AWS re:Invent

IoT401

Implementing Multi-region AWS IoT

Olawale Oladehin
Sr. Solutions Architect
AWS

Lucas Starrett
Cloud Solutions Architect
Analog Devices

AWS re:Invent

aws

What will you learn

Why multi-region?

Foundation for multi-region with AWS IoT

Variations of multi-region architectures

Why multi-region for IoT?

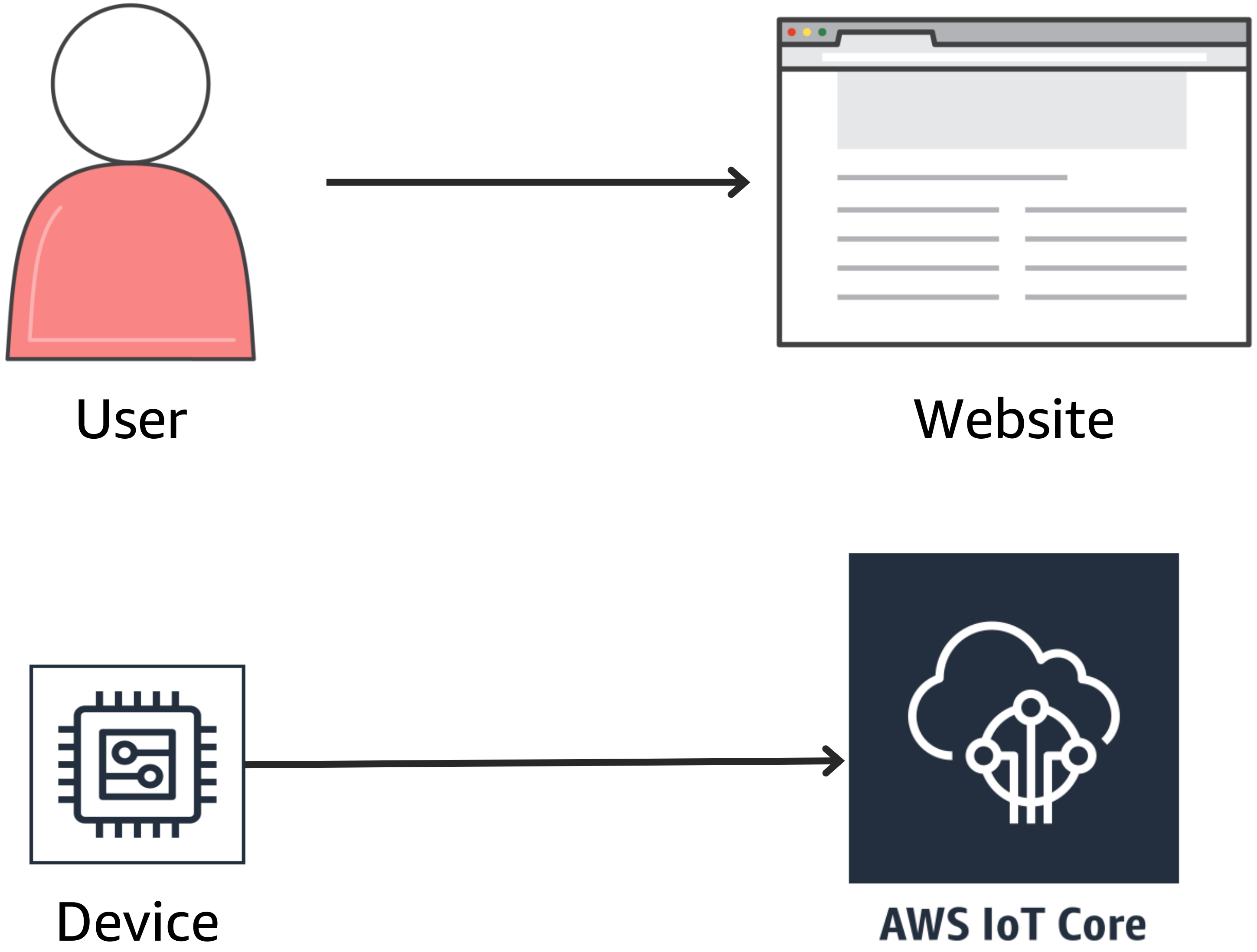
Why multi-region IoT?

Latency

Resiliency

Disaster Recovery

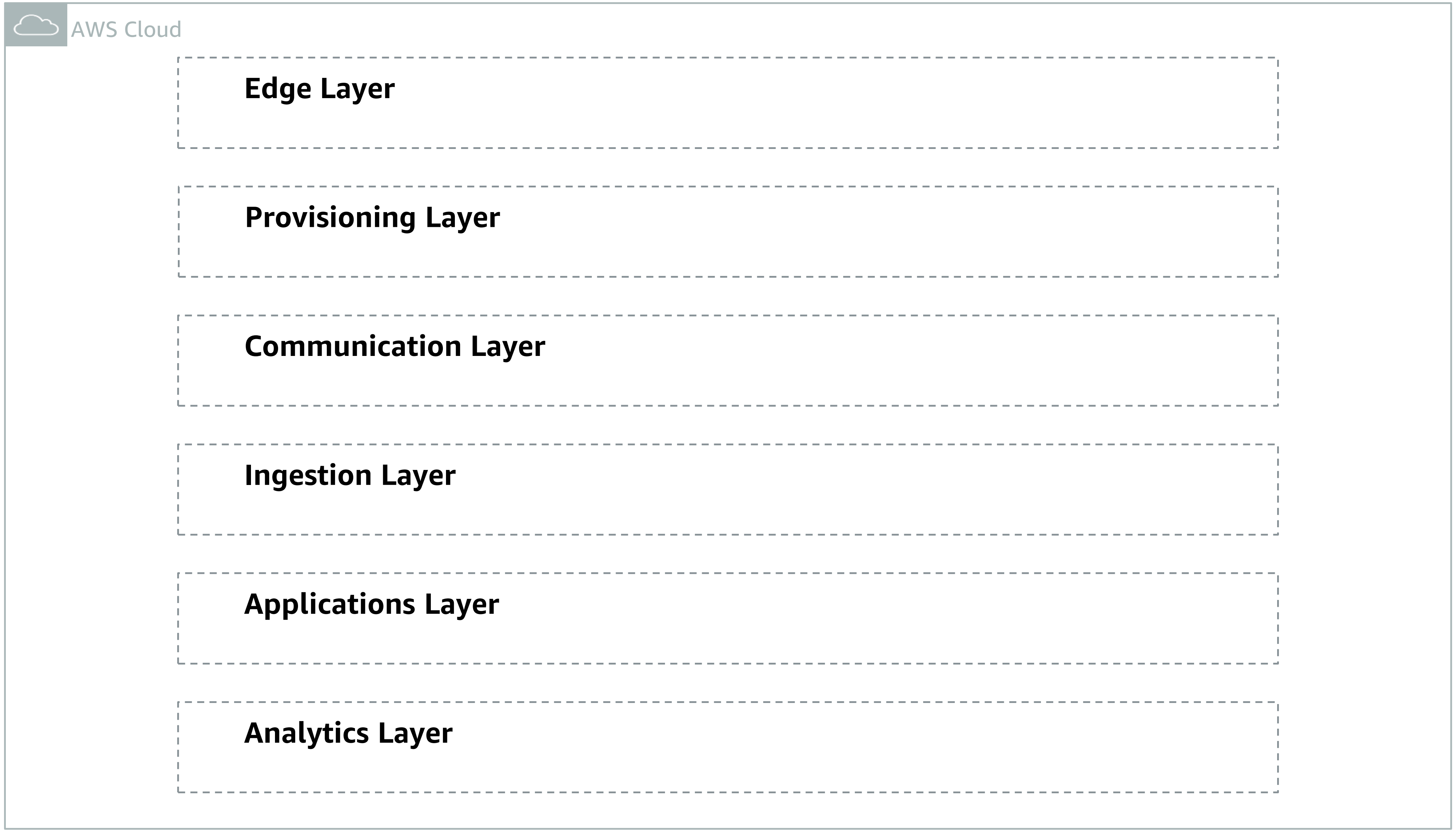
What makes IoT unique for multi-region?



- Computers and servers
- User-driven retry logic
- Web-based UI

- Constrained devices
- Recovery logic is pre-programmed
- Global logistics and provisioning

Where the focus will be?



AWS Cloud

Edge Layer

Provisioning Layer

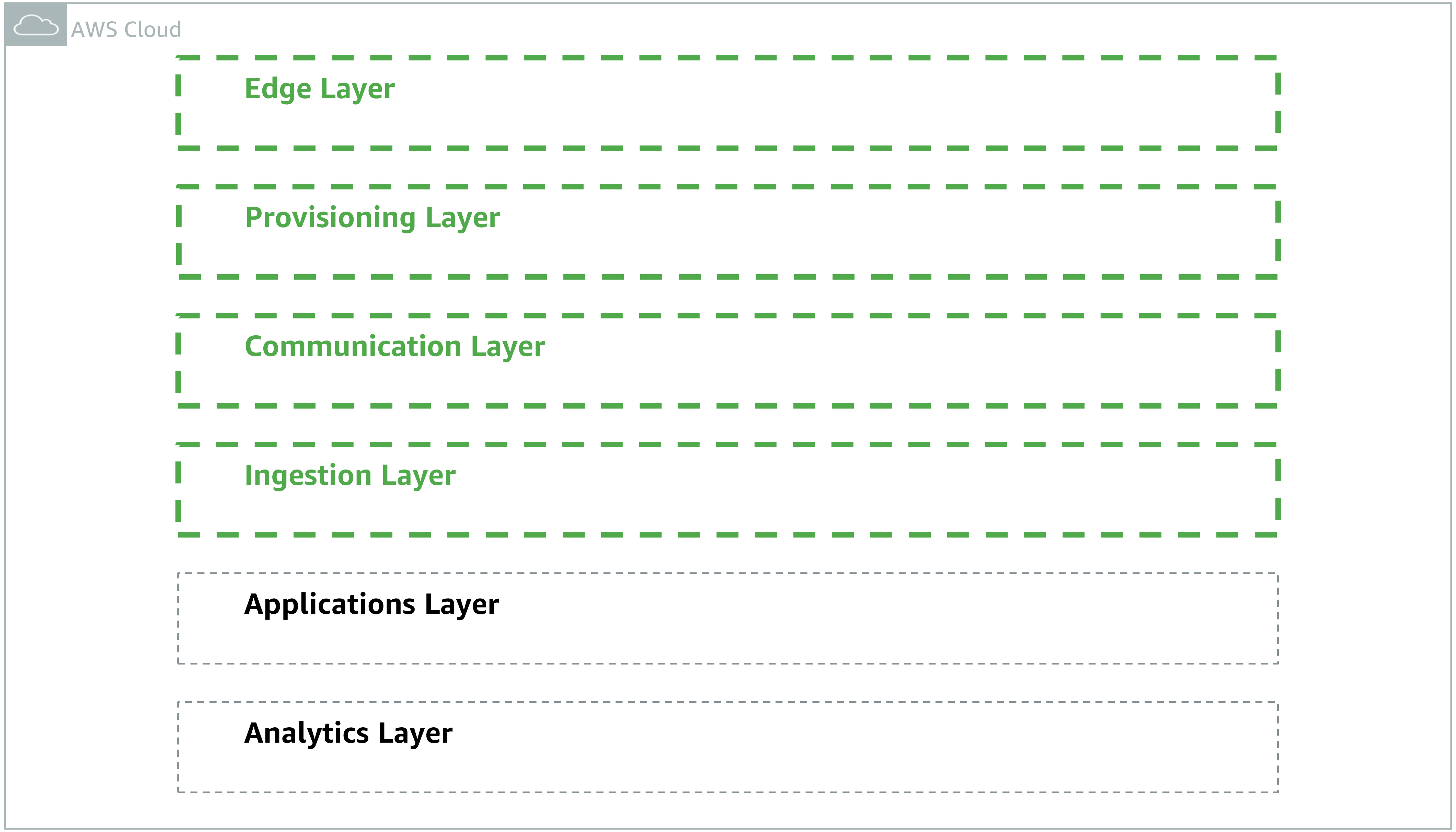
Communication Layer

Ingestion Layer

Applications Layer

Analytics Layer

Where the focus will be?


AWS Cloud

Edge Layer

Provisioning Layer

Communication Layer

Ingestion Layer

Applications Layer

Analytics Layer

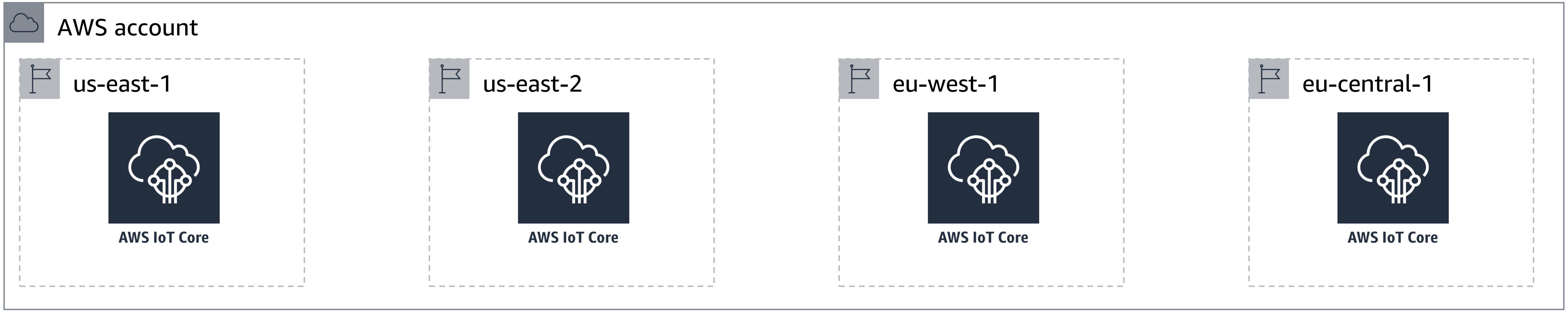
TOURNAMENT PRO SERIES
$100
SERIES

Table stakes for going multi-region

- AWS account and region structure
- Bootstrapping and device configuration
- Over-the-air updates
- Single region resiliency

How many AWS accounts do you need to deploy your IoT application?

