



G O T O

Amsterdam 2018

# Serverless Architectural Patterns

**Boaz Ziniman, Technical Evangelist – Amazon Web Service**



@ziniman



boaz.ziniman.aws



*Please*

**Ask questions  
through the app**



*Rate Session*

*Thank you!*



GOTO  
Amsterdam

# Serverless Beyond Lambda

The AWS logo, consisting of the letters "aws" in white lowercase font with a yellow curved arrow underneath, is positioned in the top left corner of the slide.

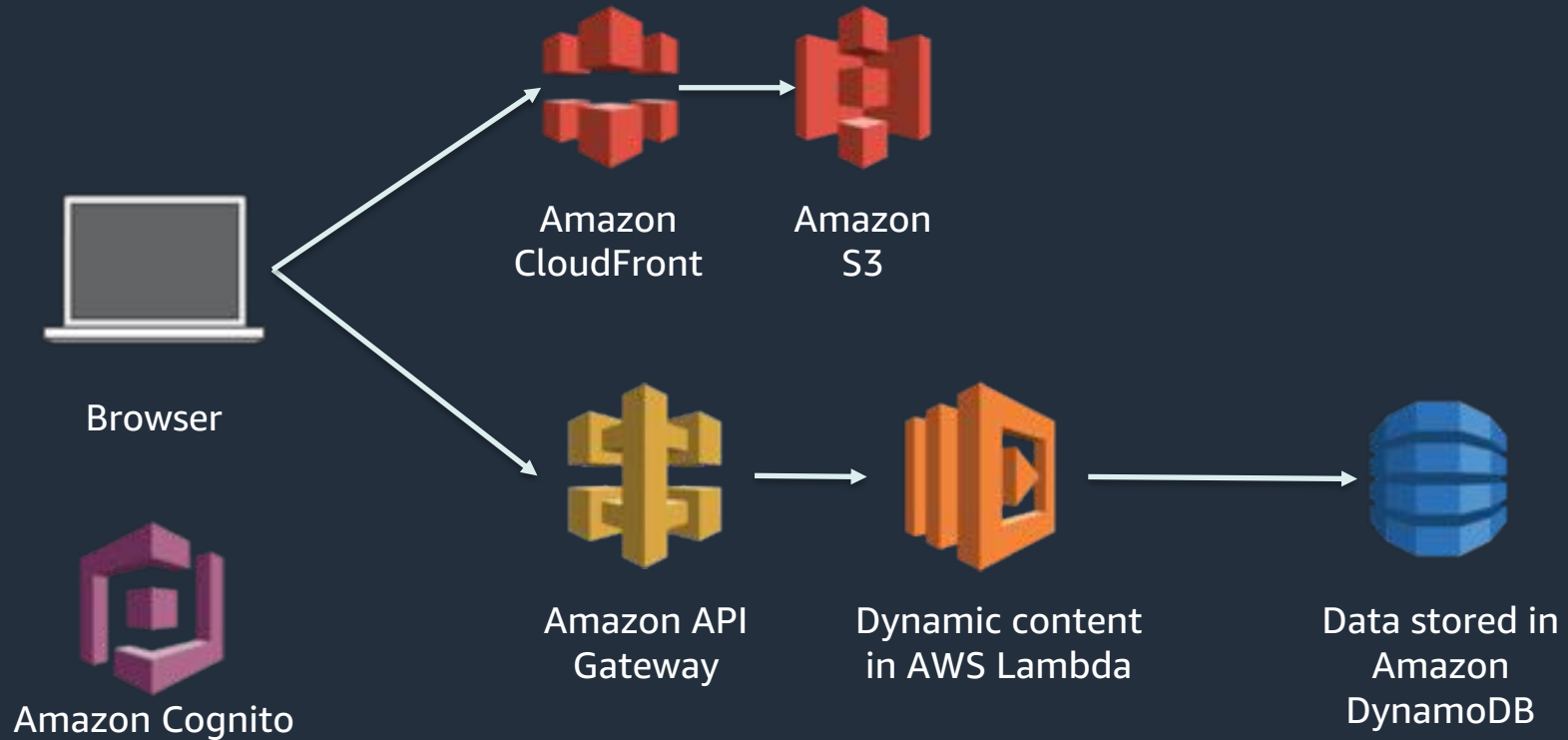
aws

GOTO

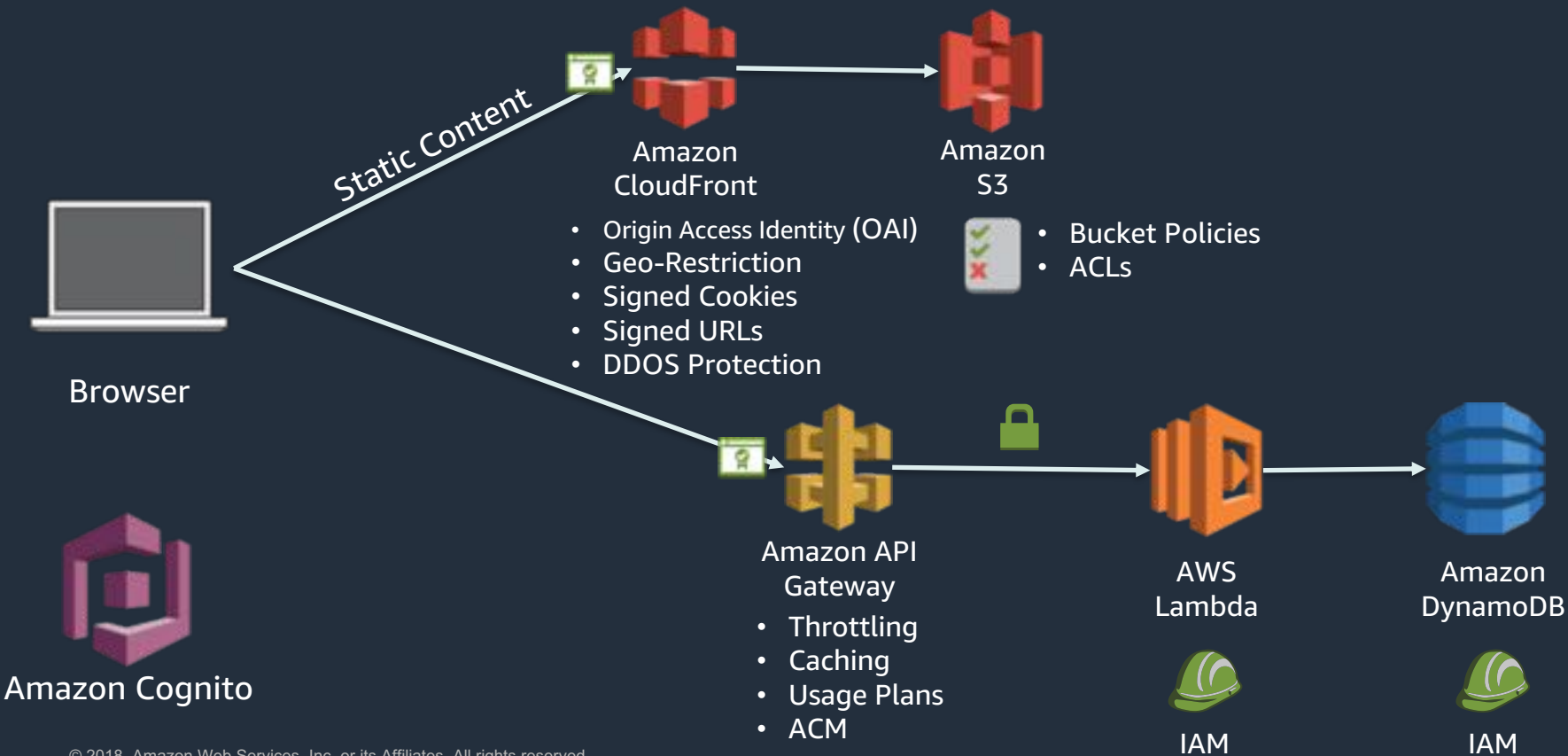
Amsterdam

# Pattern 1: Web App/Microservice/API

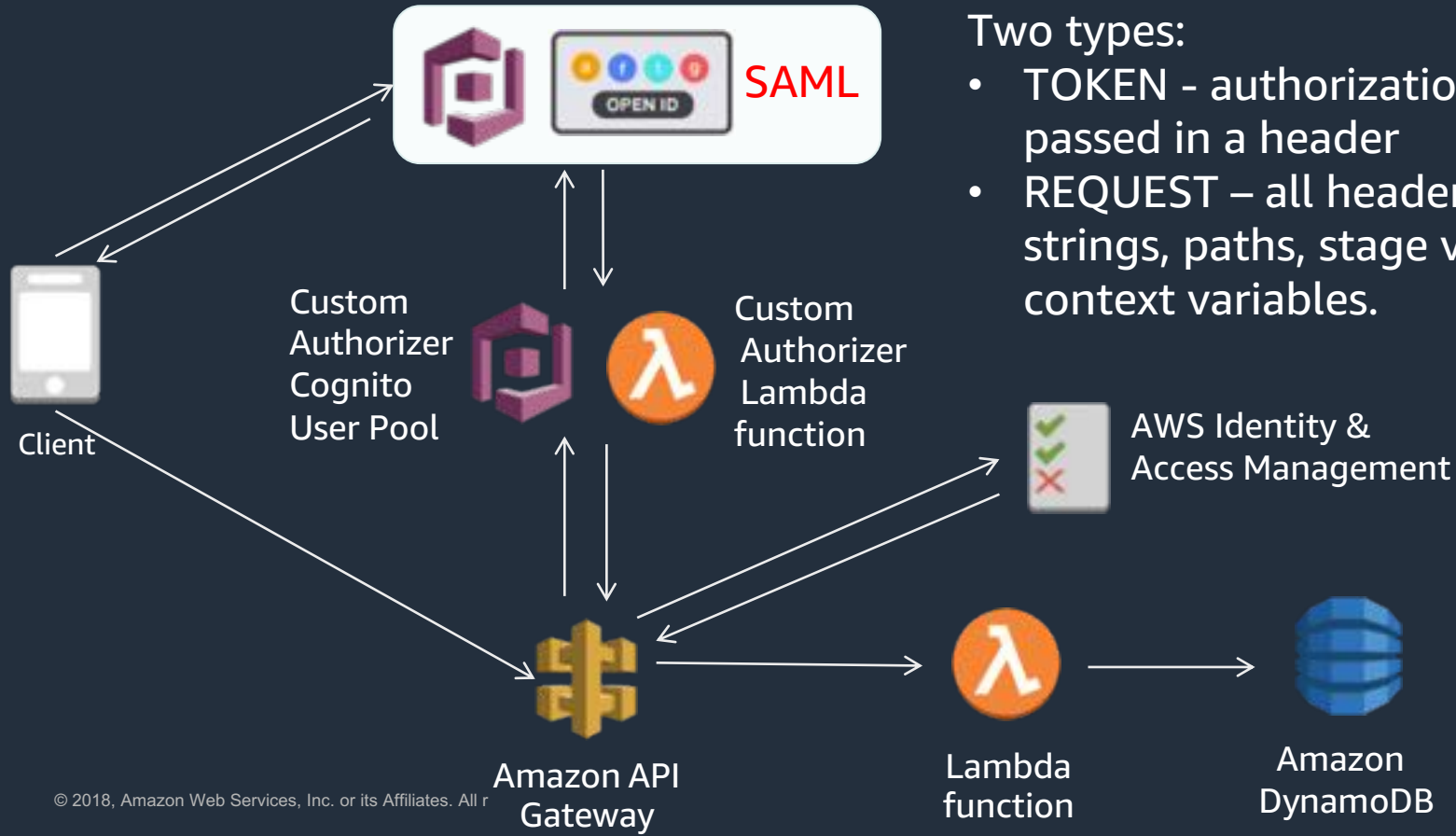
# Web application



# Serverless web app security



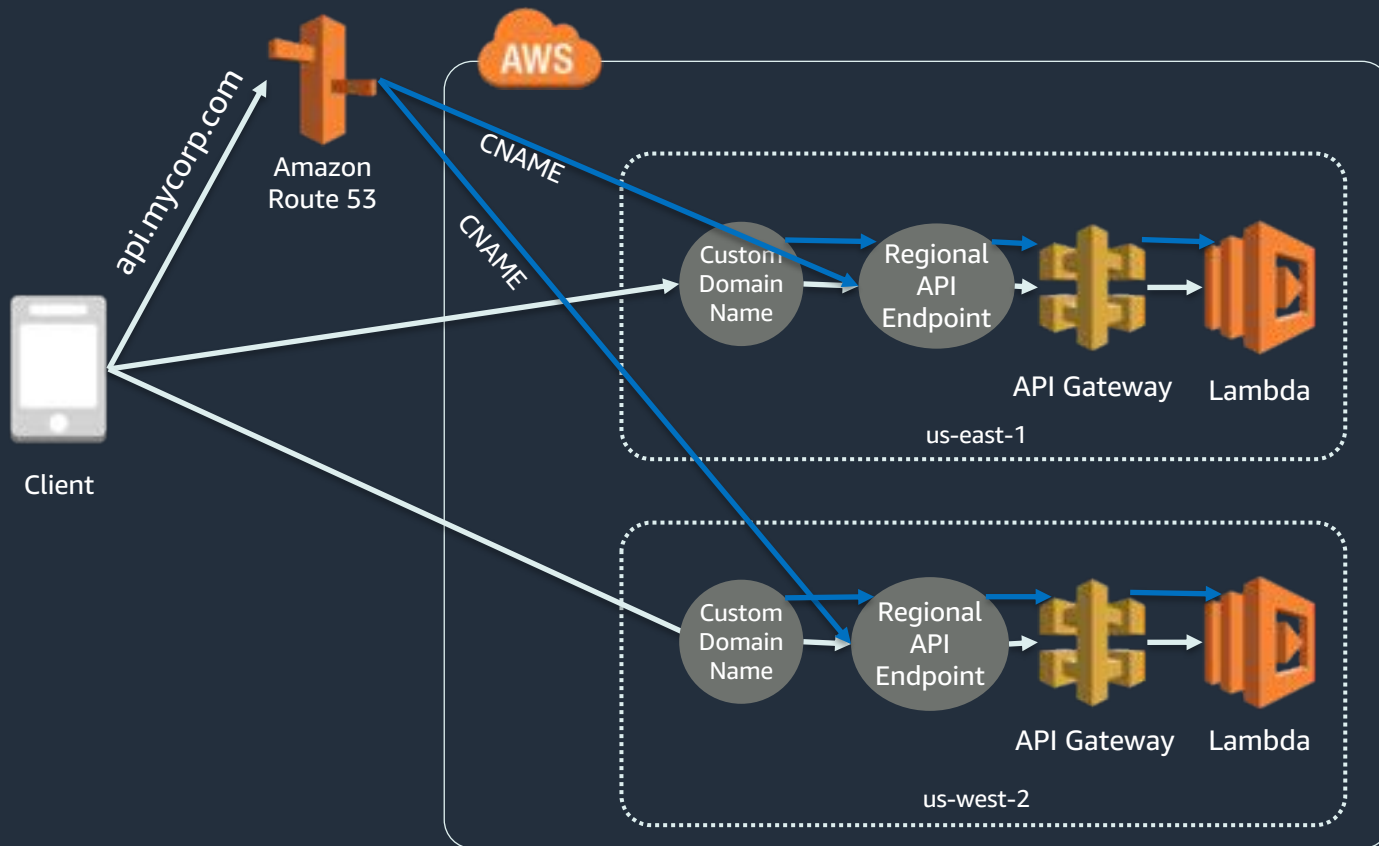
