Observability for Modern Event Driven Applications

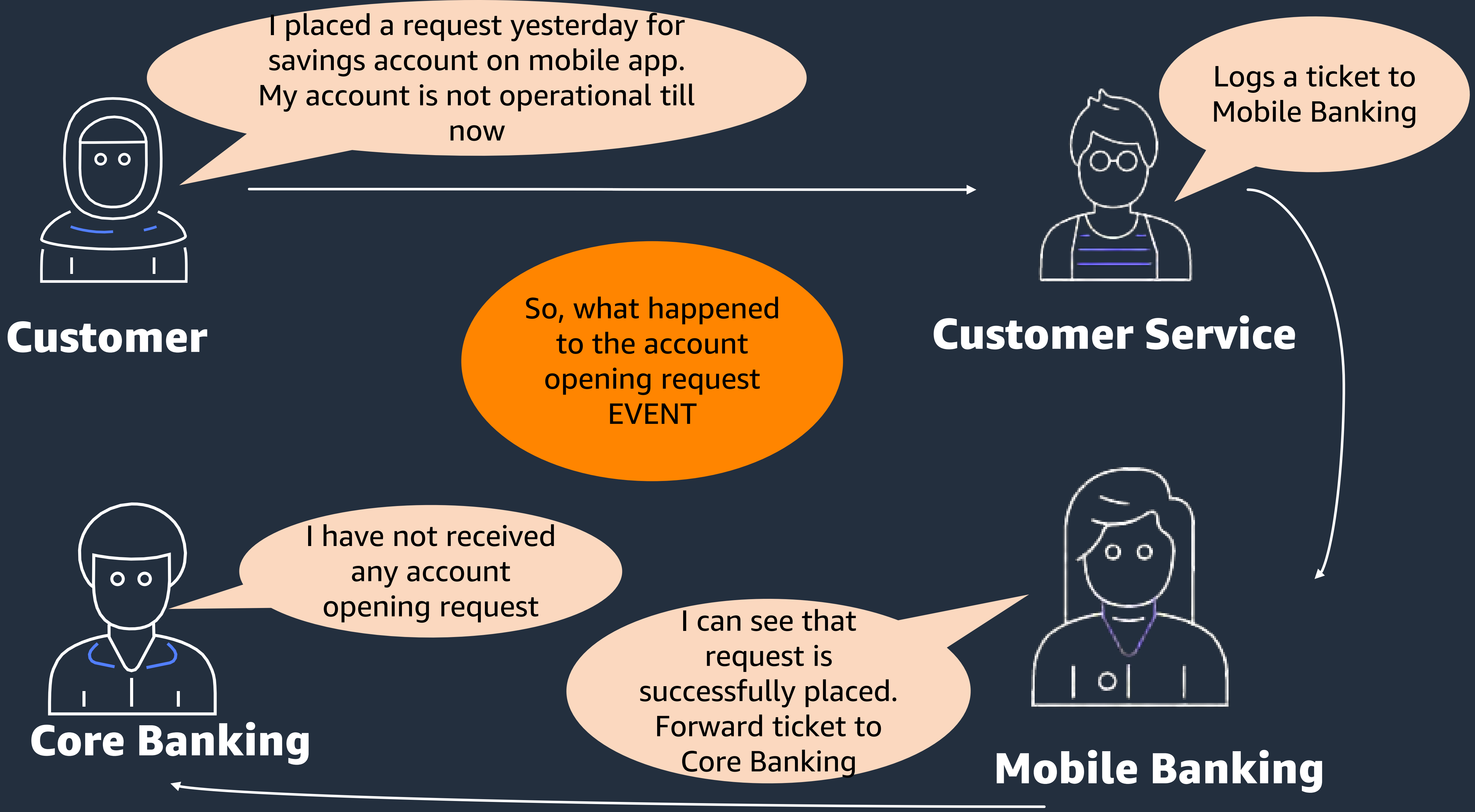
Urmila Raju (She/Her)

Senior Solutions Architect, AWS

Bank XXX - NewsLetter

“ Dear Customers - We are happy to announce that you can now open savings accounts through Mobile Banking!! “

Place the request with a few clicks on our mobile app and get your account operational in 2 hours



I placed a request yesterday for savings account on mobile app. My account is not operational till now

Customer

Logs a ticket to Mobile Banking

Customer Service

I can see that request is successfully placed. Forward ticket to Core Banking

Mobile Banking

I have not received any account opening request

Core Banking

So, what happened to the account opening request EVENT

Observability for Modern Event Driven Applications

Urmila Raju (She/Her)

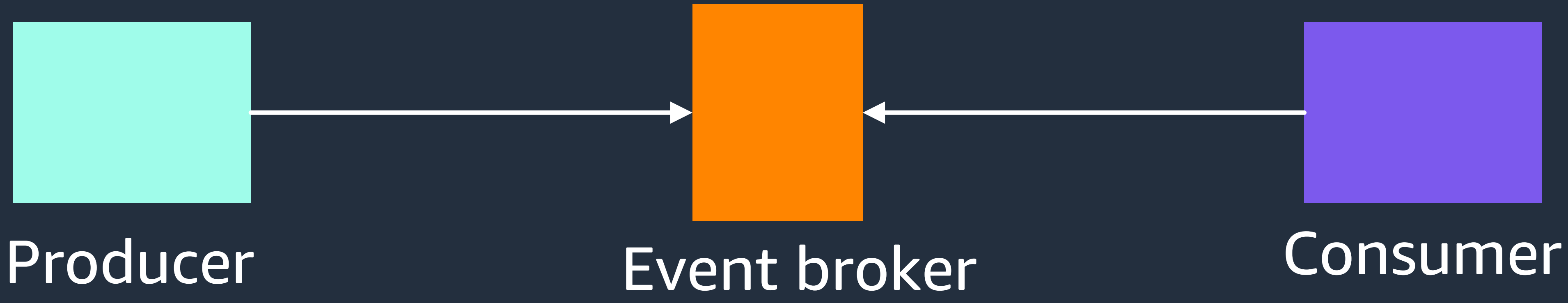
Senior Solutions Architect, AWS

Event-driven architectures (EDA)

An **architectural style** of building **loosely-coupled** software systems that work together by **emitting and responding to events.**

What is an event

An event is a change in state, or an update emitted by a *producer*, which *consumers* are interested in.



Why Customers are moving to event-driven applications

1

Speed & agility

Move faster. Build and deploy services independently.

2

Resiliency

Loosely coupled systems can run and fail independently.

3

Scalability

Minimize waiting time through async and parallel processing.

4

Work backwards your business process

Align business and technology stakeholders

But…… EDA is hard to get it right!

1

Eventual Consistency

2

End to End Performance

3

Meeting Business SLAs

OBSERVABILITY

What is observability?

A measure of how well we can understand a system from the work it does

90% of the methods in this service complete in under 200 milliseconds

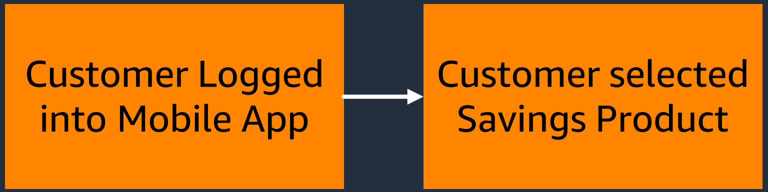
This API had 100 ‘403’ errors today

Our application is 35% slower than last week after this configuration change

Example Business Problem : Open Savings A/C for an existing customer through Mobile Banking App

Customer Logged into Mobile App

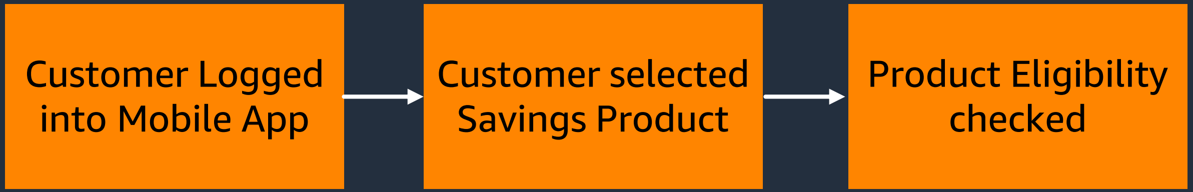
Example Business Problem : Open Savings A/C for an existing customer through Mobile Banking App



Customer Logged into Mobile App

Customer selected Savings Product

Example Business Problem : Open Savings A/C for an existing customer through Mobile Banking App

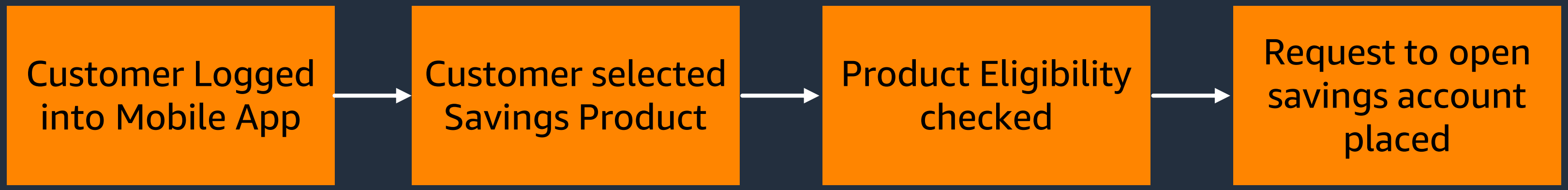


Customer Logged into Mobile App

Customer selected Savings Product

Product Eligibility checked

Example Business Problem : Open Savings A/C for an existing customer through Mobile Banking App



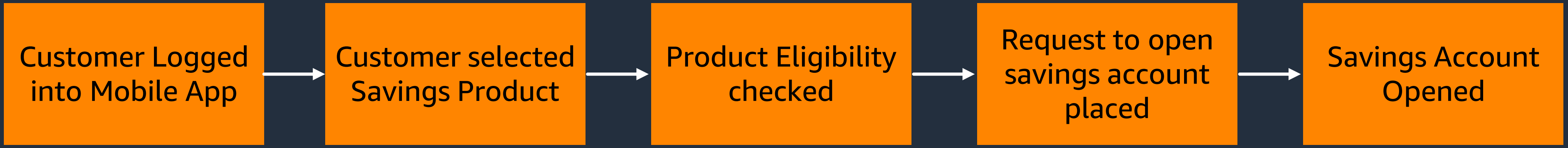
Customer Logged into Mobile App

Customer selected Savings Product

Product Eligibility checked

Request to open savings account placed

Example Business Problem : Open Savings A/C for an existing customer through Mobile Banking App



Customer Logged into Mobile App

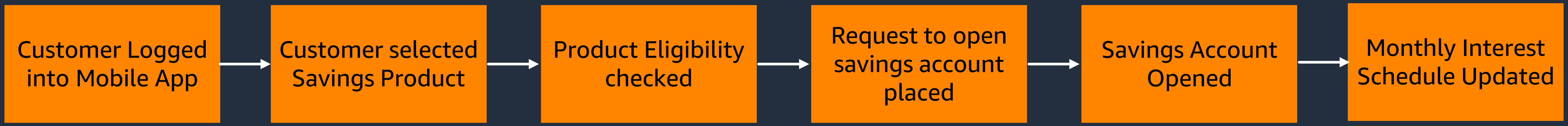
Customer selected Savings Product

Product Eligibility checked

Request to open savings account placed

Savings Account Opened

Example Business Problem : Open Savings A/C for an existing customer through Mobile Banking App



Customer Logged into Mobile App

Customer selected Savings Product

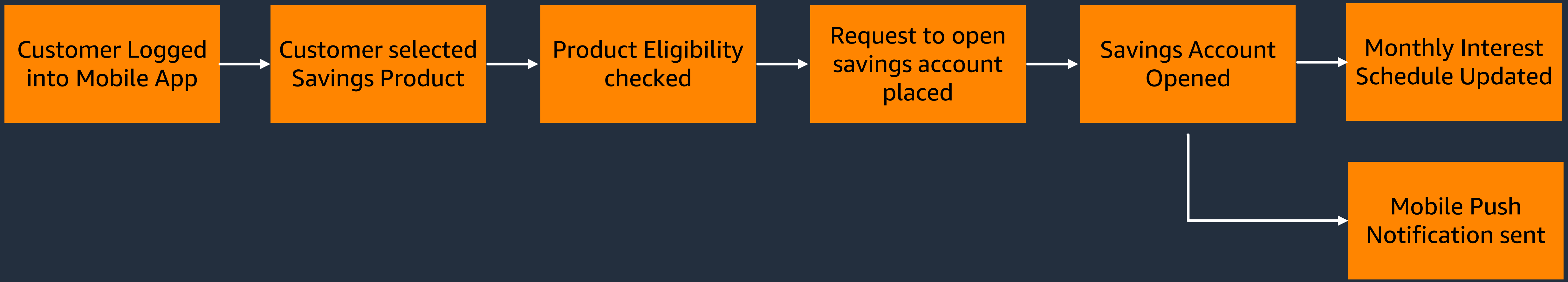
Product Eligibility checked

Request to open savings account placed

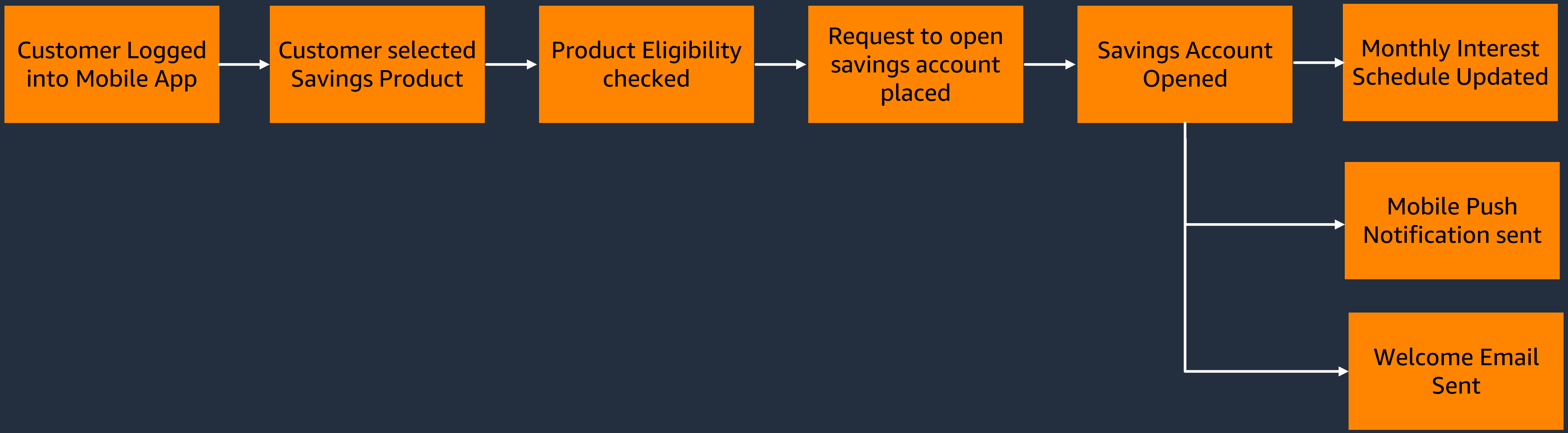
Savings Account Opened

Monthly Interest Schedule Updated

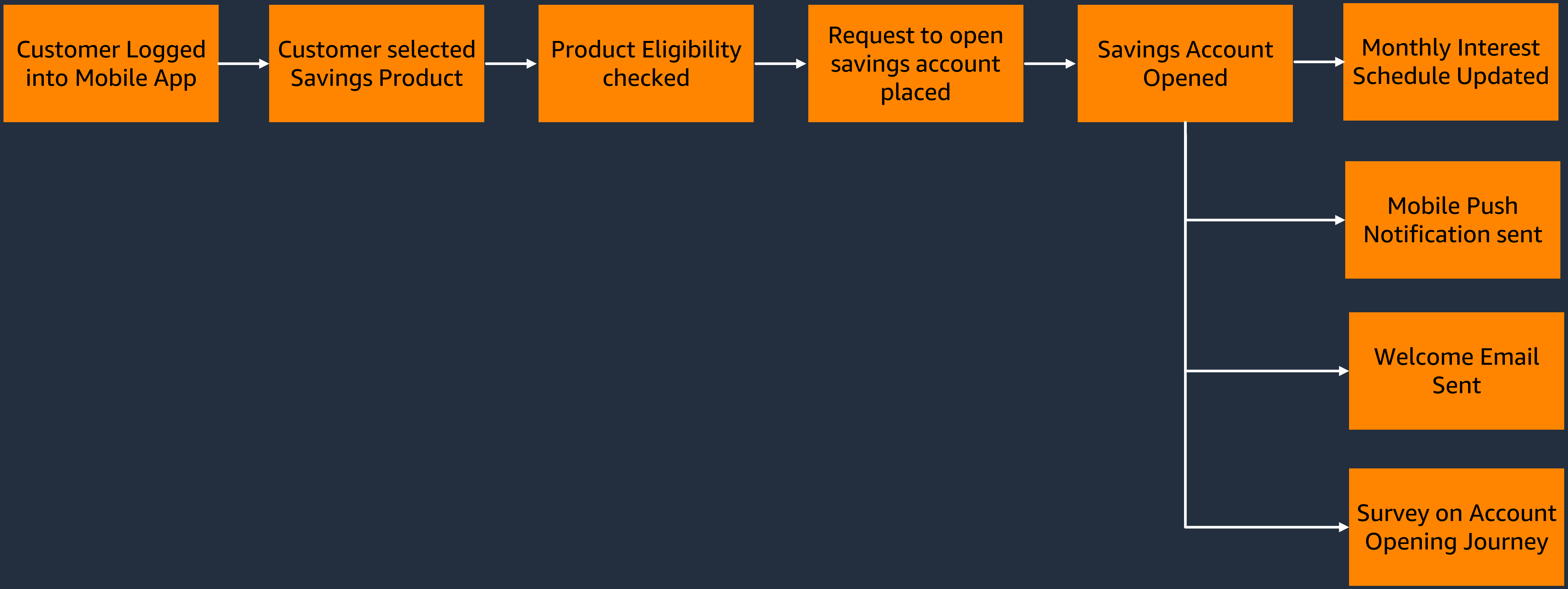
Example Business Problem : Open Savings A/C for an existing customer through Mobile Banking App


Customer Logged into Mobile App
Customer selected Savings Product
Product Eligibility checked
Request to open savings account placed
Savings Account Opened
Monthly Interest Schedule Updated
Mobile Push Notification sent

Example Business Problem : Open Savings A/C for an existing customer through Mobile Banking App


Customer Logged into Mobile App
Customer selected Savings Product
Product Eligibility checked
Request to open savings account placed
Savings Account Opened
Monthly Interest Schedule Updated
Mobile Push Notification sent
Welcome Email Sent

Example Business Problem : Open Savings A/C for an existing customer through Mobile Banking App



