Securing AWS Microservices in a Distributed Environment

CASSANDRA YOUNG AKA MUTEKI

@MUTEKI_RTW || CMORGANYOUNG@GMAIL.COM

Securing AWS [Serverless] Microservices [Applications] in a Distributed [Development] Environment

…that’s too many words

Securing AWS [Serverless] Microservices [Applications] in a Distributed [Development] Environment

What does ‘Microservices’ mean?

”The microservices architecture is defined as collection of smaller services arranged together in a workflow, often utilizing a serverless platform. Microservices architecture … [connects] these services or functions in a loosely coupled manner, often hosted separately and communicating via REST APIs or other lightweight protocols.”

- Microservices Architecture is a design pattern
- Collection of decoupled components
- Development and upgrade focus shifted to component level

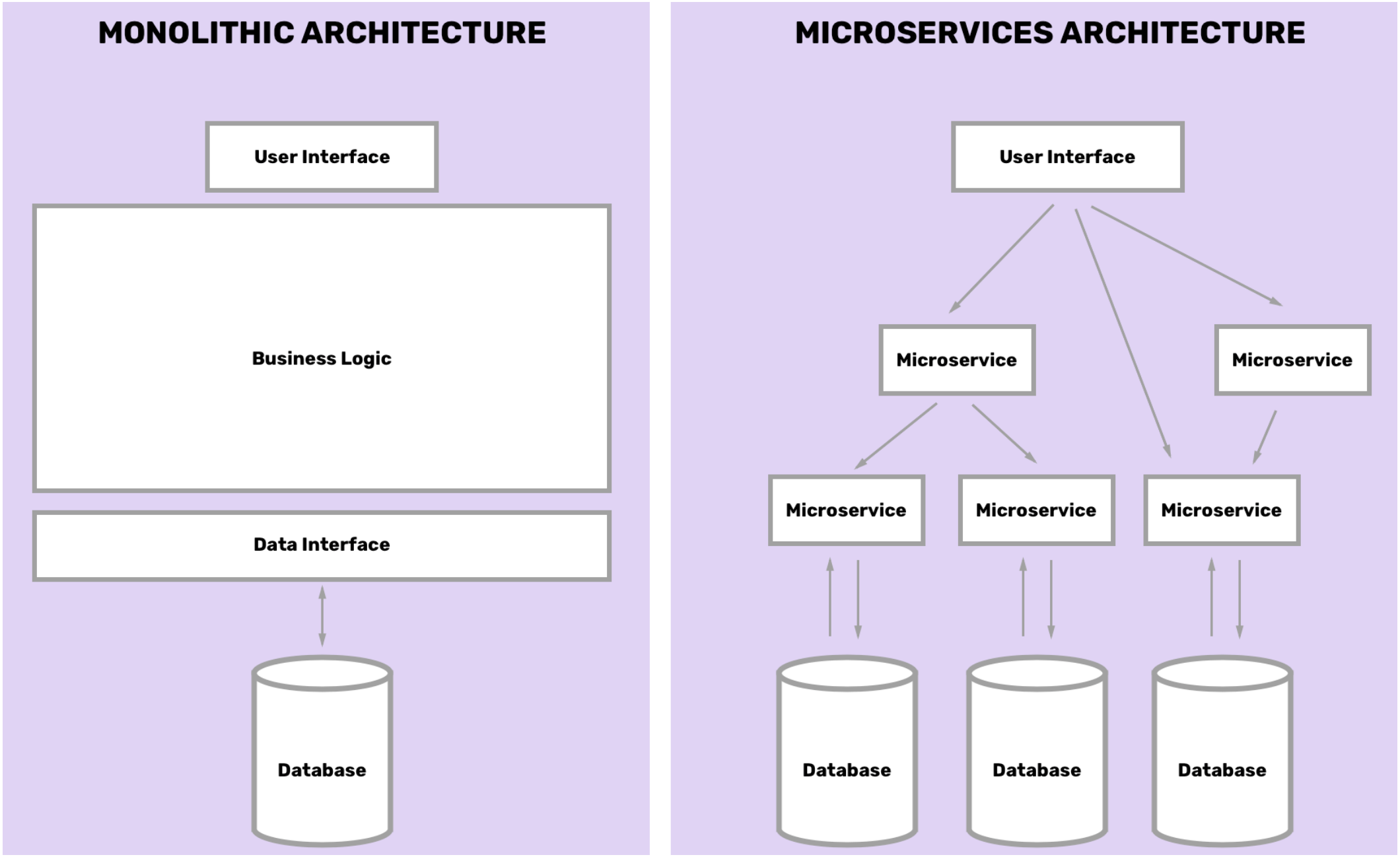


Image Credit: algorithmia.com/blog/introduction-to-serverless-microservices

Securing AWS [Serverless] Microservices [Applications] in a Distributed [Development] Environment

What is Serverless?

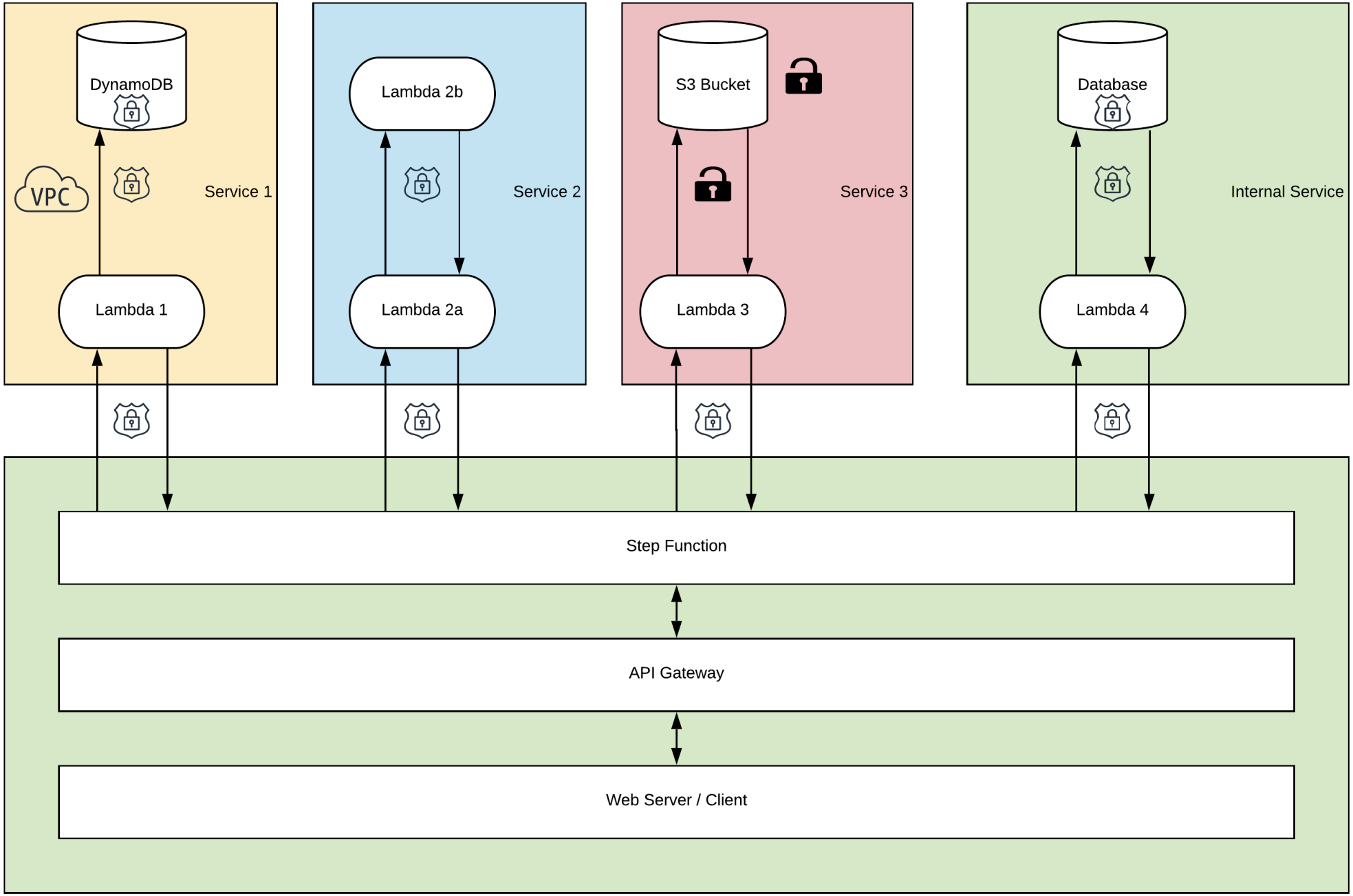
- Execution model which abstracts away infrastructure
- Code-focused, obfuscating OS & dependencies
- Simplified development process
- Event-driven, primarily accessed via APIs