# Custom Authorizers



Two types:

- **TOKEN** - authorization token passed in a header
- **REQUEST** – all headers, query strings, paths, stage variables or context variables.

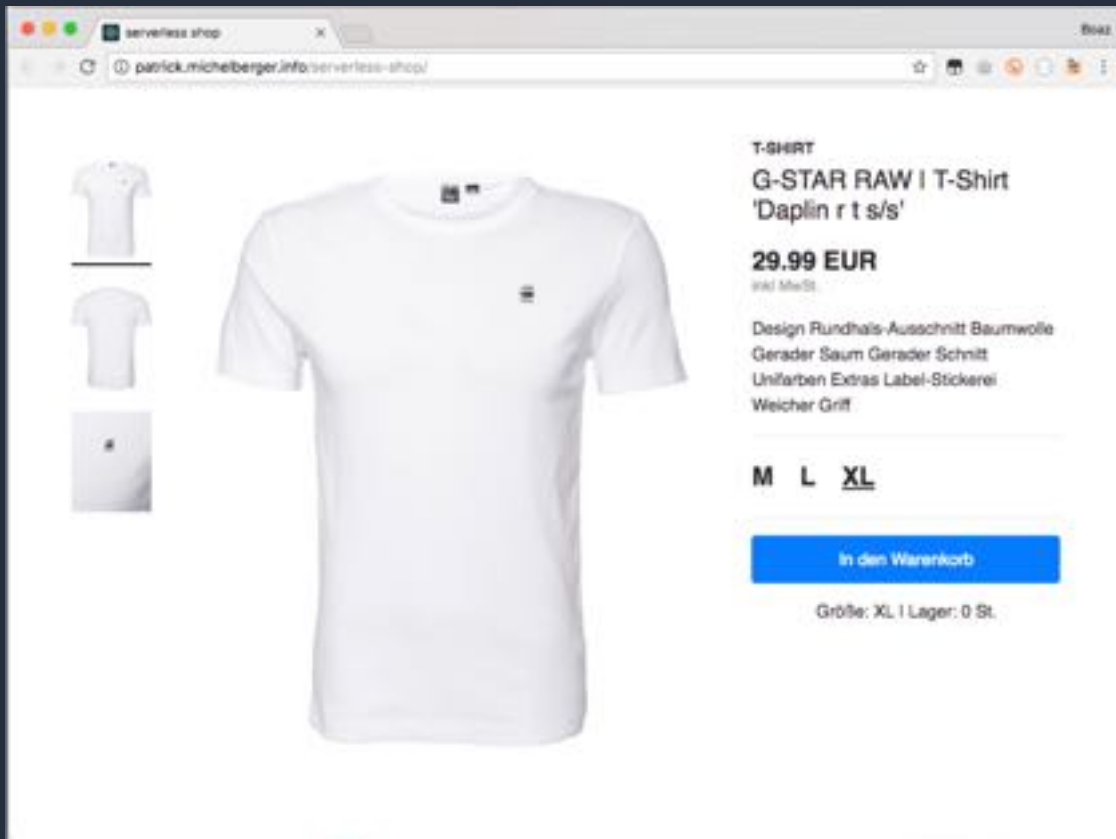
# Multi-Region with API Gateway





The AWS logo, consisting of the letters "aws" in white on a dark blue square background, with a yellow curved arrow underneath.