Customer Logged into Mobile App

Customer selected Savings Product

Product Eligibility checked

Request to open savings account placed

Savings Account Opened

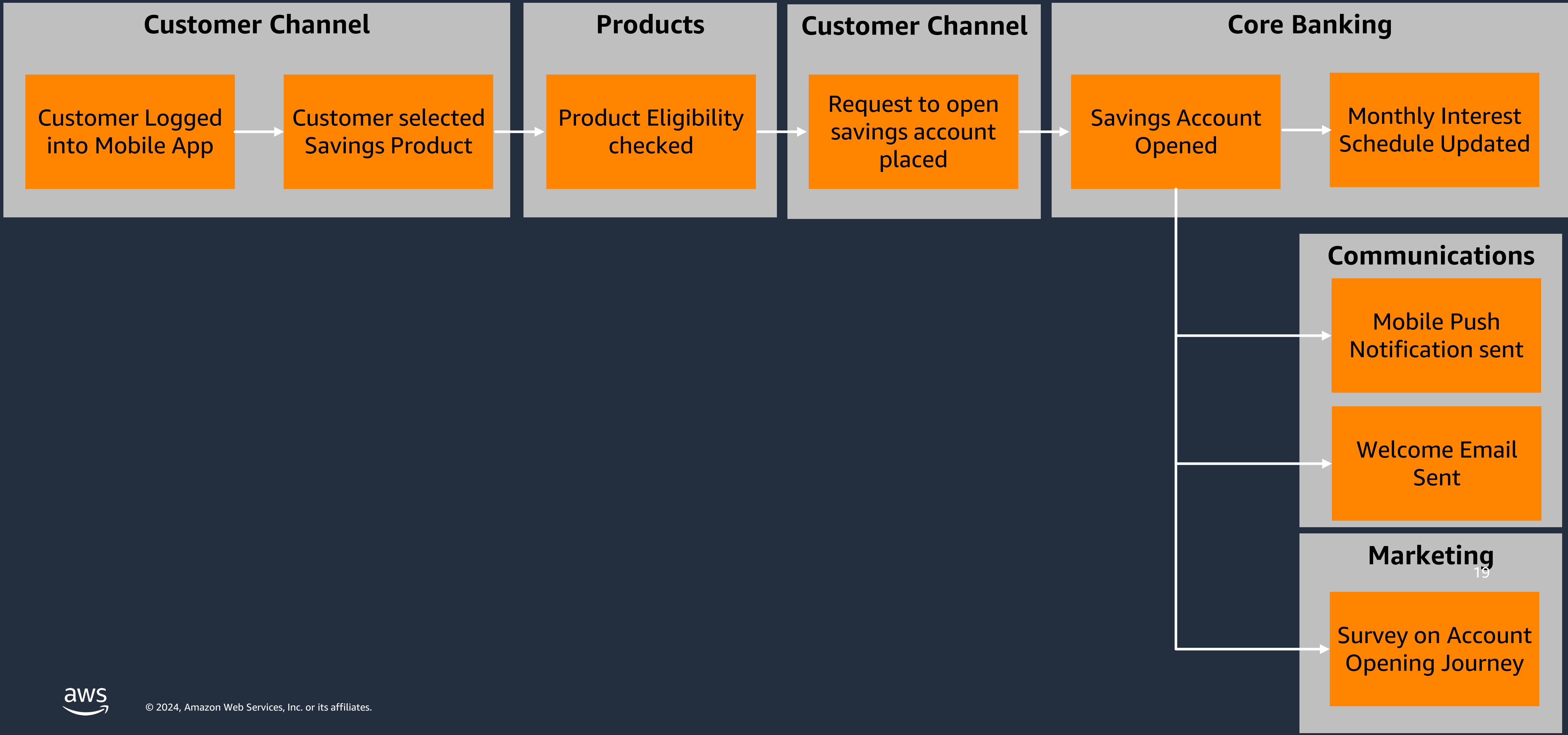
Monthly Interest Schedule Updated

Mobile Push Notification sent

Welcome Email Sent

Survey on Account Opening Journey

EDA: Based on Domain Driven Design (DDD) and Events

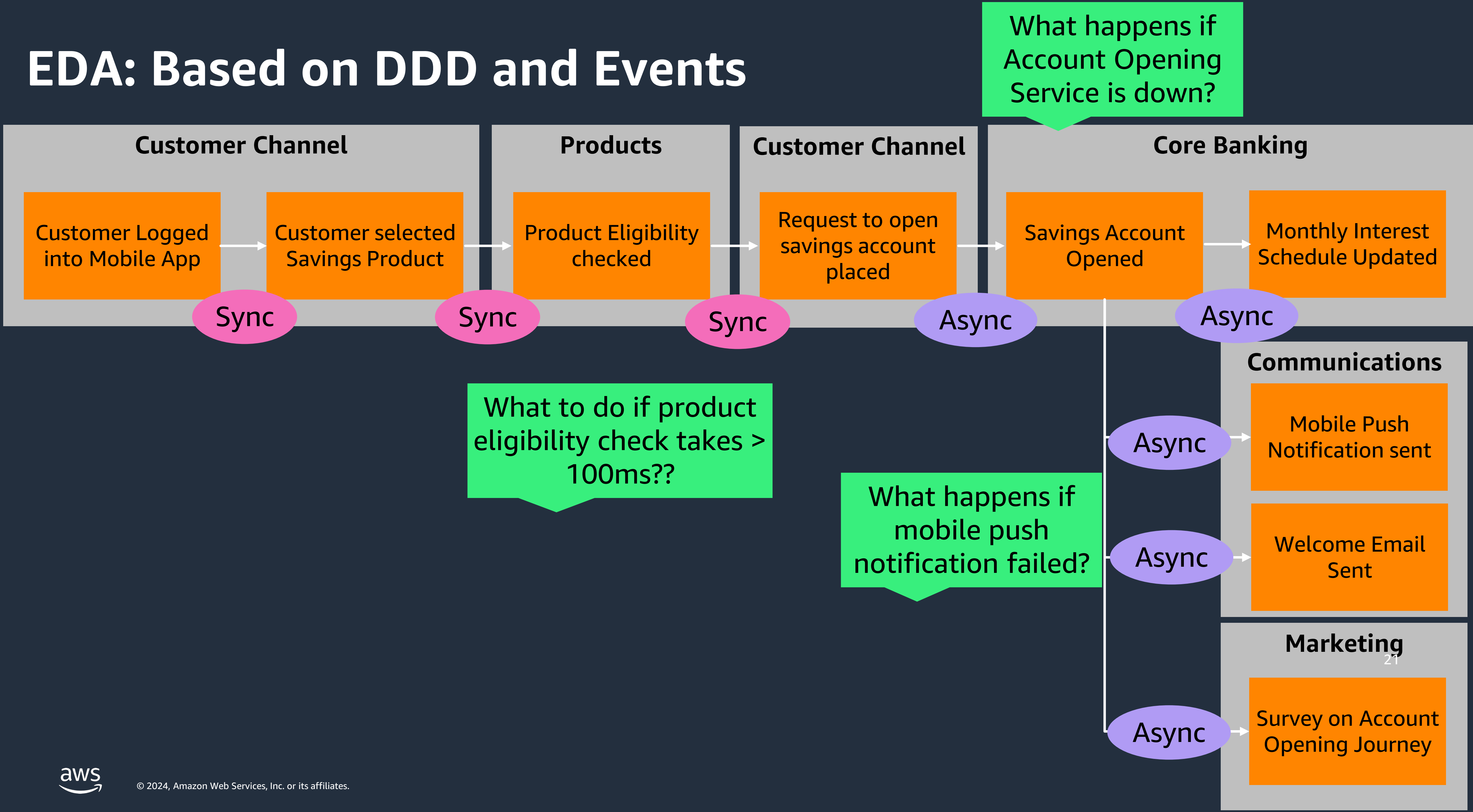

Customer Channel
Customer Logged into Mobile App
Customer selected Savings Product
Products
Product Eligibility checked
Customer Channel
Request to open savings account placed
Core Banking
Savings Account Opened
Monthly Interest Schedule Updated
Communications
Mobile Push Notification sent
Welcome Email Sent
Marketing
Survey on Account Opening Journey

Example Business Problem : Open ISA (Instant Savings Account) for an existing customer through Mobile Banking App

Business SLAs:

- Customer can request to **open account 24/7**.
- Customer can **check eligibility in real time (100 ms)** and place account opening request.
- Account should be **operational in 2 hours** of request being placed.
- Customer should receive **mobile notification in 2 hours** of request being placed.
- Customer should **receive welcome email in 1 day** of account being opened.

EDA: Based on DDD and Events



What happens if Account Opening Service is down?

Customer Channel

Customer Logged into Mobile App

Sync

Customer selected Savings Product

Sync

Products

Product Eligibility checked

Sync

Customer Channel

Request to open savings account placed

Async

Core Banking

Savings Account Opened

Async

Monthly Interest Schedule Updated

What to do if product eligibility check takes > 100ms??

What happens if mobile push notification failed?

Communications

Async

Mobile Push Notification sent

Async

Welcome Email Sent

Marketing

Async

Survey on Account Opening Journey

aws

End to End Observability is key for a successful EDA

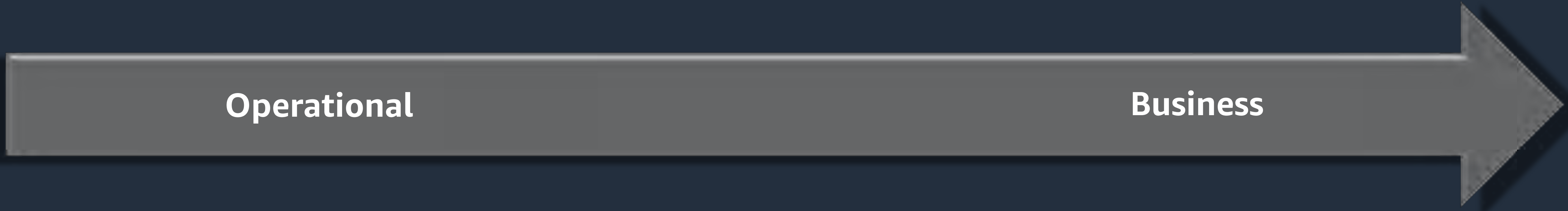
Observability matters because

Visibility

Real-time troubleshooting

Customer experience

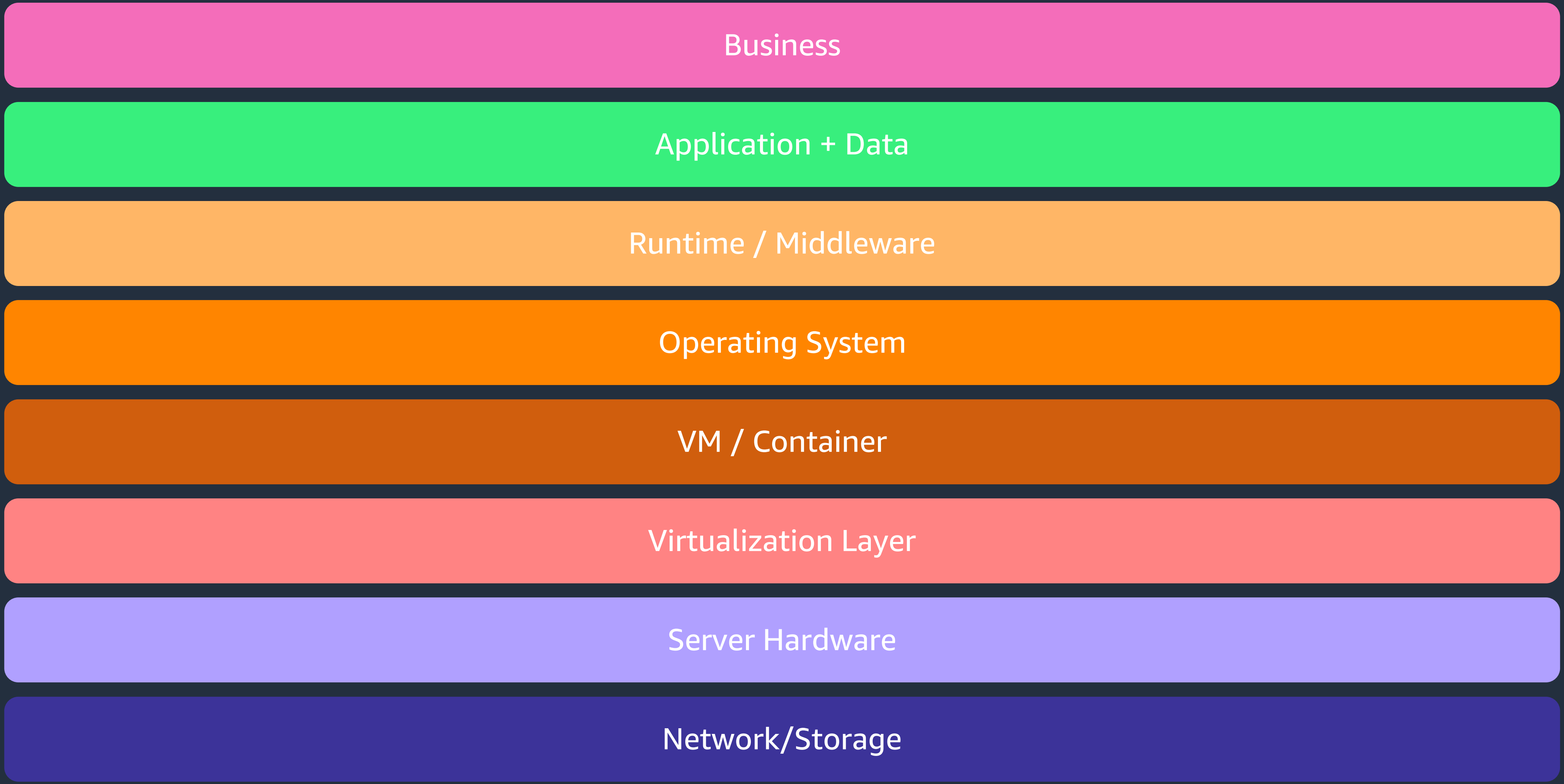
Applications = $$



Operational

Business

Traditional monitoring layers



Business

Application + Data

Runtime / Middleware

Operating System

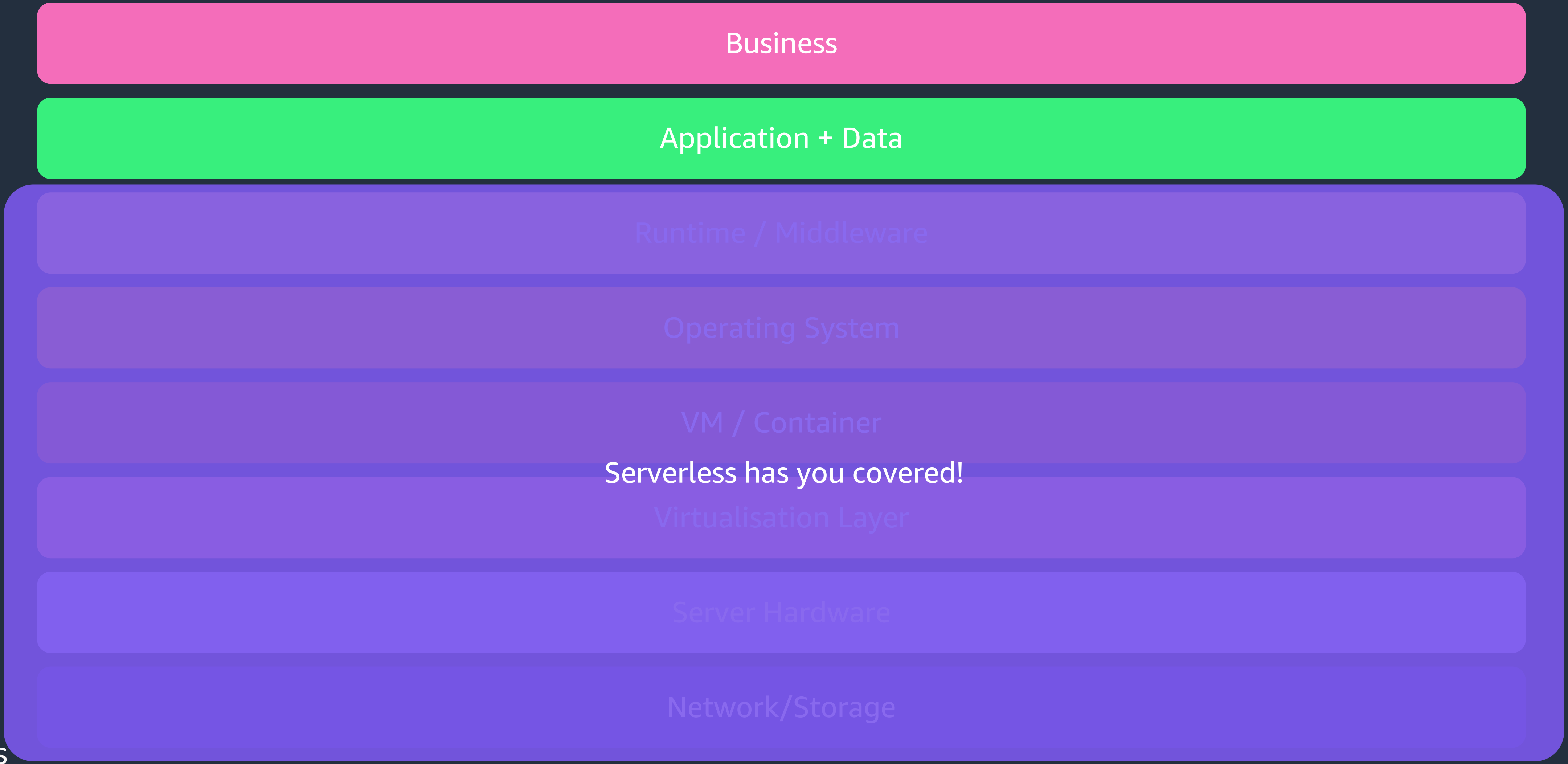
VM / Container

Virtualization Layer

Server Hardware

Network/Storage

When you do EDA on AWS....



Business

Application + Data

Runtime / Middleware

Operating System

VM / Container

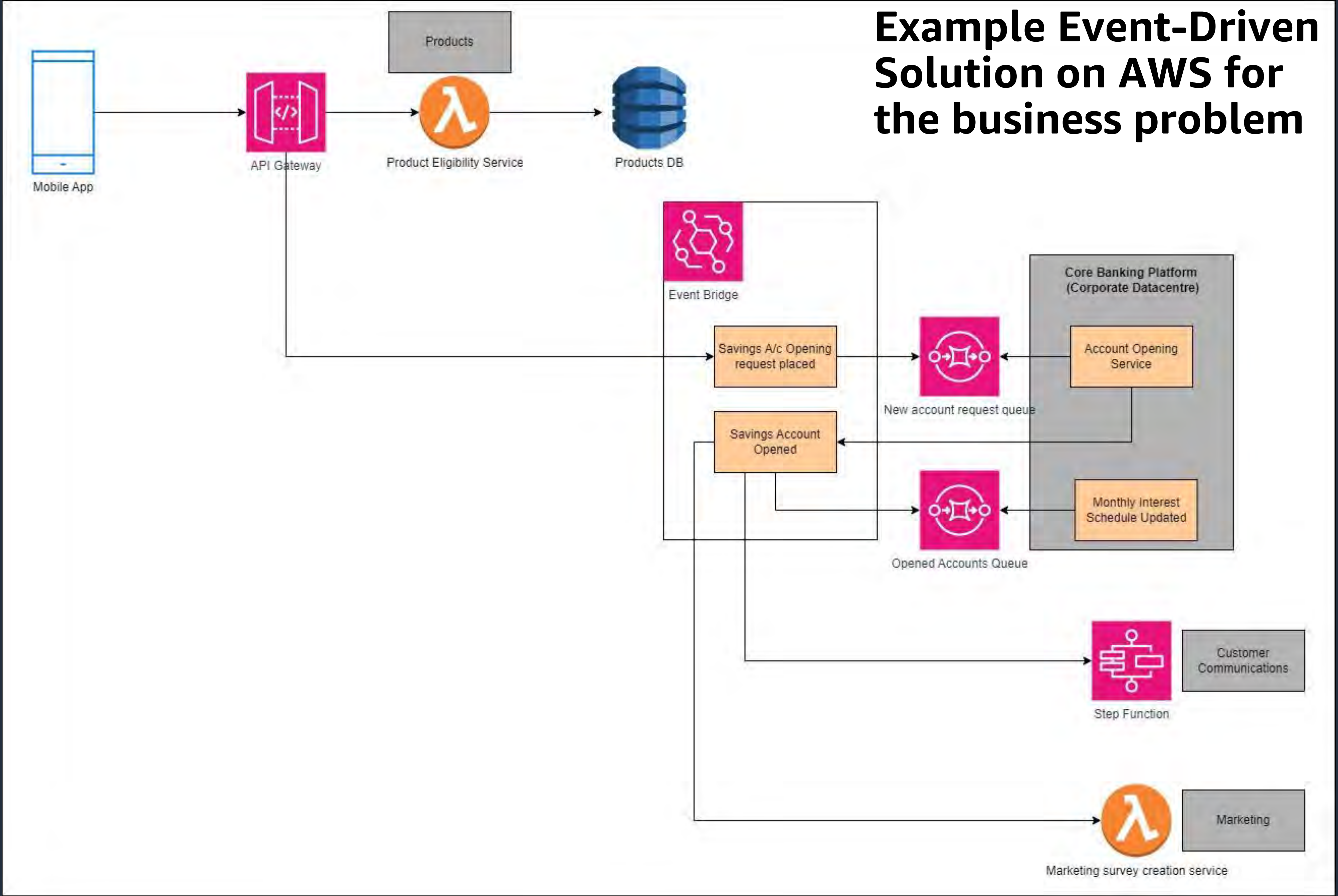
Serverless has you covered!

Virtualisation Layer

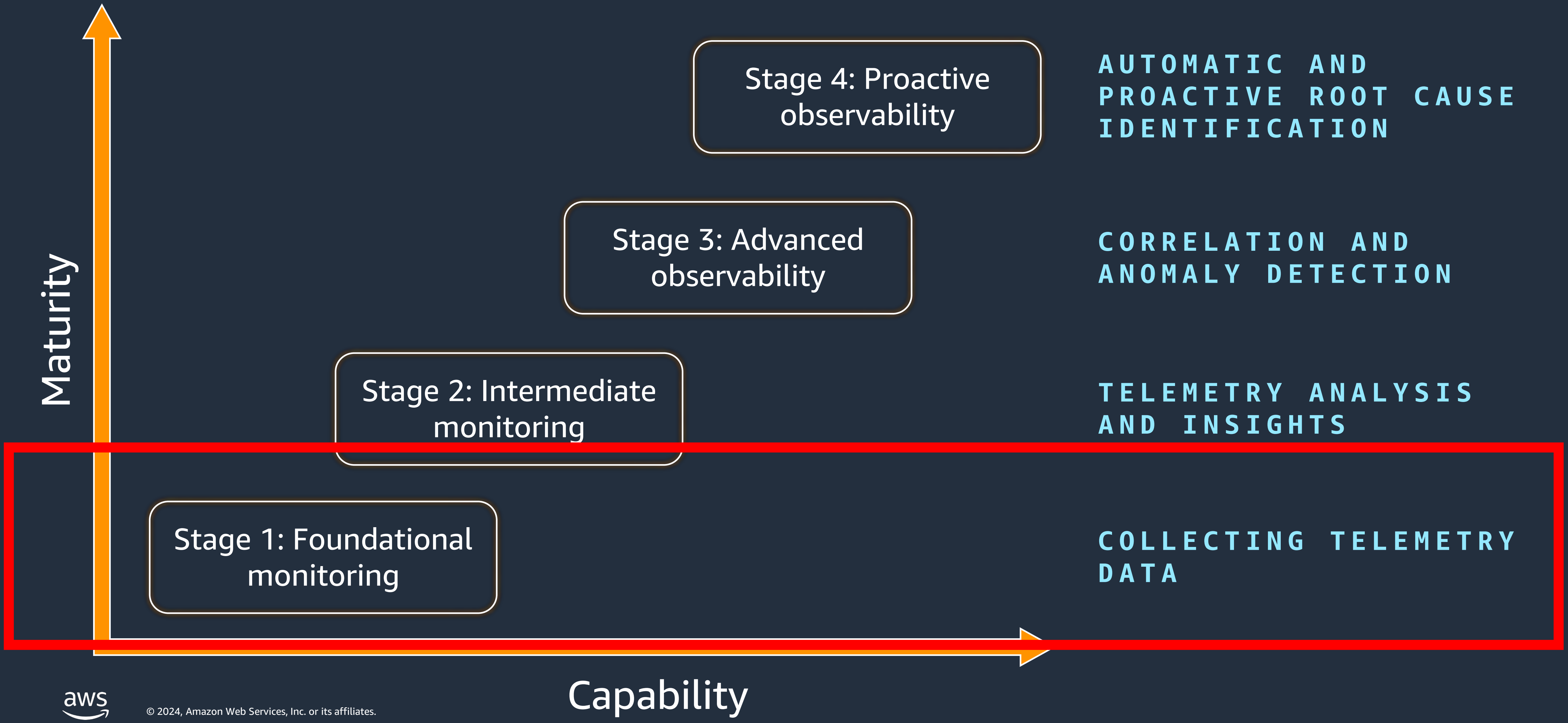
Server Hardware

Network/Storage

Example Event-Driven Solution on AWS for the business problem


Products
Mobile App
API Gateway
Product Eligibility Service
Products DB
Event Bridge
Core Banking Platform (Corporate Datacentre)
Savings A/c Opening request placed
New account request queue
Account Opening Service
Savings Account Opened
Monthly Interest Schedule Updated
Opened Accounts Queue
Step Function
Customer Communications
Marketing
Marketing survey creation service