One account for all regions



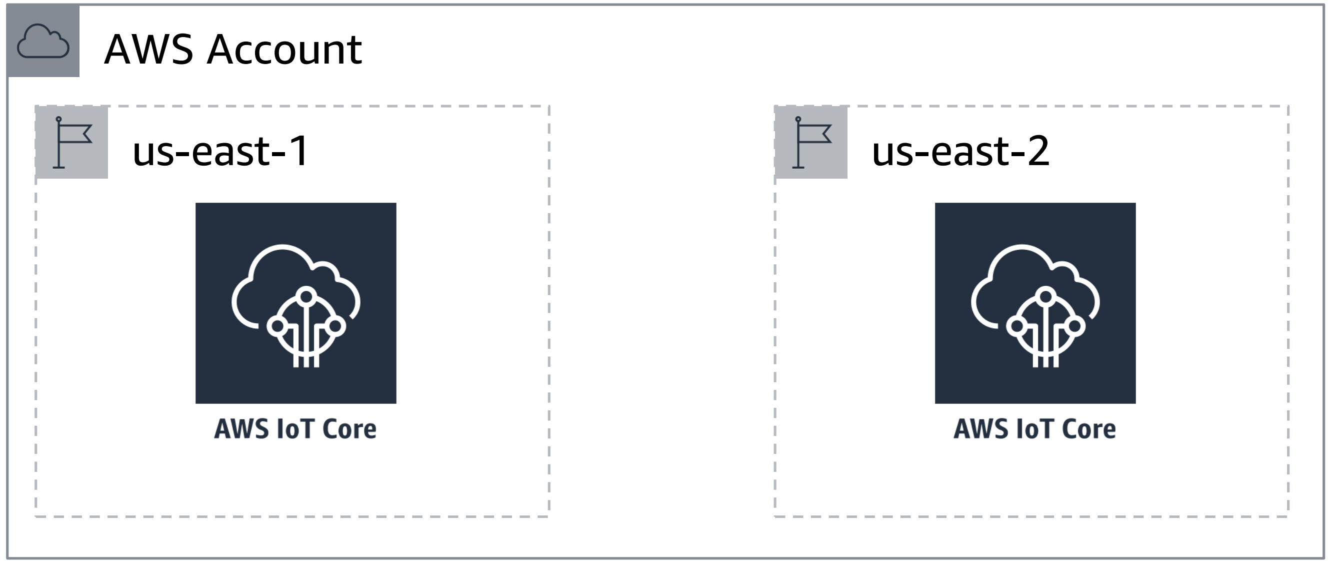
Pros

- Decreased replication complexity

Cons

- Increased blast radius for account users
- Implicit mapping of failover regions

Isolate Accounts By Regions



AWS Account

us-east-1

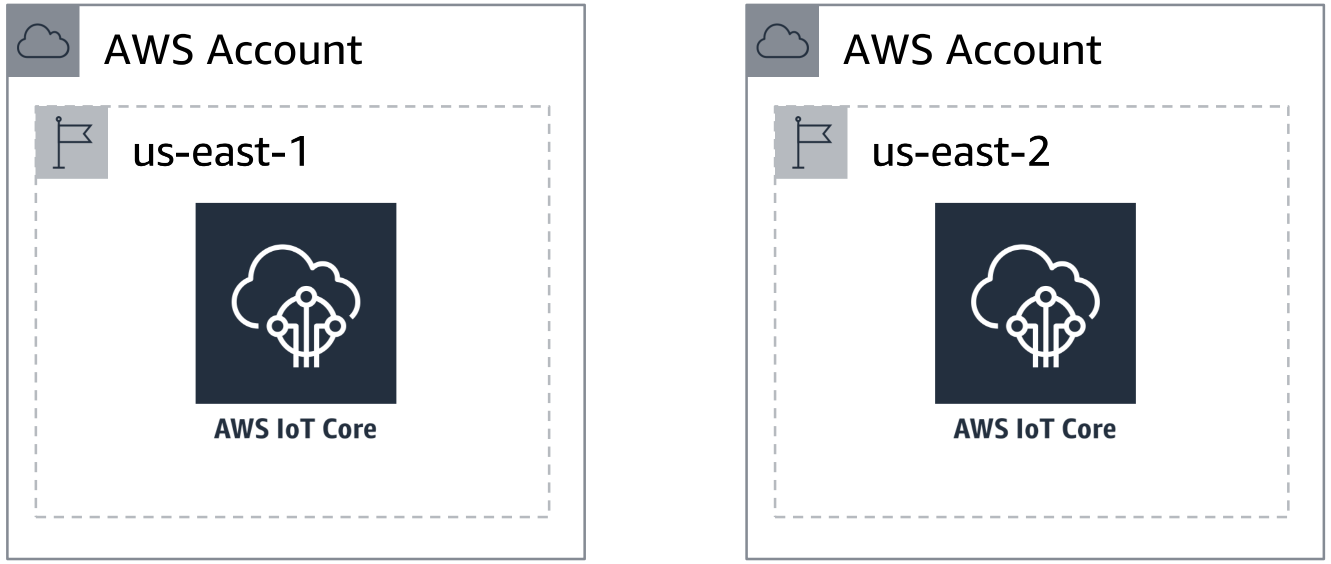
AWS IoT Core

us-east-2

AWS IoT Core

One Account Per Deployment

- Discrete Mapping of Accounts to Regions
- Smaller Blast Radius for Account Users



AWS Account

us-east-1

AWS IoT Core

AWS Account

us-east-2

AWS IoT Core

One Account Per Region

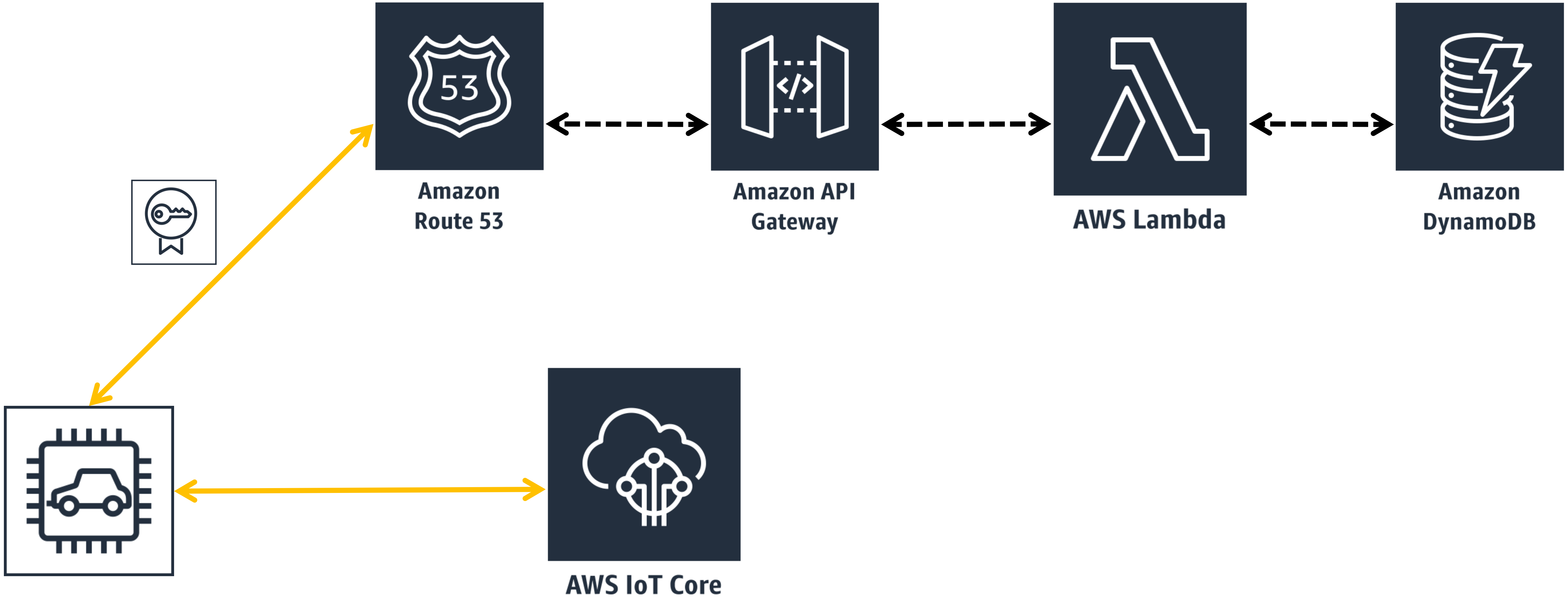
- Smallest Blast Radius for Account Users
- Increased Complexity for Replication

How do you programmatically configure your device settings and identity?

Bootstrapping and Device Configuration

- Global Endpoint
- Device Configuration
- Use Your Own Certificate Authority (CA) for Device Certificates

Bootstrapping—HTTP architecture



Bootstrapping – HTTP Response

```
{
    "endpoints":  [{
         "endpont":"XXXXX.iot.us-east-1.amazonaws.com",
         "CAs": [ "-----BEGIN CERTIFICATE-----cert-contents-----END CERTIFICATE-----"],
         "primary":true,
         "retry":5
       }],
    "topics": [
       "configurations": "cmd/123/config/456/",
       "sensor": ”data/123/456/"
      ]
 }
```

Bootstrapping – HTTP Response

```
{
    "endpoints":  [{
         "endpont":"XXXXX.iot.us-east-1.amazonaws.com",
         "CAs": [ "-----BEGIN CERTIFICATE-----cert-contents-----END CERTIFICATE-----"],
         "primary":true,
         "retry":5
       }],
    "topics": [
       "configurations": "cmd/123/config/456/",
       "sensor": "data/123/456/"
      ]
 }
```

AWS re:Invent

Bootstrapping – HTTP Response

```
{
    "endpoints":  [{
         "endpont":"XXXXX.iot.us-east-1.amazonaws.com",
         "CAs": [ "-----BEGIN CERTIFICATE-----cert-contents-----END CERTIFICATE-----"],
         "primary":true,
         "retry":5
       }],
    "topics": [
      "configurations": "cmd/123/config/456/",
      "sensor": "data/123/456/"
     ]
 }
```

Why use your own CA?

- AWS IoT-generated certificates are regional
- Customer-generated certificates can be provisioned in multiple regions

Bring your own certificate
Just-in-time registration
Just-in-time provisioning
Bulk provisioning templates

How do you build resiliency into your IoT application?

Single-region resiliency

AWS IoT Rules Engine Error Action

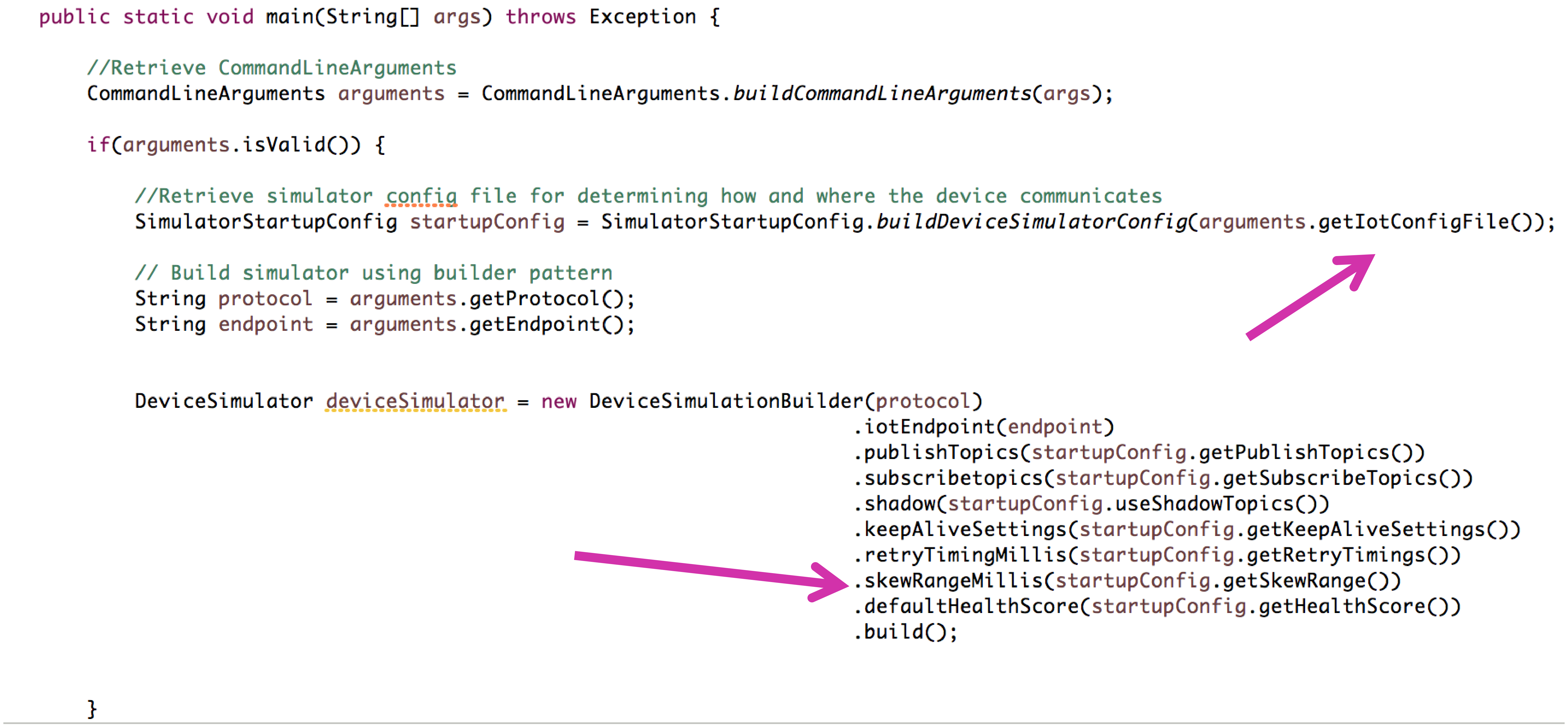
Using multiple upstream AWS IoT rules

Using services that leverage multiple Availability Zones (AZs)