- AWS Lambda is Serverless Function-as-a-Service

Securing AWS [Serverless] Microservices [Applications] in a Distributed [Development] Environment

What does Distributed Development Environment mean?

- Functions can be outsourced
- Why build it all in-house when you can use someone else's?
- APIs simplify this process


DynamoDB
VPC
Service 1
Lambda 1
Lambda 2b
Service 2
Lambda 2a
S3 Bucket
Service 3
Lambda 3
Database
Internal Service
Lambda 4
Step Function
API Gateway
Web Server / Client

Exposure += Risks

Vulnerabilities: AWS Lambda Runtime

- Lambda architecture:
  - Code executed in sandboxed container on VM
  - 512mb of read/write storage in /tmp
  - Warm start enables persistence of execution environment
  - Event data passed in upon Lambda execution
- String injection exposes underlying runtime environment

- Want more? Watch this talk: “Gone in 60 Milliseconds” - Rich Jones

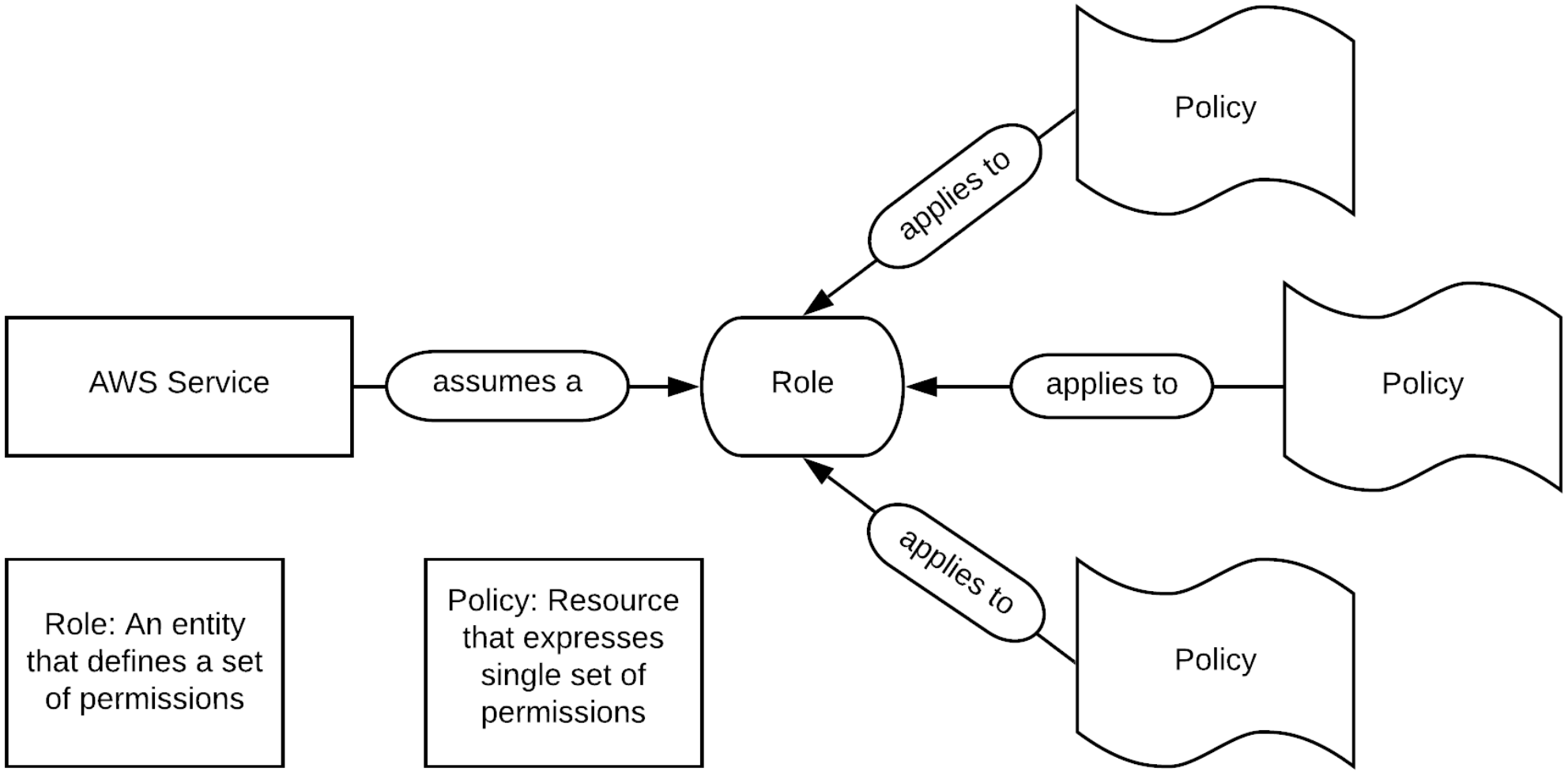
Data Exposure: In-Transit & At Rest

IN-TRANSIT

- Most AWS services are HTTPS only
  - Lambda, RDS, API Gateway, Step Functions, KMS
- Notable exceptions:
  - DynamoDB, EC2, S3
- Detailed in AWS General Reference
  -> Service Endpoints and Quotas

AT REST

- Some services encrypted by default:
  - Lambda (RAM), DynamoDB
- Most services provide options to use default or CMK (Customer Managed Key) via KMS



IAM Roles & Policies

Lenient IAM Policy

Policy granting a Lambda full access to a DynamoDB table:

```
{
    "Version": "2012-10-17",
    "Statement": [
    {
        "Effect": "Allow",
        "Action": "dynamodb:*",
        "Resource": "arn:aws:dynamodb:us-east-2:663910366299:table/Payments"
    }
    ]
}
```

Adjusted IAM Policy

Policy granting a Lambda tightly scoped access to a DynamoDB table:

```
{
    "Version": "2012-10-17",
    "Statement": [
    {
        "Effect": "Allow",
        "Action": [
            "dynamodb:PutItem",
            "dynamodb:UpdateItem"
        ]
        "Resource": "arn:aws:dynamodb:us-east-2:663910366299:table/Payments"
    }
    ]
}
```