AWS observability maturity model

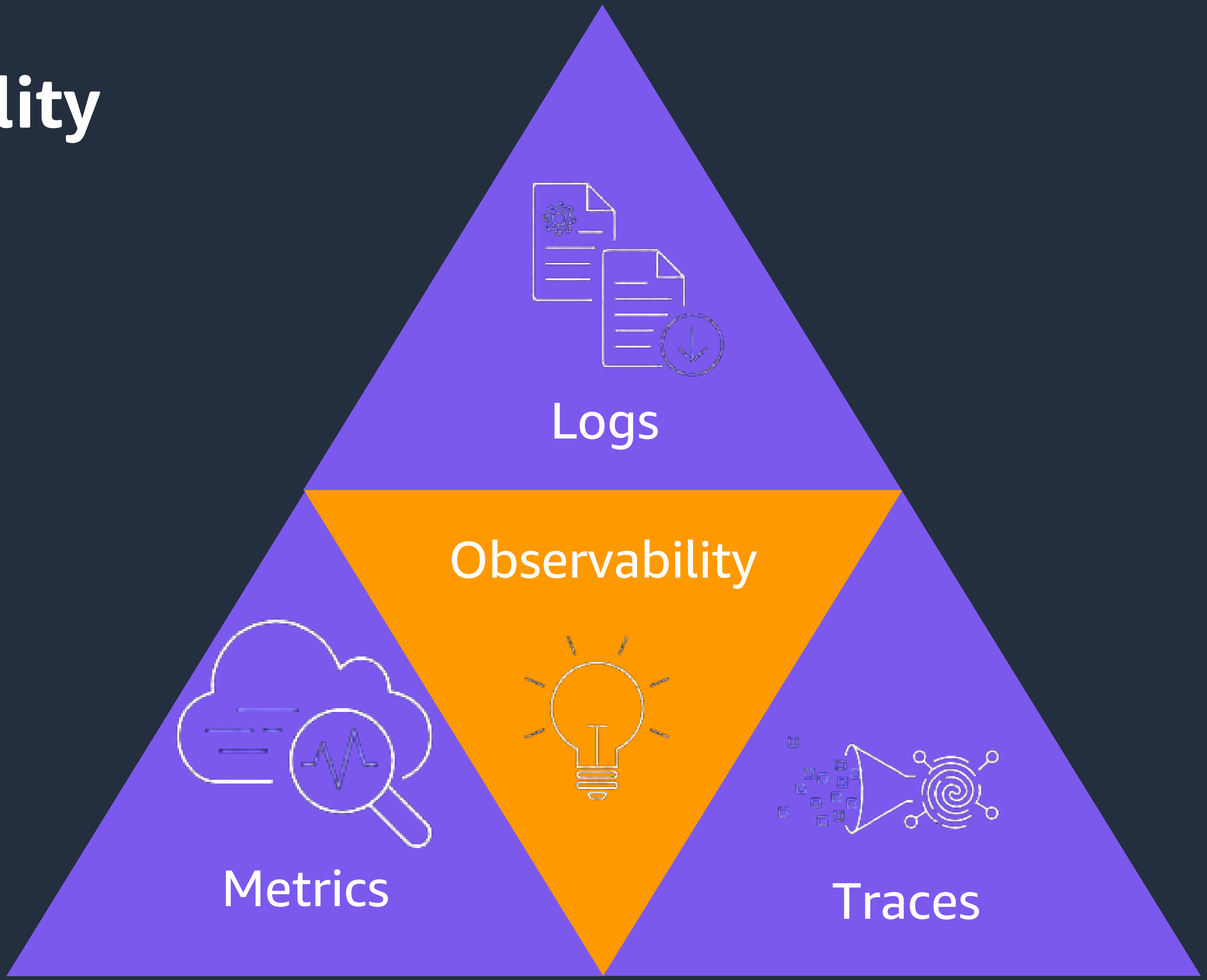

Stage 4: Proactive observability
AUTOMATIC AND PROACTIVE ROOT CAUSE IDENTIFICATION
Stage 3: Advanced observability
CORRELATION AND ANOMALY DETECTION
Stage 2: Intermediate monitoring
TELEMETRY ANALYSIS AND INSIGHTS
Stage 1: Foundational monitoring
COLLECTING TELEMETRY DATA
Maturity
Capability

Three pillars of observability

Metrics	Logs	Traces
Numeric data measured at various time intervals (time series data); SLIs (request rate, error rate, duration etc.)	Timestamped records of discrete events that happened within an application or system, such as a failure, an error, or a state transformation	A trace represents a single user’s journey across multiple applications and systems (usually microservices)

Definitions from: Distributed Systems Observability
https://www.oreilly.com/library/view/distributed-systems-observability/9781492033431/

Observability



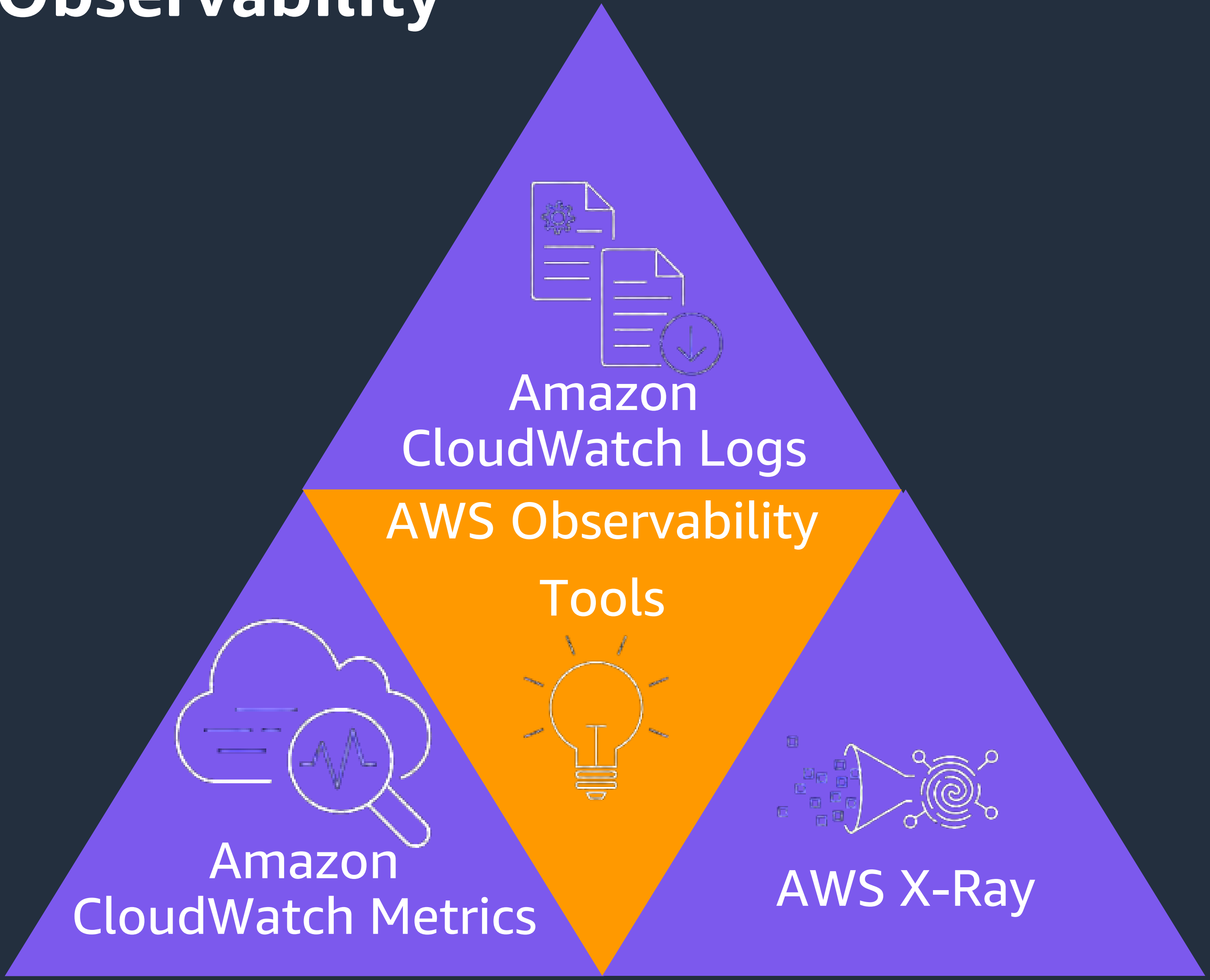
Logs

Observability

Metrics

Traces

AWS Native Observability



Amazon CloudWatch Logs

AWS Observability Tools

Amazon CloudWatch Metrics

AWS X-Ray

Viewing standard metrics

Amazon CloudWatch – Inbuilt Metrics

Collects Standard Metrics

Amazon SQS

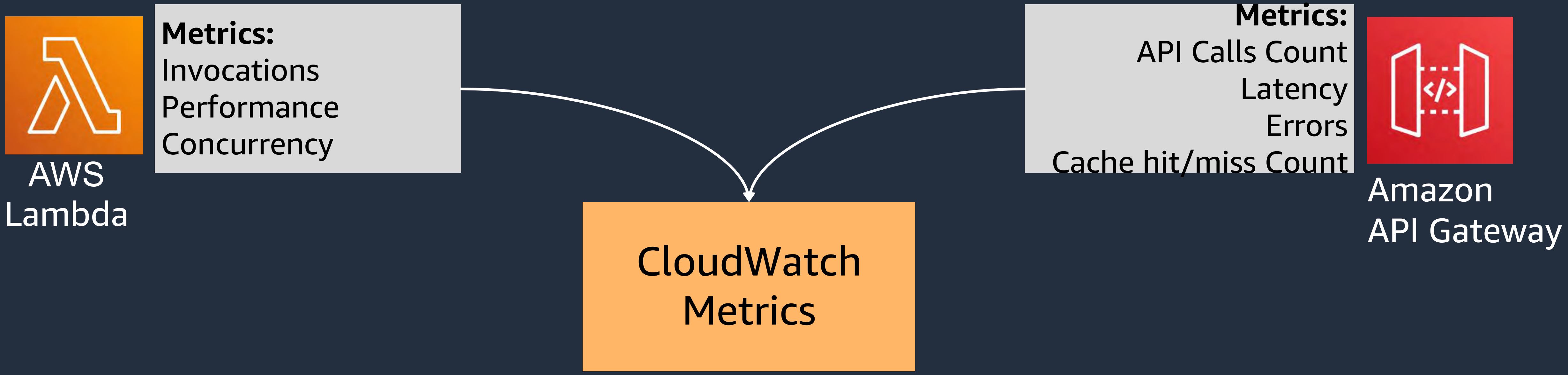
Amazon EventBridge

AWS Step functions

AWS Lambda

Amazon API Gateway

CloudWatch built-in metrics



AWS Lambda

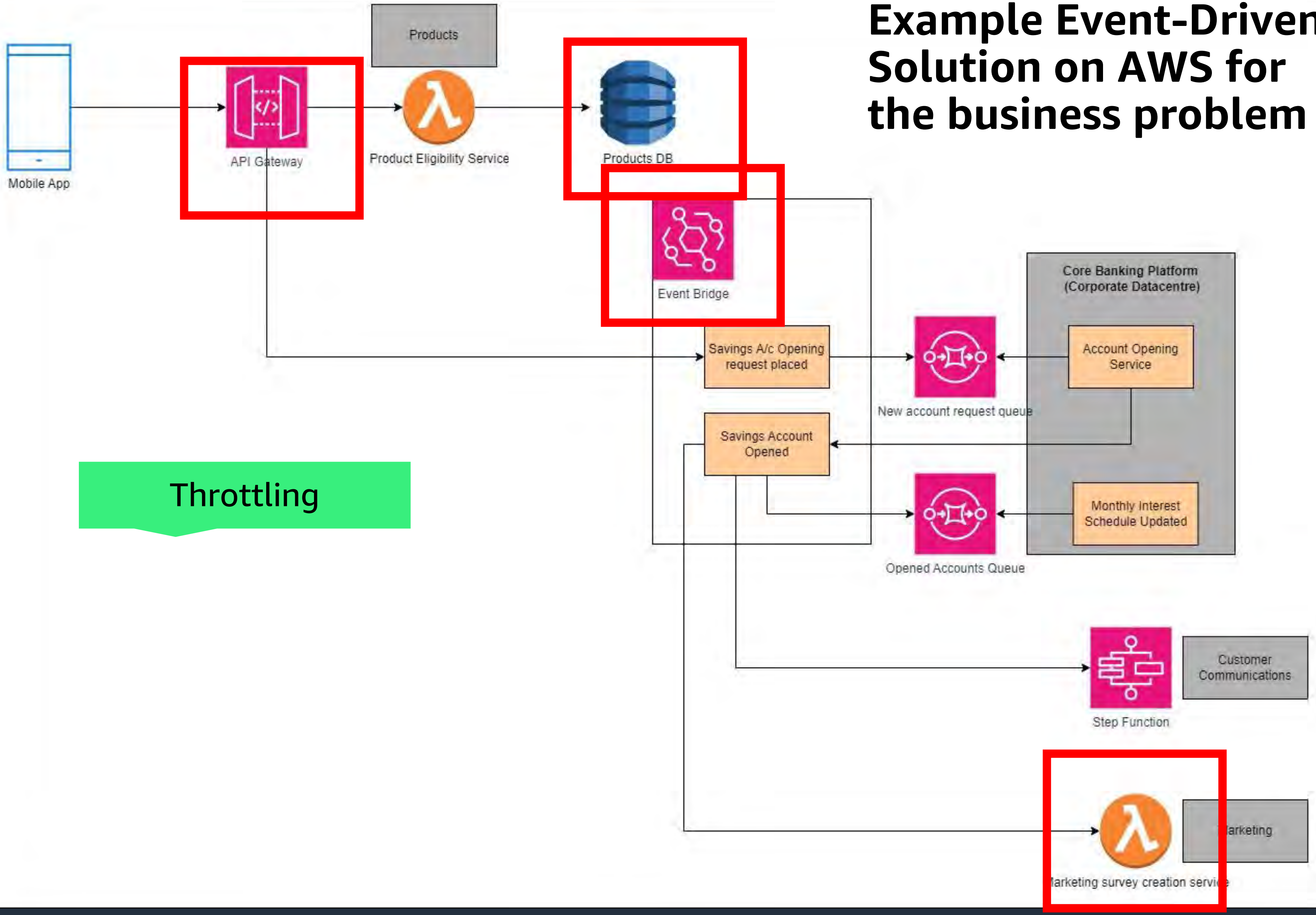
Metrics:
Invocations
Performance
Concurrency

Metrics:
API Calls Count
Latency
Errors
Cache hit/miss Count

Amazon API Gateway

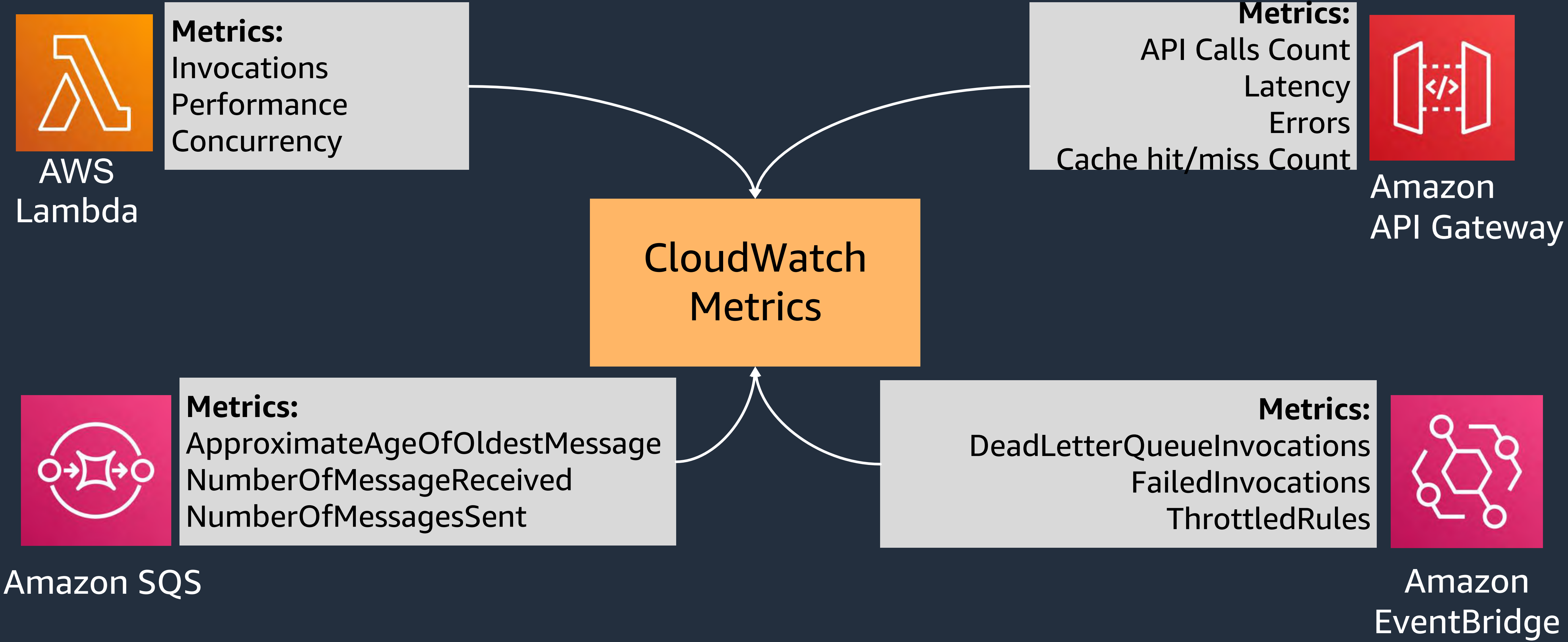
CloudWatch Metrics

Example Event-Driven Solution on AWS for the business problem


Mobile App
API Gateway
Products
Product Eligibility Service
Products DB
Event Bridge
Savings A/c Opening request placed
Savings Account Opened
New account request queue
Opened Accounts Queue
Core Banking Platform (Corporate Datacentre)
Account Opening Service
Monthly Interest Schedule Updated
Step Function
Customer Communications
larketing
larketing survey creation servi e

Throttling

CloudWatch built-in metrics



AWS Lambda

Metrics:
Invocations
Performance
Concurrency

Amazon API Gateway

Metrics:
API Calls Count
Latency
Errors
Cache hit/miss Count

CloudWatch Metrics

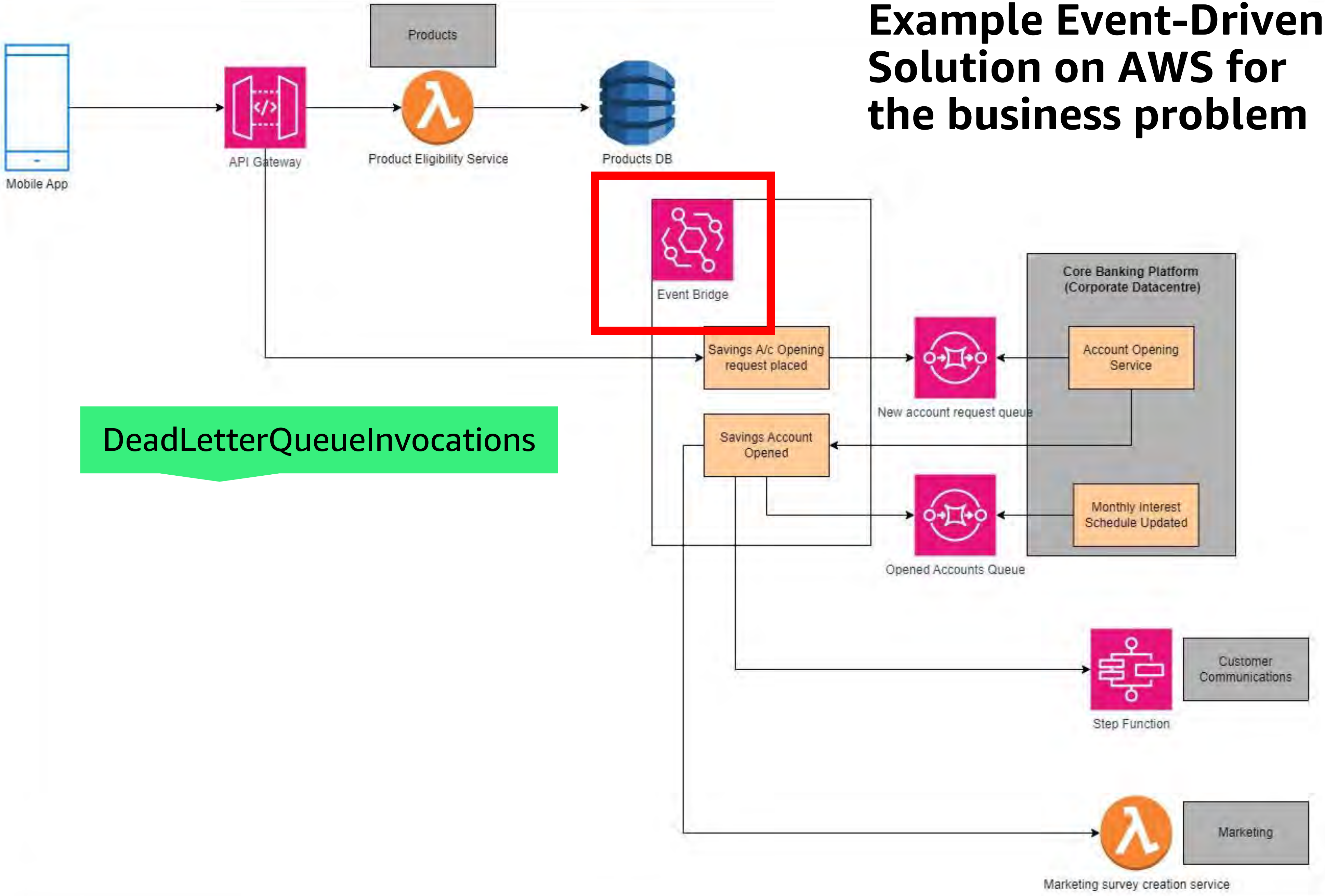
Amazon SQS

Metrics:
ApproximateAgeOfOldestMessage
NumberOfMessageReceived
NumberOfMessagesSent