Have retry logic in the cloud

Retry logic in the device SDK for the IoT broker

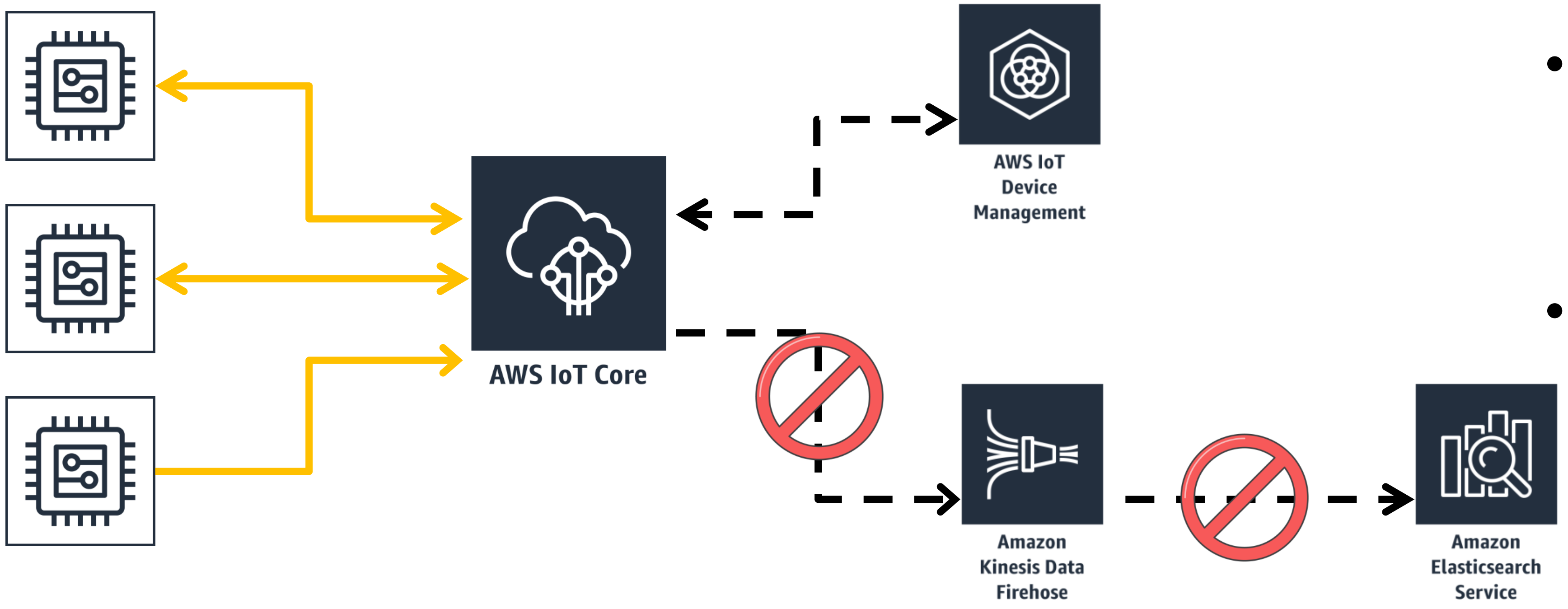
Implement retry logic in your device



```
public static void main(String[] args) throws Exception {

    //Retrieve CommandLineArguments
    CommandLineArguments arguments = CommandLineArguments.buildCommandLineArguments(args);

    if(arguments.isValid()) {

        //Retrieve simulator config file for determining how and where the device communicates
        SimulatorStartupConfig startupConfig = SimulatorStartupConfig.buildDeviceSimulatorConfig(arguments.getIotConfigFile());

        // Build simulator using builder pattern
        String protocol = arguments.getProtocol();
        String endpoint = arguments.getEndpoint();


        DeviceSimulator deviceSimulator = new DeviceSimulationBuilder(protocol)
                                                                    .iotEndpoint(endpoint)
                                                                    .publishTopics(startupConfig.getPublishTopics())
                                                                    .subscribetopics(startupConfig.getSubscribeTopics())
                                                                    .shadow(startupConfig.useShadowTopics())
                                                                    .keepAliveSettings(startupConfig.getKeepAliveSettings())
                                                                    .retryTimingMillis(startupConfig.getRetryTimings())
                                                                    .skewRangeMillis(startupConfig.getSkewRange())
                                                                    .defaultHealthScore(startupConfig.getHealthScore())
                                                                    .build();


    }
```

How do you update already deployed devices?

Overview the air (OTA) Updates


AWS IoT Device Management
AWS IoT Core
Amazon Kinesis Data Firehose
Amazon Elasticsearch Service

- Ability to update devices proactively
- Ability to failover devices based on cloud-side logic

Recap on the table stakes

Account structure

Bootstrapping

OTA updates

Single-region resiliency

Analog Devices, Inc.

ANALOG DEVICES

AHEAD OF WHAT'S POSSIBLE™

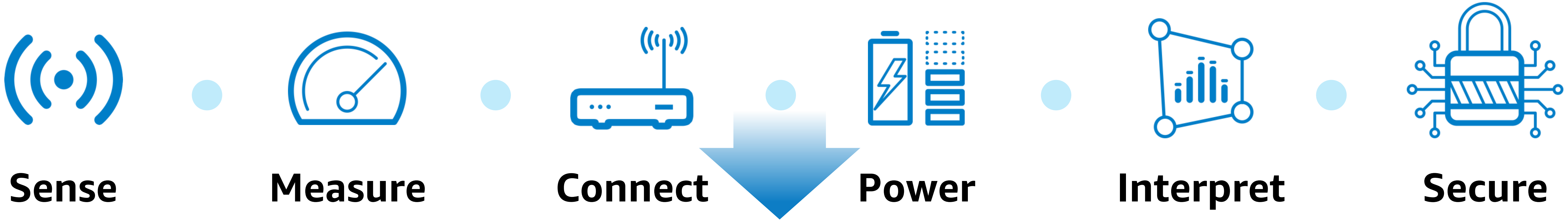
Who Is ADI?

A leader with a cutting-edge portfolio of analog and mixed-signal technologies and solutions

A fusion of deep domain knowledge and technical expertise

A collaborative partner and trusted ally dedicated to helping customers succeed

Bridging the Physical and Digital Worlds



Sense

Measure

Connect

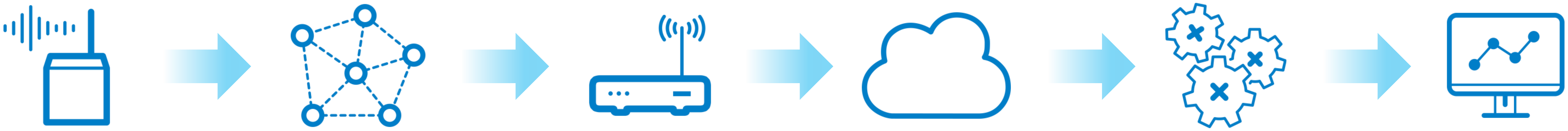
Power

Interpret

Secure

More valuable and **trusted** insights to make decisions, take action, and see better outcomes.

ADI Argus-M: Machine Health Monitoring



Edge Node

- Edge processing
- Battery operated: reliable, long lifetime, and low maintenance

Wireless Network

- High reliability
- Operates in harsh industrial environments

Gateway

- Multiple protocol support (Ethernet and cellular)
- Data aggregation for easy integration

ADI Cloud

- Secure device management
- Scalable infrastructure

Analytics

- Continuous machine monitoring and characterization
- Ongoing adaptation for improved uptime

User Interface

- Intuitive user experience from deployment to operation
- View status and interpret results easily

ANALOG DEVICES
AHEAD OF WHAT'S POSSIBLE™

Argus-M Deployment

Edge nodes monitor water pump motors at ADI’s manufacturing facility

- 15 edge nodes monitoring 13 assets
- Bearing vibration summary data sent semi frequently; raw vibration data sent infrequently
- Mesh networked edge nodes with multiyear battery life

Some of our multi-region IoT requirements

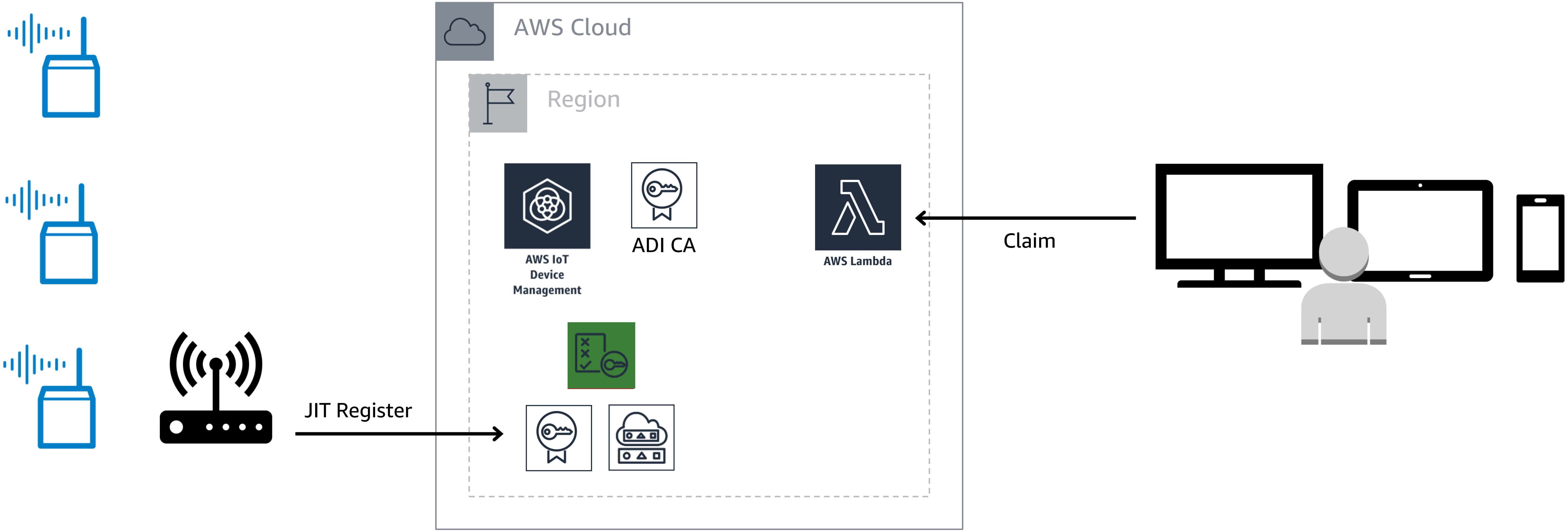
- Must be able to operate devices in multiple AWS regions best suited to customer or installation geolocation

- Must be able to operate devices in multiple AWS regions without re-provisioning device keys and certificates

- Must be able to migrate devices and dispatch OTA updates to devices in multiple AWS regions without physically interacting with devices

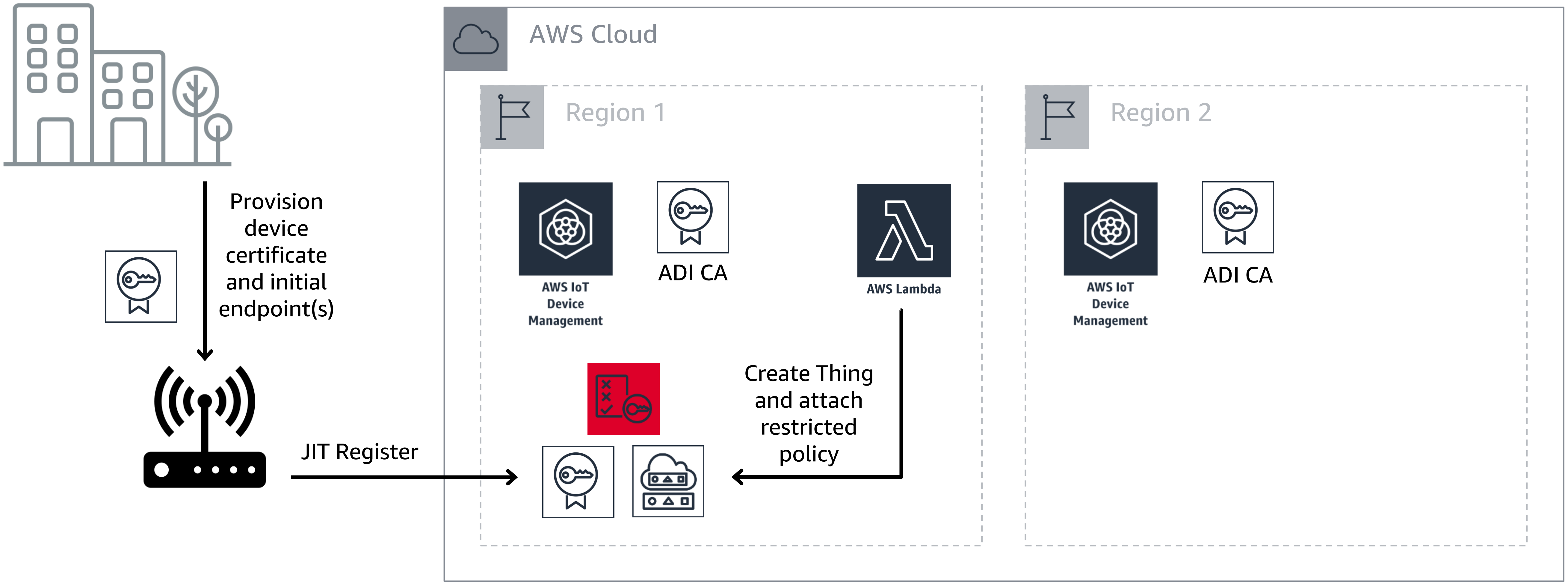
IoT Device Registration and Claiming Pattern

