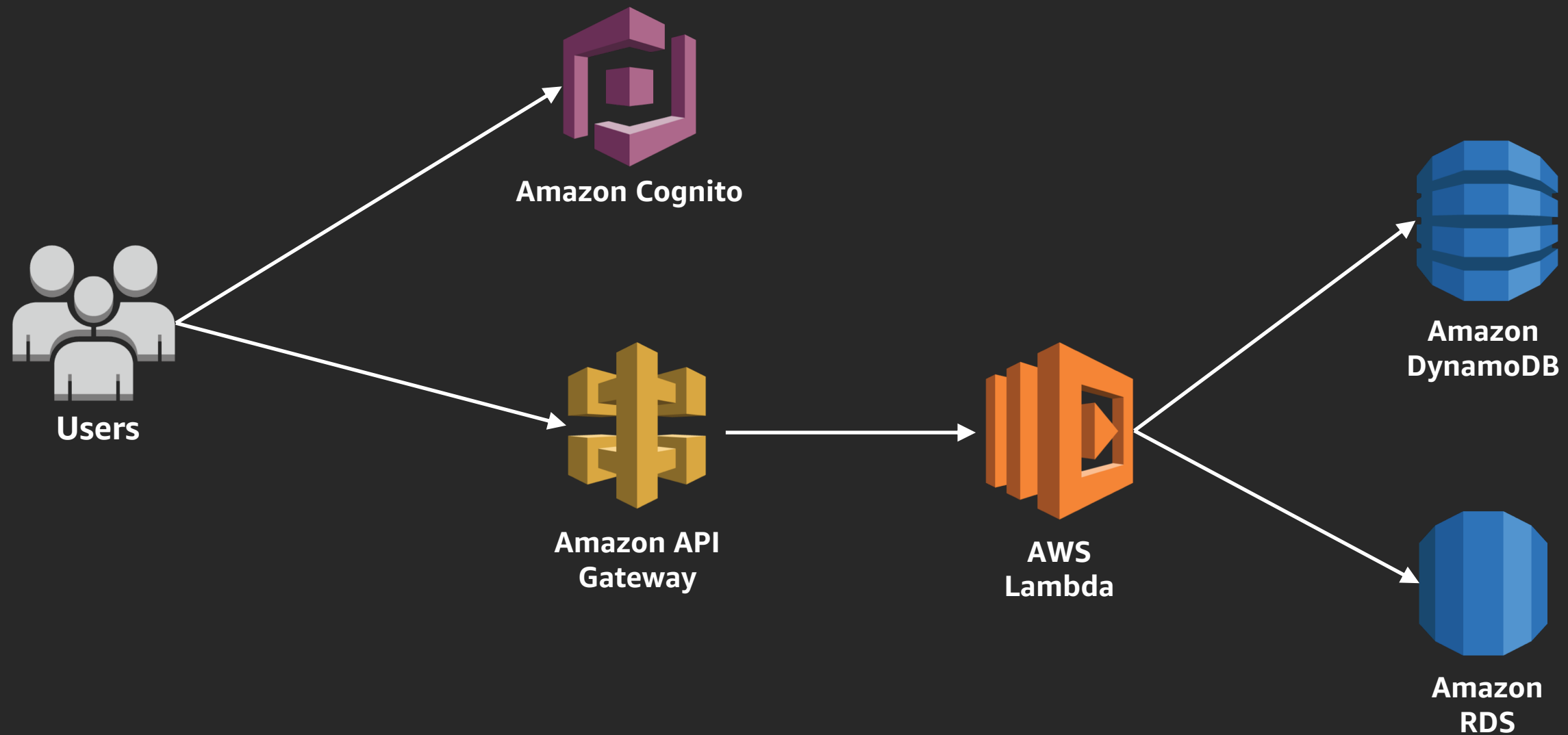
SRV314

Workshop: Securing Serverless Applications and AWS Lambda

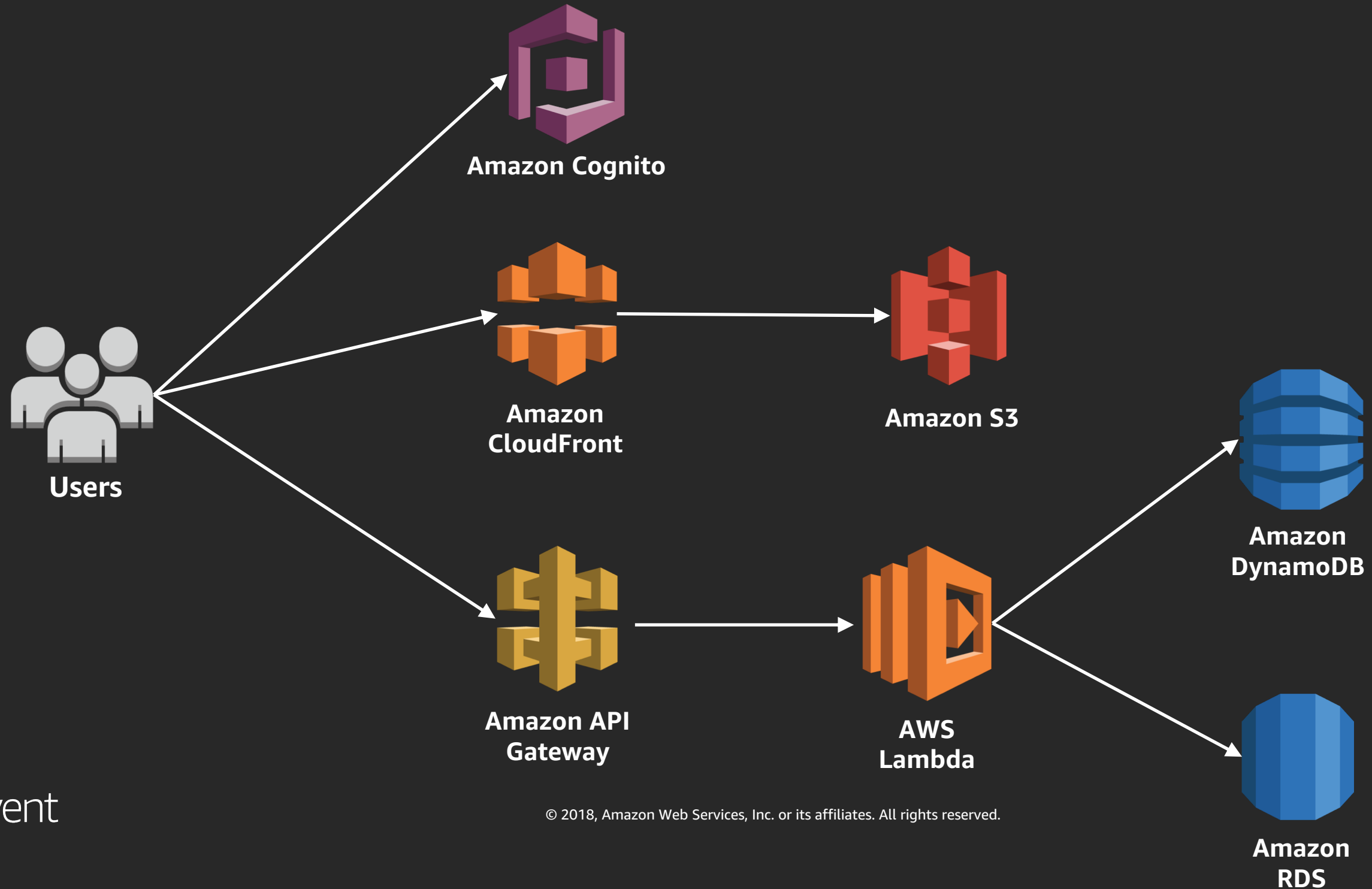
Agenda

- Serverless security – is it different?
- Security domains for serverless applications
- Workshop scenario
- How to secure serverless applications
- Hands-on

Sample architecture for serverless API endpoint



Sample architecture for serverless web app



How is serverless security different?