Amazon EventBridge

Metrics:
DeadLetterQueueInvocations
FailedInvocations
ThrottledRules

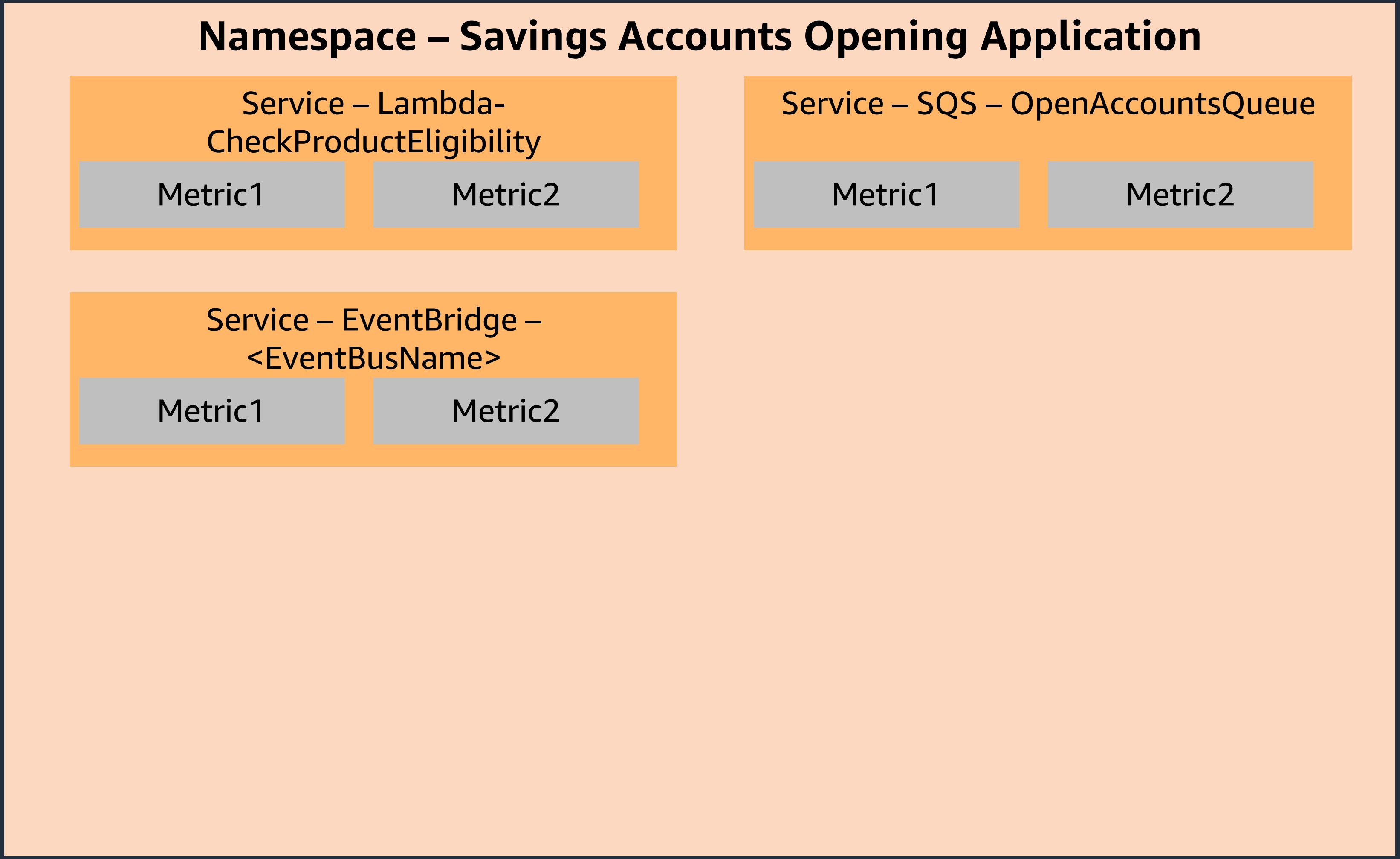
Example Event-Driven Solution on AWS for the business problem



Mobile App
API Gateway
Products
Product Eligibility Service
Products DB
Event Bridge
Savings A/c Opening request placed
Savings Account Opened
New account request queue
Opened Accounts Queue
Core Banking Platform (Corporate Datacentre)
Account Opening Service
Monthly Interest Schedule Updated
Step Function
Customer Communications
Marketing survey creation service
Marketing

DeadLetterQueueInvocations

Metrics – Namespaces and Dimensions



Namespace – Savings Accounts Opening Application

Service – Lambda-CheckProductEligibility

Metric1 Metric2

Service – SQS – OpenAccountsQueue

Metric1 Metric2

Service – EventBridge – <EventBusName>

Metric1 Metric2

Creating custom metrics

Application, business, and operations metrics

Built-in metrics often not enough

- **What about business / customer metrics?**

  Measure application performance against business goals

  Revenue, sign-ups, tracks uploaded, perceived latency, page views, etc.

- **How operationally stable is the application?**

  Continuous integration/deployment feedback time, mean time between failure/recovery, number of on-call pages and time to resolution, etc.

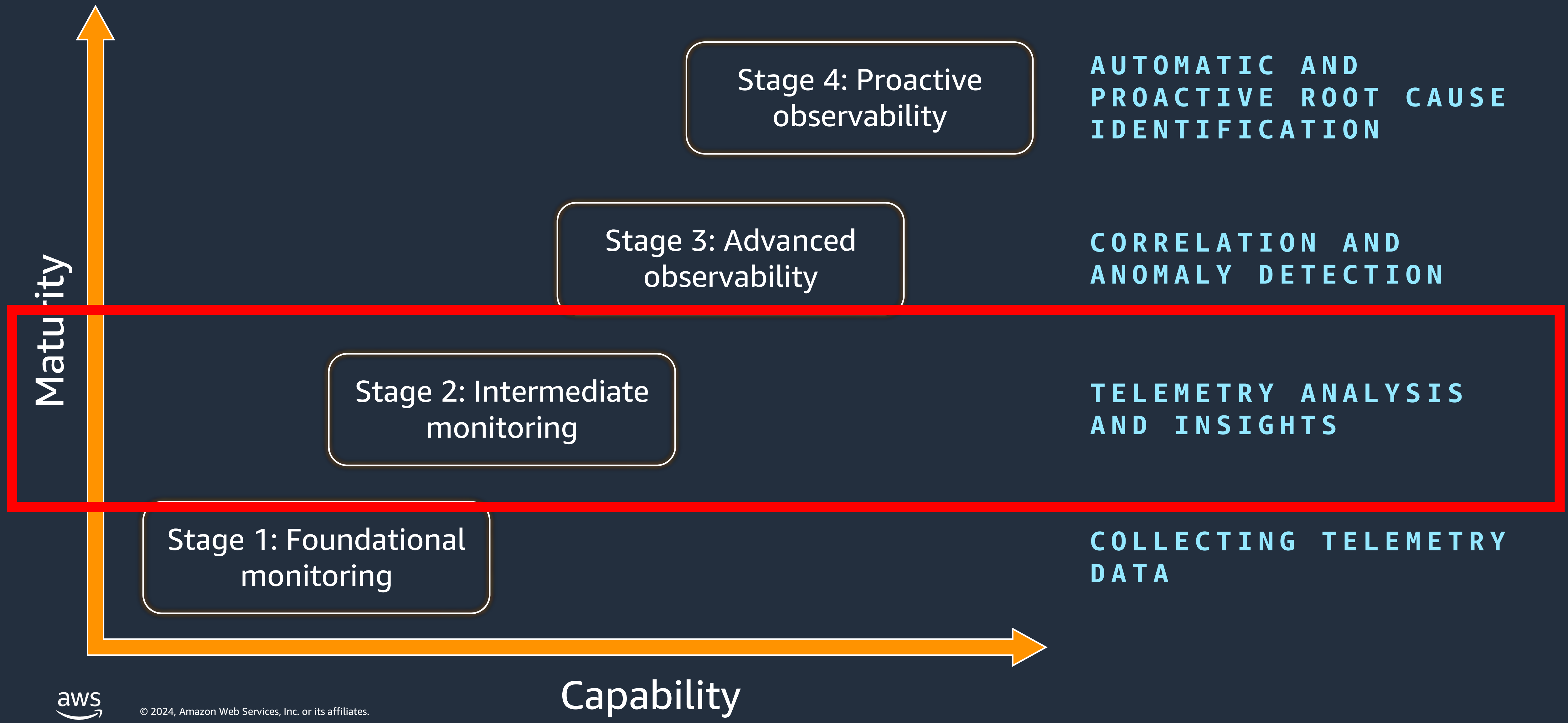
- **What about caught errors, warnings?**

  Caught exceptions are not counted as Errors on AWS Lambda.

- **What if I want to use other dimensions?**

  User ID, category, item, tags, environment, etc.

AWS observability maturity model


Maturity
Stage 4: Proactive observability
Stage 3: Advanced observability
Stage 2: Intermediate monitoring
Stage 1: Foundational monitoring
Capability

AUTOMATIC AND PROACTIVE ROOT CAUSE IDENTIFICATION

CORRELATION AND ANOMALY DETECTION

TELEMETRY ANALYSIS AND INSIGHTS

COLLECTING TELEMETRY DATA

Lambda Insights

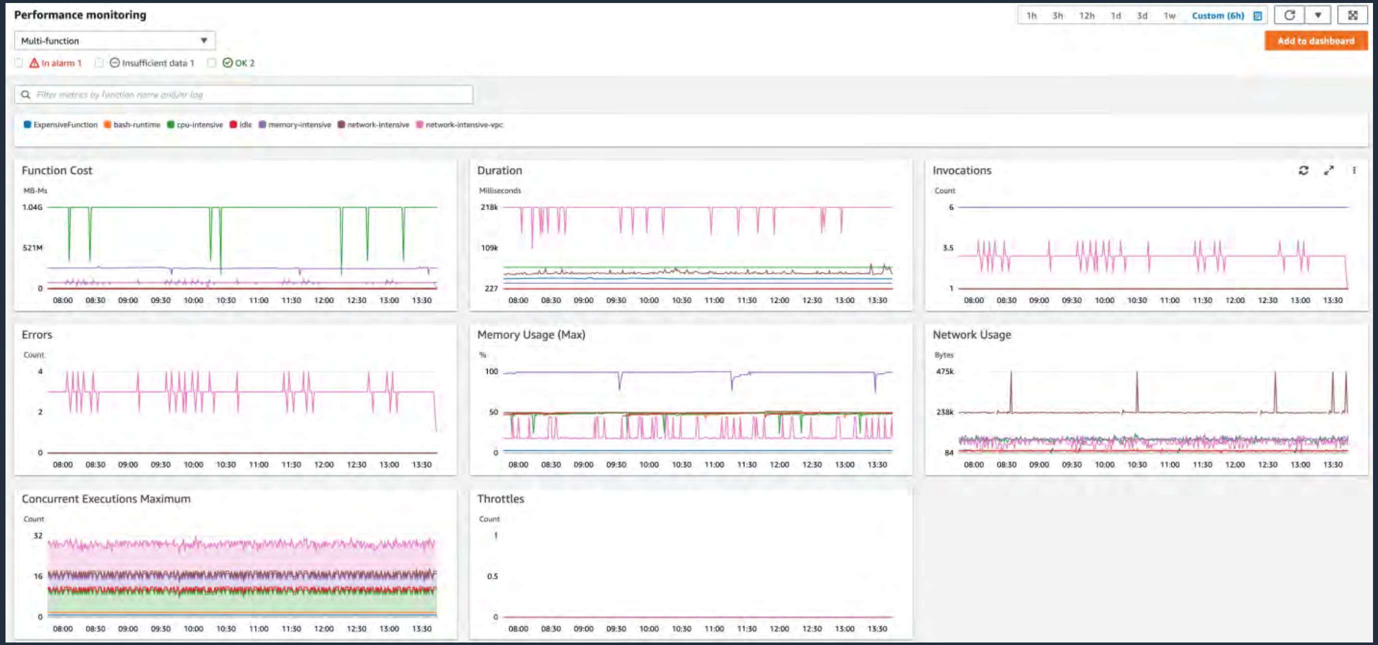
Lambda insights

Monitor, troubleshoot, and optimize the performance of AWS Lambda functions

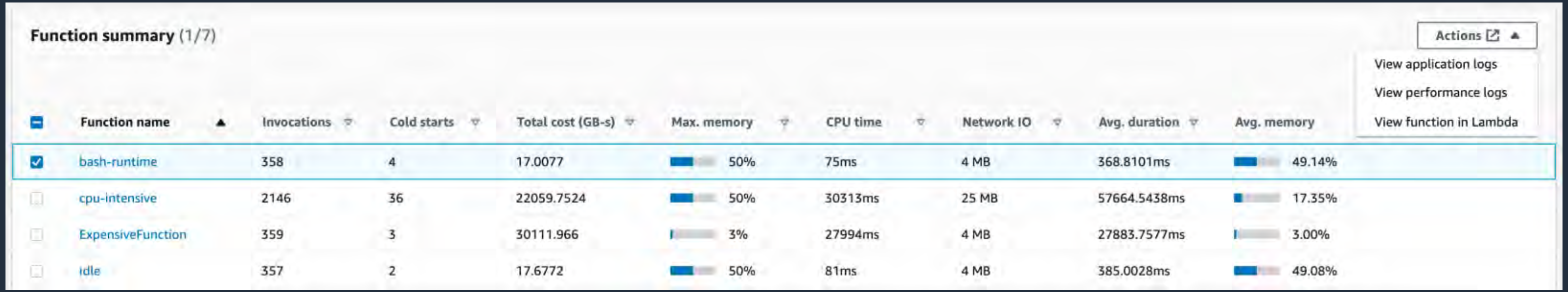
Use Cases

- Identify function issues such as memory leaks
- Identify high-cost functions
- Identify performance changes caused by new function versions
- Understand latency drivers in functions

Lambda insights



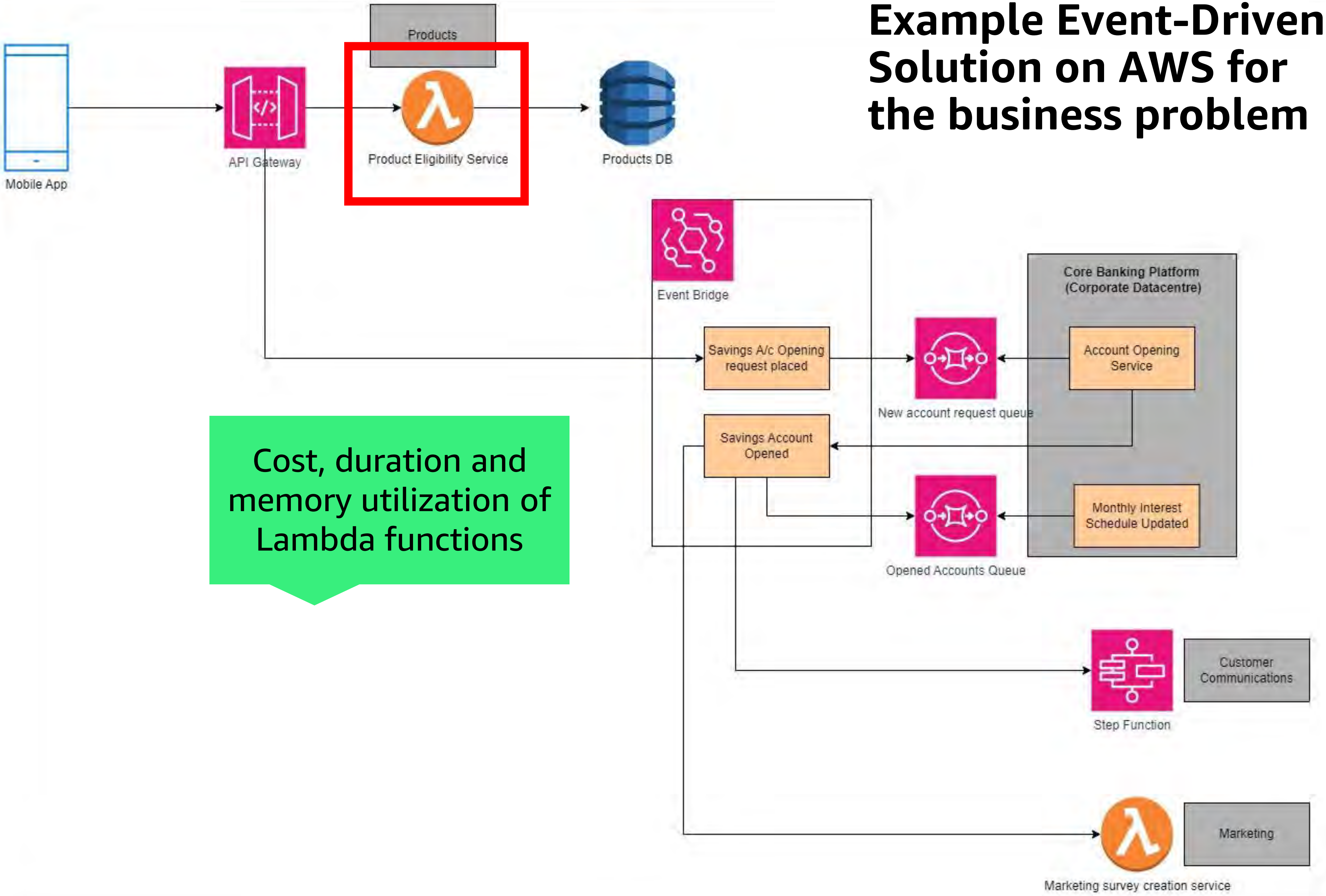
- Aggregated view across multiple Lambda functions as well as single function views
- Search by “name” or “tag”
- Optimize Lambda function utilization
- Predefined performance metrics


Function summary (1/7)

Actions
- View application logs
- View performance logs
- View function in Lambda

☐	Function name	Invocations	Cold starts	Total cost (GB-s)	Max. memory	CPU time	Network IO	Avg. duration	Avg. memory
☑	bash-runtime	358	4	17.0077	50%	75ms	4 MB	368.8101ms	49.14%
☐	cpu-intensive	2146	36	22059.7524	50%	30313ms	25 MB	57664.5438ms	17.35%
☐	ExpensiveFunction	359	3	30111.966	3%	27994ms	4 MB	27883.7577ms	3.00%
☐	idle	357	2	17.6772	50%	81ms	4 MB	385.0028ms	49.08%

Example Event-Driven Solution on AWS for the business problem


Products
Mobile App
API Gateway
Product Eligibility Service
Products DB
Event Bridge
Core Banking Platform (Corporate Datacentre)
Savings A/c Opening request placed
New account request queue
Account Opening Service
Savings Account Opened
Monthly Interest Schedule Updated
Opened Accounts Queue
Step Function
Customer Communications
Marketing
Marketing survey creation service

Cost, duration and memory utilization of Lambda functions

Structured and Centralized Logging

CloudWatch Logs

API Gateway REST API Logging

2 Levels of logging, ERROR and INFO

Optionally log method request/body content

Set globally in stage, or override per method

Metric Filters

Build metrics based on log filters, e.g: “count of 4xx errors.”

Jump to logs that generated metrics

Export logs to Amazon OpenSearch or S3

Explore with Kibana or Athena/QuickSight respectively

Lambda Logging

Logging directly what is emitted from your code to `stdout`, e.g: your language’s equivalent of `console.log()` - basic request information included

via `PutMetric` API or `Embedded Metrics Format (EMF)` of JSON structured logs. logs include invocation information

Error occurred during init or restore phase are also sent to CloudWatch Logs.