GOTO  
Amsterdam



<http://bit.ly/ServerlessShop>

 <https://github.com/patrick-michelberger/serverless-shop>



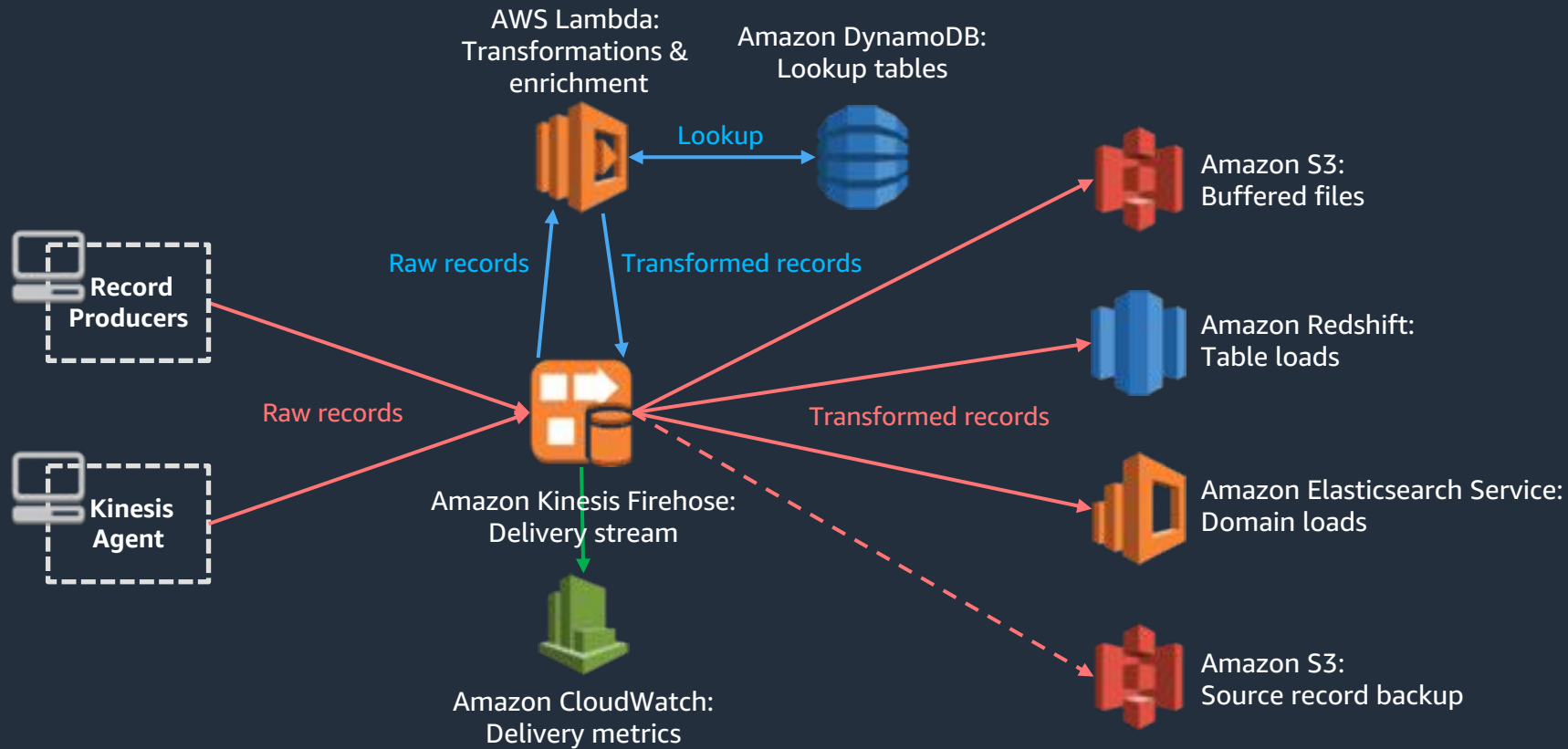
GOTO  
Amsterdam

# Pattern 2: Stream Processing

# Stream processing characteristics

- High ingest rate
- Near real-time processing (low latency from ingest to process)
- Spiky traffic (lots of devices with intermittent network connections)
- Message durability
- Message ordering

# Streaming data ingestion



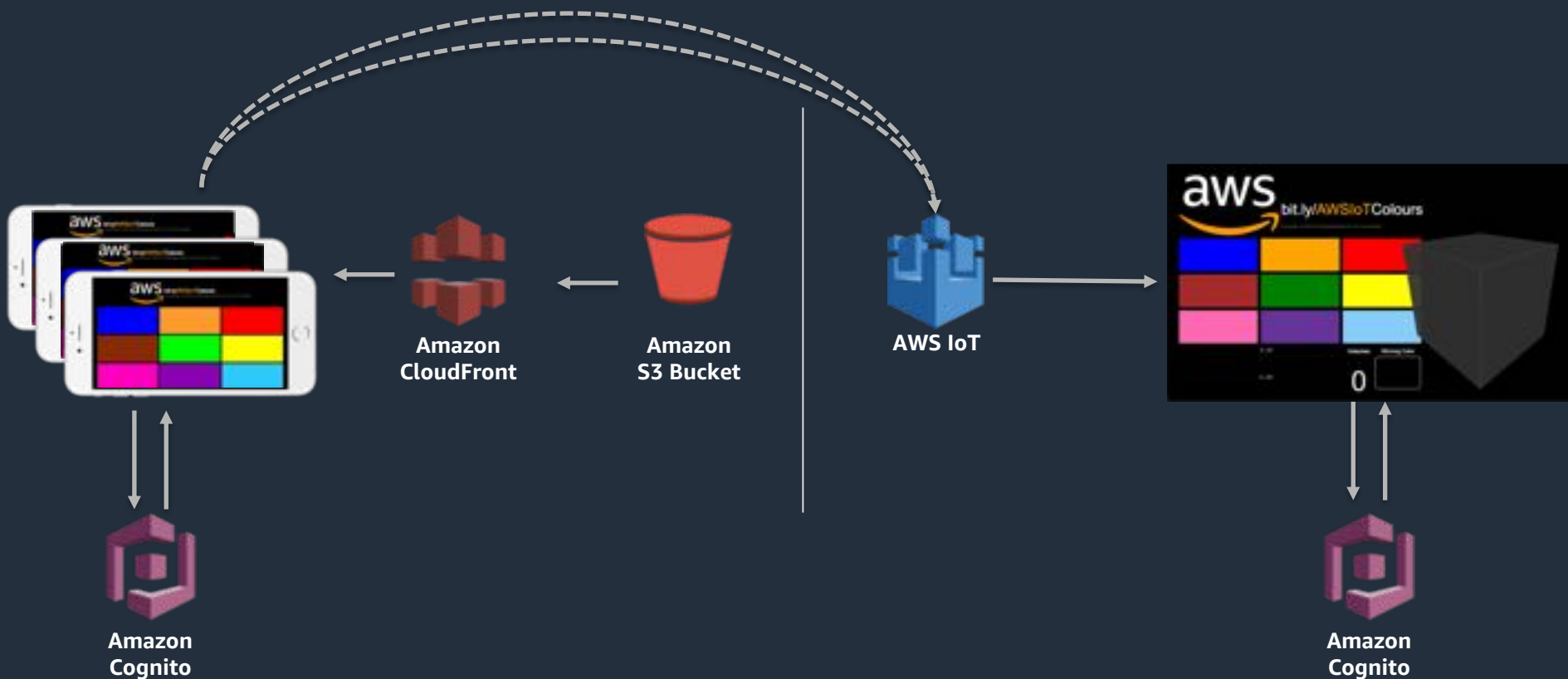
# Best practices

- Tune Firehose **buffer size** and **buffer interval**
  - Larger objects = fewer Lambda invocations, fewer S3 PUTs
- Enable **compression** to reduce storage costs
- Enable **Source Record Backup** for transformations
  - Recover from transformation errors
- Follow Amazon Redshift Best Practices for Loading Data
  - How to handle time series, sorted data