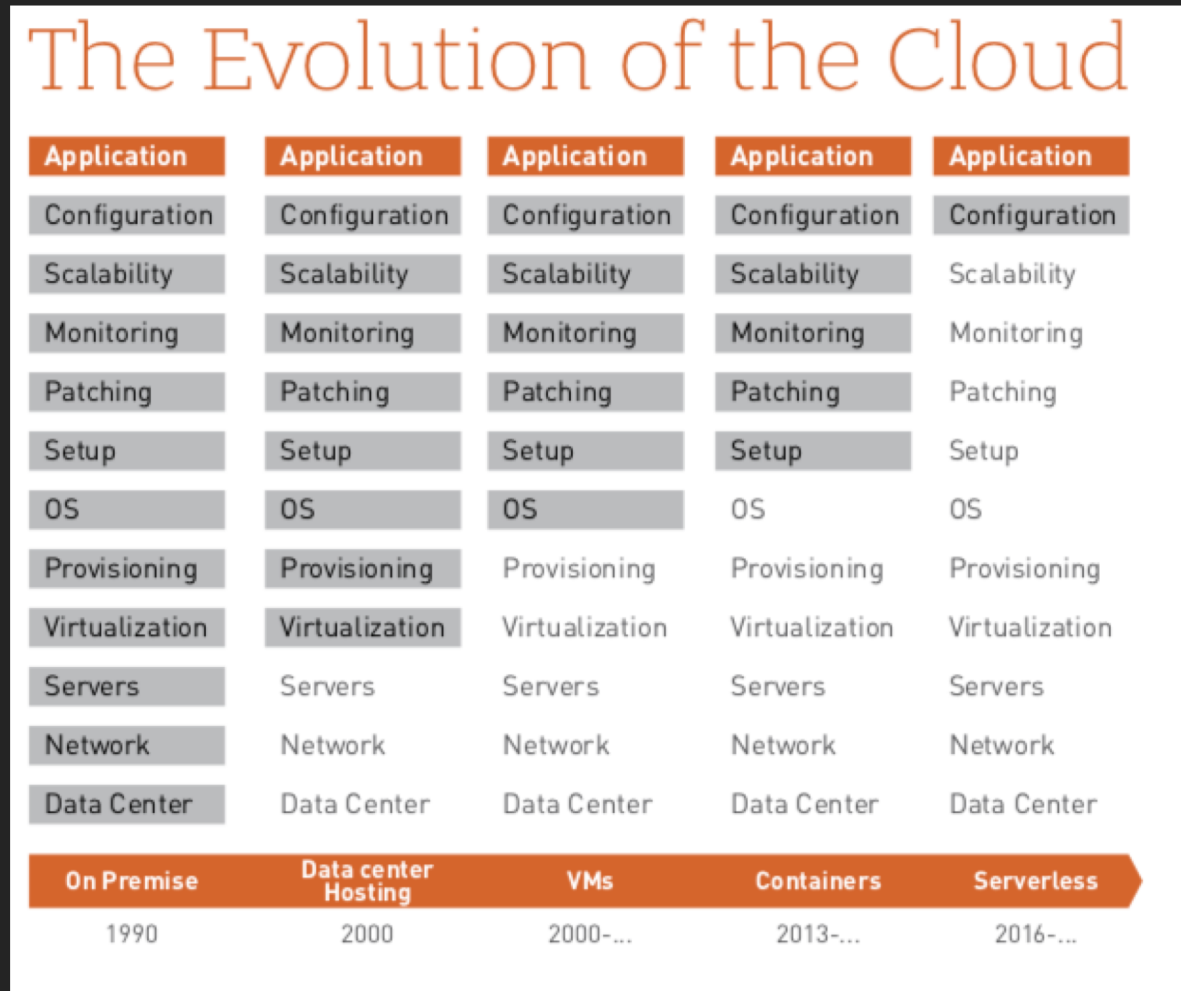


Image credit: Protego Serverless Security Primer eBook

<https://www.protego.io/ebook/>

Different:

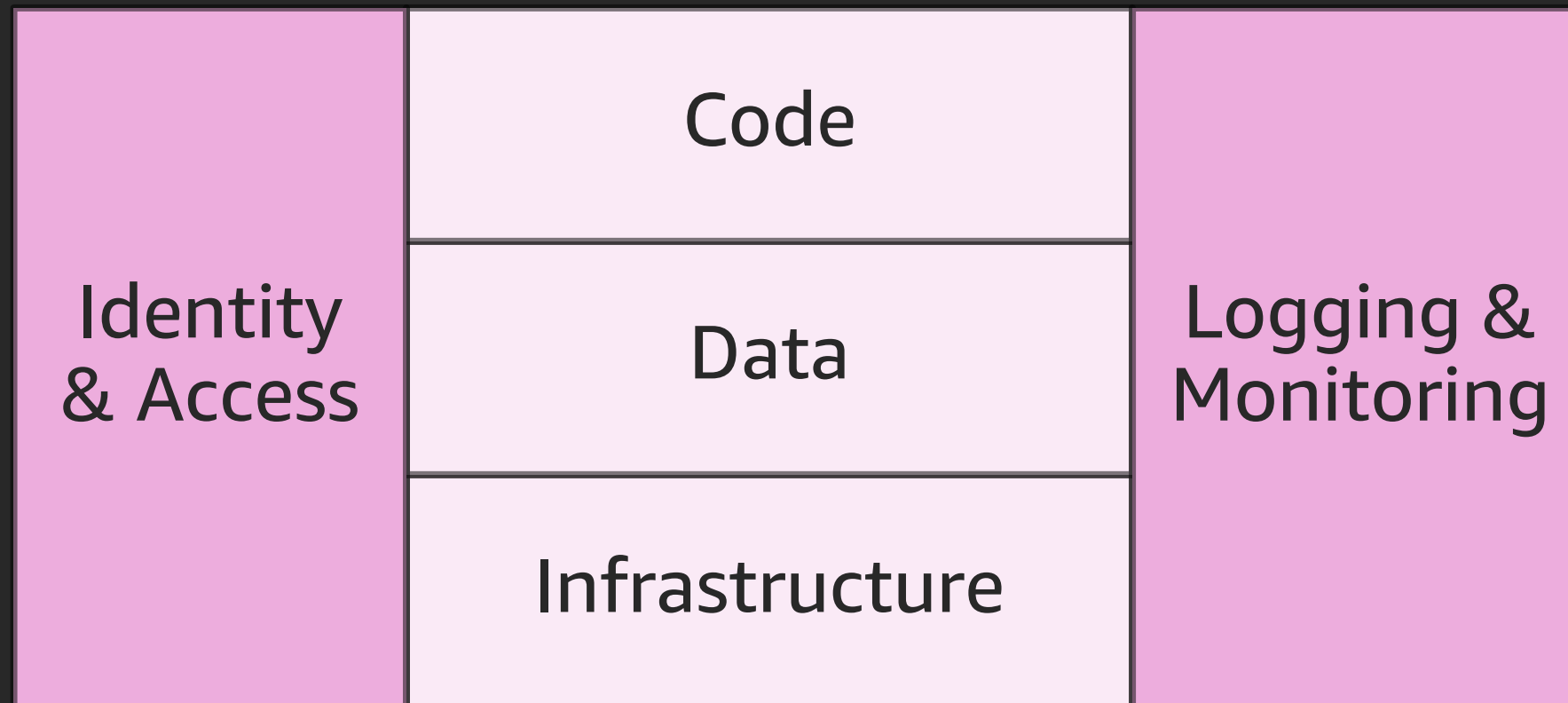
- Reduced scope
- Ephemeral environment
- More events can trigger your lambda
- Old techniques might not be relevant

But still...

- Need to secure databases, s3 buckets, etc.
- Need to secure your code.
- Need to use minimum privilege access.
- Need to monitor usage and data flow.