Standard structured logging

```
message =
{
        "PrinceInCart": 100,
        "QuantityInCart": 2,
        "ProductId": "a23390f",
        "CategoryId": "bca4cec1",
        "UserId": "31ba3930",
        "CartId": "58dd189f",
        "Environment": "prod",
        "LogLevel": "INFO",
        "Timestamp": "2020-01-01 00:00:00.000000",
        "Message": "Product added to cart"
}
```

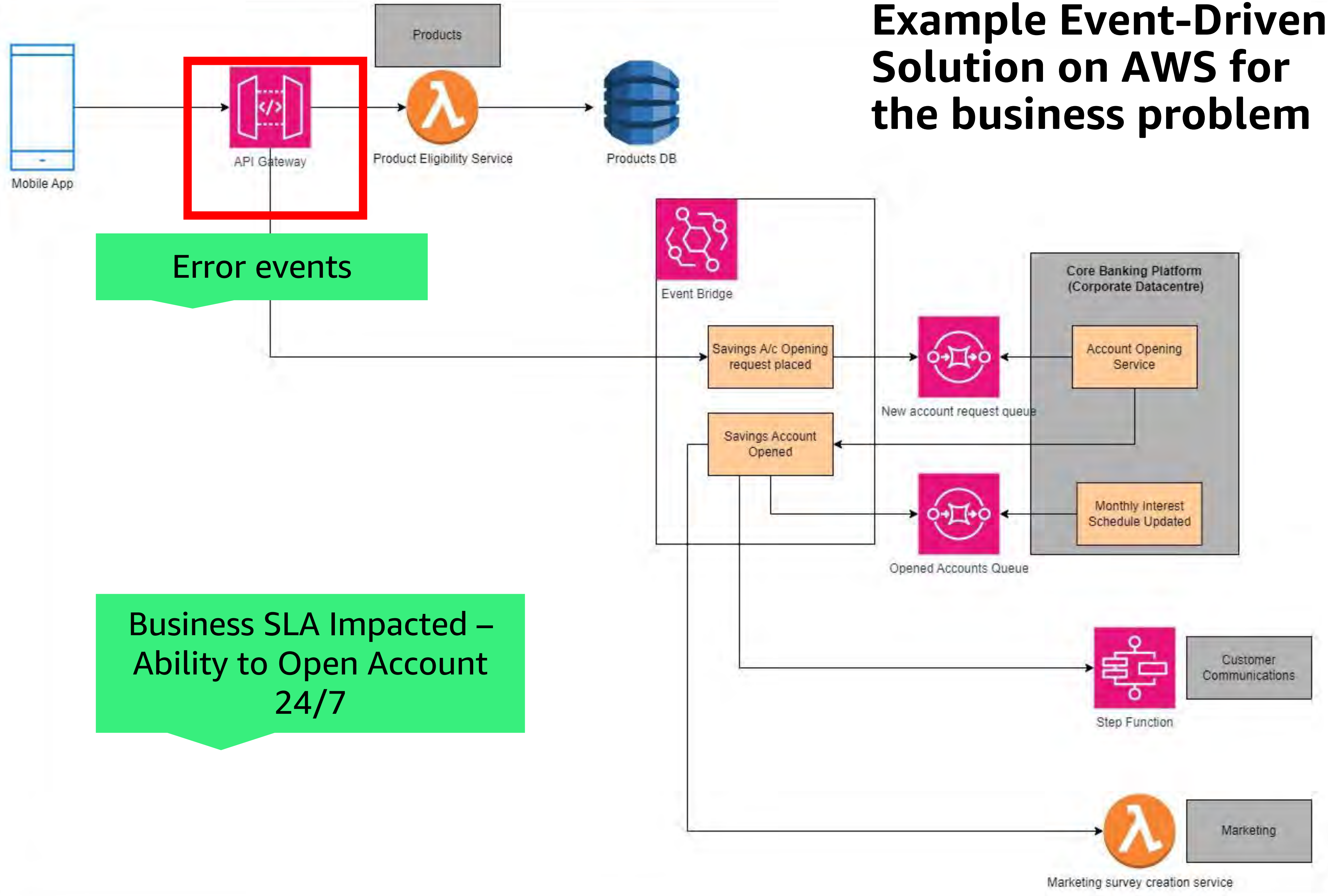
```
# Python
print(json.dumps(message))


# JavaScript
console.log(JSON.stringify(message))


# Go
enc := json.NewEncoder(os.Stdout)
enc.Encode(message)


# Java
log.info(objectMapper.writeObjectAsString(message));
```

Example Event-Driven Solution on AWS for the business problem


Products
Mobile App
API Gateway
Product Eligibility Service
Products DB
Event Bridge
Savings A/c Opening request placed
Savings Account Opened
New account request queue
Opened Accounts Queue
Core Banking Platform (Corporate Datacentre)
Account Opening Service
Monthly Interest Schedule Updated
Step Function
Customer Communications
Marketing
Marketing survey creation service

Error events

Business SLA Impacted – Ability to Open Account 24/7

Query CloudWatch Logs Insights

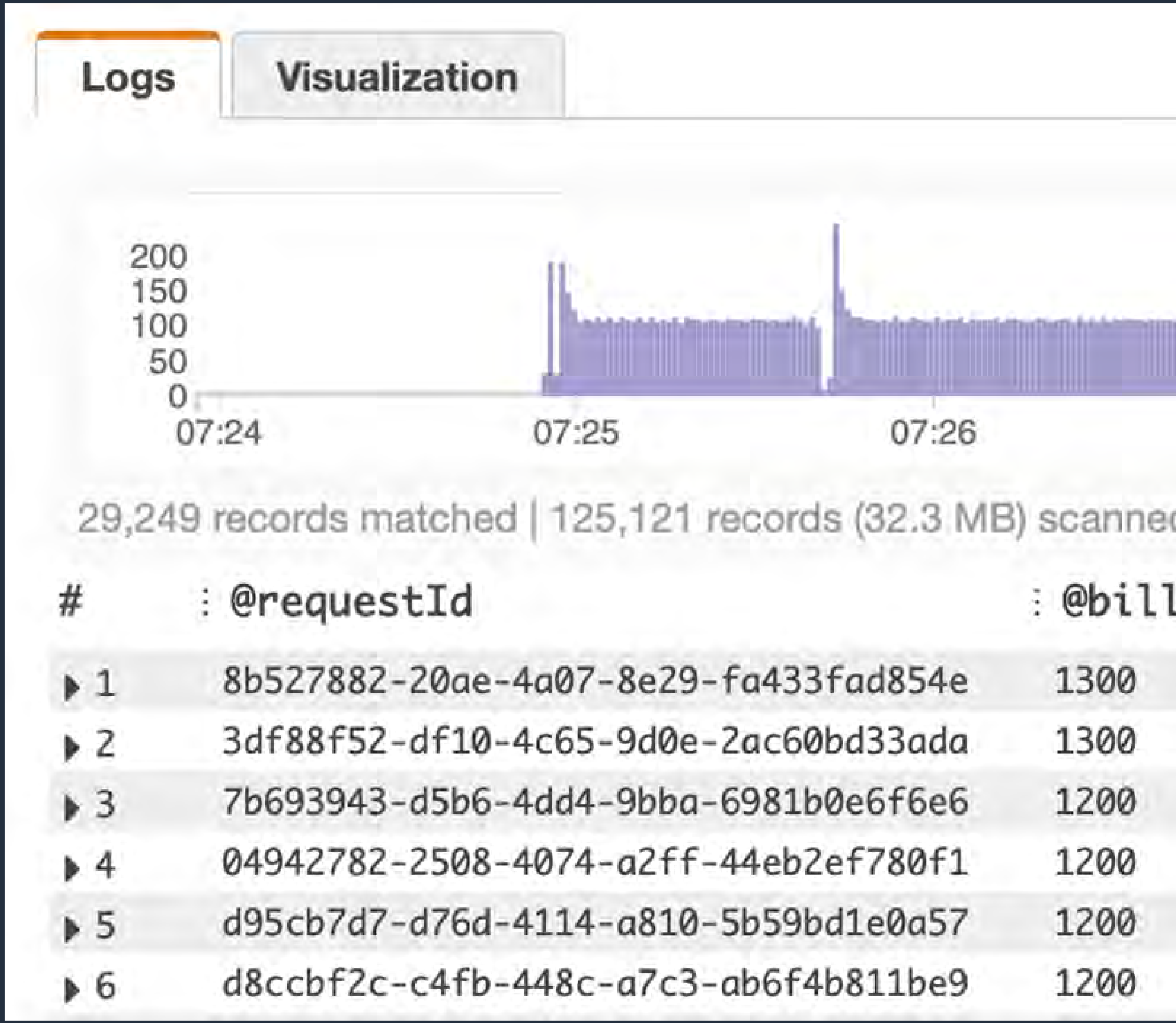
Top 100 most expensive executions

```
filter @type = "REPORT"

| fields @requestId, @billedDuration

| sort by @billedDuration desc

| limit 100
```



29,249 records matched | 125,121 records (32.3 MB) scanne

#	@requestId	@bill
▸ 1	8b527882-20ae-4a07-8e29-fa433fad854e	1300
▸ 2	3df88f52-df10-4c65-9d0e-2ac60bd33ada	1300
▸ 3	7b693943-d5b6-4dd4-9bba-6981b0e6f6e6	1200
▸ 4	04942782-2508-4074-a2ff-44eb2ef780f1	1200
▸ 5	d95cb7d7-d76d-4114-a810-5b59bd1e0a57	1200
▸ 6	d8ccbf2c-c4fb-448c-a7c3-ab6f4b811be9	1200

Query CloudWatch Logs Insights

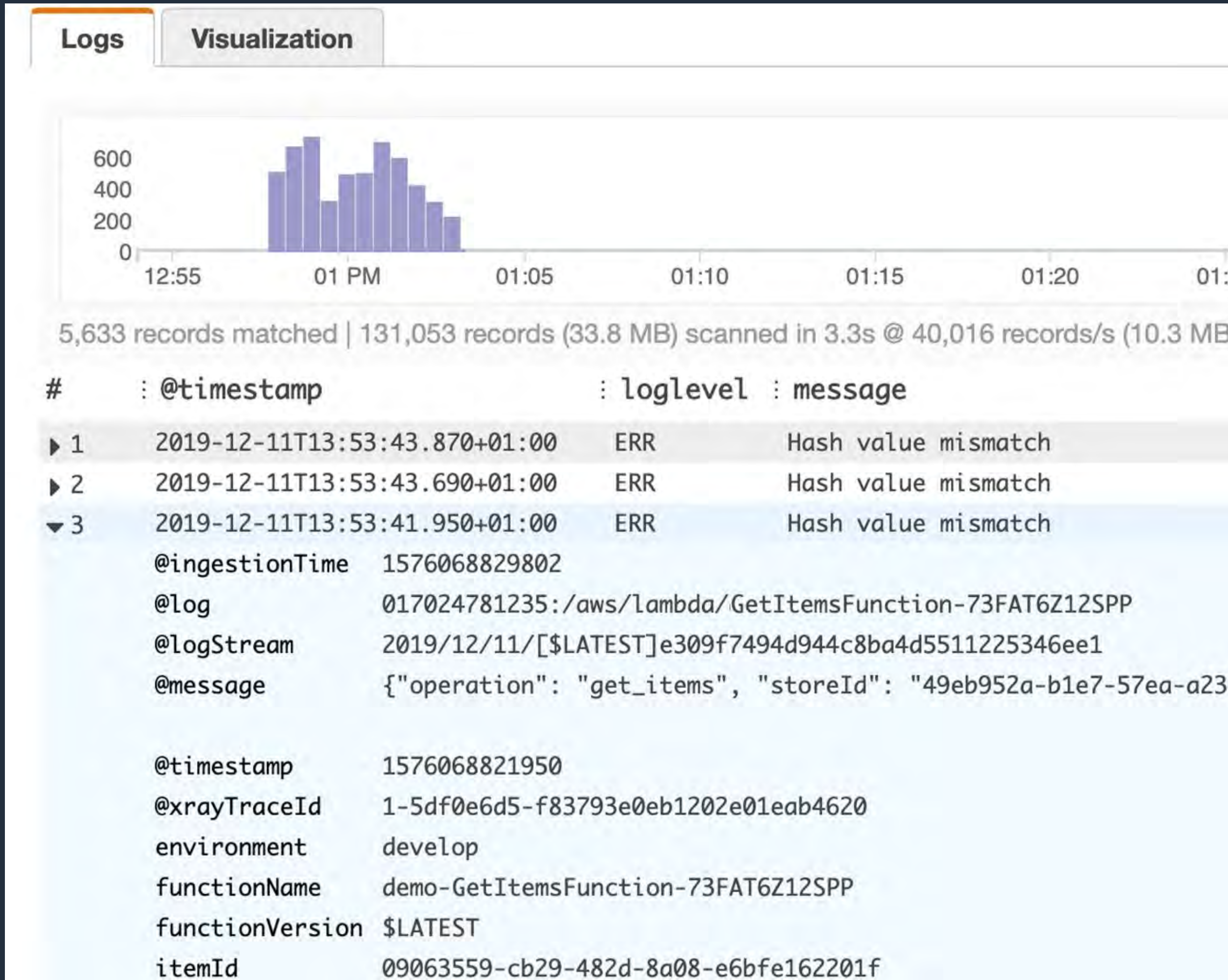
Get the last 100 error messages

```
fields Timestamp, LogLevel, Message

| filter LogLevel == "ERR"

| sort @timestamp desc

| limit 100
```



```
5,633 records matched | 131,053 records (33.8 MB) scanned in 3.3s @ 40,016 records/s (10.3 MB

#      : @timestamp                       : loglevel  : message
▸ 1      2019-12-11T13:53:43.870+01:00     ERR          Hash value mismatch
▸ 2      2019-12-11T13:53:43.690+01:00     ERR          Hash value mismatch
▾ 3      2019-12-11T13:53:41.950+01:00     ERR          Hash value mismatch
         @ingestionTime   1576068829802
         @log             017024781235:/aws/lambda/GetItemsFunction-73FAT6Z12SPP
         @logStream       2019/12/11/[$LATEST]e309f7494d944c8ba4d5511225346ee1
         @message         {"operation": "get_items", "storeId": "49eb952a-b1e7-57ea-a23

         @timestamp       1576068821950
         @xrayTraceId     1-5df0e6d5-f83793e0eb1202e01eab4620
         environment      develop
         functionName     demo-GetItemsFunction-73FAT6Z12SPP
         functionVersion  $LATEST
         itemId           09063559-cb29-482d-8a08-e6bfe162201f
```

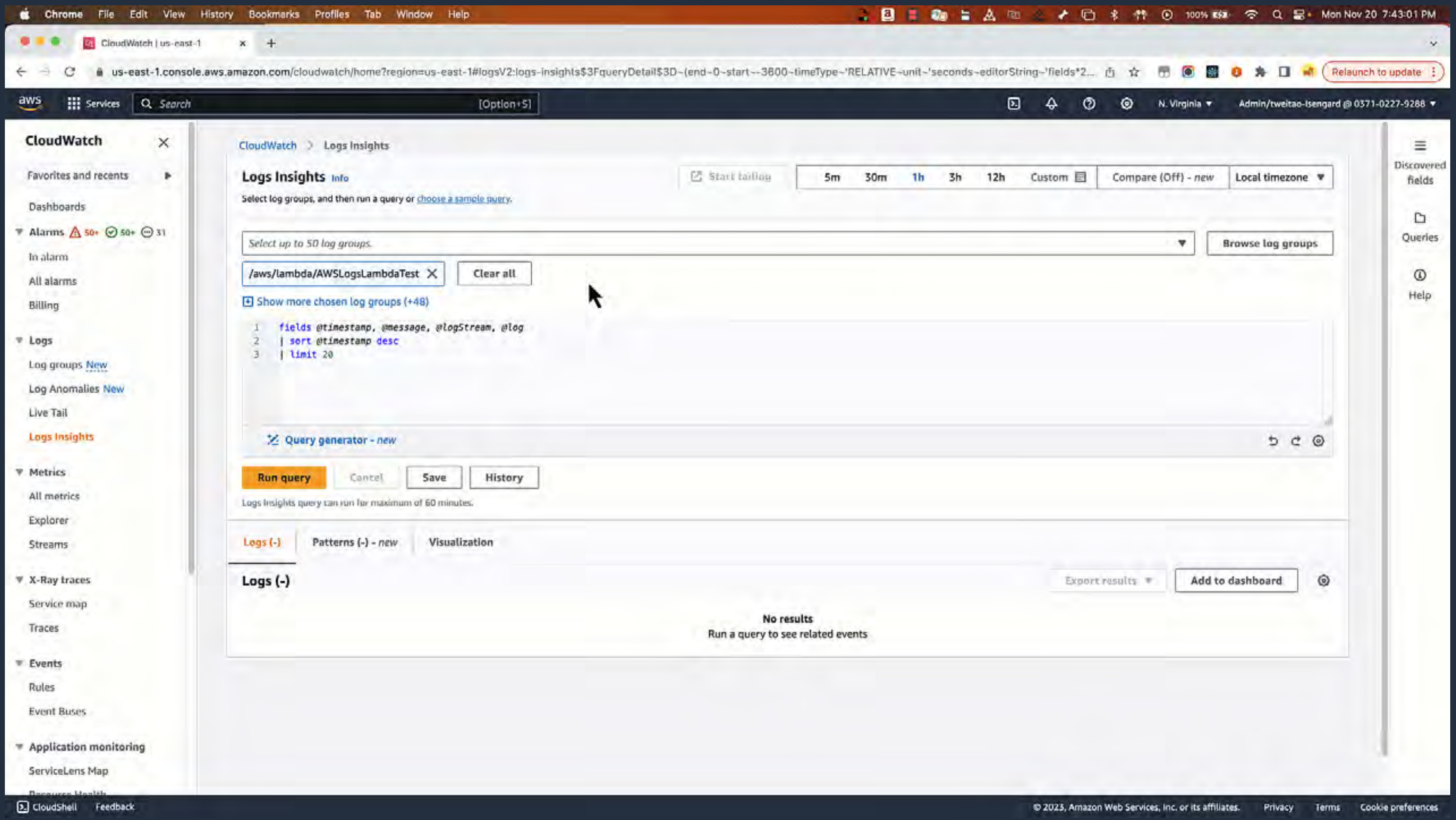
AI-powered natural language query generation PREVIEW

Easy getting started: Generate queries to interact with your Logs and Metrics by asking questions in natural language

Develop query expertise*:* Provides line by line explanation of the generated query to help you learn the syntax

Iterative deep dives*:* Update existing queries with natural language instructions for guided query iteration

Preview available in US East (Virginia) and US West (Oregon)



Logs Insights Info

Select log groups, and then run a query or choose a sample query.

Select up to 50 log groups.

/aws/lambda/AWSLogsLambdaTest

```
fields @timestamp, @message, @logStream, @log
| sort @timestamp desc
| limit 20
```

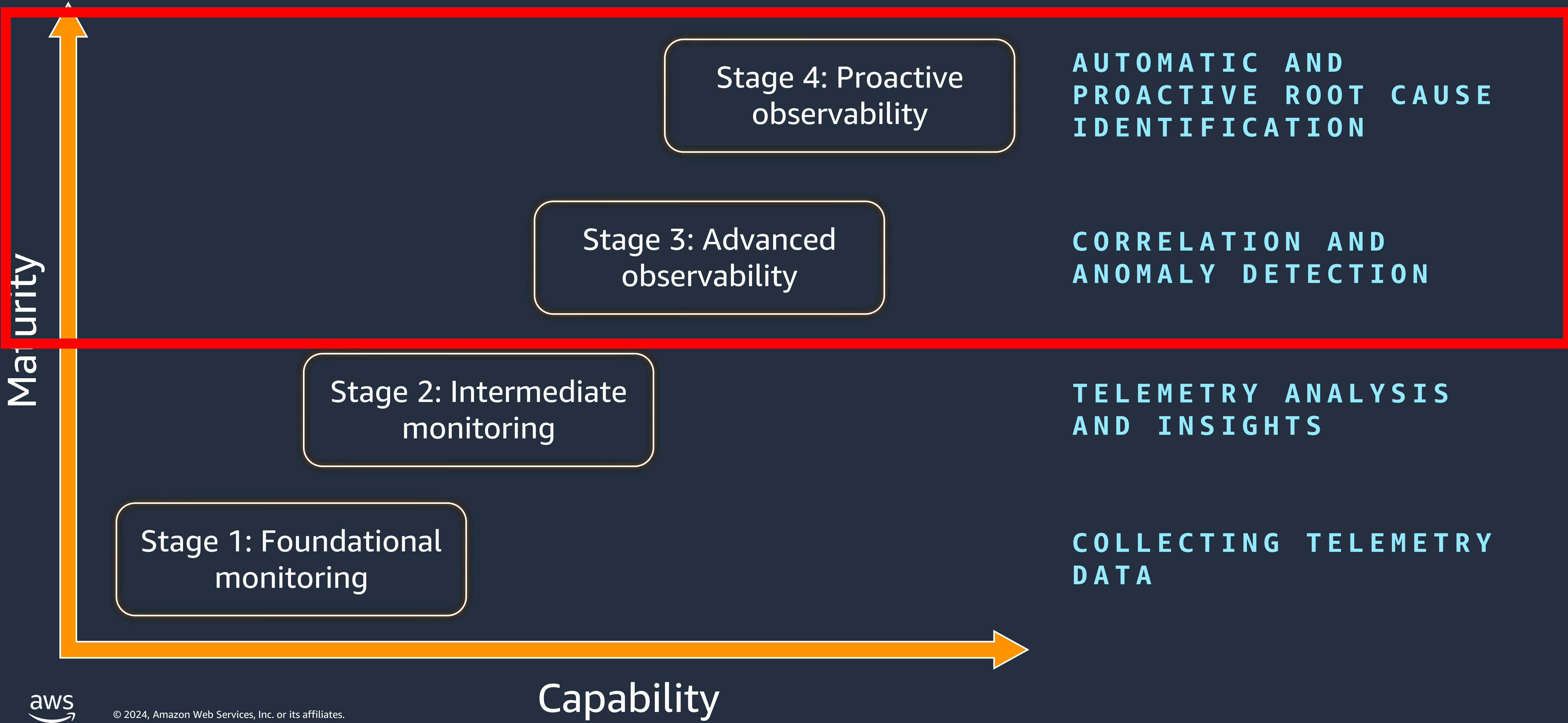
Logs Insights query can run for maximum of 60 minutes.