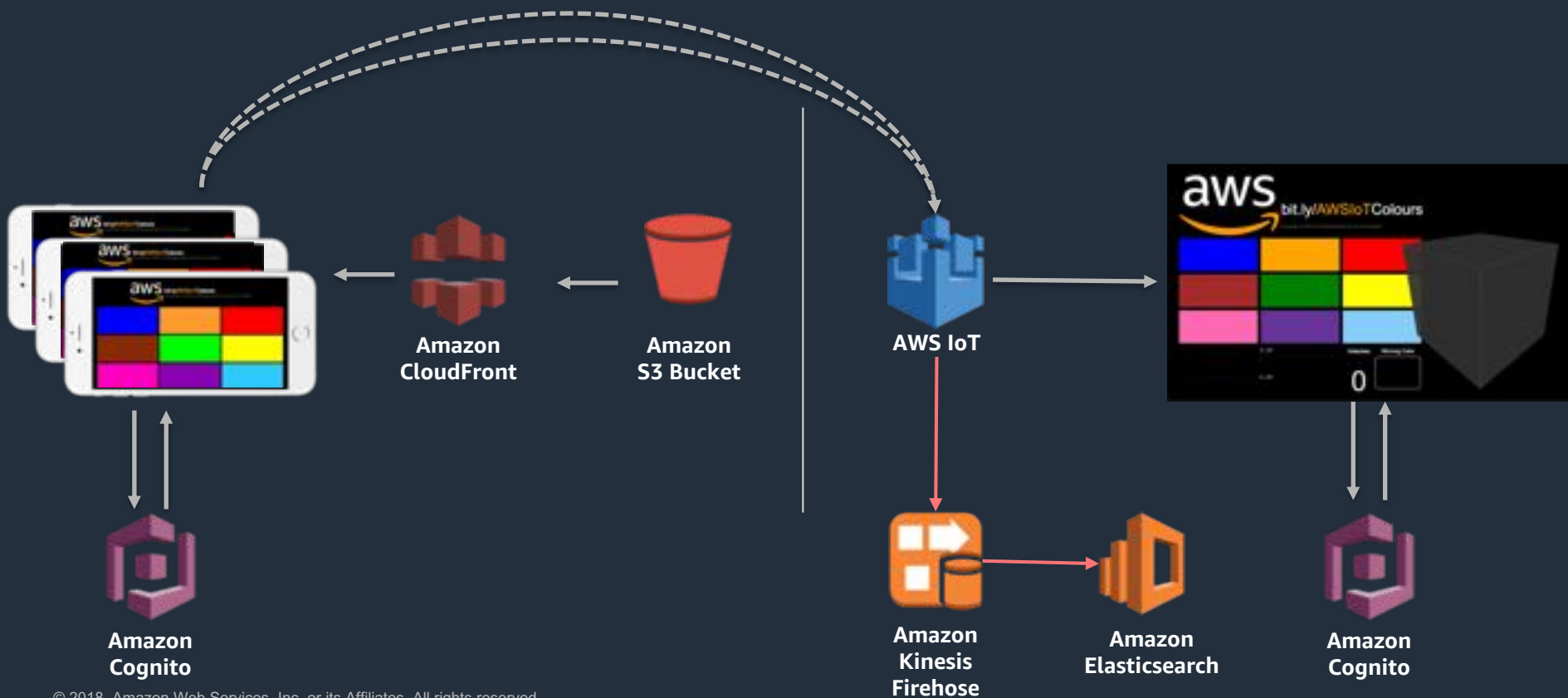


[bit.ly/AWSIoTColours](https://bit.ly/AWSIoTColours)