Device connection to AWS IoT and device activation and association with an application are separated into two distinct steps



IoT Device Registration Flow

Device is equipped from manufacturing to register with any operational region using its manufacturing-provisioned certificate



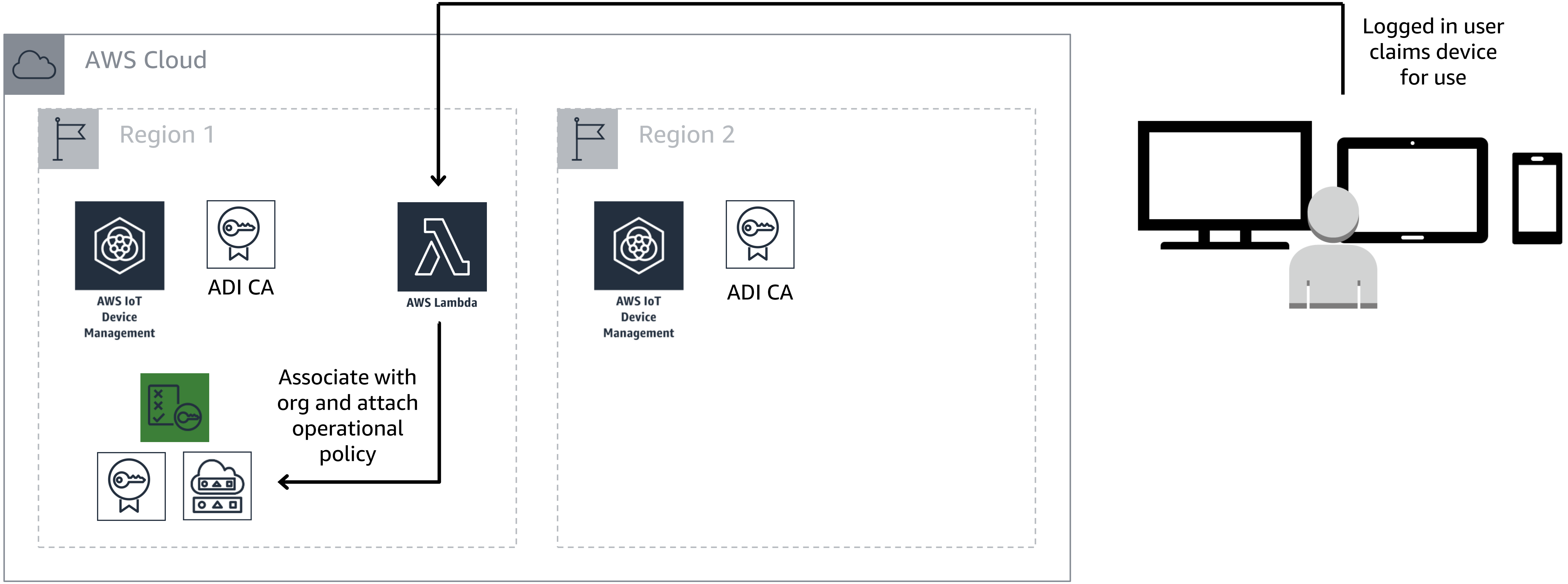
Provision device certificate and initial endpoint(s)
JIT Register
AWS Cloud
Region 1
AWS IoT Device Management
ADI CA
AWS Lambda
Create Thing and attach restricted policy
Region 2
AWS IoT Device Management
ADI CA

Extensible Common Name Field for JIT registration

```
Issuer: C=US, ST=MA, L=Boston, O=ADI, OU=IoT, CN=RegisteredCA
    Subject: C=US, ST=MA, L=Boston, O=ADI, OU=IoT,
             CN={\"id\":\"uniqueDeviceId\"\, \”sku\":\"2563X01189\"}
    Subject Public Key Info:
        Public Key Algorithm: id-ecPublicKey
            Public-Key: (384 bit)
            pub:
                04:17:53:67:a9:eb:be:c6:10:c2:d8:67:df:55:4a:
```

IoT Device Claiming Flow

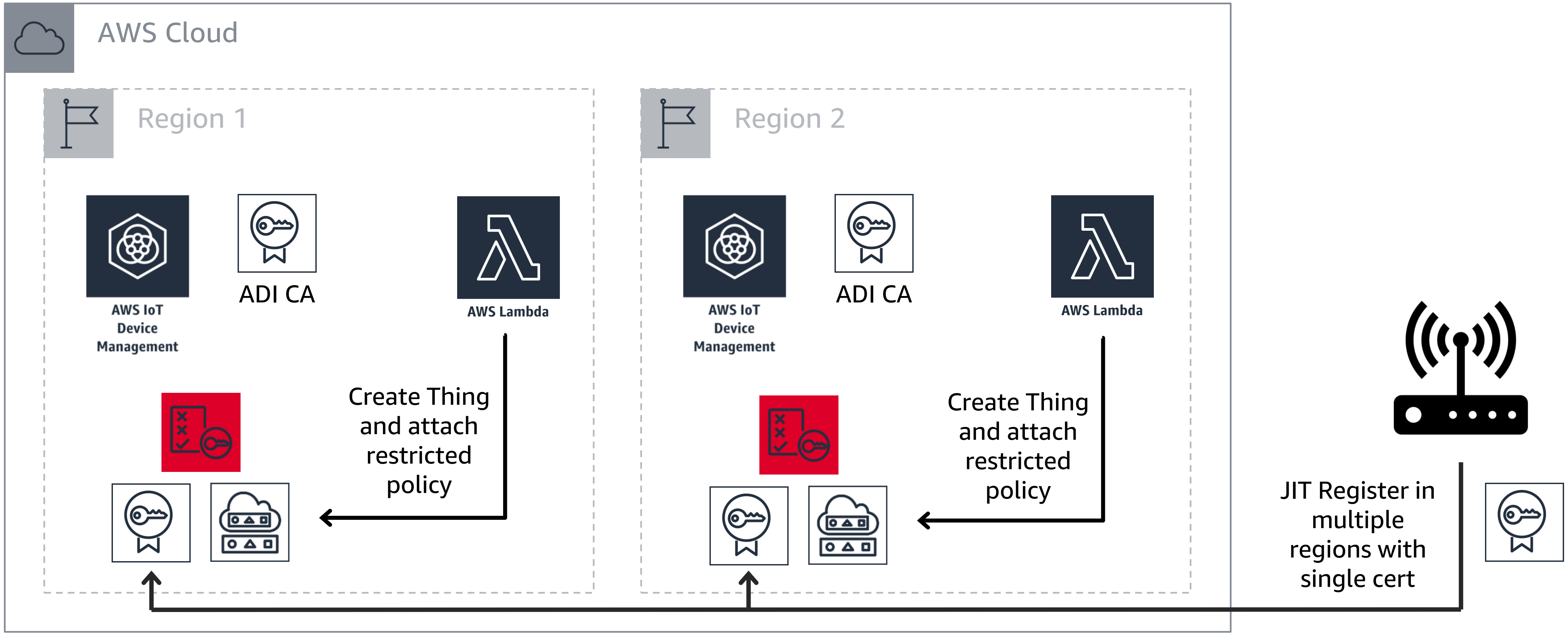
Registered device is associated with an owner and activated for use via claiming


AWS Cloud
Region 1
AWS IoT Device Management
ADI CA
AWS Lambda
Associate with org and attach operational policy
Region 2
AWS IoT Device Management
ADI CA
Logged in user claims device for use

ANALOG DEVICES
AHEAD OF WHAT'S POSSIBLE™

Registration and Claiming Multi-Region Agility

Registering in multiple regions does not require multiple certificates

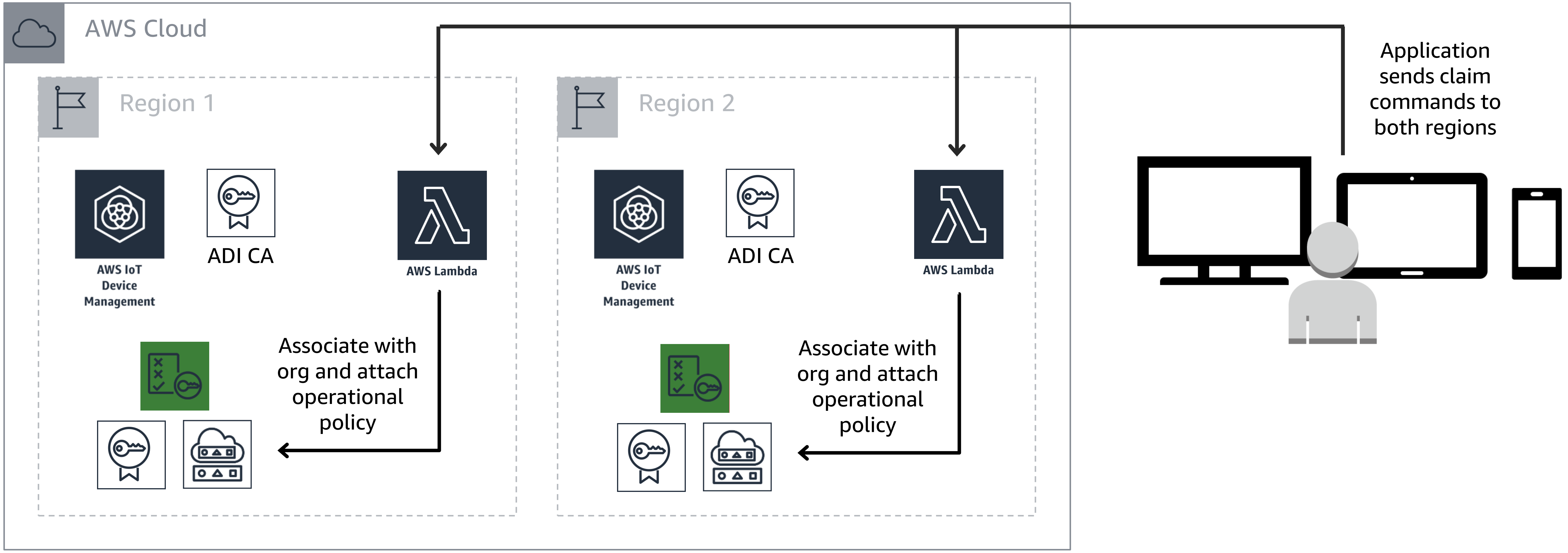

AWS Cloud
Region 1
AWS IoT Device Management
ADI CA
AWS Lambda
Create Thing and attach restricted policy
Region 2
AWS IoT Device Management
ADI CA
AWS Lambda
Create Thing and attach restricted policy
JIT Register in multiple regions with single cert

AWS re:Invent

aws

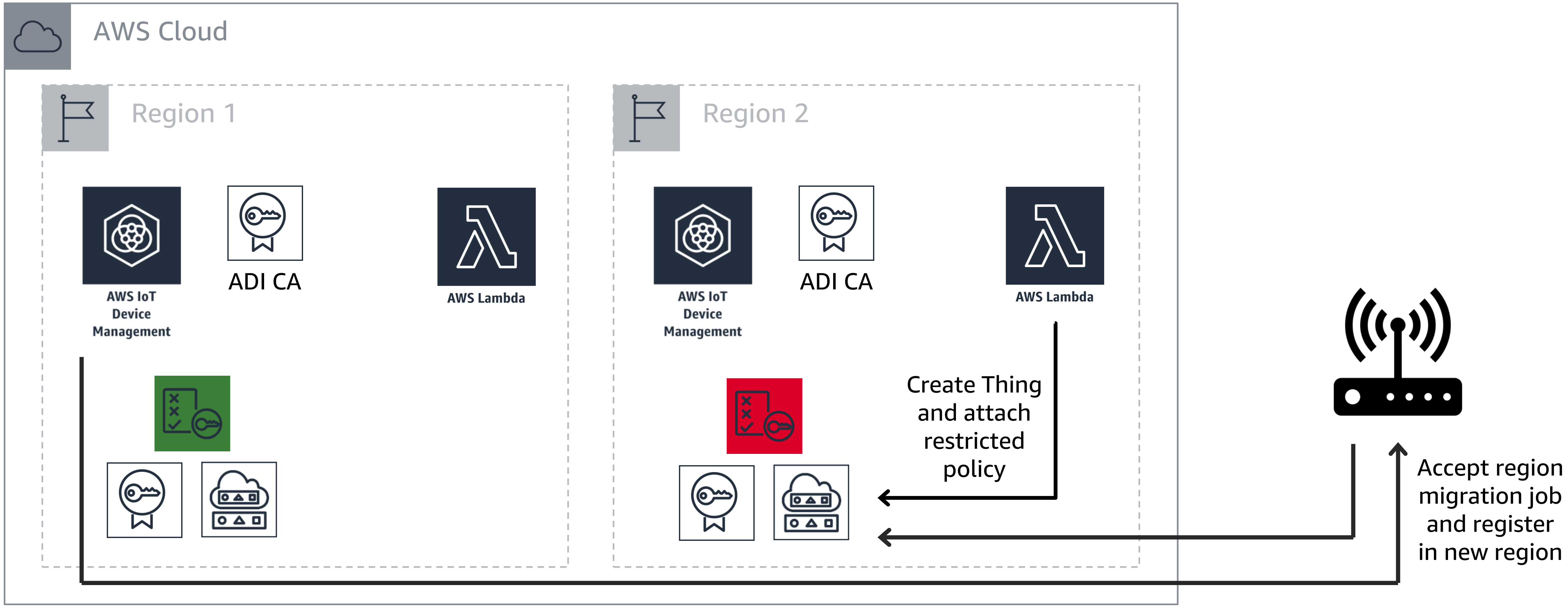
Registration and Claiming Multi-Region Agility