Logs (-)

No results
Run a query to see related events

aws

AWS observability maturity model


Stage 4: Proactive observability
Stage 3: Advanced observability
Stage 2: Intermediate monitoring
Stage 1: Foundational monitoring
Maturity
Capability

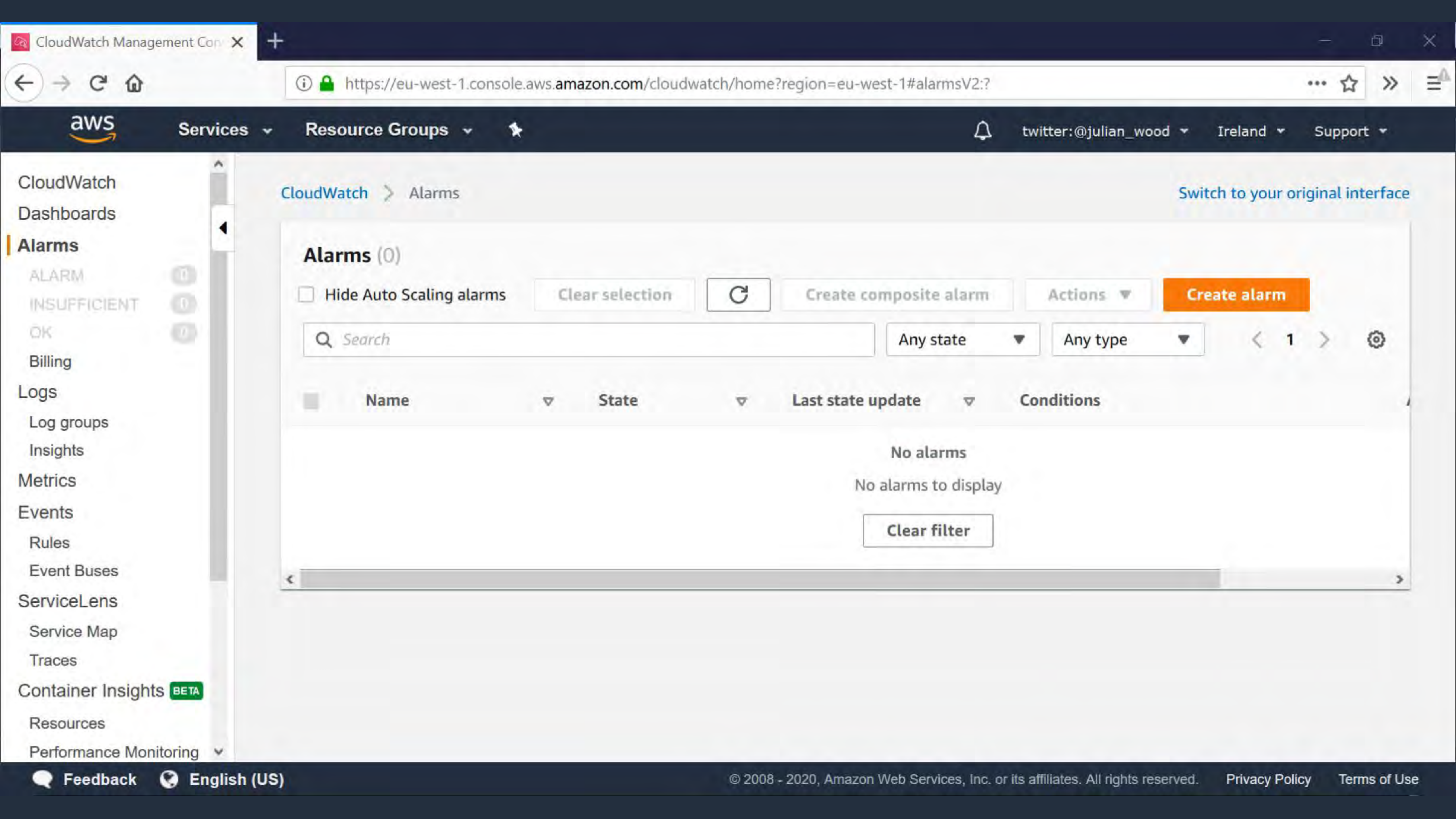
AUTOMATIC AND PROACTIVE ROOT CAUSE IDENTIFICATION

CORRELATION AND ANOMALY DETECTION

TELEMETRY ANALYSIS AND INSIGHTS

COLLECTING TELEMETRY DATA

Creating alerts

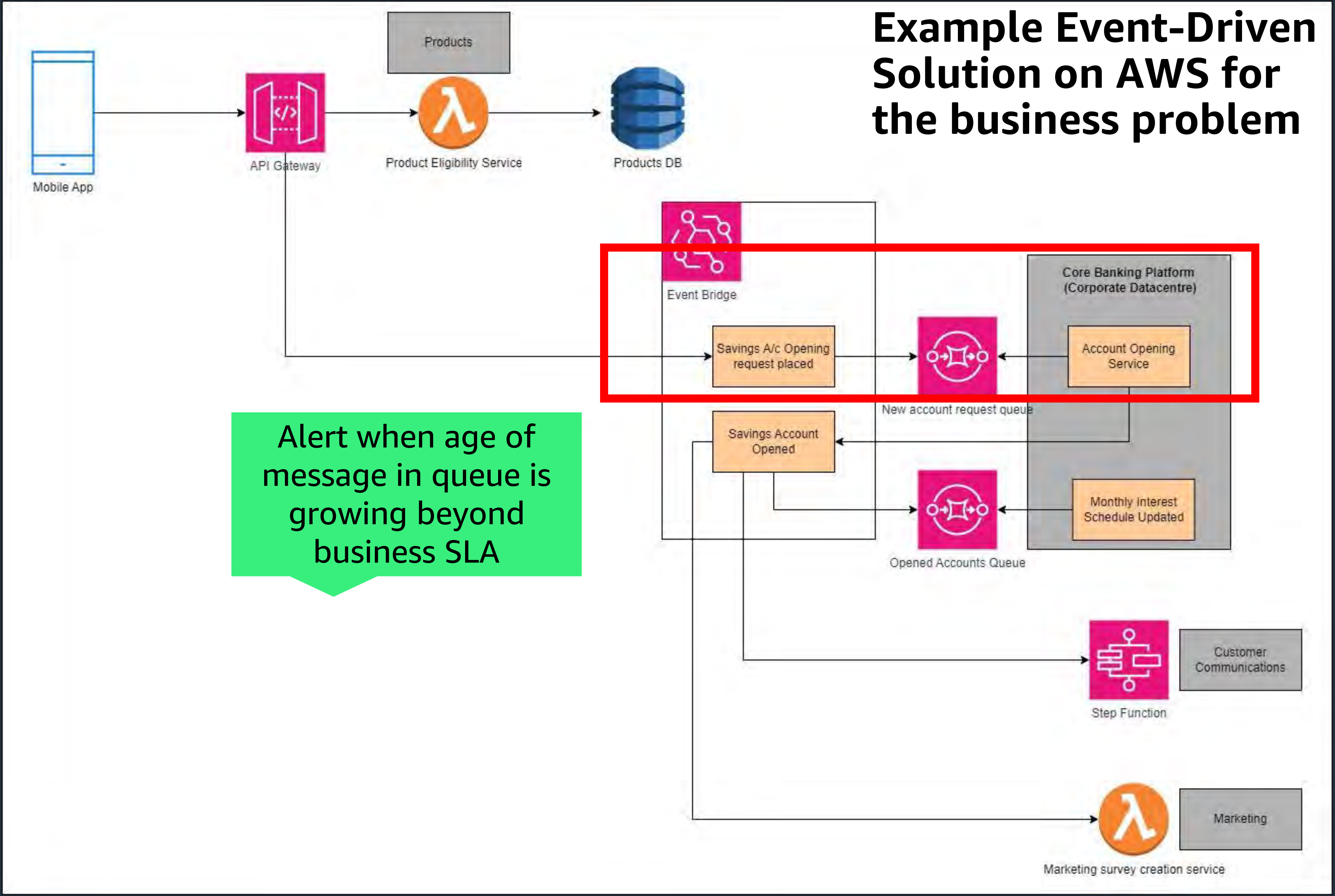


No alarms

No alarms to display

© 2008 - 2020, Amazon Web Services, Inc. or its affiliates. All rights reserved.

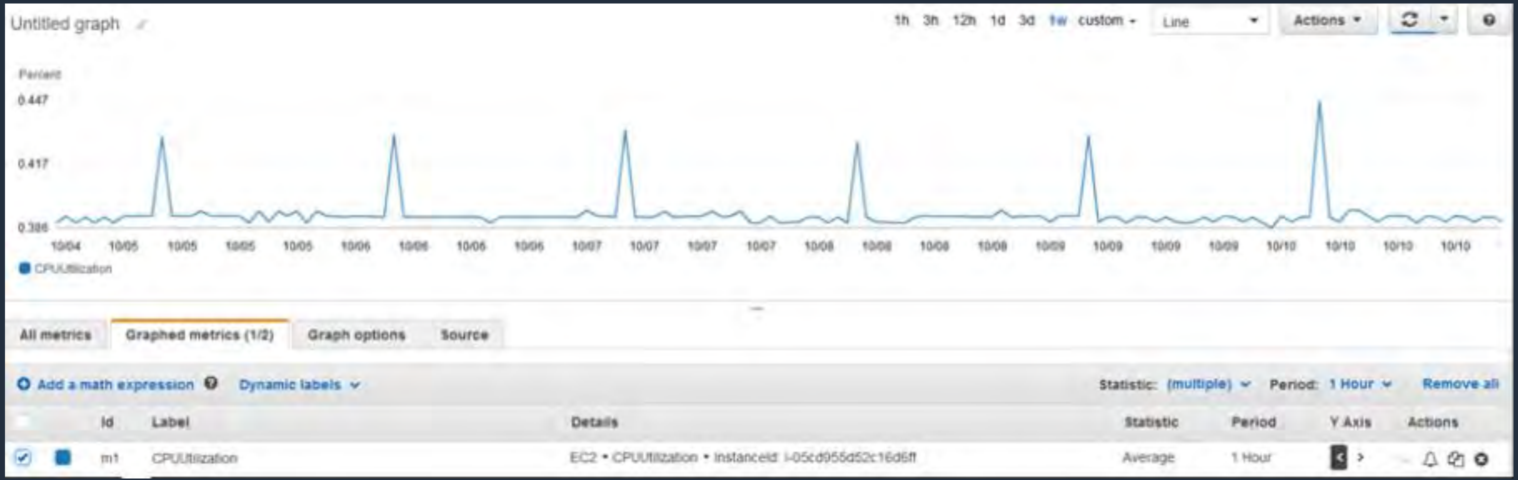
Example Event-Driven Solution on AWS for the business problem



Products
Mobile App
API Gateway
Product Eligibility Service
Products DB
Event Bridge
Core Banking Platform (Corporate Datacentre)
Savings A/c Opening request placed
Account Opening Service
New account request queue
Savings Account Opened
Monthly Interest Schedule Updated
Opened Accounts Queue
Step Function
Customer Communications
Marketing
Marketing survey creation service

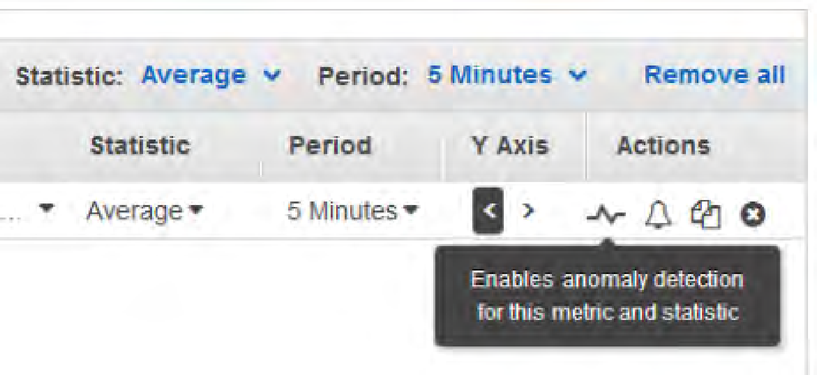
Alert when age of message in queue is growing beyond business SLA

Use CloudWatch anomaly detection alarms



Statistic: (multiple) Period: 1 Hour

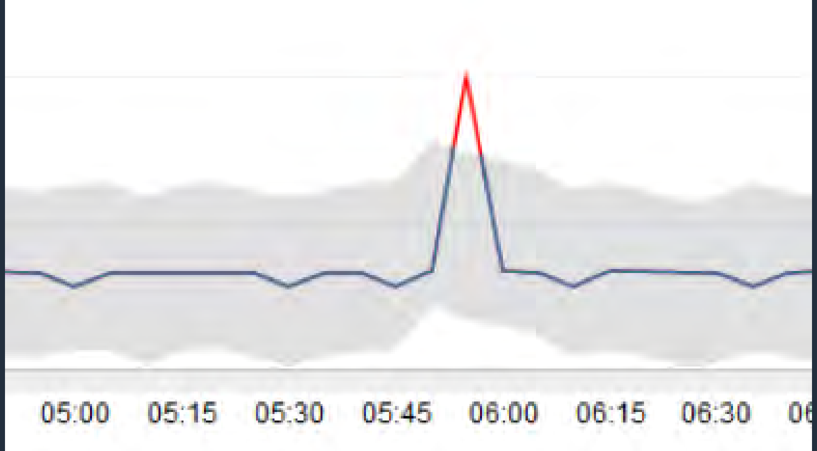
		Id	Label	Details	Statistic	Period	Y Axis	Actions
☑	🟦	m1	CPUUtilization	EC2 • CPUUtilization • InstanceId: i-05cd955d52c16d6ff	Average	1 Hour	< >	

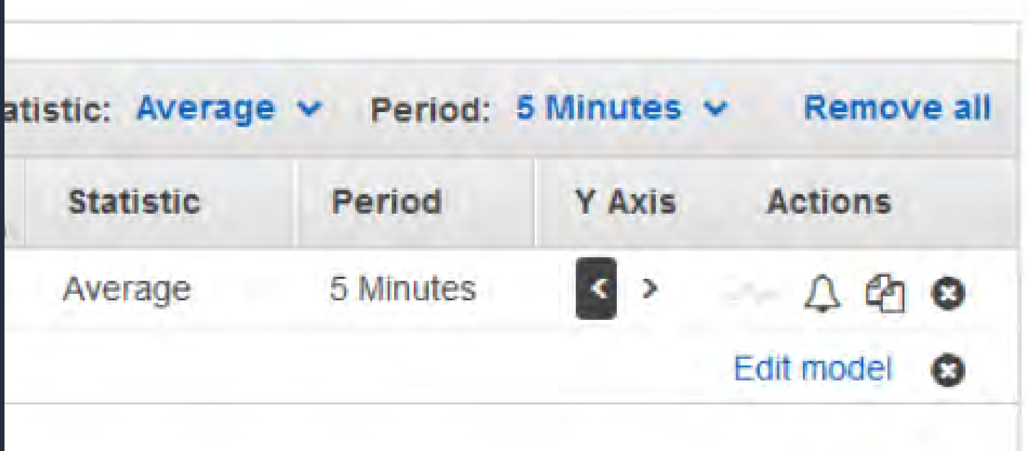


Statistic: Average Period: 5 Minutes

Statistic	Period	Y Axis	Actions
Average	5 Minutes	< >	

Enables anomaly detection for this metric and statistic

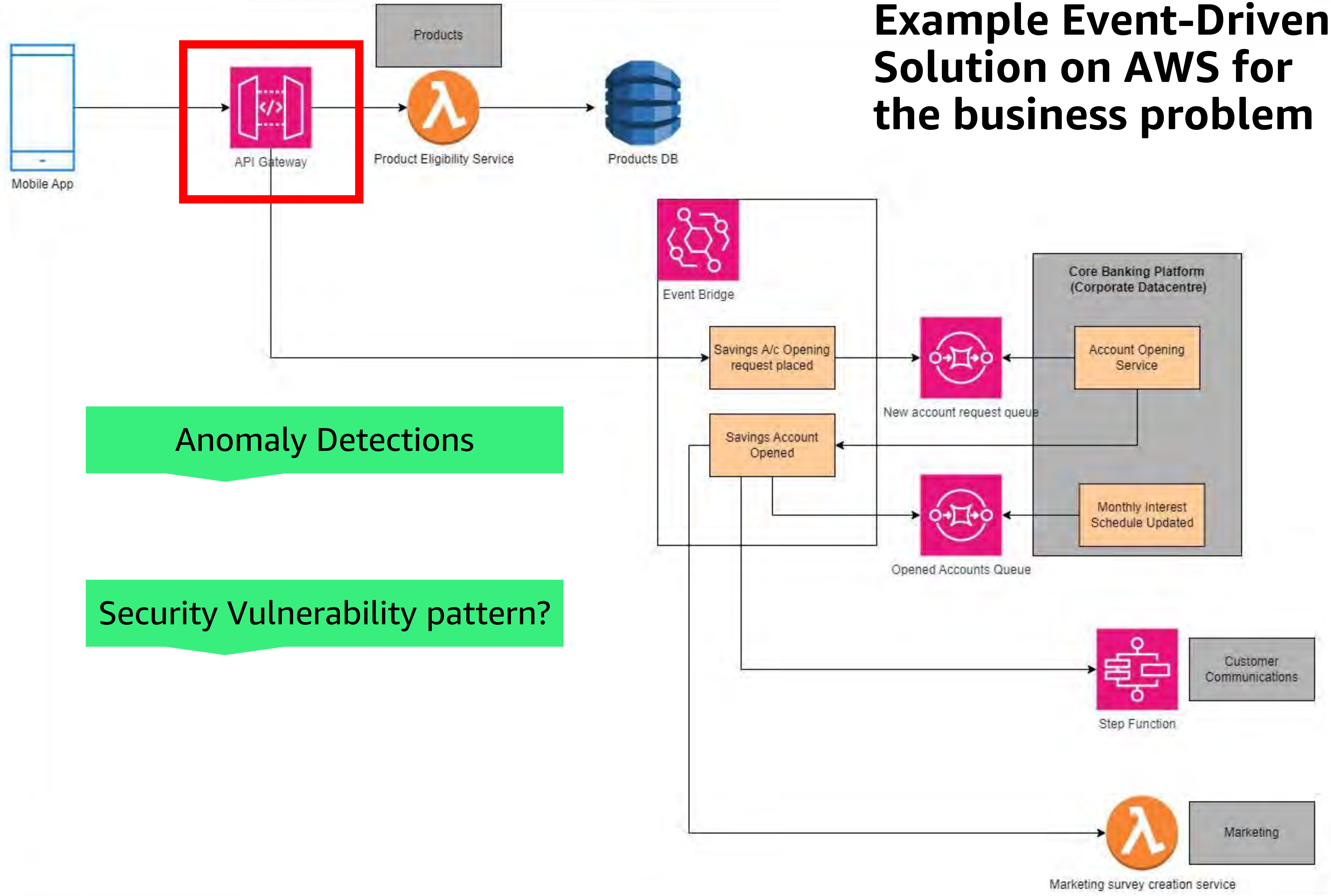




atistic: Average Period: 5 Minutes

Statistic	Period	Y Axis	Actions
Average	5 Minutes	< >	

Example Event-Driven Solution on AWS for the business problem


Mobile App
API Gateway
Products
Product Eligibility Service
Products DB
Event Bridge
Savings A/c Opening request placed
Savings Account Opened
New account request queue
Opened Accounts Queue
Core Banking Platform (Corporate Datacentre)
Account Opening Service
Monthly Interest Schedule Updated
Step Function
Customer Communications
Marketing
Marketing survey creation service

Anomaly Detections

Security Vulnerability pattern?

Log Patterns

Challenges with Logs Analysis

Too much data

Ever-increasing volume and diversity of log data

What changed?

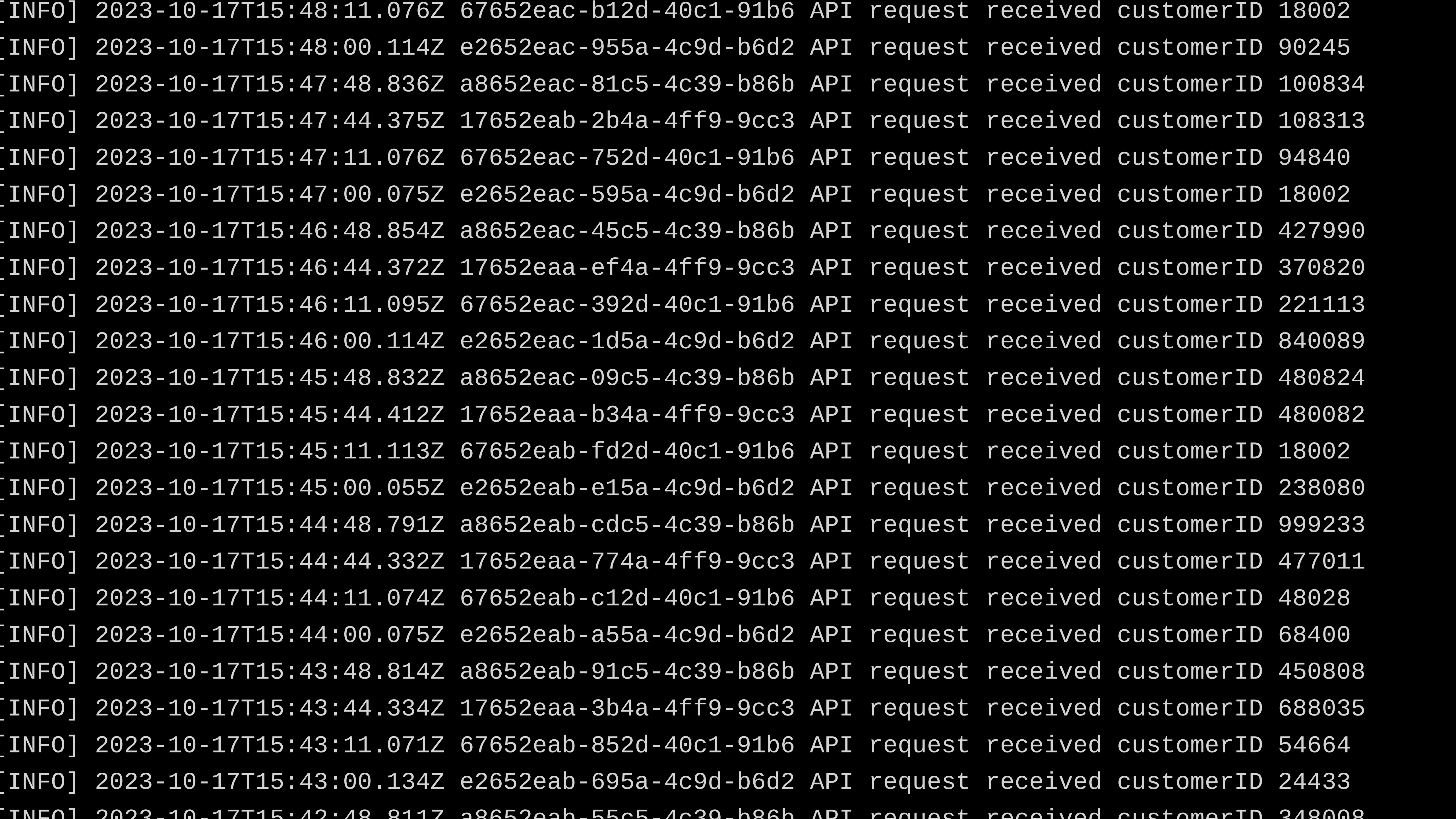
Difficult to identify how logs have changed over time