Security domains for serverless applications

Domains of security for (serverless) applications



OWASP 2017- Top 10 Web Application Security Risks

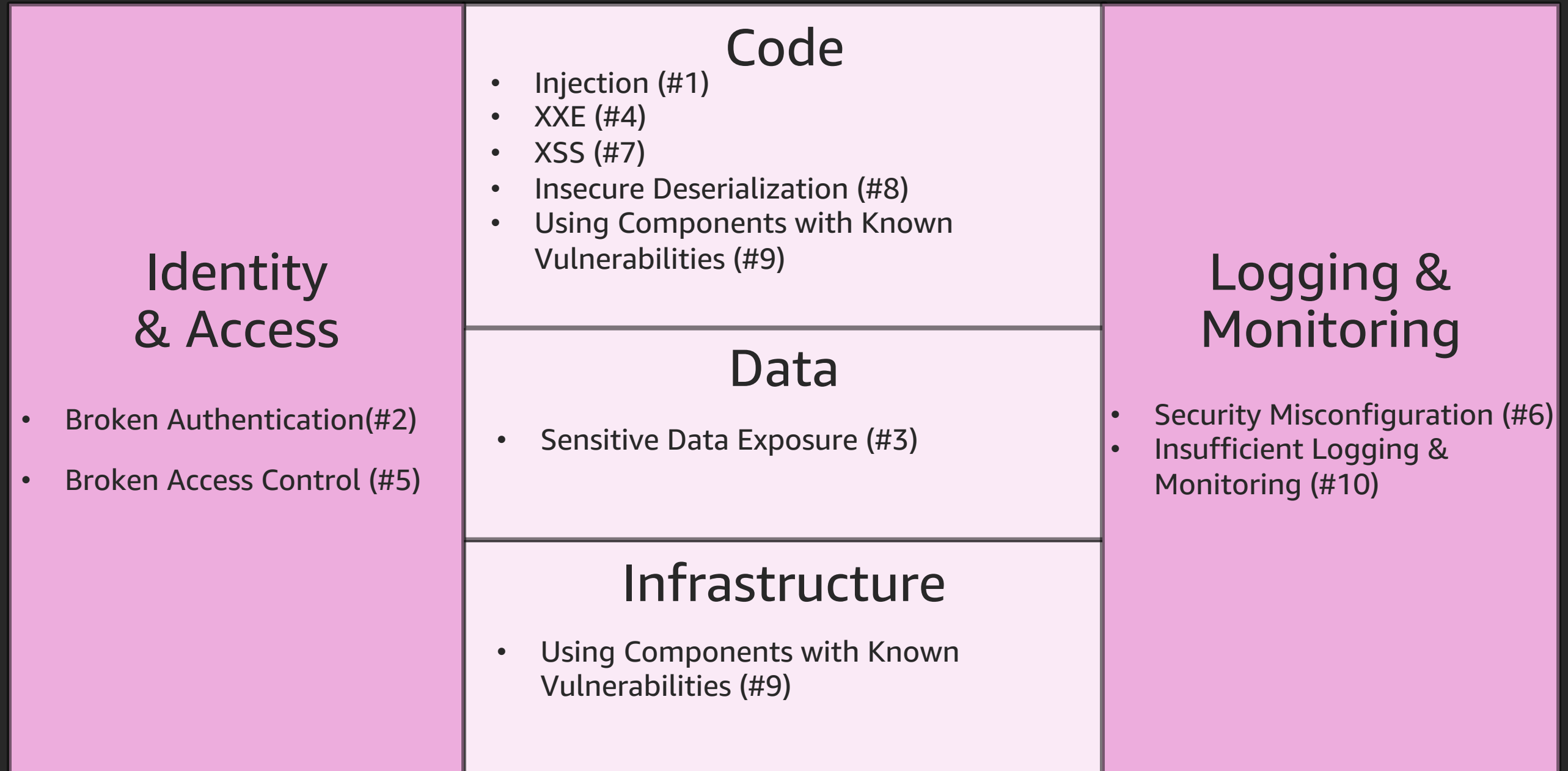
- **Exploitability**
- **Prevalence**
- **Detectability**
- **Technical impact**



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



Workshop scenario

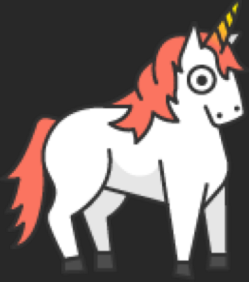
Scenario: Wild Rydes (www.wildrydes.com)



3rd party functionality– unicorn customization



3rd party API: Unicorn customization



List customization options and prices:

GET /capes



GET /glasses



GET /horns



GET /socks



3rd party API: Unicorn customization



Create and manage customizations

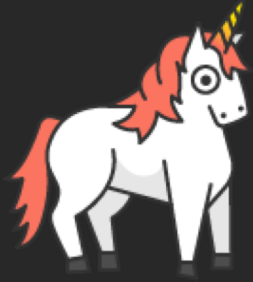
POST /customizations

GET /customizations

GET /customizations/{id}

DELETE /customizations/{id}

Admin API: register 3rd party partners



Register new partners

POST /partners

Workshop architecture – starting point



Deployed using SAM
(Serverless Application Model)