Devices can be registered and claimed in multiple regions to support failover


AWS Cloud
Region 1
AWS IoT Device Management
ADI CA
AWS Lambda
Associate with org and attach operational policy
Region 2
AWS IoT Device Management
ADI CA
AWS Lambda
Associate with org and attach operational policy
Application sends claim commands to both regions

Registration and Claiming Multi-Region Agility

Devices can be instructed to migrate to a new region via an IoT Job



AWS Cloud
Region 1
AWS IoT Device Management
ADI CA
AWS Lambda
Region 2
AWS IoT Device Management
ADI CA
AWS Lambda
Create Thing and attach restricted policy
Accept region migration job and register in new region

ANALOG DEVICES
AHEAD OF WHAT'S POSSIBLE™

Current Multi-Region Limitations and Challenges

- The JIT multi-region registration pattern requires the device to connect first to each region in order to register
- Using AWS IoT Jobs for region switching events requires tricky choreography
- To leverage bootstrapping pattern for dynamically configuring device MQTT endpoints on first boot, device must support HTTP
- Devices that connect through a gateway and not directly to AWS IoT via a certificate cannot take advantage of the JIT registration pattern
- Extended attributes of devices and groups stored in DynamoDB tables must be accounted for in preparation for migration of failover events

ANALOG DEVICES
AHEAD OF WHAT'S POSSIBLE™

Summary

- Managing and registering our own CA with AWS IoT in multiple regions and taking advantage of the JIT registration pattern enhances multi-region flexibility for our devices
- Separating device registration and device claiming into two distinct steps further enhances multi-region flexibility
- IoT Jobs can be leveraged to instruct devices to migrate between regions or update failover configurations
- **Building on AWS IoT makes deploying our IoT applications at a global scale possible!**

Implementing multi-region

Multi-Region Strategy - Comparison

Active/Passive

- Registry replication
- Certificate replication
- Pilot light infrastructure
- Data pinned to region
- Process data in the active region

Active/Active

- Registry replication
- Certificate replication
- Dual infrastructure
- Devices pinned to a region
- Process data once in any region

Multi-Region Strategy - Comparison

Active/Passive

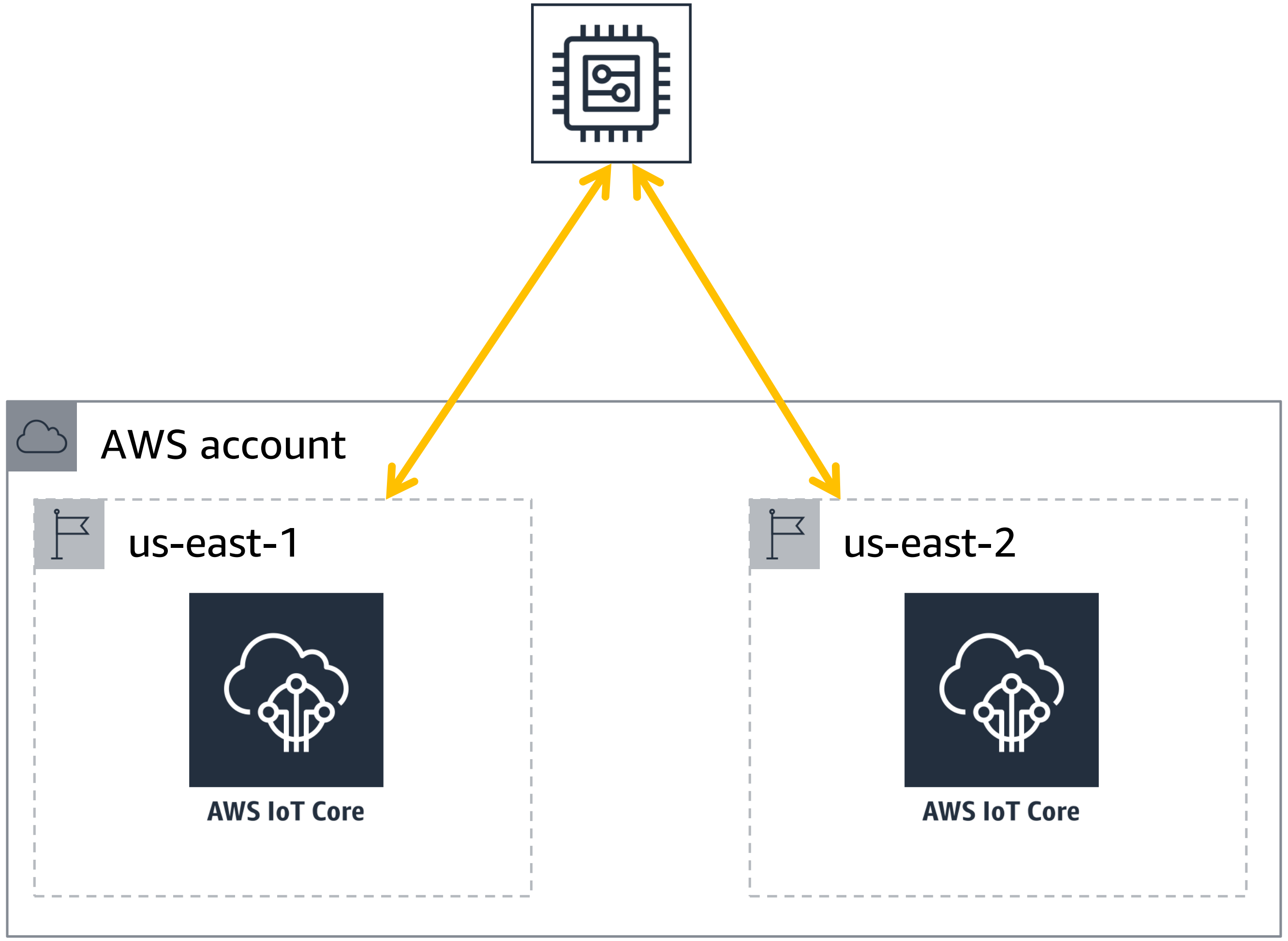
- Registry replication
- Certificate replication
- Pilot light infrastructure
- Data pinned to region
- Process data in the active region

Active/Active

- Registry replication
- Certificate replication
- Dual infrastructure
- Devices pinned to a region
- Process data once in any region

Active/Passive multi-region deployments

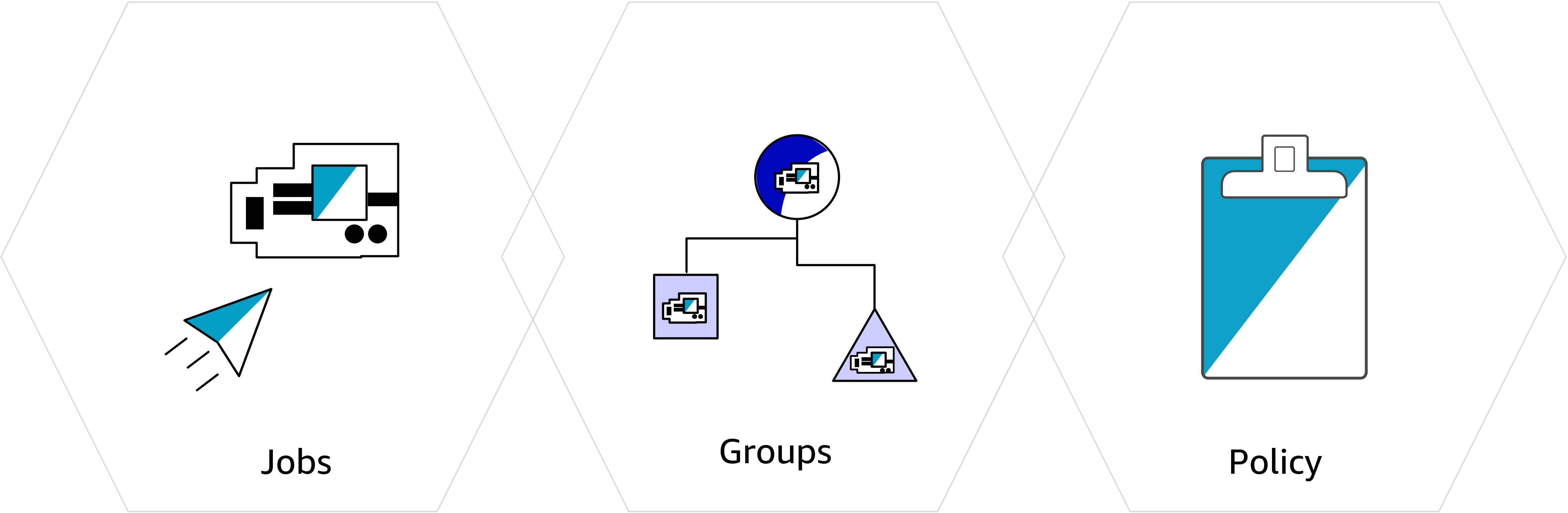
- Disaster recovery
- Primary and standby region
- All devices are pinned to a single region



Multi-Region Strategy – Active/Passive

- Registry replication
- Certificate replication
- Pilot light infrastructure
- Data pinned to region
- Process data in the active region

Replicate registry changes cross-region



Jobs

Groups

Policy

Registry Events

$aws/events/thing/name>/+

Thing Events

$aws/events/thingType/<name>/+

$aws/events/thingTypeAssociation/thing/<name>/<type>

ThingType Events

$aws/events/thingGroup/<name>/+

$aws/events/thingGroupMembership/thingGroup/<name>/thing/<name>

$aws/events/thingGroupHierarchy/thingGroup/<parentName>/childThingGroupName/<childName>/added

Thing Group Events

Registry Events

```
{
  "eventType": "THING_EVENT",
  "eventId": "9212cdaa2dd75b2a6c95236816ea8d69",
  "timestamp": 1542786890366,
  "operation": "UPDATED",
  "accountId": "377913865018",
  "thingId": "4d5b7dff-aaa1-46b5-92ae-ce603ba9822e",
  "thingName": "device12345678",
  "versionNumber": 2,
  "thingTypeName": "ElectricCar",
  "billinGroupName": null,
  "attributes": {
   "serialNumber": "Number"
  }
 }
```

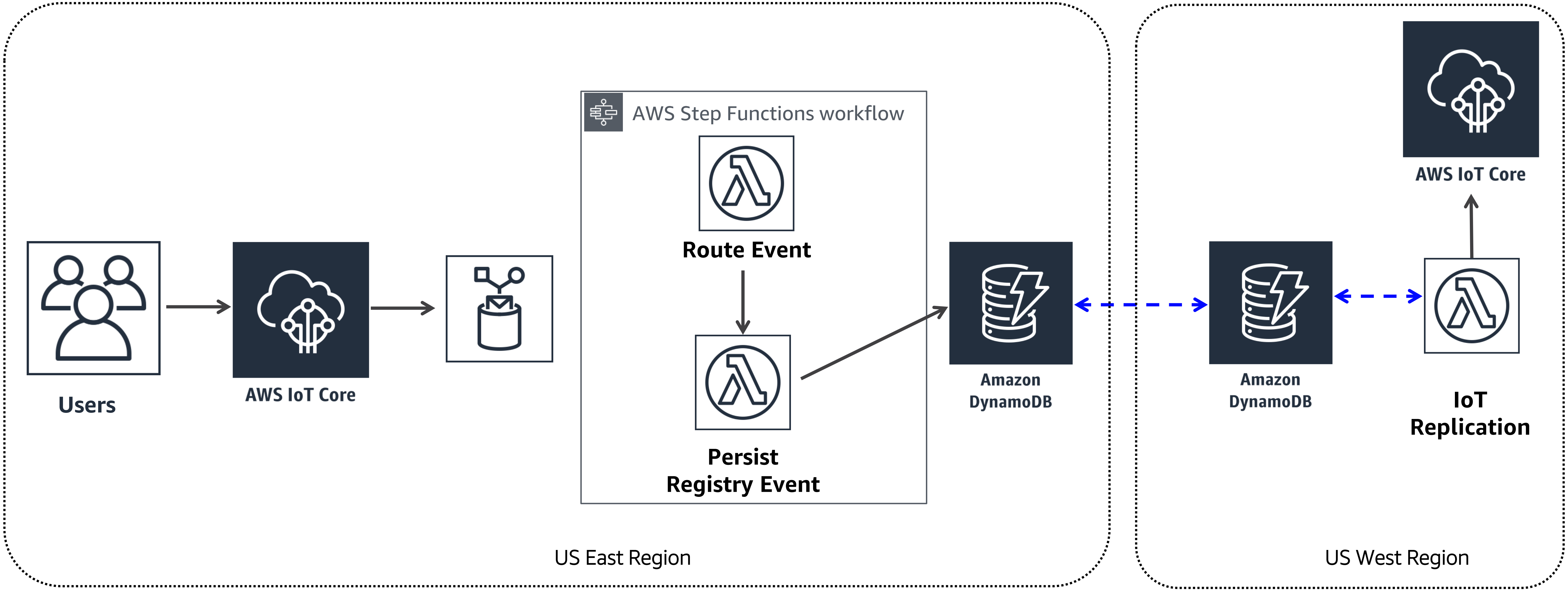
Registry Events

```
{
  "eventType": "THING_EVENT",
  "eventId": "9212cdaa2dd75b2a6c95236816ea8d69",
  "timestamp": 1542786890366,
  "operation": "UPDATED",
  "accountId": "377913865018",
  "thingId": "4d5b7dff-aaa1-46b5-92ae-ce603ba9822e",
  "thingName": "device12345678",
  "versionNumber": 2,
  "thingTypeName": "ElectricCar",
  "billinGroupName": null,
  "attributes": {
    "serialNumber": "Number"
  }
 }
```