Proactive detection

Identifying unusual changes in application logs

Unknown unknowns

Monitoring for unforeseen issues

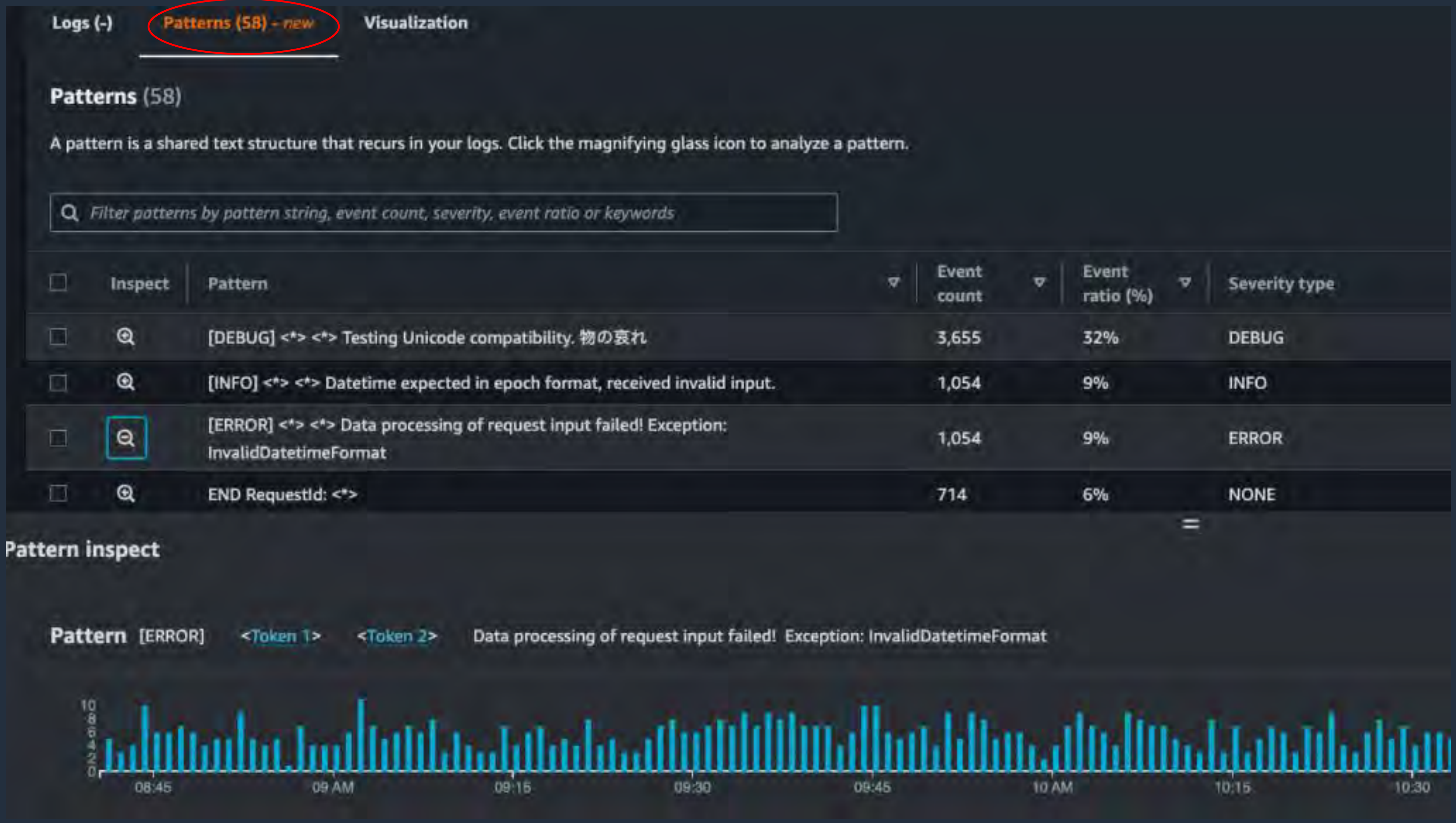


```
[INFO] 2023-10-17T15:48:11.076Z 67652eac-b12d-40c1-91b6 API request received customerID 18002
[INFO] 2023-10-17T15:48:00.114Z e2652eac-955a-4c9d-b6d2 API request received customerID 90245
[INFO] 2023-10-17T15:47:48.836Z a8652eac-81c5-4c39-b86b API request received customerID 100834
[INFO] 2023-10-17T15:47:44.375Z 17652eab-2b4a-4ff9-9cc3 API request received customerID 108313
[INFO] 2023-10-17T15:47:11.076Z 67652eac-752d-40c1-91b6 API request received customerID 94840
[INFO] 2023-10-17T15:47:00.075Z e2652eac-595a-4c9d-b6d2 API request received customerID 18002
[INFO] 2023-10-17T15:46:48.854Z a8652eac-45c5-4c39-b86b API request received customerID 427990
[INFO] 2023-10-17T15:46:44.372Z 17652eaa-ef4a-4ff9-9cc3 API request received customerID 370820
[INFO] 2023-10-17T15:46:11.095Z 67652eac-392d-40c1-91b6 API request received customerID 221113
[INFO] 2023-10-17T15:46:00.114Z e2652eac-1d5a-4c9d-b6d2 API request received customerID 840089
[INFO] 2023-10-17T15:45:48.832Z a8652eac-09c5-4c39-b86b API request received customerID 480824
[INFO] 2023-10-17T15:45:44.412Z 17652eaa-b34a-4ff9-9cc3 API request received customerID 480082
[INFO] 2023-10-17T15:45:11.113Z 67652eab-fd2d-40c1-91b6 API request received customerID 18002
[INFO] 2023-10-17T15:45:00.055Z e2652eab-e15a-4c9d-b6d2 API request received customerID 238080
[INFO] 2023-10-17T15:44:48.791Z a8652eab-cdc5-4c39-b86b API request received customerID 999233
[INFO] 2023-10-17T15:44:44.332Z 17652eaa-774a-4ff9-9cc3 API request received customerID 477011
[INFO] 2023-10-17T15:44:11.074Z 67652eab-c12d-40c1-91b6 API request received customerID 48028
[INFO] 2023-10-17T15:44:00.075Z e2652eab-a55a-4c9d-b6d2 API request received customerID 68400
[INFO] 2023-10-17T15:43:48.814Z a8652eab-91c5-4c39-b86b API request received customerID 450808
[INFO] 2023-10-17T15:43:44.334Z 17652eaa-3b4a-4ff9-9cc3 API request received customerID 688035
[INFO] 2023-10-17T15:43:11.071Z 67652eab-852d-40c1-91b6 API request received customerID 54664
[INFO] 2023-10-17T15:43:00.134Z e2652eab-695a-4c9d-b6d2 API request received customerID 24433
[INFO] 2023-10-17T15:42:48.811Z a8652eab-55c5-4c39-b86b API request received customerID 348008
```

```
[INFO] 2023-10-17T15:48:11.076Z 67652eac-b12d-40c1-91b6 API request received customerID 18002
[INFO] 2023-10-17T15:48:00.114Z e2652eac-955a-4c9d-b6d2 API request received customerID 90245
[INFO] 2023-10-17T15:47:48.836Z a8652eac-81c5-4c39-b86b API request received customerID 100834
[INFO] 2023-10-17T15:47:44.375Z 17652eab-2b4a-4ff9-9cc3 API request received customerID 108313
[INFO] 2023-10-17T15:47:11.076Z 67652eac-752d-40c1-91b6 API request received customerID 94840
[INFO] 2023-10-17T15:47:00.075Z e2652eac-595a-4c9d-b6d2 API request received customerID 18002
[INFO] 2023-10-17T15:46:48.854Z a8652eac-45c5-4c39-b86b API request received customerID 427990
[INFO] 2023-10-17T15:46:44.372Z 17652eaa-ef4a-4ff9-9cc3 API request received customerID 370820
[INFO] 2023-10-17T15:46:11.095Z 67652eac-392d-40c1-91b6 API request received customerID 221113
[INFO
[INFO
[INFO
[INFO
[INFO] 2023-10-17T15:45:00.055Z e2652eab-e15a-4c9d-b6d2 API request received customerID 238080
[INFO] 2023-10-17T15:44:48.791Z a8652eab-cdc5-4c39-b86b API request received customerID 999233
[INFO] 2023-10-17T15:44:44.332Z 17652eaa-774a-4ff9-9cc3 API request received customerID 477011
[INFO] 2023-10-17T15:44:11.074Z 67652eab-c12d-40c1-91b6 API request received customerID 48028
[INFO] 2023-10-17T15:44:00.075Z e2652eab-a55a-4c9d-b6d2 API request received customerID 68400
[INFO] 2023-10-17T15:43:48.814Z a8652eab-91c5-4c39-b86b API request received customerID 450808
[INFO] 2023-10-17T15:43:44.334Z 17652eaa-3b4a-4ff9-9cc3 API request received customerID 688035
[INFO] 2023-10-17T15:43:11.071Z 67652eab-852d-40c1-91b6 API request received customerID 54664
[INFO] 2023-10-17T15:43:00.134Z e2652eab-695a-4c9d-b6d2 API request received customerID 24433
[INFO] 2023-10-17T15:42:48.811Z a8652eab-55c5-4c39-b86b API request received customerID 348008
```

```
[INFO] <*> <*> API request received customerID <*>
```

Pattern Analysis in Logs Insights



Logs (-)　Patterns (58) - *new*　Visualization

Patterns (58)

A pattern is a shared text structure that recurs in your logs. Click the magnifying glass icon to analyze a pattern.

Filter patterns by pattern string, event count, severity, event ratio or keywords

□	Inspect	Pattern	Event count	Event ratio (%)	Severity type
□	🔍	[DEBUG] <*> <*> Testing Unicode compatibility. 物の哀れ	3,655	32%	DEBUG
□	🔍	[INFO] <*> <*> Datetime expected in epoch format, received invalid input.	1,054	9%	INFO
□	🔍	[ERROR] <*> <*> Data processing of request input failed! Exception: InvalidDatetimeFormat	1,054	9%	ERROR
□	🔍	END RequestId: <*>	714	6%	NONE

Pattern inspect

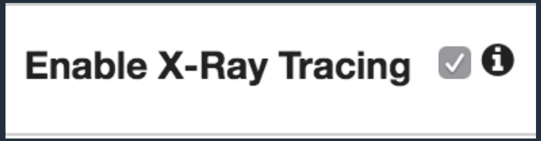
Pattern [ERROR] <Token 1> <Token 2> Data processing of request input failed! Exception: InvalidDatetimeFormat

Tracing

AWS X-Ray

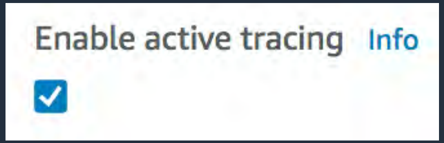
End-to-end view of requests flowing through an application

Lambda: instruments incoming requests for all supported languages and can capture calls made in code



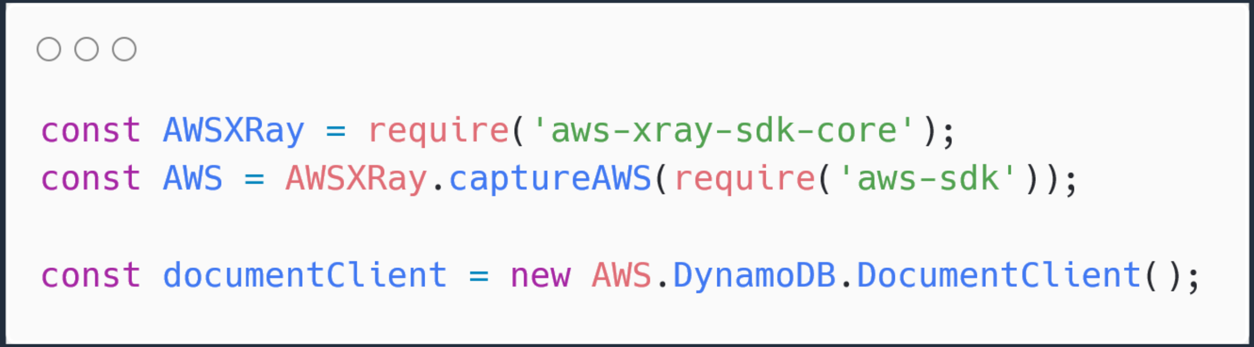
Enable X-Ray Tracing ☑

API Gateway: inserts a tracing header into HTTP calls as well as reports data back to X-Ray itself



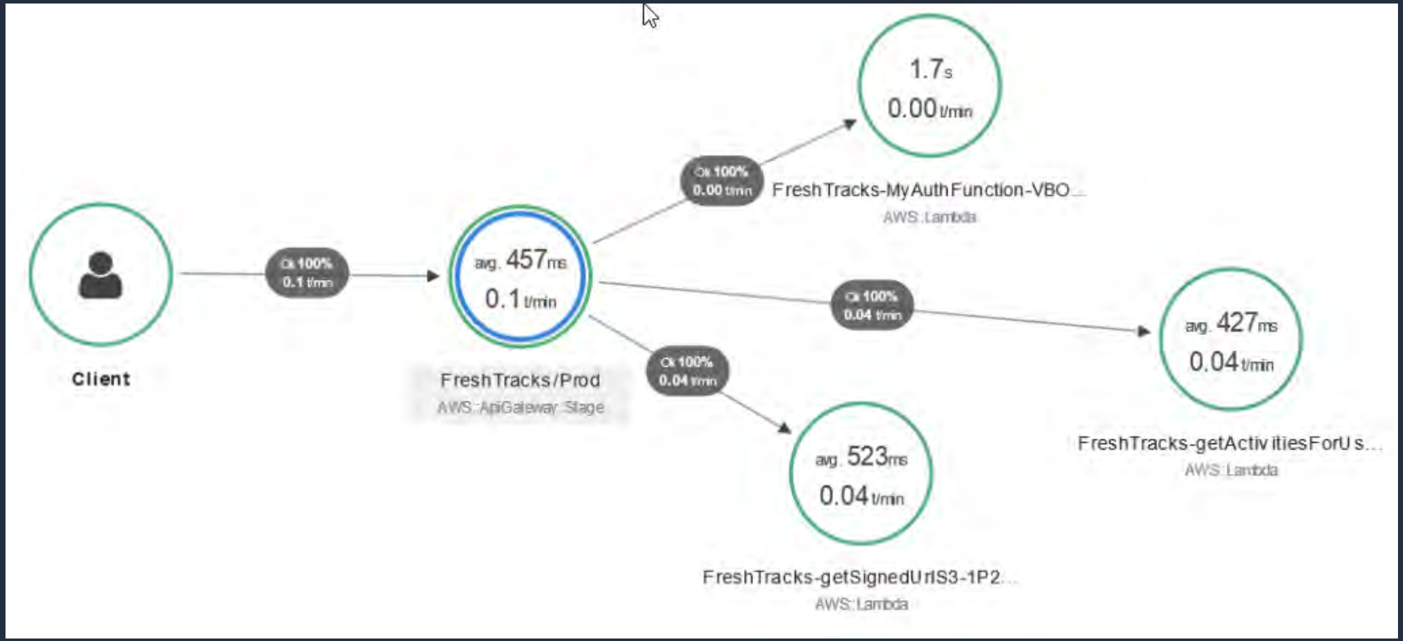
Enable active tracing Info

☑

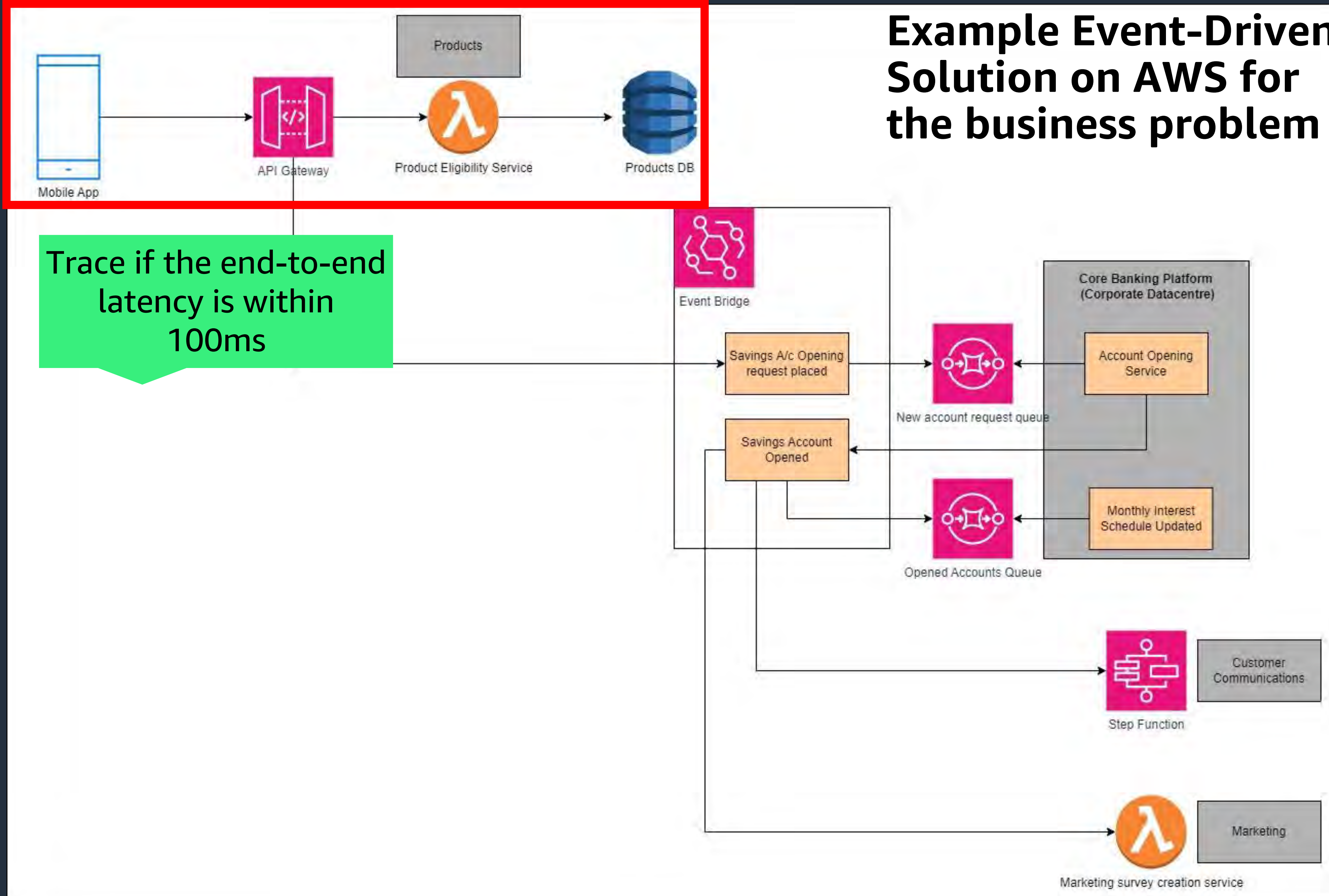


```
const AWSXRay = require('aws-xray-sdk-core');
const AWS = AWSXRay.captureAWS(require('aws-sdk'));

const documentClient = new AWS.DynamoDB.DocumentClient();
```



Example Event-Driven Solution on AWS for the business problem



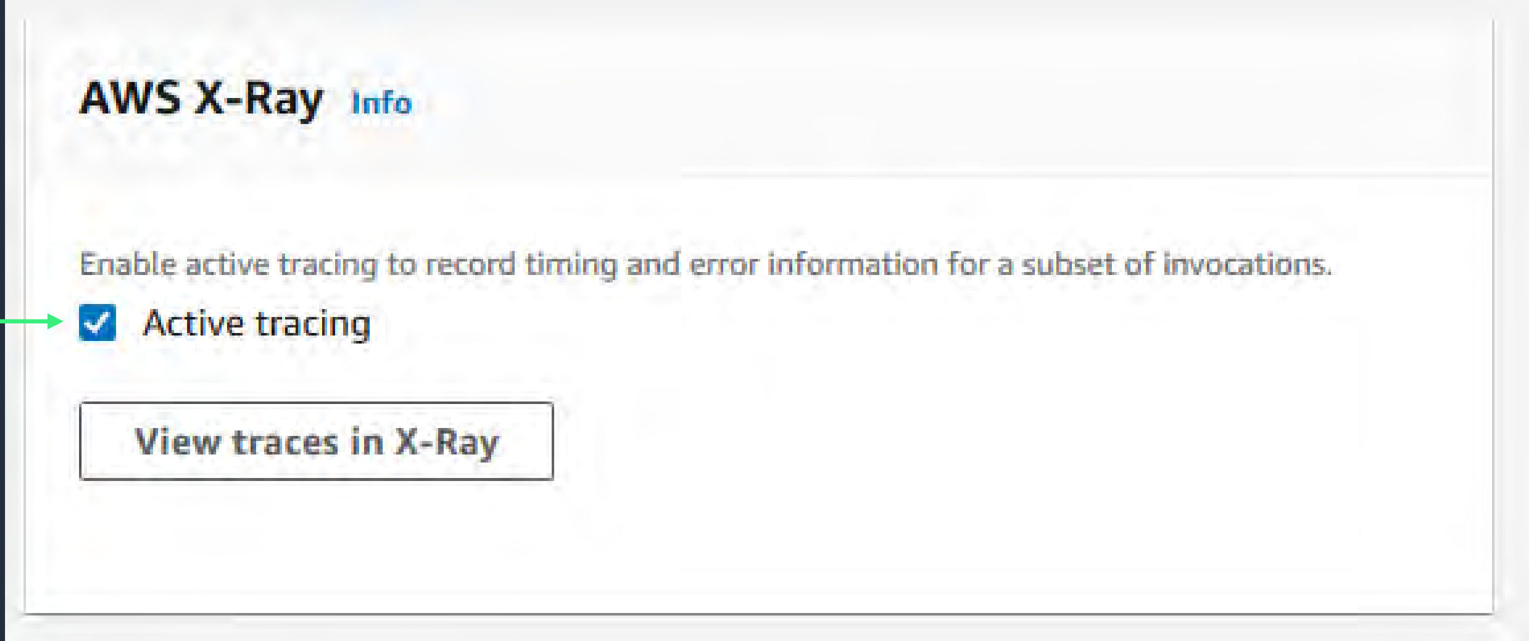
Products
Mobile App
API Gateway
Product Eligibility Service
Products DB
Event Bridge
Savings A/c Opening request placed
Savings Account Opened
New account request queue
Opened Accounts Queue
Core Banking Platform (Corporate Datacentre)
Account Opening Service
Monthly Interest Schedule Updated
Step Function
Customer Communications
Marketing
Marketing survey creation service

Trace if the end-to-end latency is within 100ms

Enable X-Ray tracing: AWS management console

AWS Lambda Console

(per function)



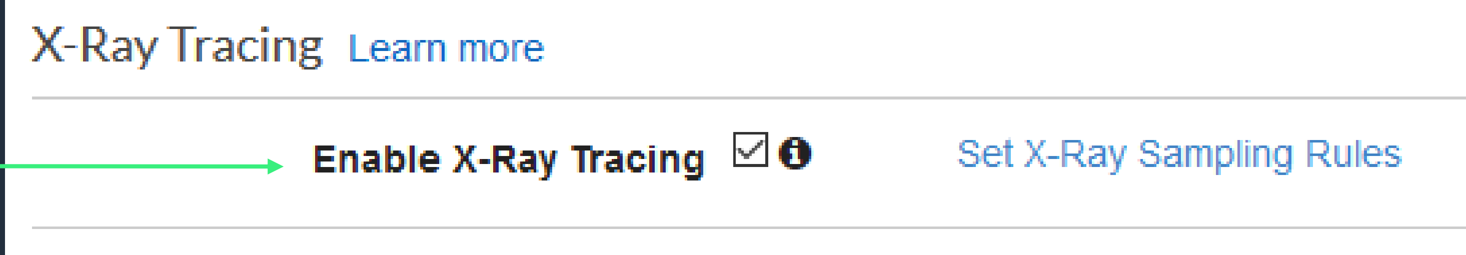
AWS X-Ray Info

Enable active tracing to record timing and error information for a subset of invocations.

