CLOUD COMPUTING

Analysis of a Technical Document

IoT Smart Home Application in the Cloud

Dr. Marlène Seif

Ahmad Hussein

Thursday, November 16, 2023

OUTLINE

What is required for the assignment

USE CASE & ELEMENTS SHARED

FUNCTIONAL & TECHNICAL CHAINS

DETAILS OF TWO TECHNO COMPONENTS

DEPLOYMENT AND SERVICE MODELS

CONCLUSION AND CRITIQUE

WIDER USE CASES FOR ALEXA

*Based on a prototype made and posted by Dan Isla on youtube
https://www.youtube.com/watch?v=tZD7qFJSjJ0

USE CASE

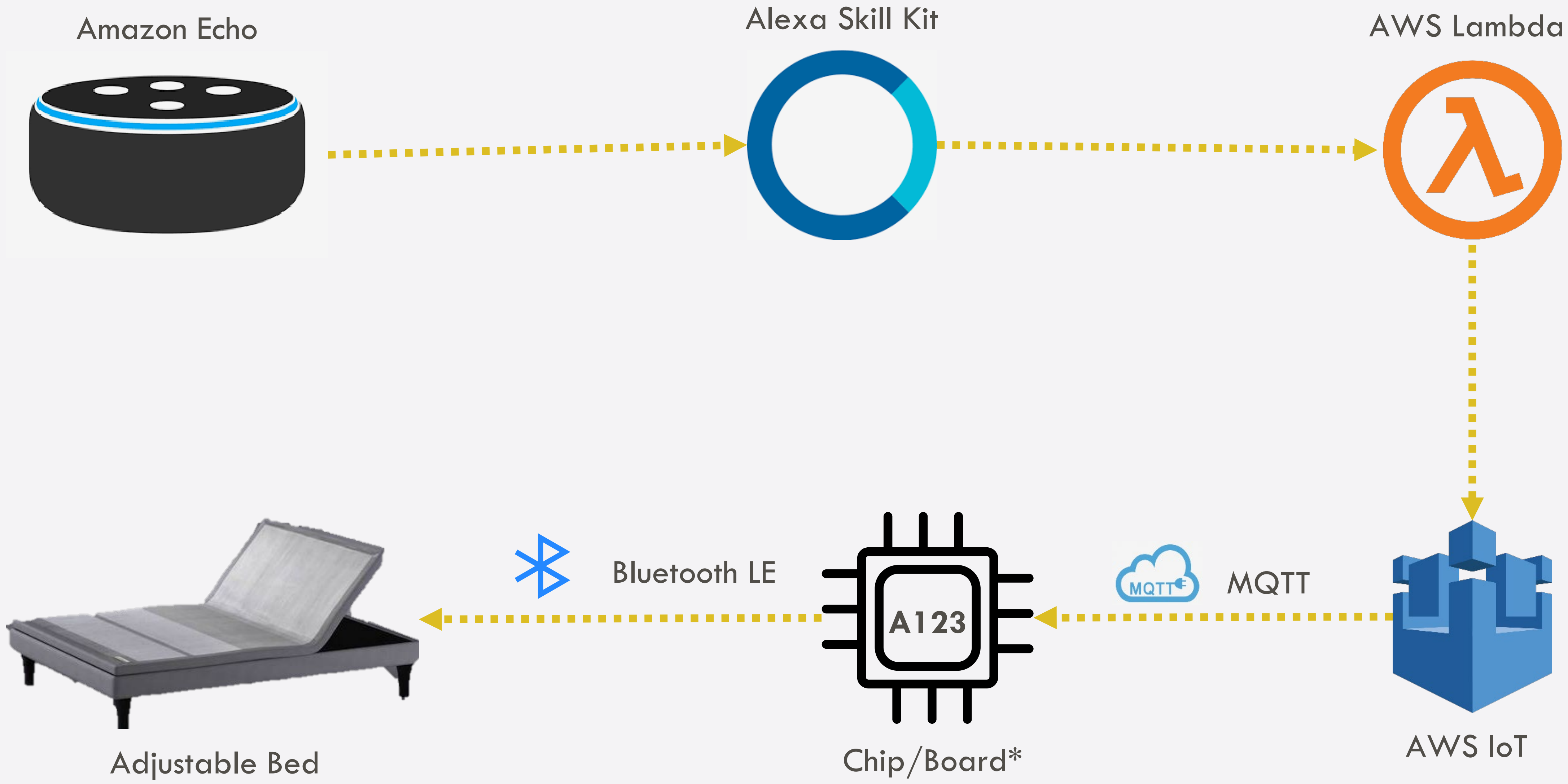
Use Case Explained

- □ This use case aim to show how to **control an adjustable bed with voice commands.**
- □ The adjustable bed used in this use case is the **Serta Motion Perfect III** adjustable base bed built by **Ergomotion** and sold by **Serta**.
- □ The bed is designed initially from the manufacturer to be controlled via Bluetooth LE from either the provided **remote controls** or the **mobile app** for Android and iOS.