Registry Events

```
{
    "eventType": "THING_EVENT",
    "eventId": "9212cdaa2dd75b2a6c95236816ea8d69",
    "timestamp": 1542786890366,
    "operation": "UPDATED",
    "accountId": "377913865018",
    "thingId": "4d5b7dff-aaa1-46b5-92ae-ce603ba9822e",
    "thingName": "device12345678",
    "versionNumber": 2,
    "thingTypeName": "ElectricCar",
    "billinGroupName": null,
    "attributes": {
     "serialNumber": "Number"
    }
  }
```

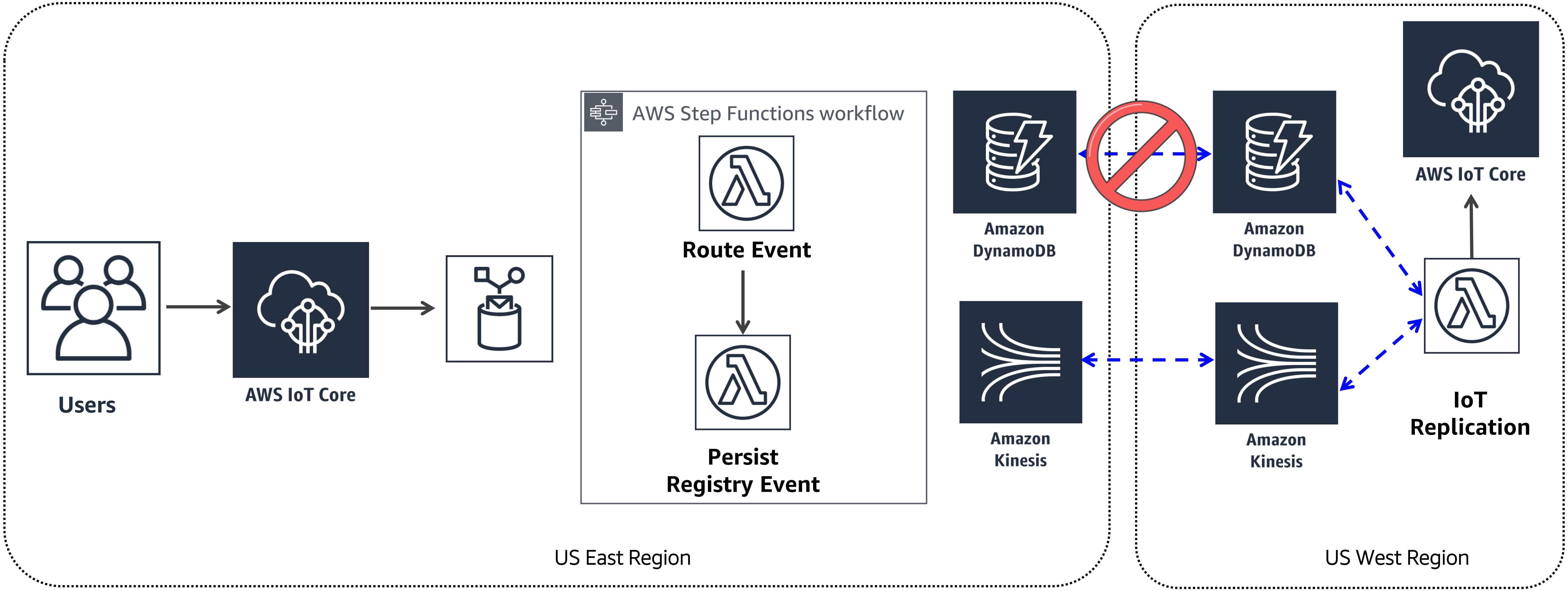
Replicate Registry with DynamoDB Global Tables



AWS re:Invent

aws

Cross Region Replication Considerations



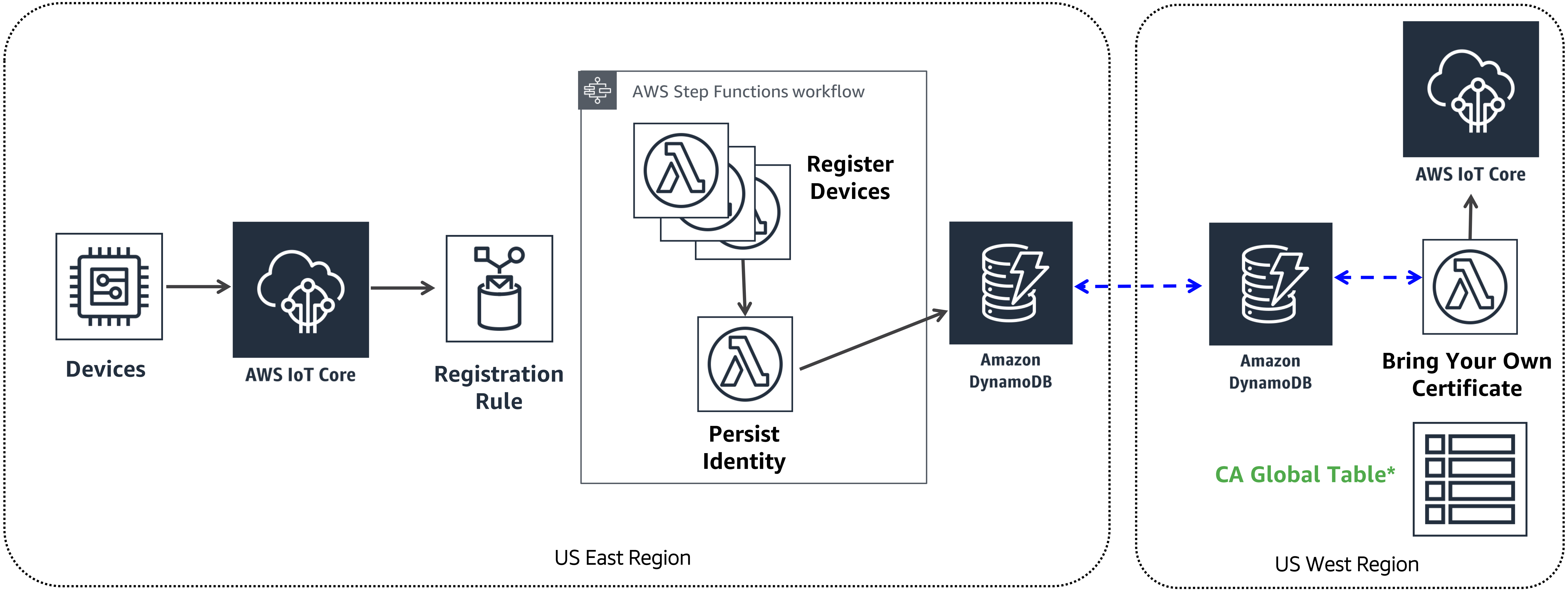
Replicate Certificate Registration Cross Region

Create rules for Just In Time Registration:

$aws/events/certificates/registered/<caCertificateID>

```
{
      "certificateId": "<certificateID>",
      "caCertificateId": "<caCertificateId>",
      "timestamp": "<timestamp>",
      "certificateStatus": "PENDING_ACTIVATION",
      "awsAccountId": "<awsAccountId>",
      "certificateRegistrationTimestamp":"<certificateRegistrationTimestamp>"
}
```

Register and Replicate Certificates across DynamoDB



Persist Identity

JITR Event

```
{
   "certificateId": "<certificateID>",
   "caCertificateId": "<caCertificateId>",
   "timestamp": "<timestamp>",
   "certificateStatus": "PENDING_ACTIVATION",
   "awsAccountId": "<awsAccountId>",
   "certificateRegistrationTimestamp":"<certificateRegistrationTimestamp>"
}
```

Custom Registry Events

Certificate Event

Policy Event

Custom IoT Events

Policy Event

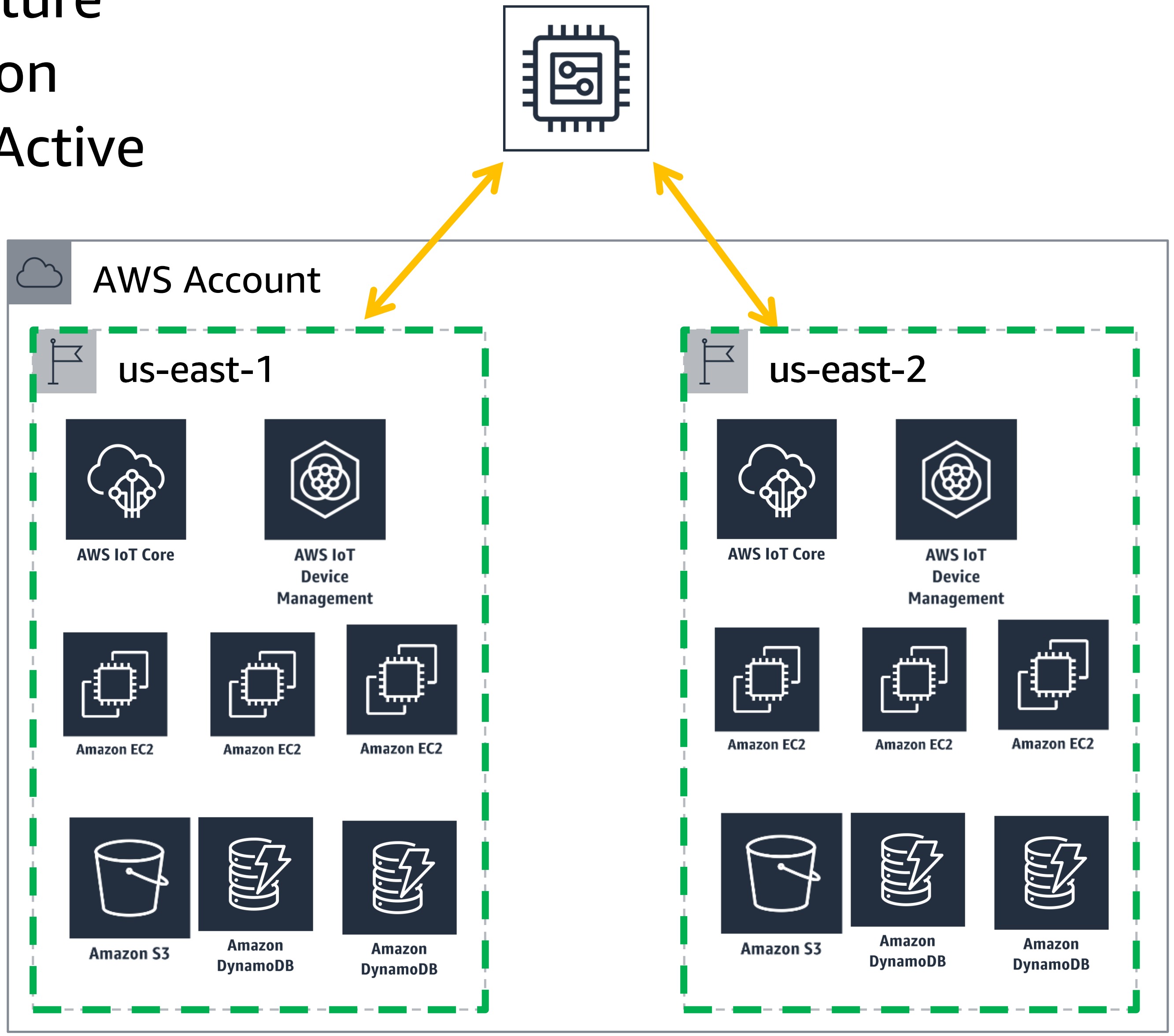
```
{
   "eventType": "POLICY_EVENT",
   "eventId": "<182983567-e09b-56d3-a123-425553>",
   "operation": "CREATED|UPDATED|DELETED",
   "accountId": "1234567890",
   "status": "ATTACHED|REMOVED",
   "globalCertIdentifier": "POLICERT_IDENTIFIERNAME",
   "thingName": "THING_NAME",
   "policyStatement": {"action": "resource"},
   "thingPolicyName": "POLICY_NAME"
}
```

Certificate Event

```
{
   "eventType": "CERTIFICATE_EVENT",
   "eventId": "<123e4567-e89b-12d3-a456-426655>",
   "operation": "CREATED|UPDATED|DELETED",
   "accountId": "1234567890",
   "status": "ACTIVE",
   "certPemFile": "-PEM--FILE",
   "thingName": "THING_NAME",
   "globalCertIdentifier": "POLICERT_IDENTIFIERNAME”
   "thingPolicyName": "POLICY_NAME”
}
```