# Colour cube voting

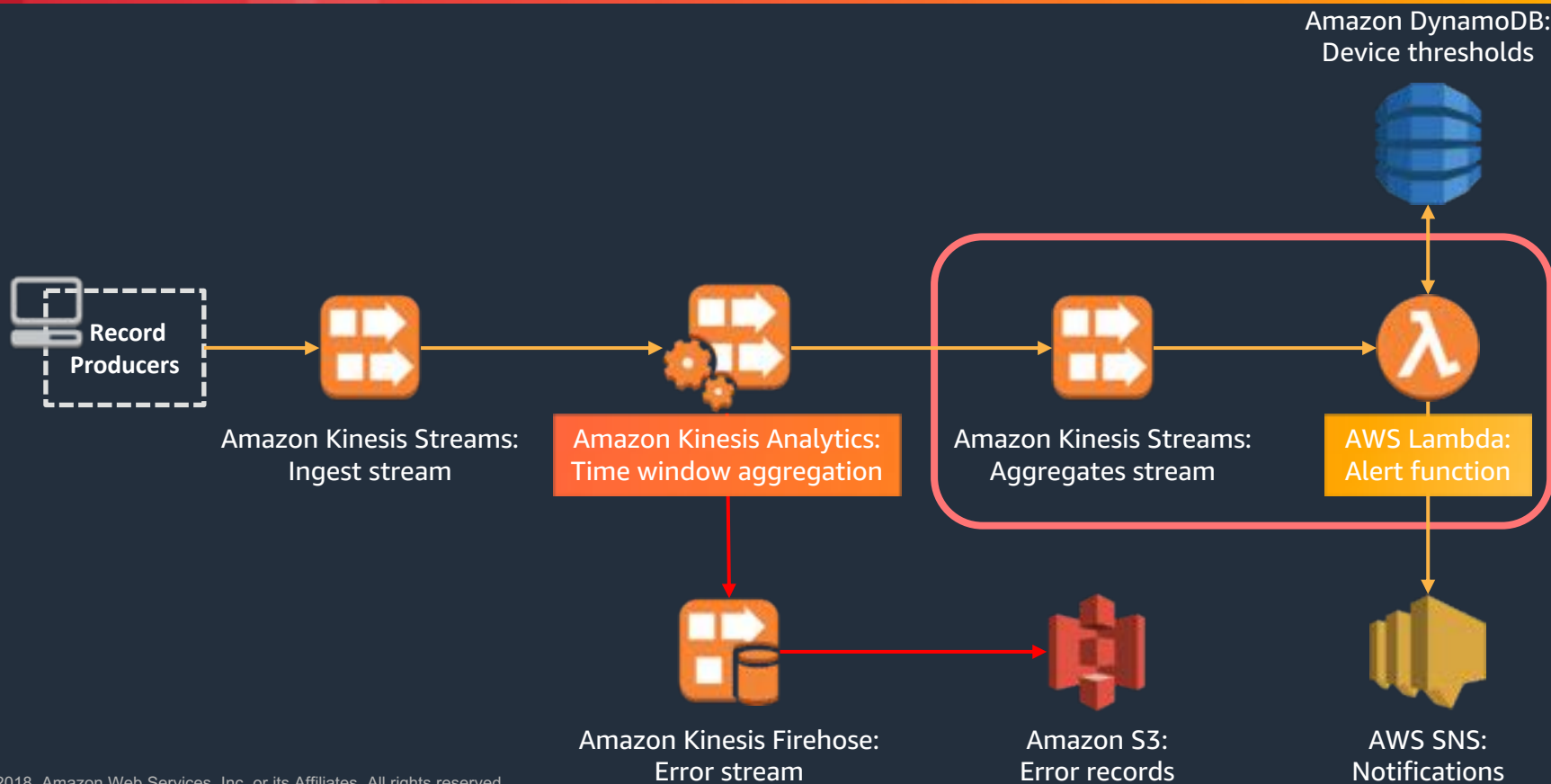


# Colour cube voting





# Real-time analytics



# Amazon Kinesis Streams and AWS Lambda

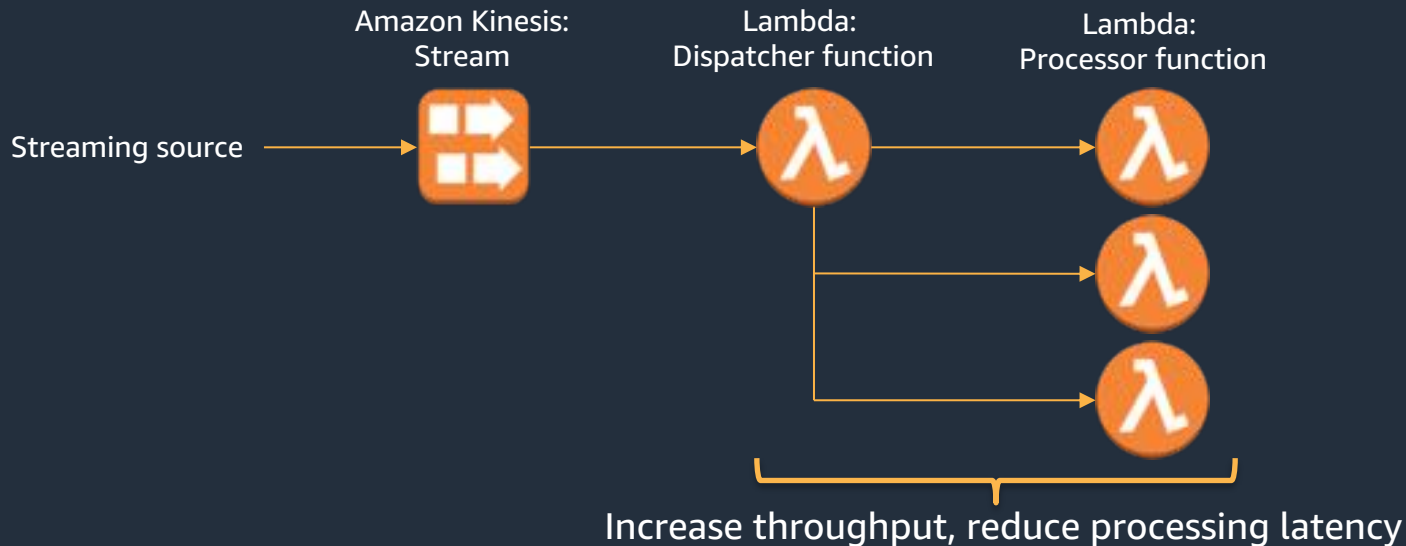


- Number of Amazon Kinesis Streams **shards** corresponds to **concurrent invocations** of Lambda function

**Batch size** sets maximum number of records per Lambda function invocation

# Fan-out pattern

Fan-out pattern trades strict message ordering vs higher throughput & lower latency



# Thomson Reuters – Product Insight

Solution for **usage analysis tracking**:

Capture, analyze, and visualize analytics data generated by offerings, providing insights to help product teams continuously improve the user experience

Throughput: Tested **4,000** requests / second

Growing to **10,000** requests / second or **25 Billion** requests / month

Latency: new events to user dashboards in less than **10 seconds**

Durable: **no data loss** since inception

The AWS logo, consisting of the letters "aws" in white on a dark blue square background, with a yellow curved arrow underneath.

GOTO

Amsterdam

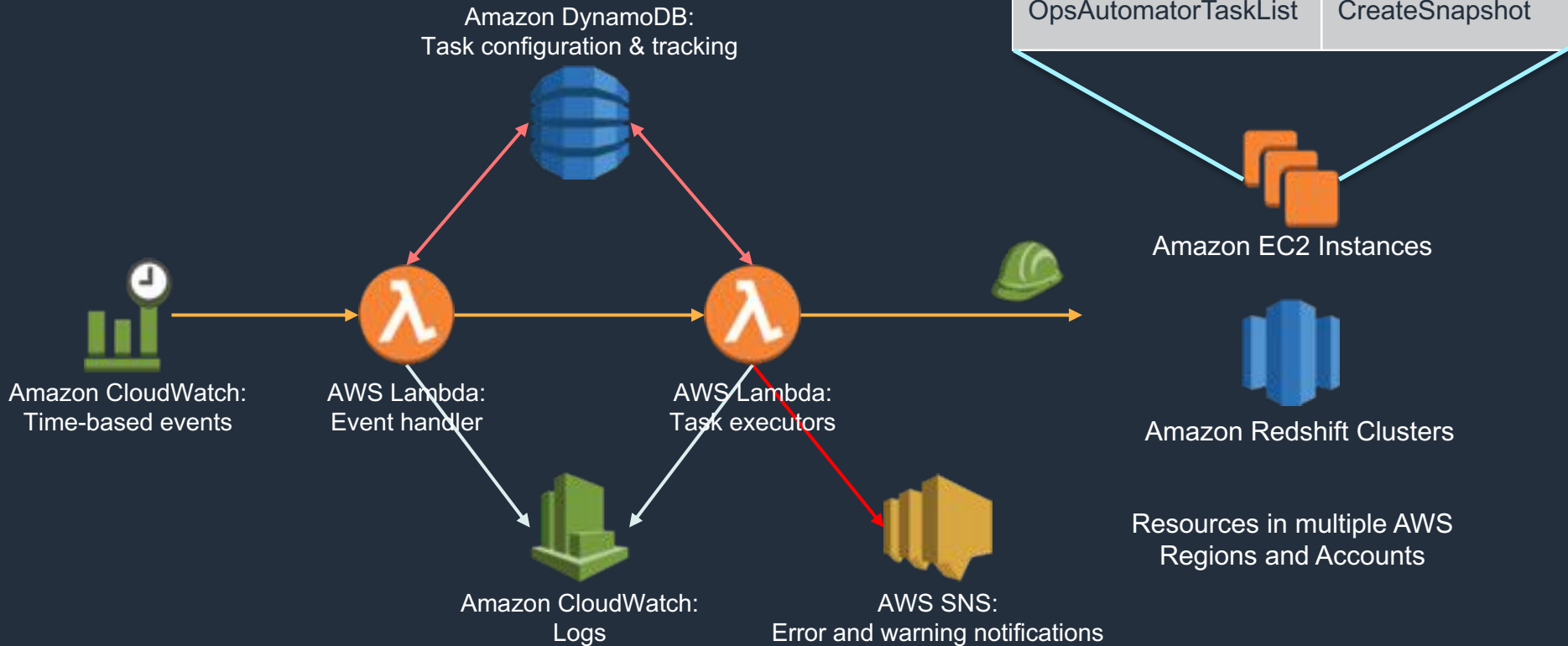
# Pattern 3: Operations Automation

# Automation characteristics

- Periodic jobs
- Event triggered workflows
- Enforce security policies
- Audit and notification
- Respond to alarms
- Extend AWS functionality