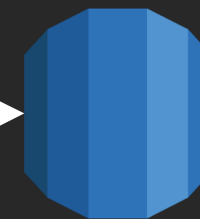
3rd party



Amazon API
Gateway



AWS
Lambda

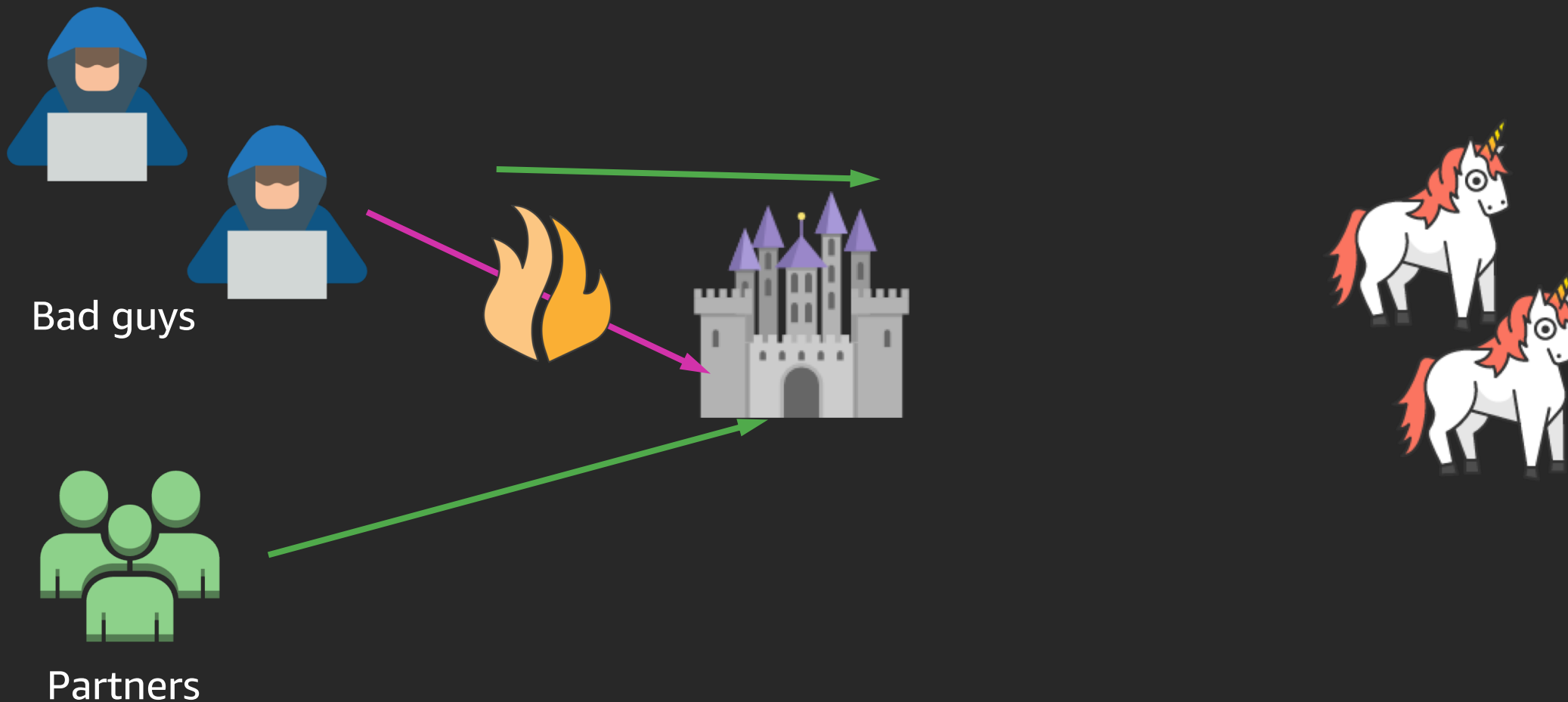


Amazon
RDS



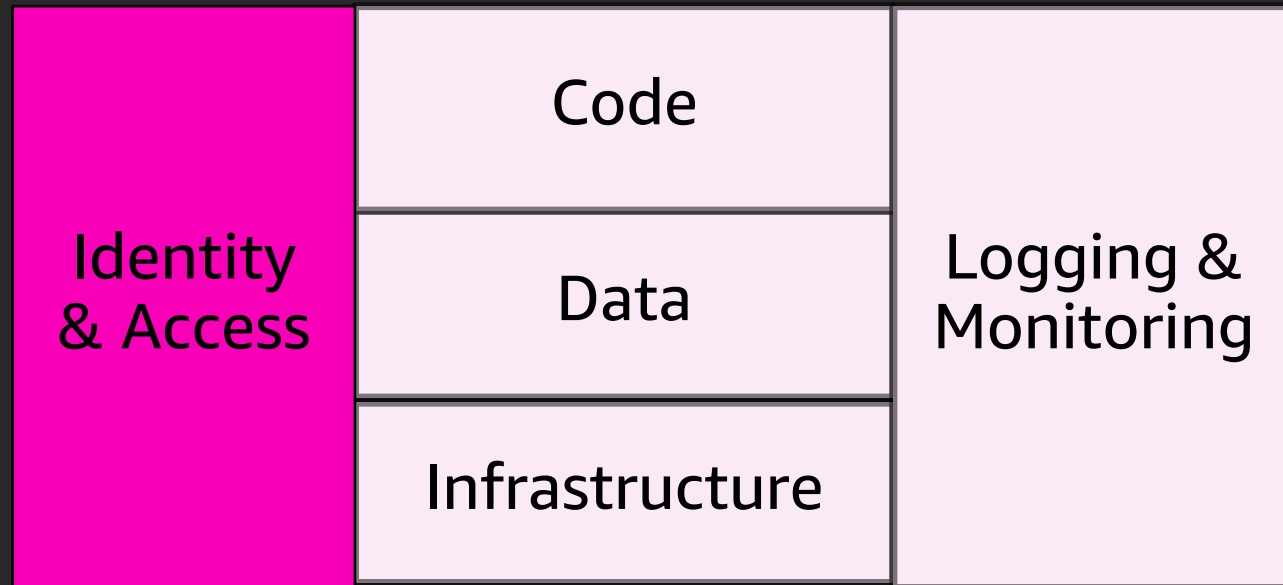
Not secure!

Your task: secure the application against attackers!



How to secure serverless applications

Identity and access management for serverless applications



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

Identity and access management for serverless applications

Authenticate & authorize end-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