Active/Passive Multi-Region Deployments

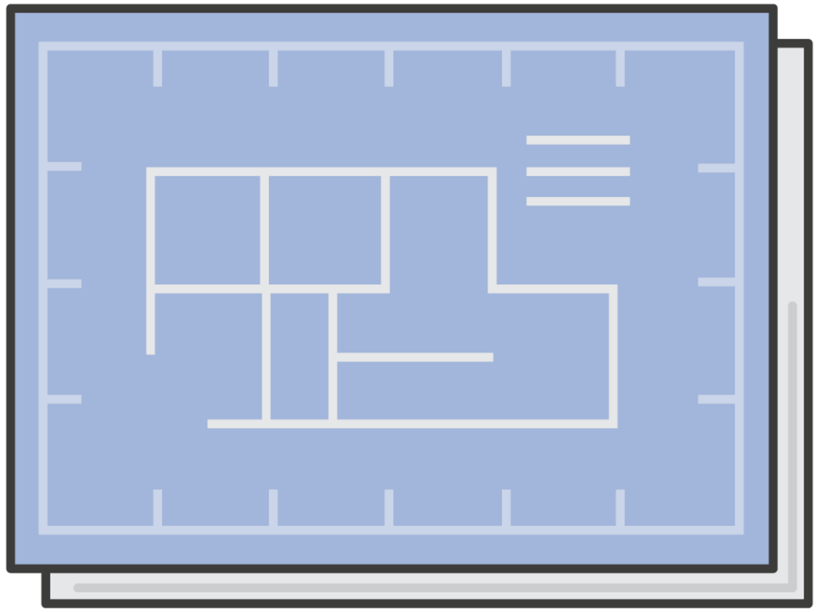
- Pilot light Infrastructure
- Data Pinned to Region
- Process Data in the Active Region



Implement Cutover Strategy

Cloud Cutover

- Device Configuration via OTA
- Bootstrapping Configuration Update



Demo

AWS re:Invent

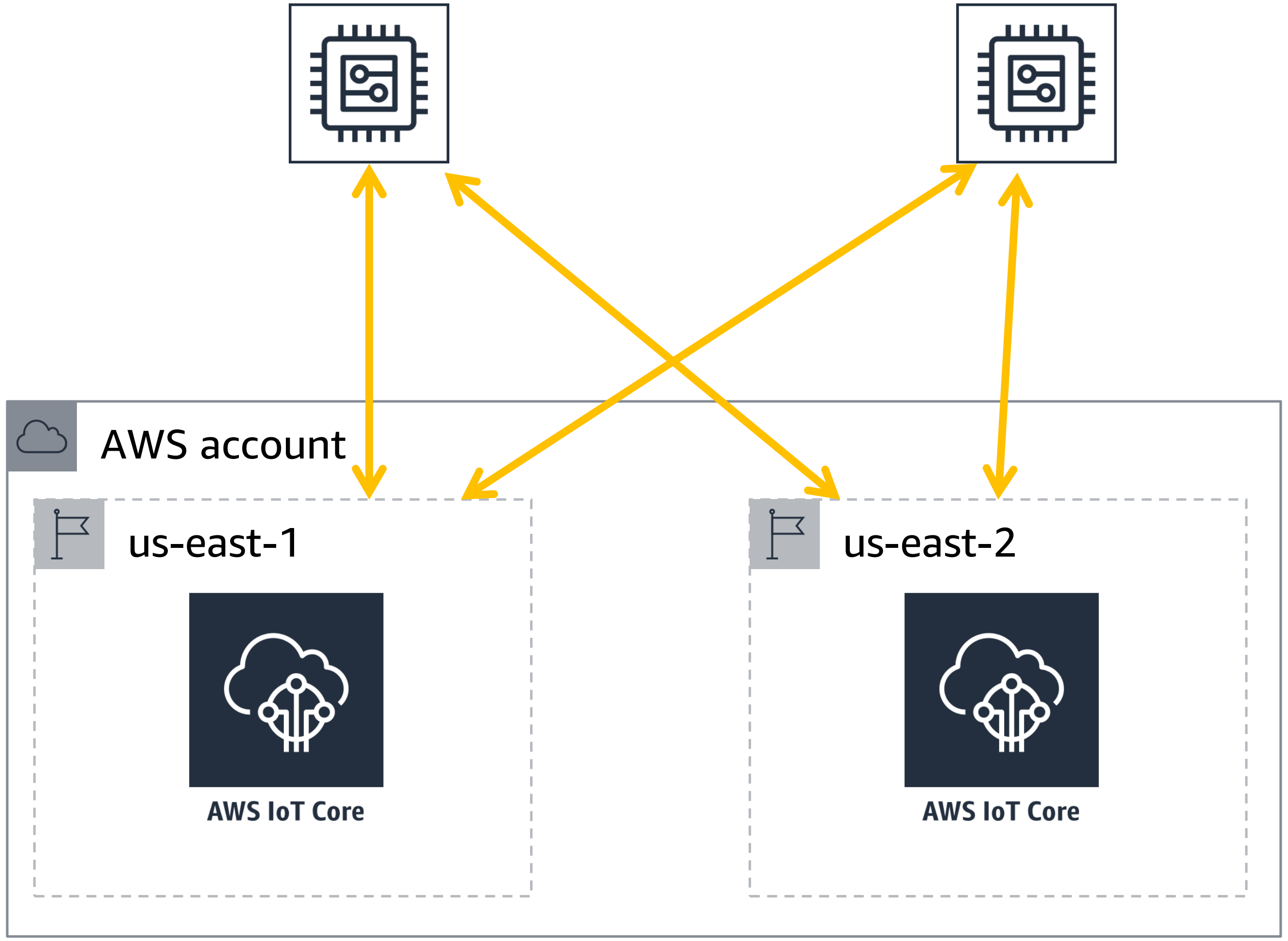
aws

> “Everyone knows that debugging is twice as hard as writing a program in the first place. So if you're as clever as you can be when you write it, how will you ever debug it?”

Brian Kernighan

Active/Active multi-region deployments

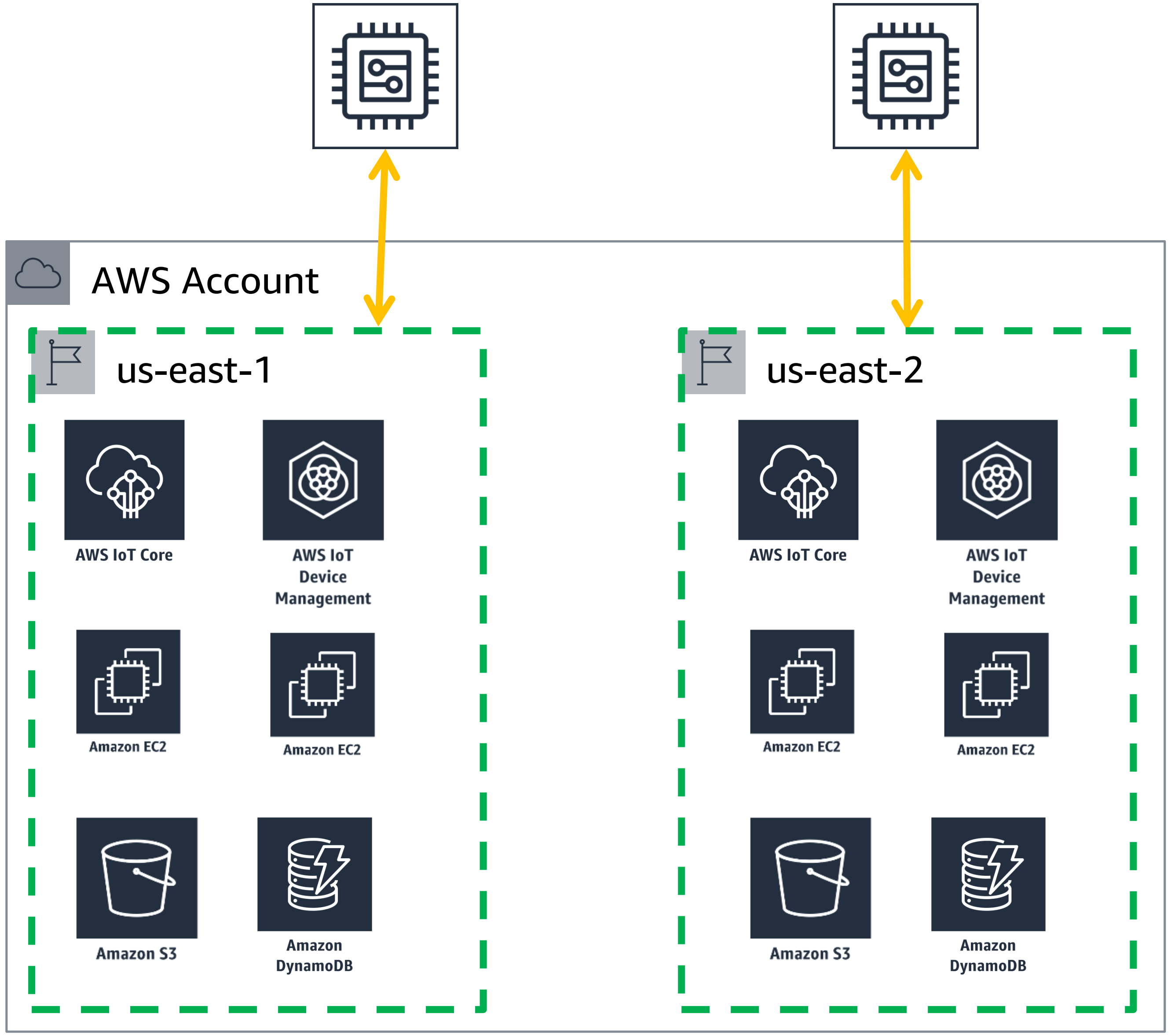
- Seamless disaster recovery
- All regions in a deployment can be primary
- Devices can communicate to any region
- Cost benefits for traditional internal applications



Multi-Region Strategy – Active/Active

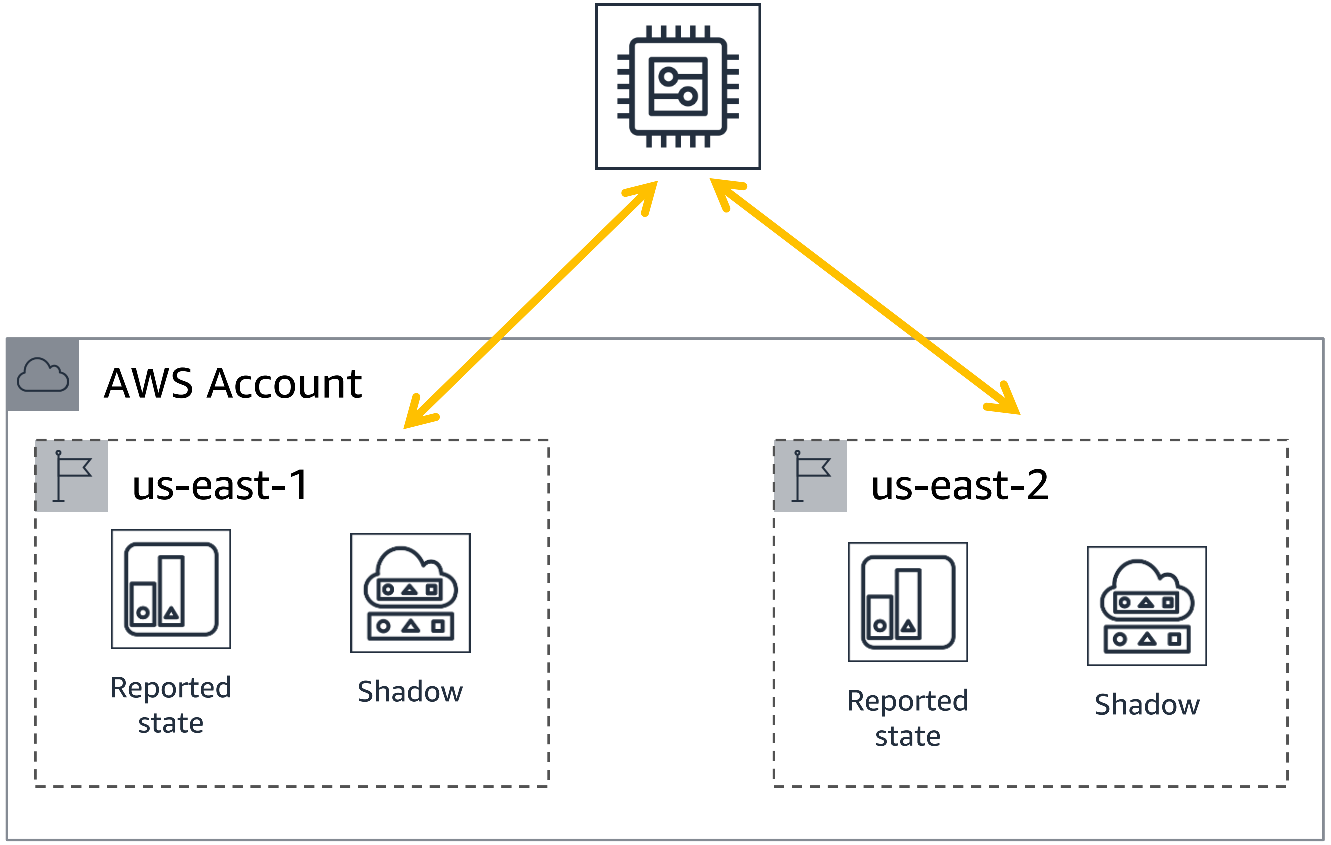
- Registry replication
- Certificate replication
- **Dual infrastructure**
- **Devices pinned to a region**
- **Process data once in any region**