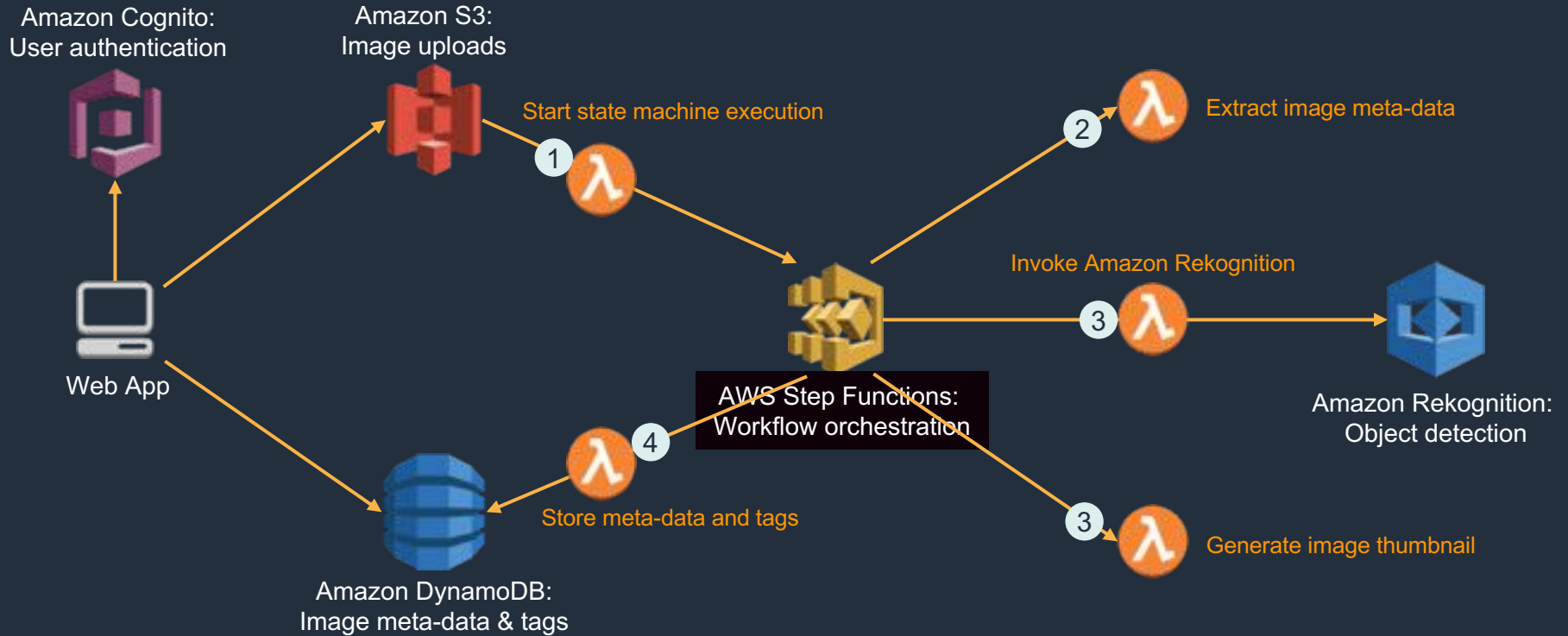
... All while being Highly Available, Scalable and Auditable

# AWS Ops Automator



<https://aws.amazon.com/answers/infrastructure-management/ops-automator/>

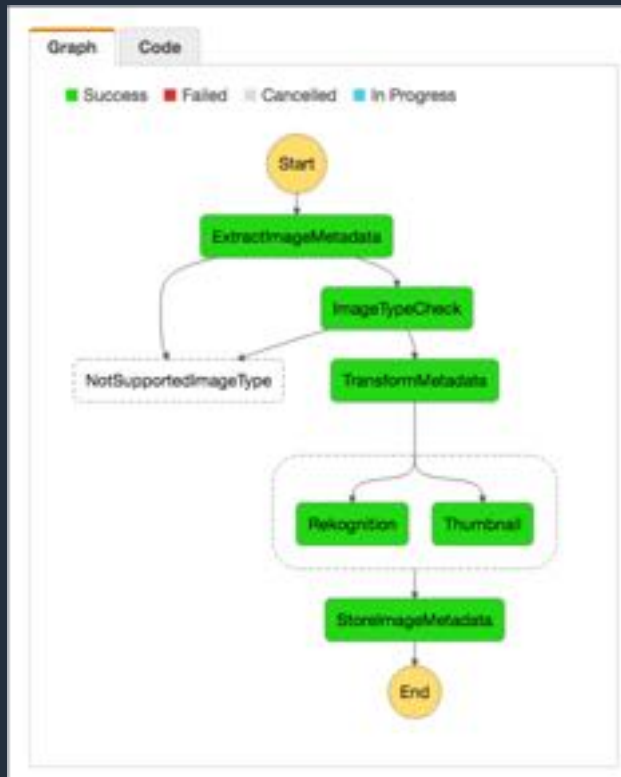
# Image recognition and processing



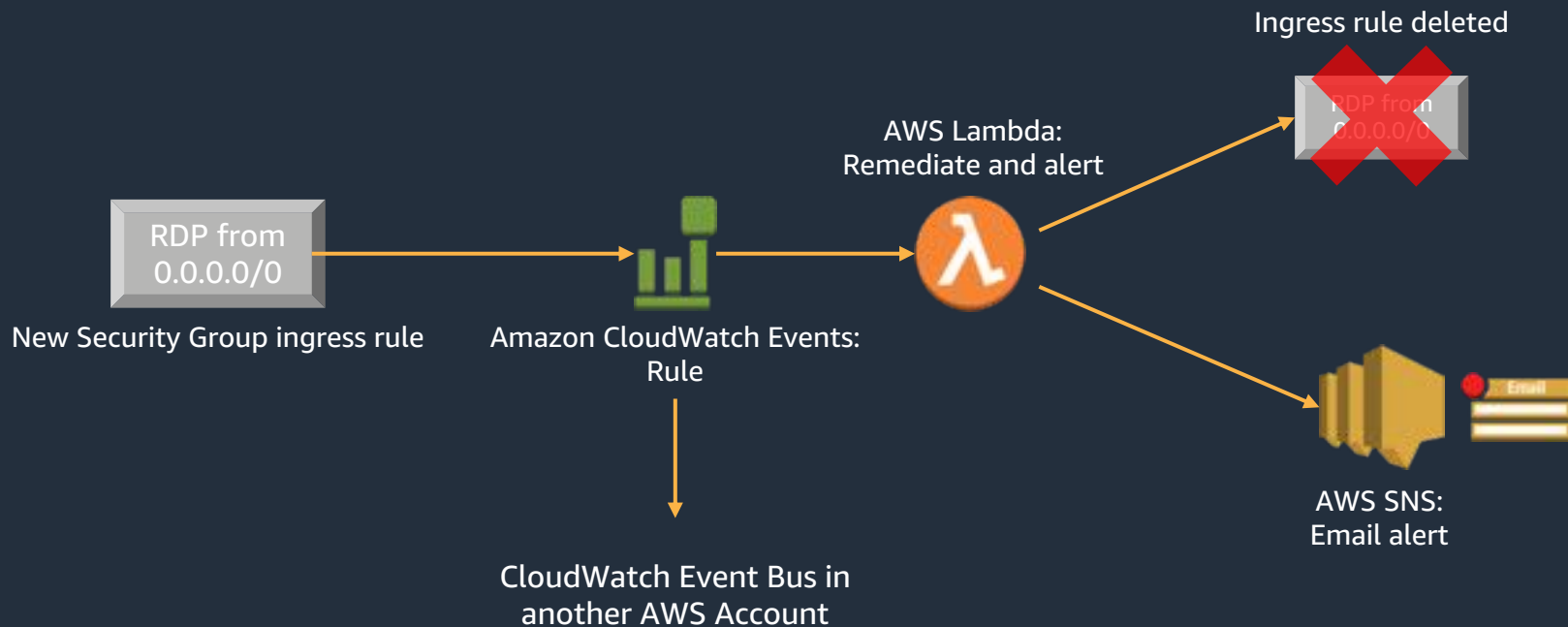
<https://github.com/aws-labs/lambda-refarch-imagerecognition>