☑ Active tracing

Amazon API Gateway Console

(per stage)



X-Ray Tracing Learn more

Enable X-Ray Tracing ☑

Also need to add IAM permissions for X-Ray

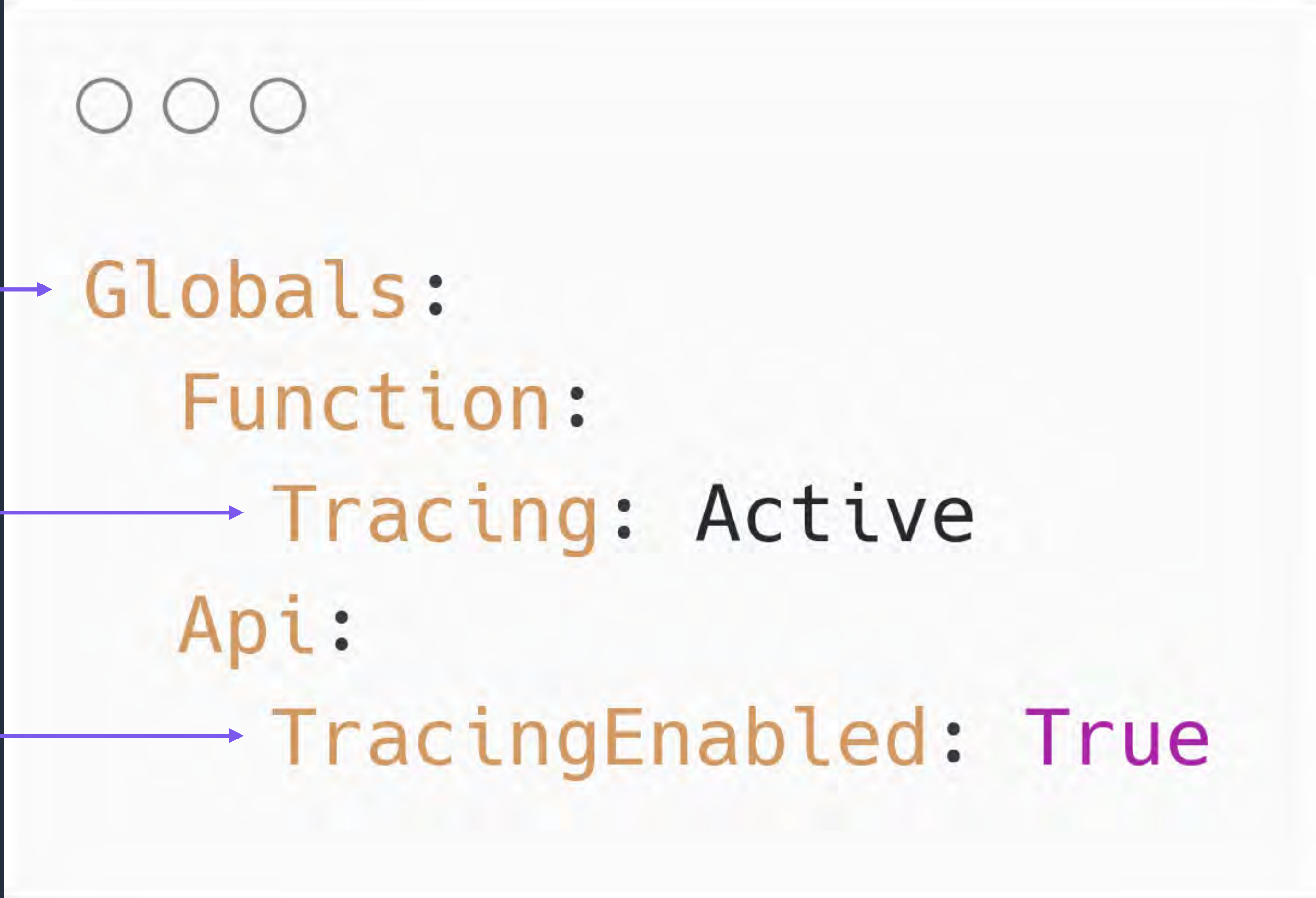
Enable X-Ray tracing: AWS SAM

90% of the work in two lines!

Globals section

All Lambda functions

All API Gateway REST APIs



```yaml
Globals:
  Function:
    Tracing: Active
  Api:
    TracingEnabled: True
```

AWS Lambda Powertools

What is AWS Lambda Powertools?

Powertools is a developer toolkit to implement Serverless best practices and increase developer velocity.

Available in Python, Java, Typescript and .Net.

Key Features

Logger

Metrics

Tracer

Idempotency

Parameters

+ Many more

A collection of utilities to ease adopting leading practices.

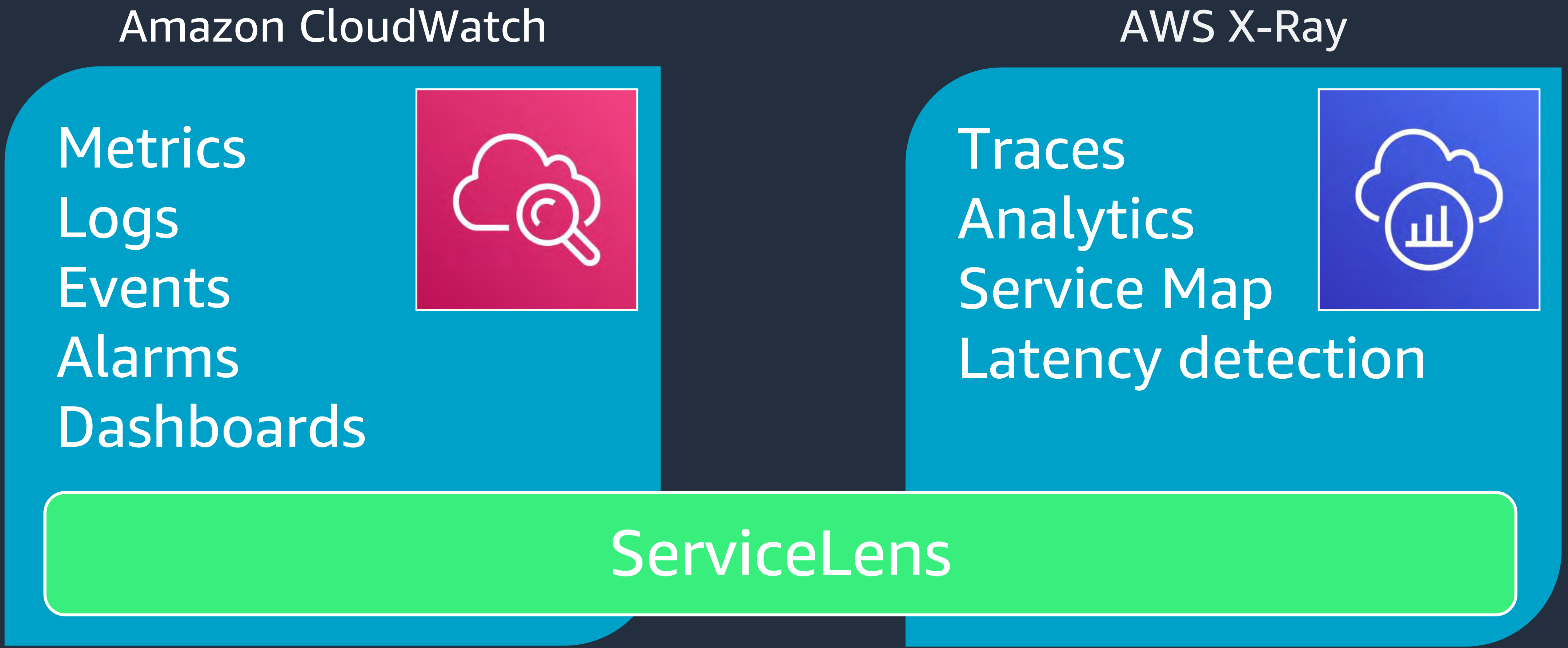
Powertools assistance

Question	Practice
SEC1	Scope access based on identity’s metadata
SEC2	Design smaller, single purpose functions
SEC2	Use temporary credentials between resources and components
SEC3	Store secrets that are used in your code securely
REL1	Use mechanisms to protect non-scalable resources
REL2	Manage transaction, partial, and intermittent failures
REL2	Manage duplicate and unwanted events
REL2	Orchestrate long-running transactions
OP1	Use distributed tracing and code is instrumented with additional context
OP1	Use structured and centralized logging
OP1	Use application, business, and operations metrics
OP2	Use infrastructure as code and stages isolated in separate environments
OP2	Use configuration management
PERF1	Measure and optimize function startup time
PERF1	Take advantage of concurrency via async and stream-based function invocations
PERF1	Optimize access patterns and apply caching where applicable
COST1	Minimize external calls and function code initialization
COST1	Optimize logging output and its retention
COST1	Use cost-aware usage patterns in code
Total	19

Bringing it all together

CloudWatch ServiceLens

- Unified access to metrics, logs, traces and canaries.
- Enabling performance monitoring from end-user interaction to infrastructure layer insights



Amazon CloudWatch

Metrics
Logs
Events
Alarms
Dashboards

AWS X-Ray

Traces
Analytics
Service Map
Latency detection

ServiceLens

Best Practices for Observing Event Driven Applications

1. Observe what matters

Example Business Problem : Open ISA (Instant Savings Account) for an existing customer through Mobile Banking App

Business SLAs:

- Customer can request to **open account 24/7**.
- Customer can **check eligibility in real time (100 ms)** and place account opening request.
- Account should be **operational in 2 hours** of request being placed.
- Customer should receive **mobile notification in 2 hours** of request being placed.
- Customer should **receive welcome email in 1 day** of account being opened.

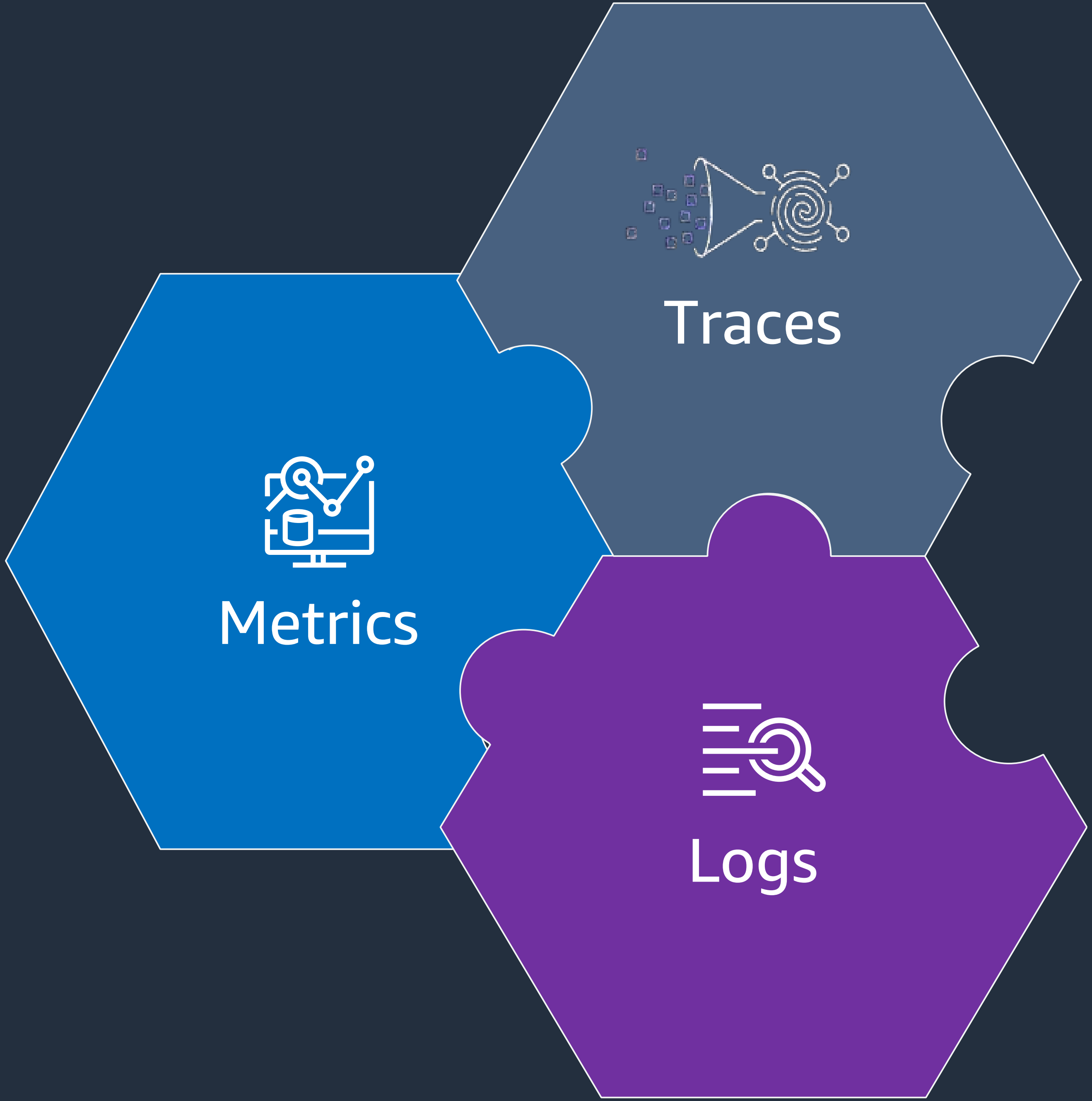
2. Measure your objectives

- Success metrics (KPIs/SLAs/SLOs/other)

- Know what **good** looks like

3. Identify sources

- Is the data available?
- Extract data
- Plan ahead



4. Alerting strategy

Define criteria (warning/critical)

Define actions

5. Dashboard strategy

Stakeholder dashboards

Cost, service audit, capacity planning

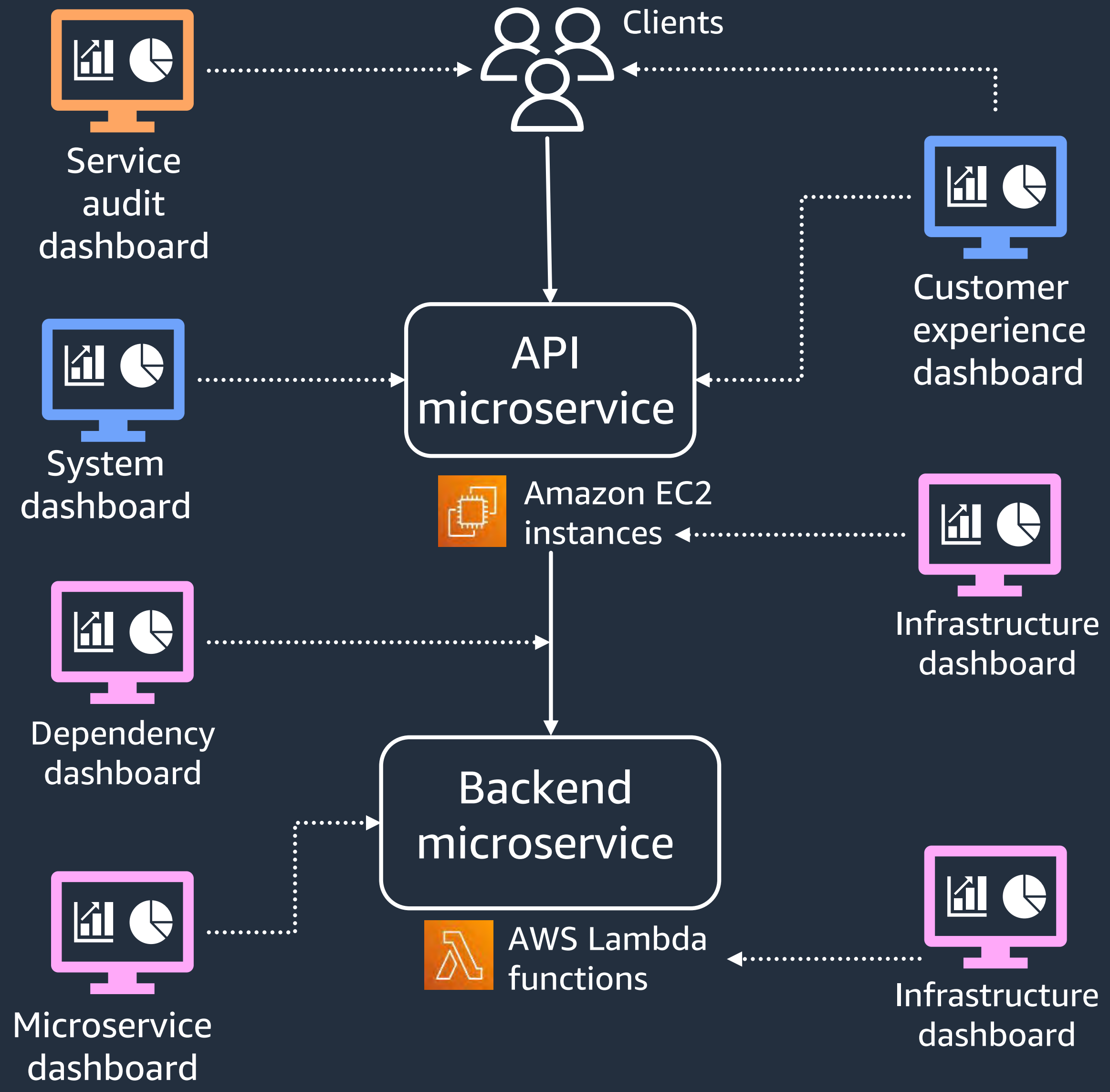
High-level dashboards

Customer experience, system level, service instance

Low-level dashboards

Infrastructure, microservice, dependency

Additional requirements



6. Tool selection

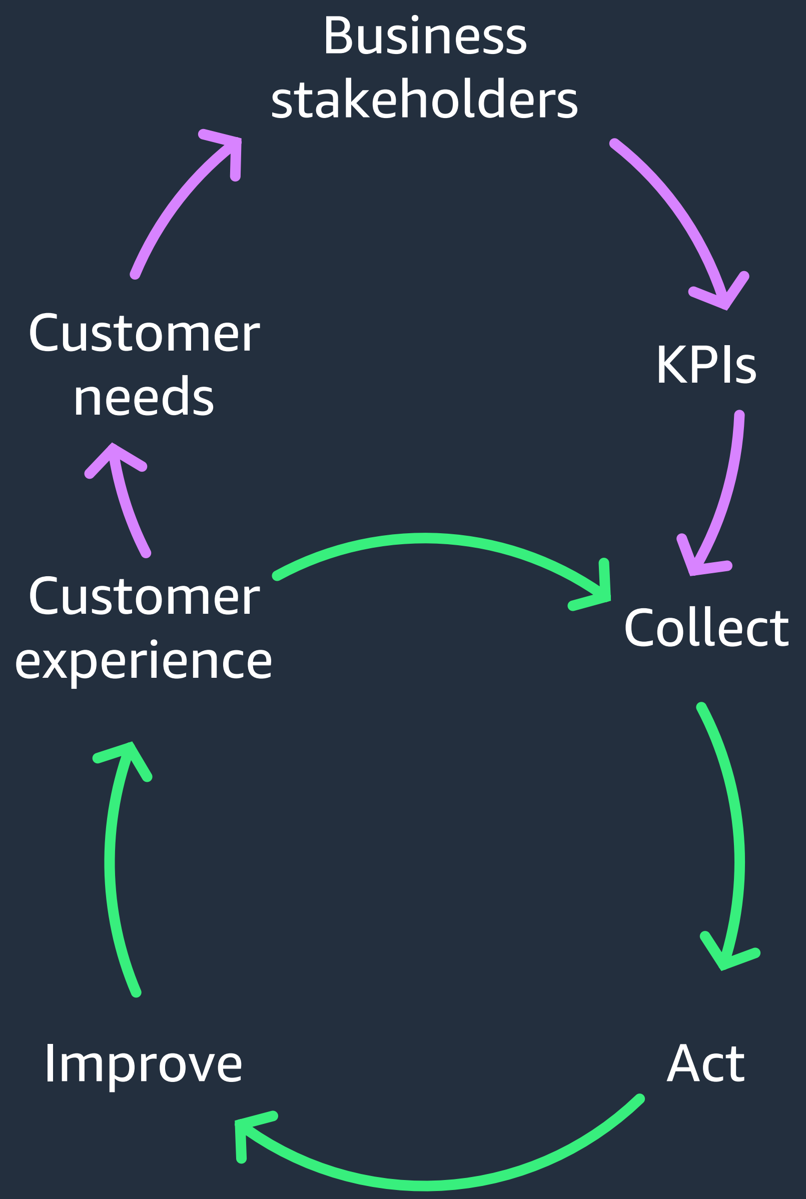
- Right tool for the job
- Pick features you **need**

7. Bring it all together

- Document
- Build into internal processes
- Operational readiness

8. Iterate

- Know your baselines
- Review routinely



Summary - Use Observability to get EDA right

1. Eventual Consistency

2. End to End Performance

3. Meeting Business SLAs

OBSERVABILITY

Further reading…

Serverless Observability

Lambda Powertools

Thank you!

Urmila Raju

Senior Solutions Architect, AWS

Connect with me on LinkedIn

Urmila Raju

Senior Solutions Architect, AWS
London