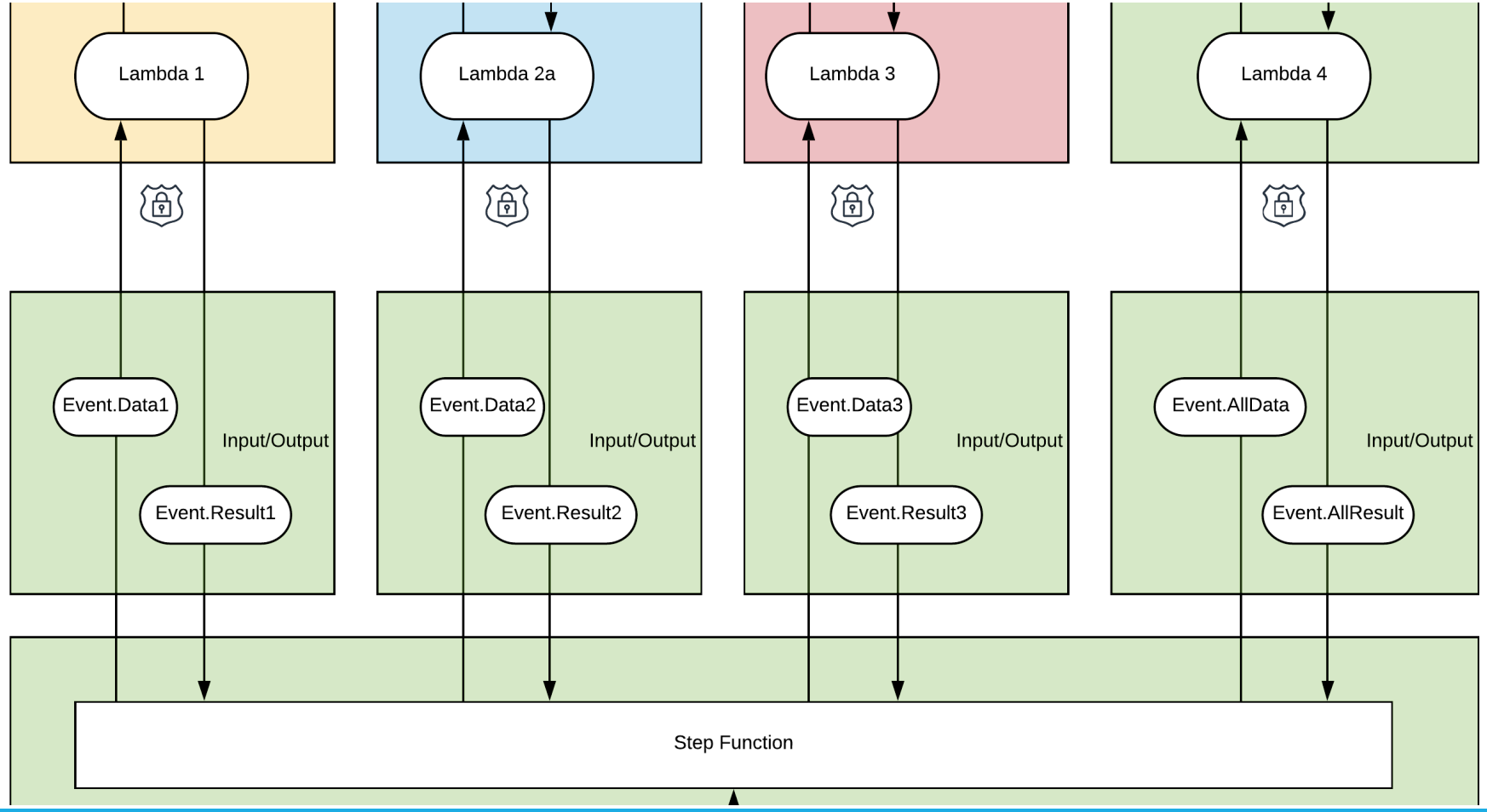
Privilege Escalation: What if Resource : * ?

Policy granting a Lambda full access to a DynamoDB table:

```
{
    "Version": "2012-10-17",
    "Statement": [
    {
        "Effect": "Allow",
        "Action": "dynamodb:*",
        "Resource": "arn:aws:dynamodb:us-east-2:663910366299:table/Payments"
    }
    ]
}
```

Considerations for Distributed Environments

Step Functions: Restricted Event Data



Cross-Account Encryption

- KMS (Key Management Service) allows cross-account key access
- Using scoped policies, can limit 3rd party account to encrypt-only, decrypt-only, etc.
- Example:
  - 3rd party stores data from multiple parties
  - encrypt using that party's KMS encryption key
  - if database is compromised from within 3rd party, data still encrypted

Notes on Logging & Monitoring

- CloudWatch captures AWS Lambda execution information
  - Can add logging statements to Lambda code
- Configure CloudWatch events with custom Rules
- CloudTrail audit logs
  - Track use of roles, KMS keys, AWS account activity
- AWS X-Ray: Service designed for monitoring microservices applications

Summary

- Code defensively
- The Principle of Least Privilege applies to Roles & Policies
- Encrypt ALL THE THINGS
- Vet 3rd parties
- Trust no one, Mr. Mulder.

Remember: The cloud is just someone else's computer!

Questions?

Comments?

Rotten tomatoes?

"The greatest mistake you can make in life is to continually be afraid you will make one."-Elbert Hubbard