# Step Functions state machine



# Enforce security policies



# Autodesk - Tailor

Serverless AWS **Account Provisioning and Management** Service:

- Automates AWS Account creation,
- Configures IAM, CloudTrail, AWS Config, Direct Connect, and VPC
- Enforces corporate standards
- Audit for compliance

Provisions new Accounts in **10 minutes** vs 10 hours in earlier manual process

Open source and extensible: <https://github.com/alanwill/aws-tailor>

# Best practices

- Gracefully handle **API throttling** by retrying with an exponential back-off algorithm (AWS SDKs do this for you)
- Publish **custom metrics** from your Lambda function that are meaningful for operations (e.g. number of EBS volumes snapshotted)
- Enable **X-Ray** tracing for your Lambda functions
- Document how to **disable** event triggers for your automation when troubleshooting



GOTO  
Amsterdam

# Additional Patterns

# Serverless Data Lake



Amazon  
DynamoDB



AWS Glue



Amazon ES

Catalog & Search



Amazon  
Kinesis  
Streams



Amazon  
Kinesis  
Firehose



AWS  
Direct  
Connect

Ingest



S3  
Bucket(s)



Amazon  
Cognito



Amazon API  
Gateway



AWS IAM

API/UI



AWS  
Lambda



Amazon  
Athena



Amazon  
QuickSight



AWS Glue



Amazon  
Redshift  
Spectrum

Analytics & Processing



AWS IAM



Key  
Management  
Service



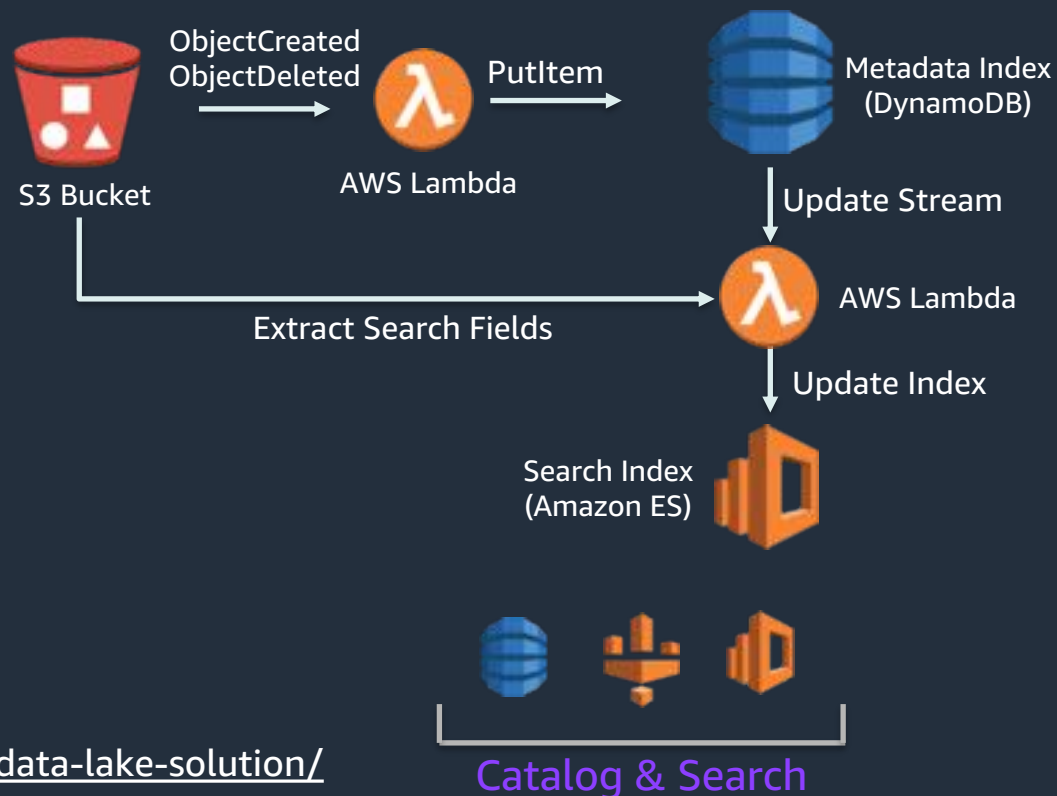
AWS  
CloudTrail



Amazon  
Macie

# Search and Data Catalog

- DynamoDB as Metadata repository
- Amazon Elasticsearch



<https://aws.amazon.com/answers/big-data/data-lake-solution/>

# Further Reading

- Optimizing Enterprise Economics with Serverless Architectures  
<https://d0.awsstatic.com/whitepapers/optimizing-enterprise-economics-serverless-architectures.pdf>
- Serverless Architectures with AWS Lambda  
<https://d1.awsstatic.com/whitepapers/serverless-architectures-with-aws-lambda.pdf>
- Serverless Applications Lens - AWS Well-Architected Framework  
<https://d1.awsstatic.com/whitepapers/architecture/AWS-Serverless-Applications-Lens.pdf>
- Streaming Data Solutions on AWS with Amazon Kinesis  
<https://d1.awsstatic.com/whitepapers/whitepaper-streaming-data-solutions-on-aws-with-amazon-kinesis.pdf>
- AWS Serverless Multi-Tier Architectures  
[https://d1.awsstatic.com/whitepapers/AWS\\_Serverless\\_Multi-Tier\\_Architectures.pdf](https://d1.awsstatic.com/whitepapers/AWS_Serverless_Multi-Tier_Architectures.pdf)



# Summary

Use DevOps tools to **automate** your serverless deployments

Apply serverless patterns for common use-cases:

- Web application
- Data Lake Foundation
- Stream processing
- Operations automation

What will **you** build with Serverless?



*Please*

**Remember to  
rate this session**

*Thank you!*

Boaz Ziniman – Technical Evangelist, AWS

 @ziniman

 boaz.ziniman.aws