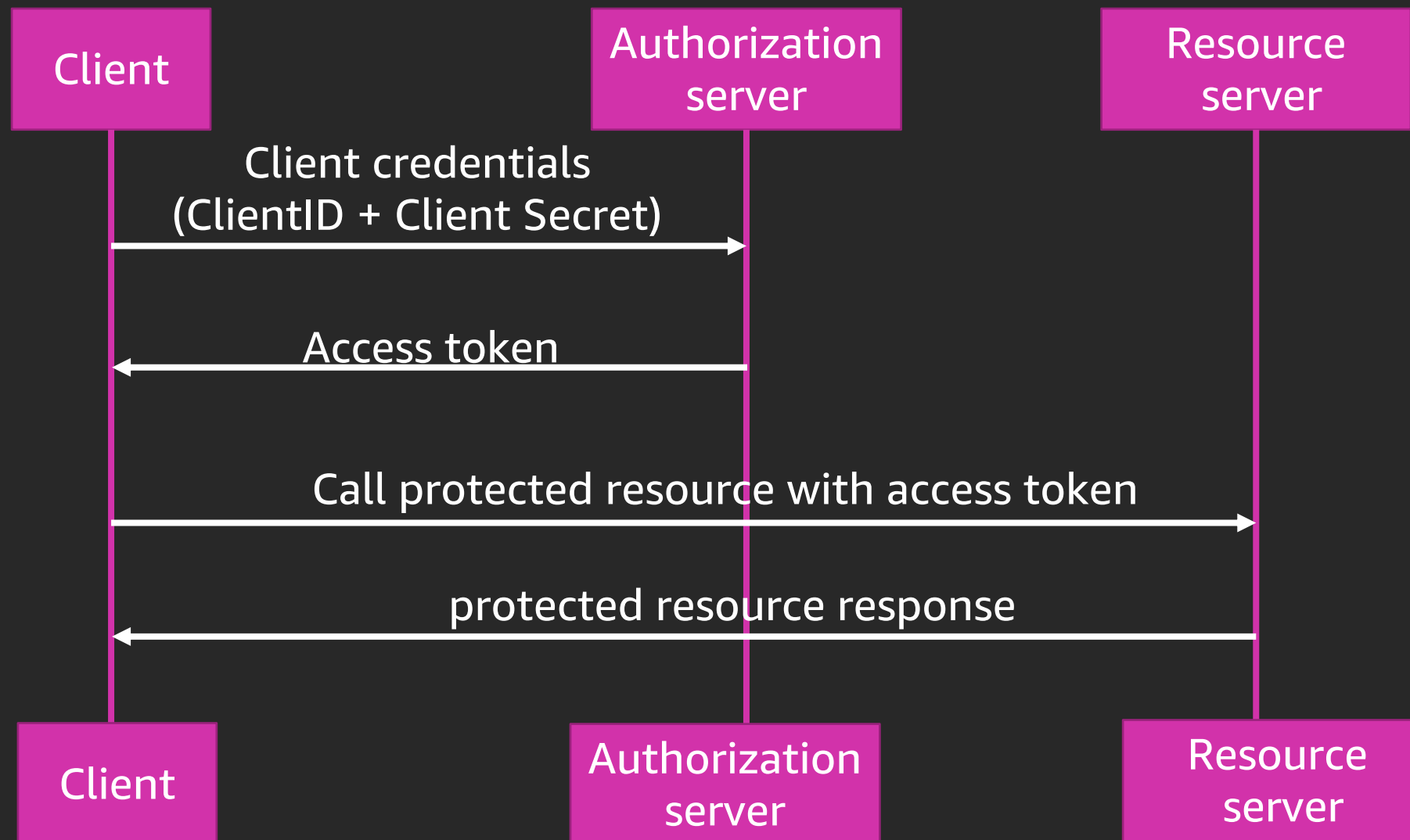
Access control between services



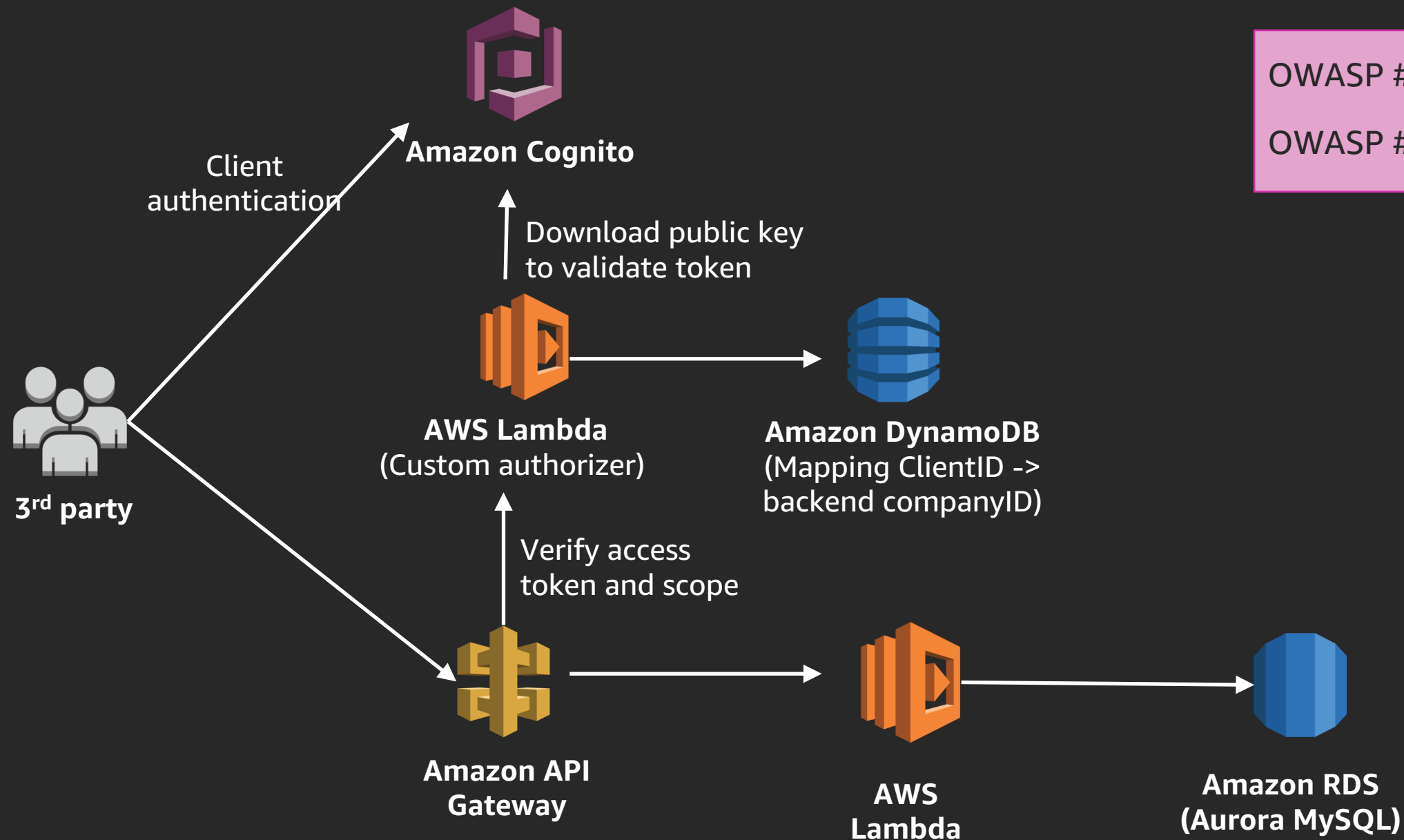
AWS Lambda:

- 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

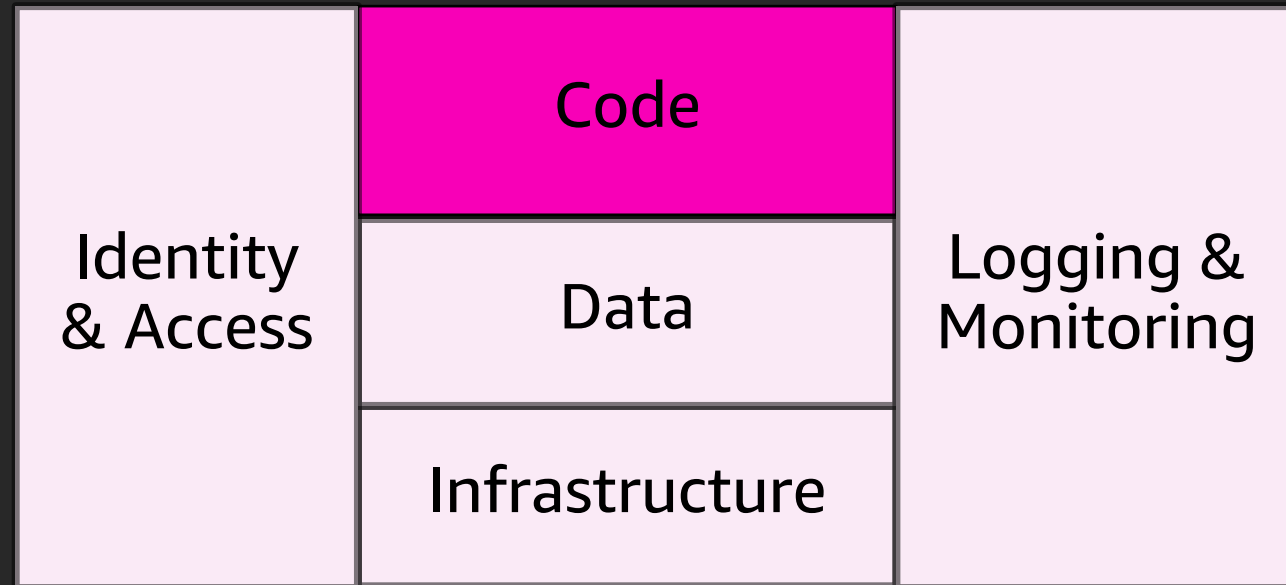


Amazon Aurora

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*



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



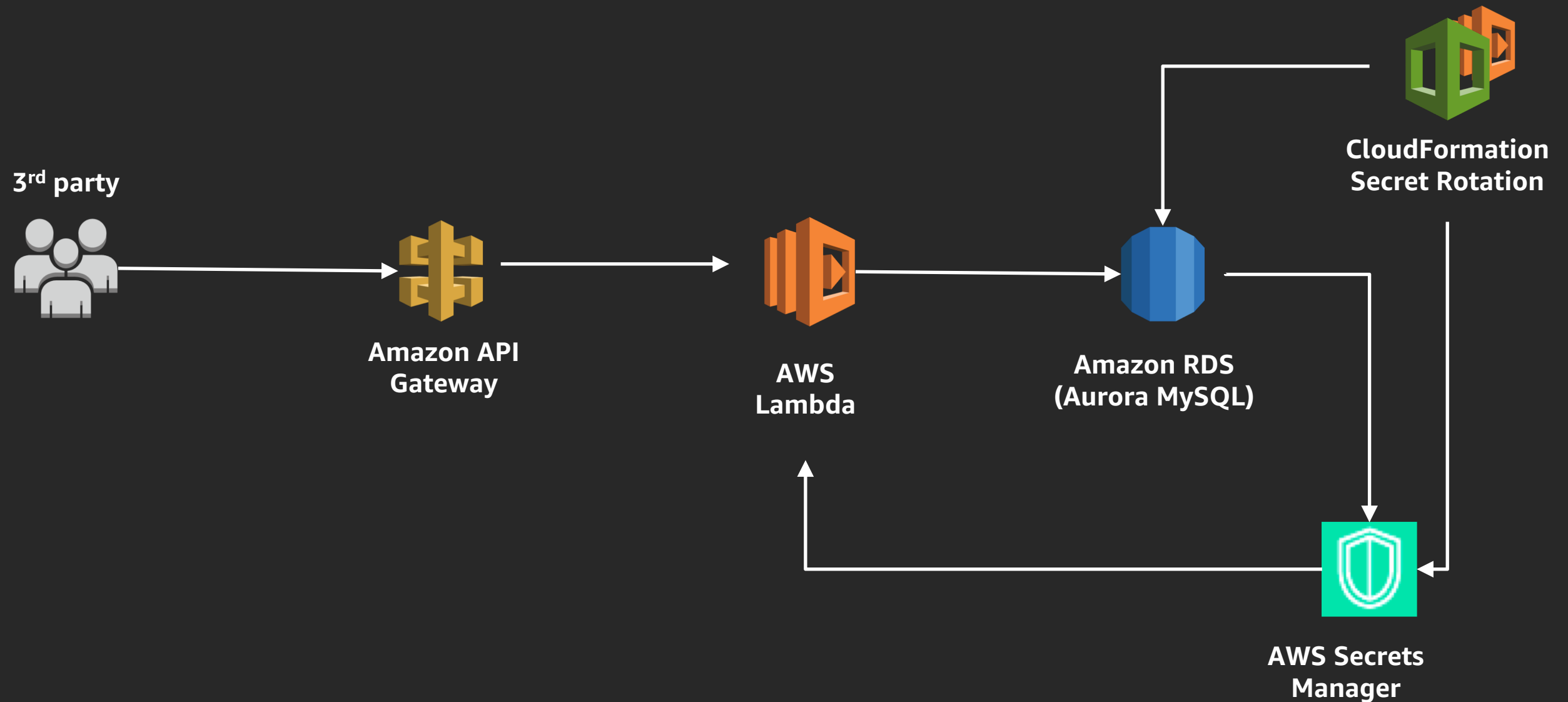
Systems Manager Parameter Store



AWS Lambda encrypted environment variables

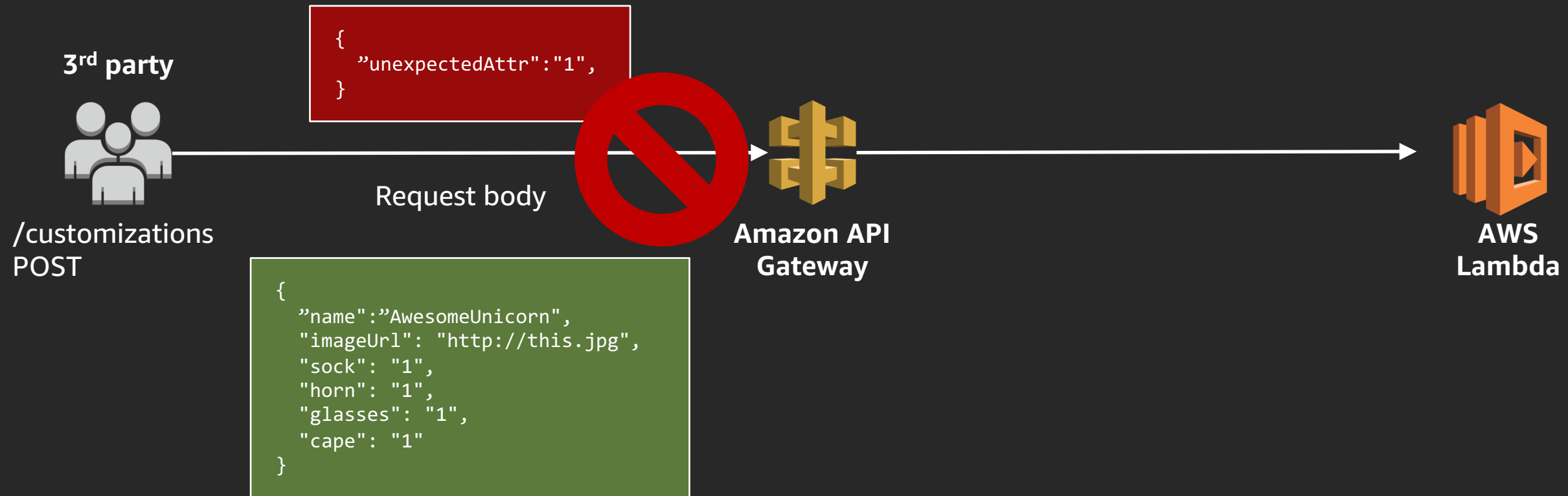
Module 2: Secret Manager

OWASP #3: Sensitive Data Exposure



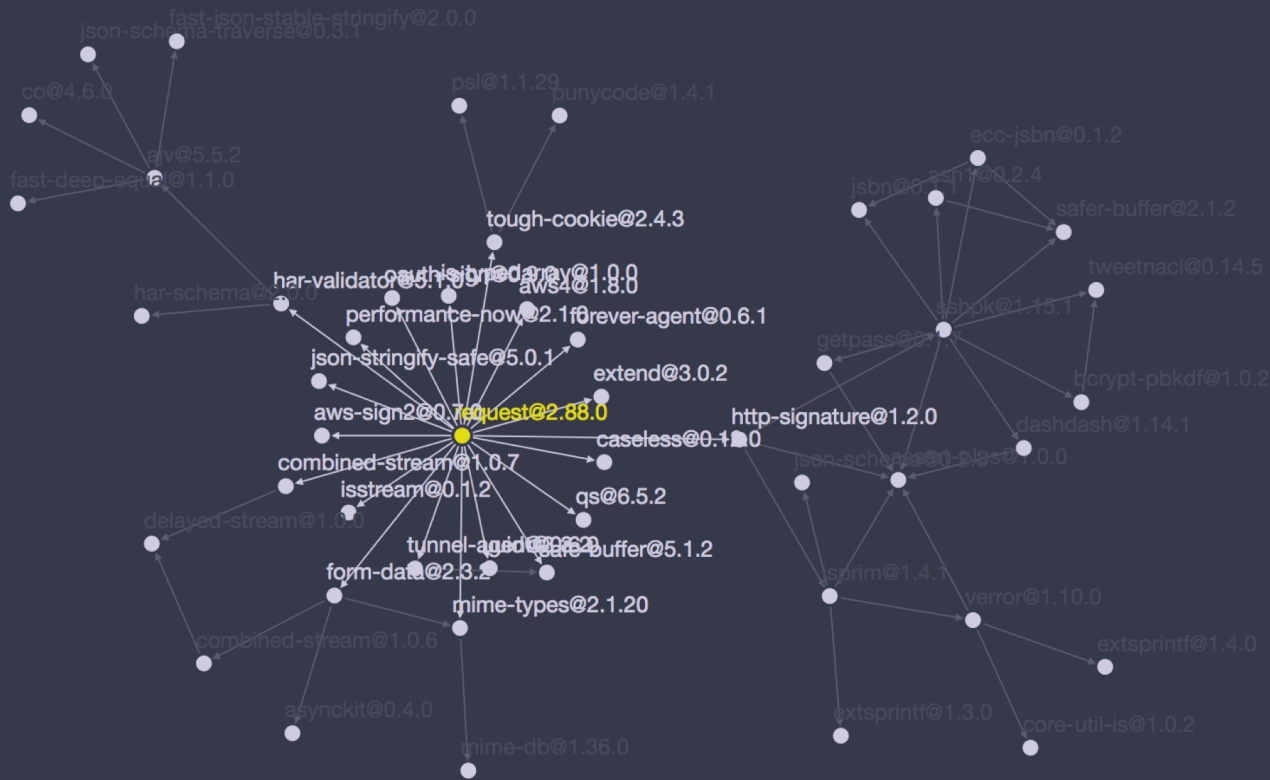
Module 3: Input Validation

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



Module 7: Dependency Vulnerability