Active/Active - Dual Infrastructure



Active/Active - Devices pinned to a region

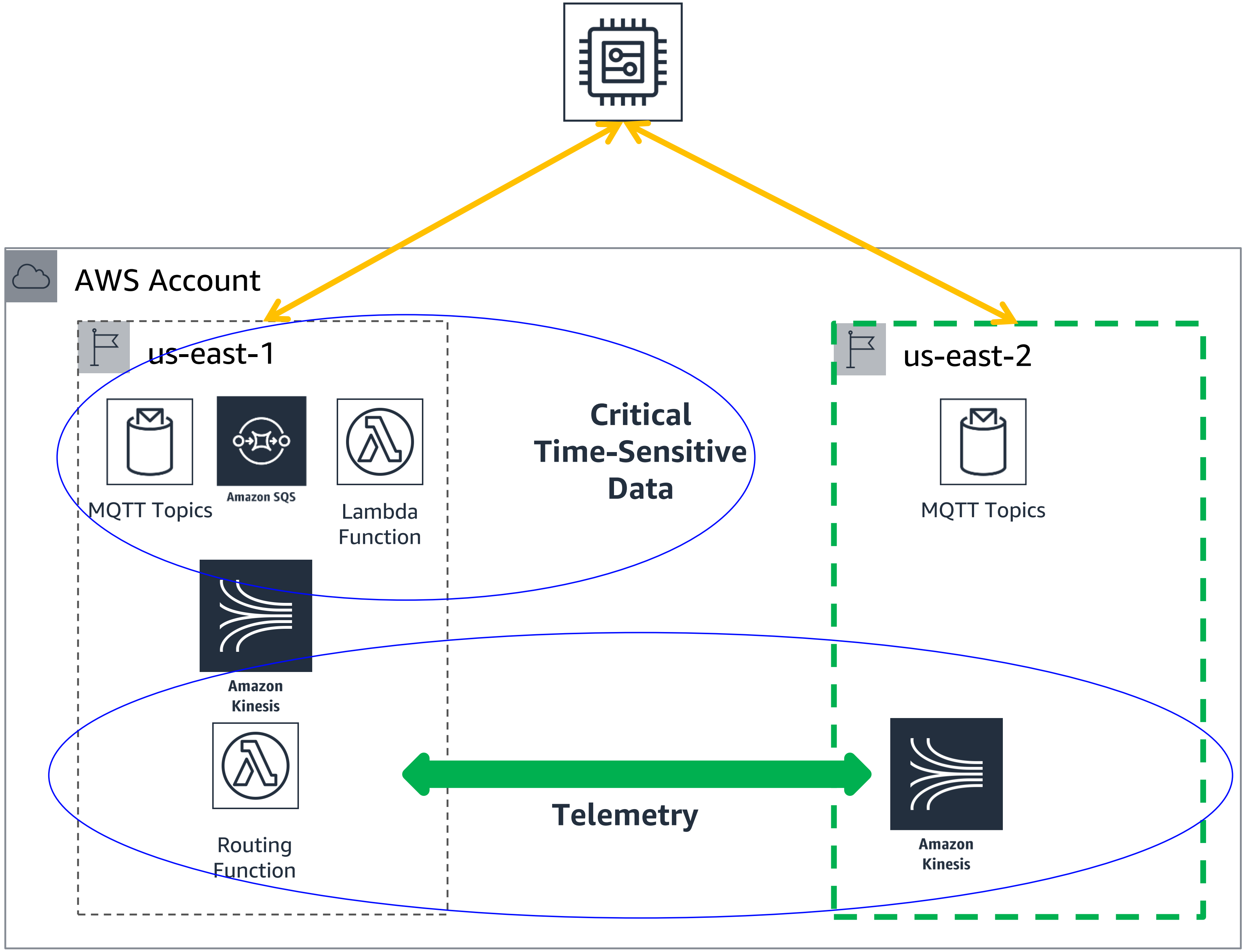
- Shadow data is updated in one region
- Syncs Shadow on region cutover
- Publishes data to connected region



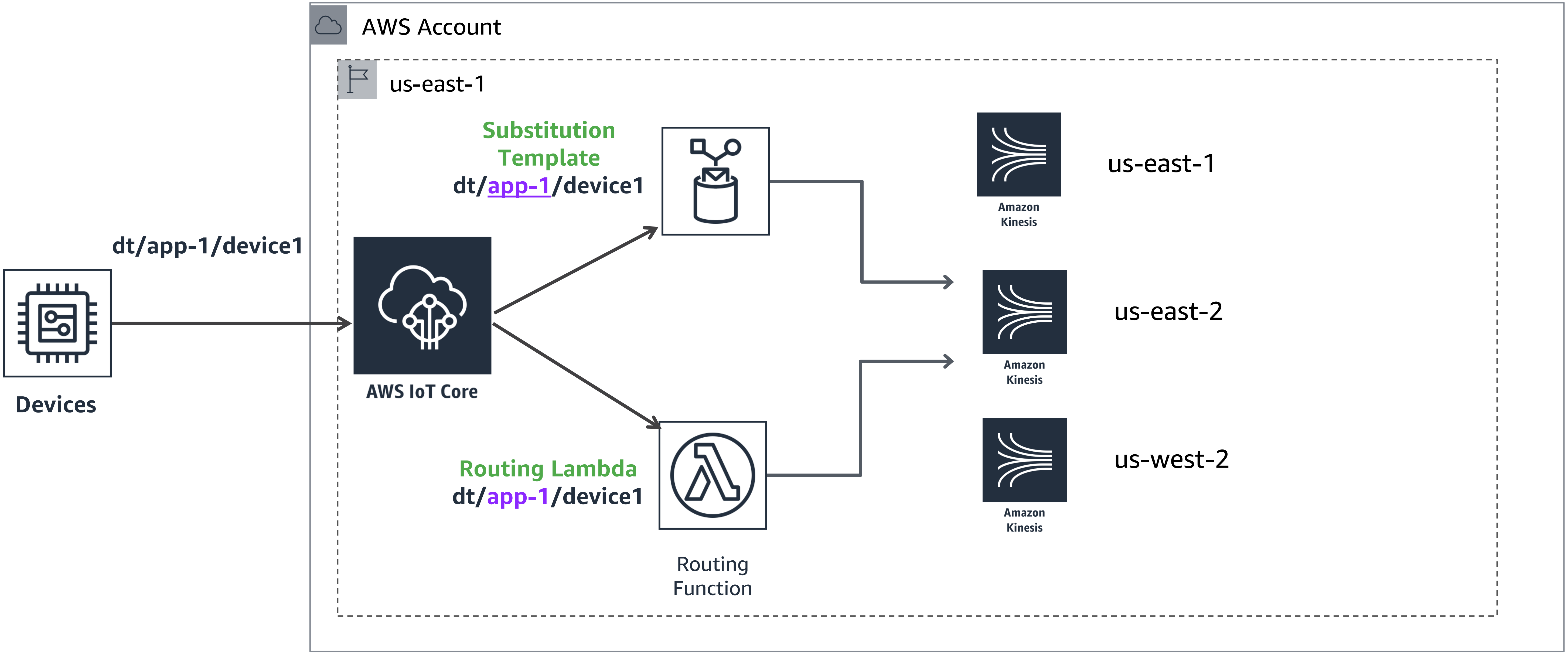
Active/Active – Process Data Once

Considerations:

- Criticality and Latency
- Telemetry
- Internal and External Consumers



Active/Active – Process Data Once



Active/Active – Process Data Once

IoT Rule Engine

Arn: ${Substitution Template}

AWS Lambda

SELECT *, topic(2) as applicationId

Kinesis Action

```
streamName: "${case topic(2)
                  when 'app-1' then 'east1kinesis'
                 when 'app-2' then 'east2kinesis'
                 when 'app-3' then 'west2kinesis'
                 end
                 }"
```

Pseudo Code

```
var appId = event.applicationId
Route route = routingService.findRoutingForId(appId)
route.sendEvent(event)
```

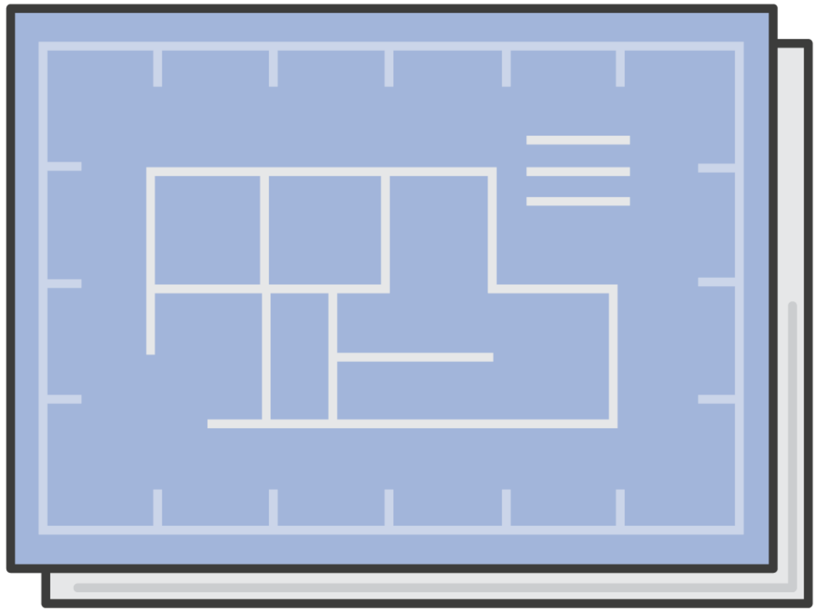
Implement Cutover Strategy

Cloud Cutover

- Send a command to the device containing the endpoint, server certificates, and configuration settings

Device Cutover

- Device has multiple IoT endpoints in firmware and swaps between endpoints after multiple errors



Multi-region choices

Active/Passive

- Easier to implement
- Replicate device registry
- Devices connect to primary region in group
- Shadow data is sticky to a region
- Idle capacity on Amazon EC2

Active/Active

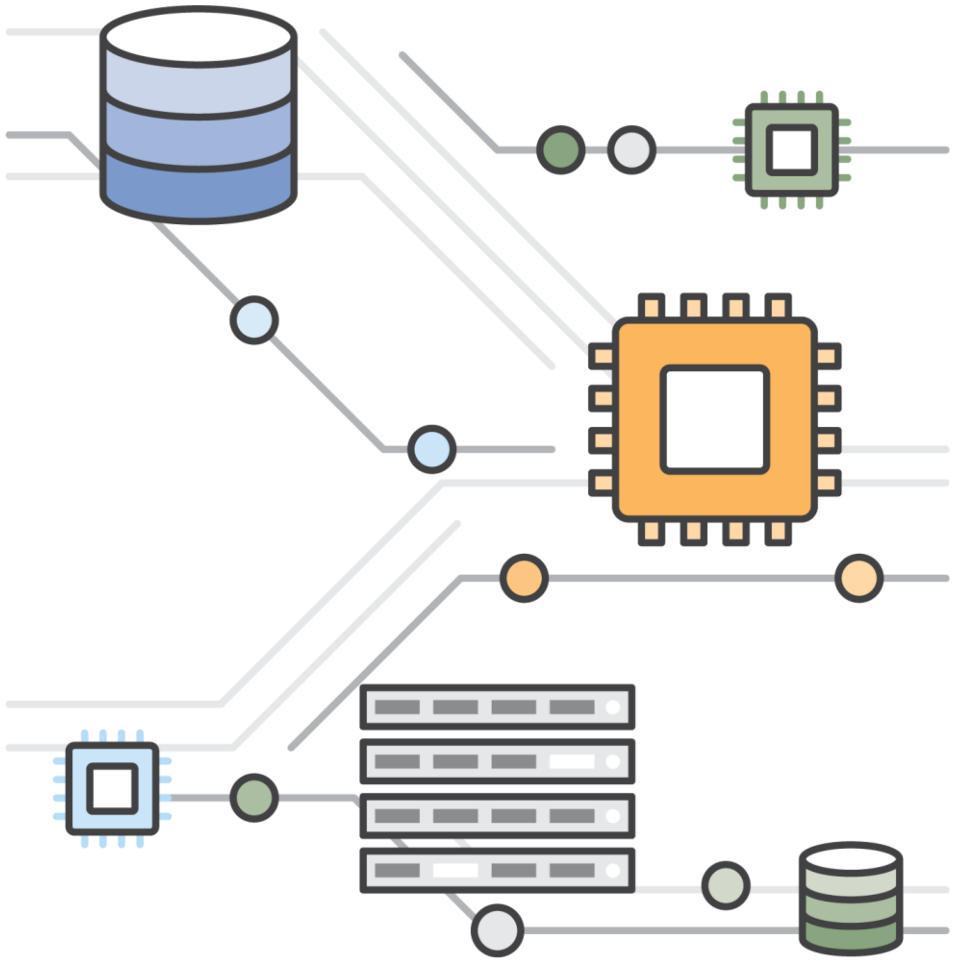
- More difficult to implement
- Replicate device registry
- Devices connect to any region in group
- Shadow data is periodically synced
- Telemetry data can be routed across region

Summary

Multi-Region Considerations

Table Stakes for Multi-Region

Active/Passive and Active/Active



Thank you!

Olawale Oladehin
Sr. Solutions Architect
AWS

Lucas Starrett
Cloud Solutions Architect
Analog Devices

Please complete the session survey in the mobile app.

AWS re:Invent

aws