- □ The business case behind this project is to give the user more **flexibility in controlling the bed** without requiring him to move around to find the remote and hunt for the buttons in the dark or to find his phone, unlock it, open the bed control app, pair the device with the bed and then start adjusting the bed!

ELEMENTS SHARED

Different Elements Shared by Dan Isla in this use case



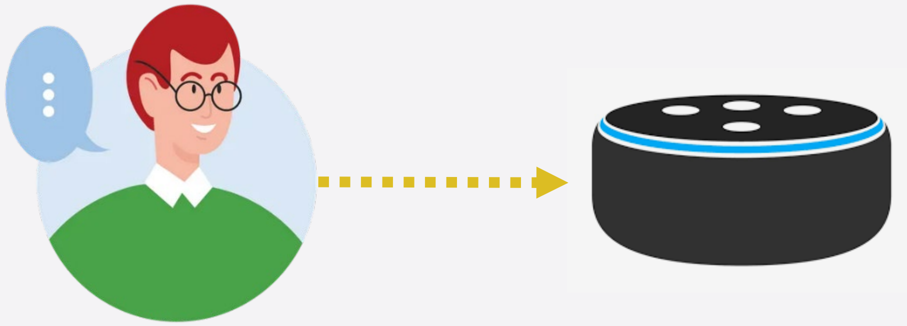
*A123 is just an example of a unique chip # for the bed for this user registered at AWS IOT in a topic.

FUNCTIONAL CHAIN

End-to-End Functional Chain of this achievement | 1

□ Step 1: Activate User

The user activates the system by a voice command to Amazon Echo. For example, the user, **Mike**, say "Alexa..", consequently, the user activates Echo. Then, he mentions the desired bed function. For example, "Tell bed to turn head and foot massage on."



"**Alexa**, tell **bed** to **turn head and foot massage on.**"

□ Step 2: Receive Route & Voice Command

Echo receives the voice command and sends it to the Alexa Skills Kit (ASK) cloud service.

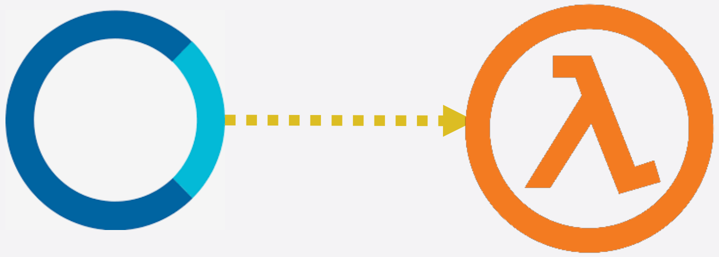
"New **voice** command from **Mike.**"