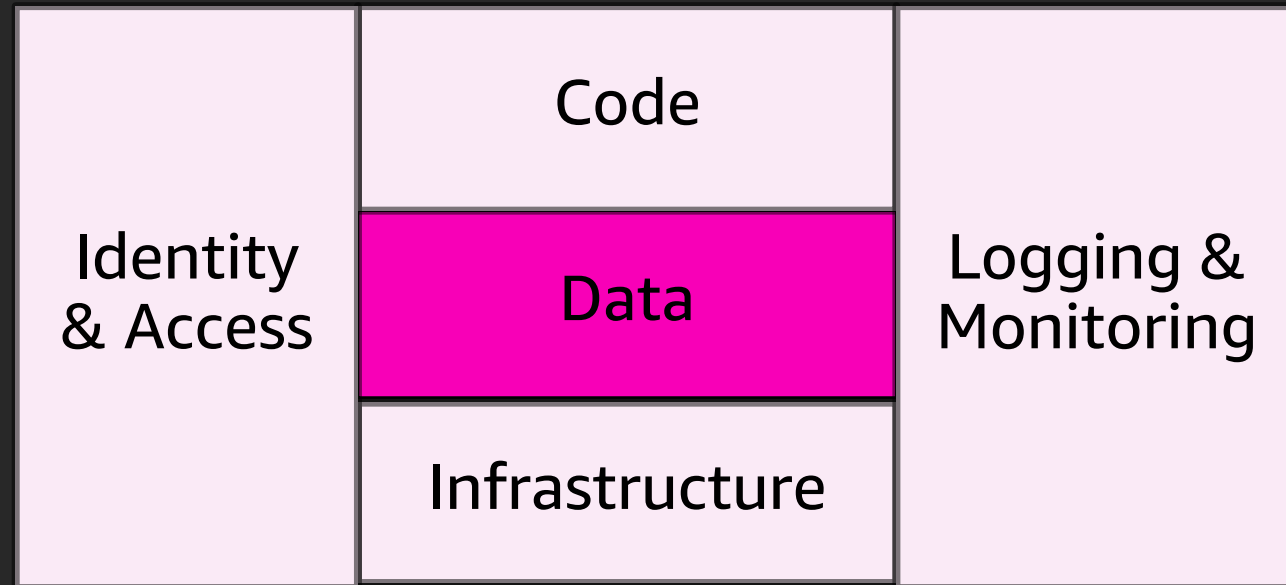
- OWASP #9: Using Components with Known Vulnerabilities



<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 platform takes care of:



Automatic replication of data across availability zones for high durability



Managed backups/encryption

Securing data for serverless applications

Data Classification



Data Flow



Data Tokenization

- DIY
- AWS Marketplace

Data Encryption at rest



AWS KMS:

- Server-side encryption
 - S3, DynamoDB, RDS
 - ...
- Client-side encryption

Data Encryption in transit



API Gateway : HTTPS only



Amazon Certificate Manager:

- Manage SSL certs for custom domains

Data backup/Replication



S3

- Versioning
- MFA delete
- Cross-region replication



DynamoDB

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



RDS

- Automated backups

Module 4: encryption in transit

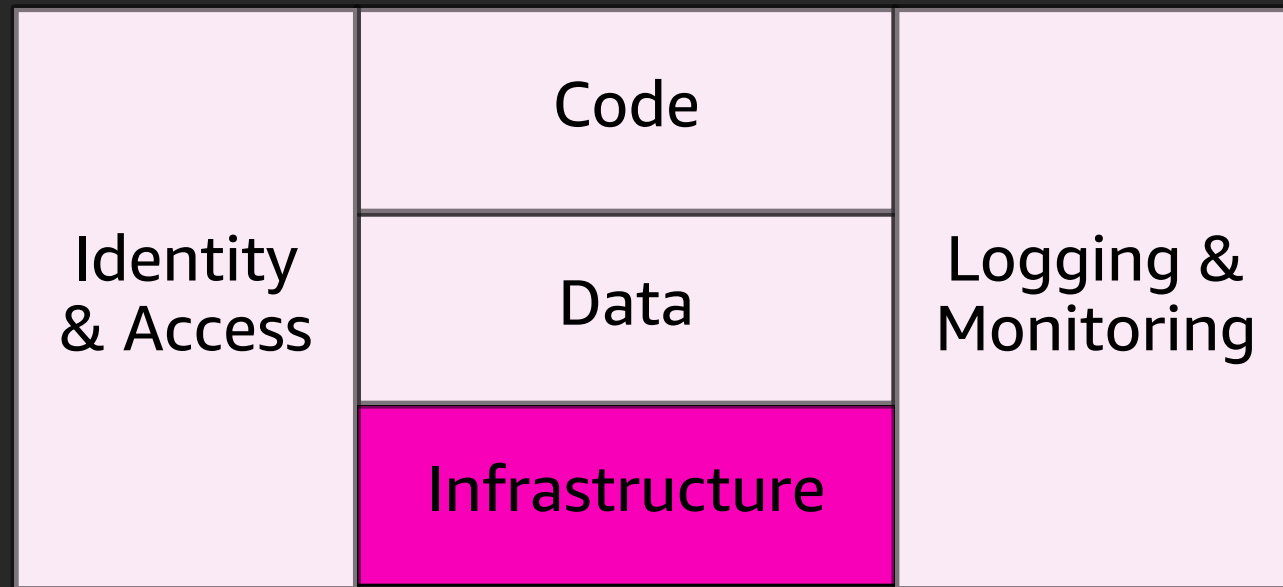
OWASP #3: Sensitive Data Exposure

```
{  
  host: "database.host.com",  
  user: "admin",  
  password: "xxxxxxxx",  
  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/ Rate limiting

 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

Network boundaries

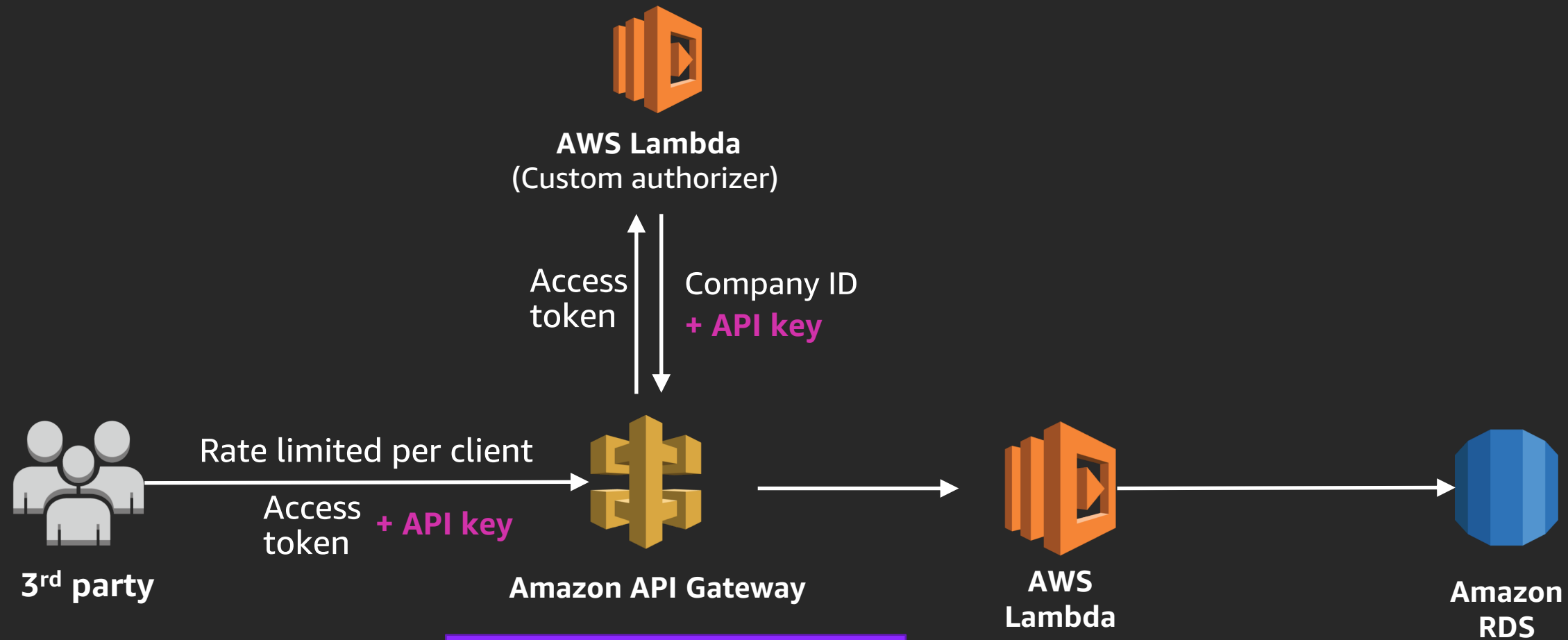
 API Gateway:

- Private VPC endpoints
- Resource policy

 AWS Lambda:

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






Module 5: Usage Plans



Basic usage plan

- 5 requests/s
- 1000 requests/month

Module 6: WAF

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



Logging & monitoring for serverless applications

Identity & Access	Code	Logging & Monitoring
	Data	
	Infrastructure	

- 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



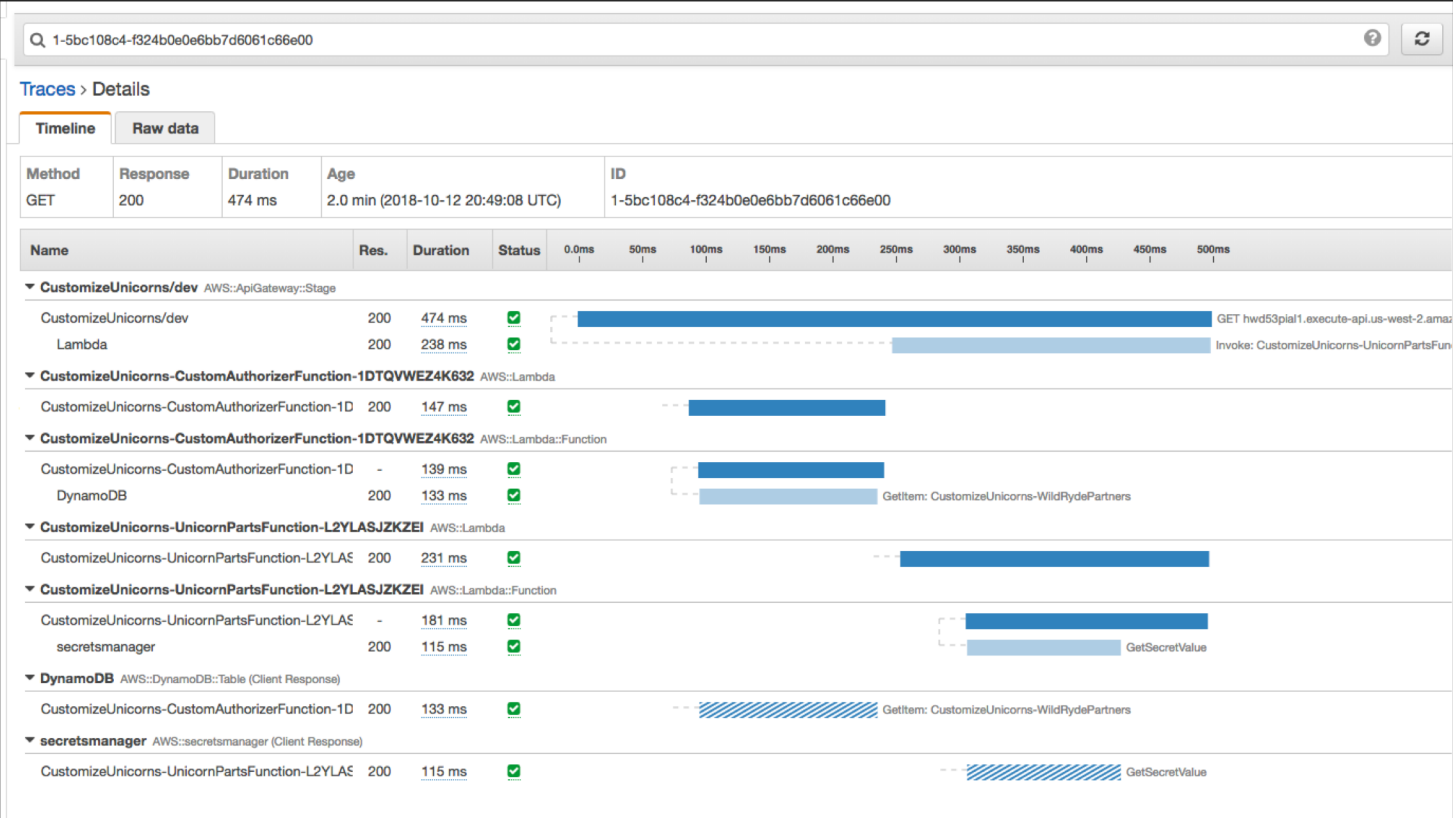
CloudWatch Events



AWS Budgets

Module 8: XRay

OWASP #10: Insufficient Logging & Monitoring

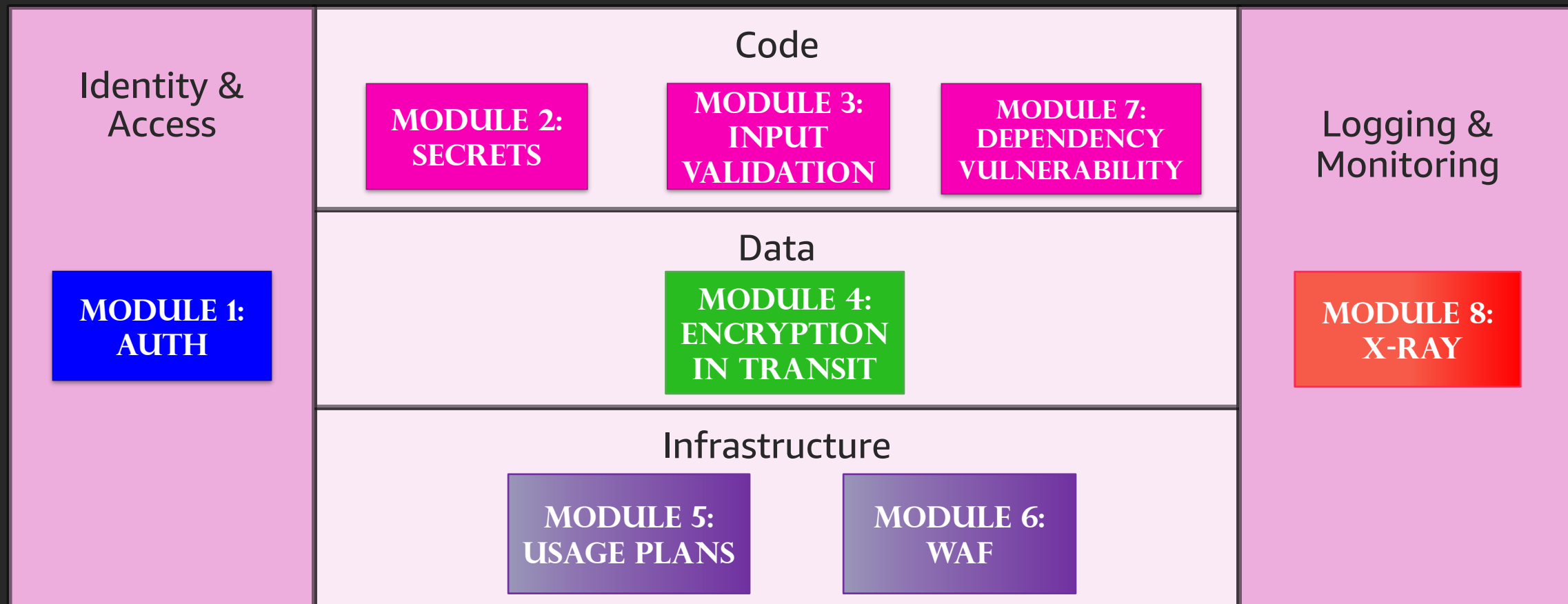


Workshop

Link to the workshop: <https://amzn.to/serverless-security>

Module 0 mandatory

Module 1-8: Pick your own battle!



Thank you!