FUNCTIONAL CHAIN

End-to-End Functional Chain of this achievement | 2

□ Step 3: Translate Voice Command

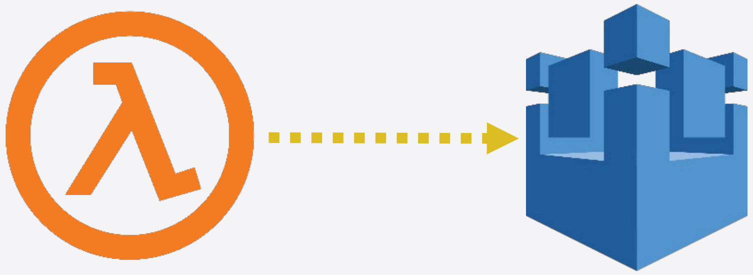
ASK parses the voice command and determines the desired action. In this case, the desired action is to turn massage on. Then, it triggers an AWS Lambda function that is responsible for controlling the user’s bed.



“**Mike** wants to **turn head and foot massage on** in his **bed.**”

□ Step 4: Find the User’s Specific Bed

The Lambda function uses the AWS IoT Core service to find the correct user’s bed chip's topic previously connected with it to know where to send the message of the user.



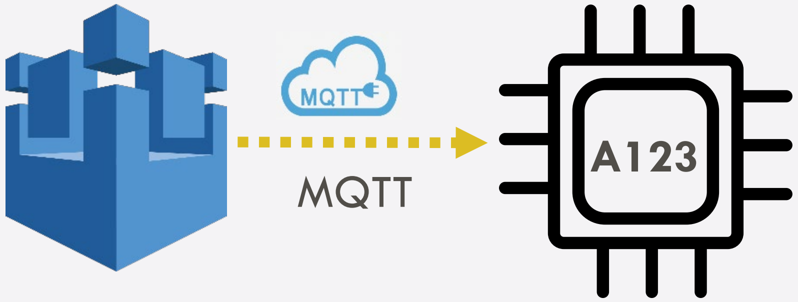
“I need to send a message to the **bed of Mike** connected to your service.”

FUNCTIONAL CHAIN

End-to-End Functional Chain of this achievement | 3

□ Step 5: Send Command to User's Bed Chip

The Lambda function uses the AWS IoT Core service to publish an MQTT message to the bed chip's topic. When the bed chip receives an MQTT message, it parses the message to determine the desired action. The MQTT message should contains a command to turn head and foot massage on.

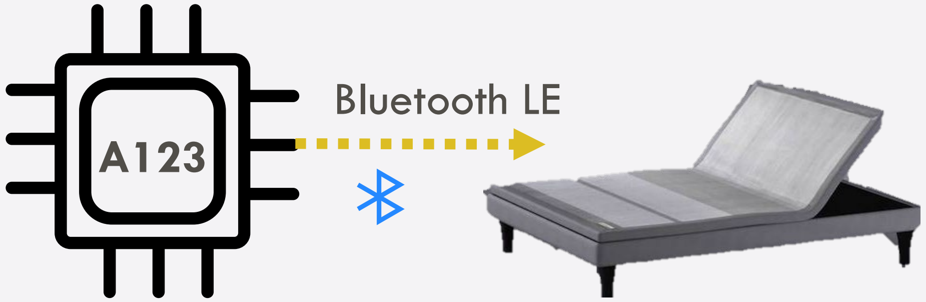


"Chip A123, turn head and foot massage on."

□ Step 6: Send Command to the Bed

The bed's chip send the command to the bed via Bluethooth LE, just like the command is being transmitted through the Remote Controller button.

*e5fe160001000005 → Head and Foot Massage On



"e5fe160001000005"

TECHNICAL CHAIN

End-to-End Technical Chain of this achievement | 1

□ Step 1: Extract the BLE transmitted code of the Remote-Control buttons

e5fe1600080000fe → Massage Head Add

e5fe160000800086 → Massage Head Min

e5fe1600000008fe → Flat Preset

e5fe1600100000f6 → ZeroG Preset

e5fe160004000002 → Massage Foot Add

e5fe160000000105 → Massage Foot Min

e5fe160001000005 → Head and Foot Massage On

e5fe160002000004 → Massage Timer

e5fe160100000005 → Lift Head

e5fe160200000004 → Lower Head

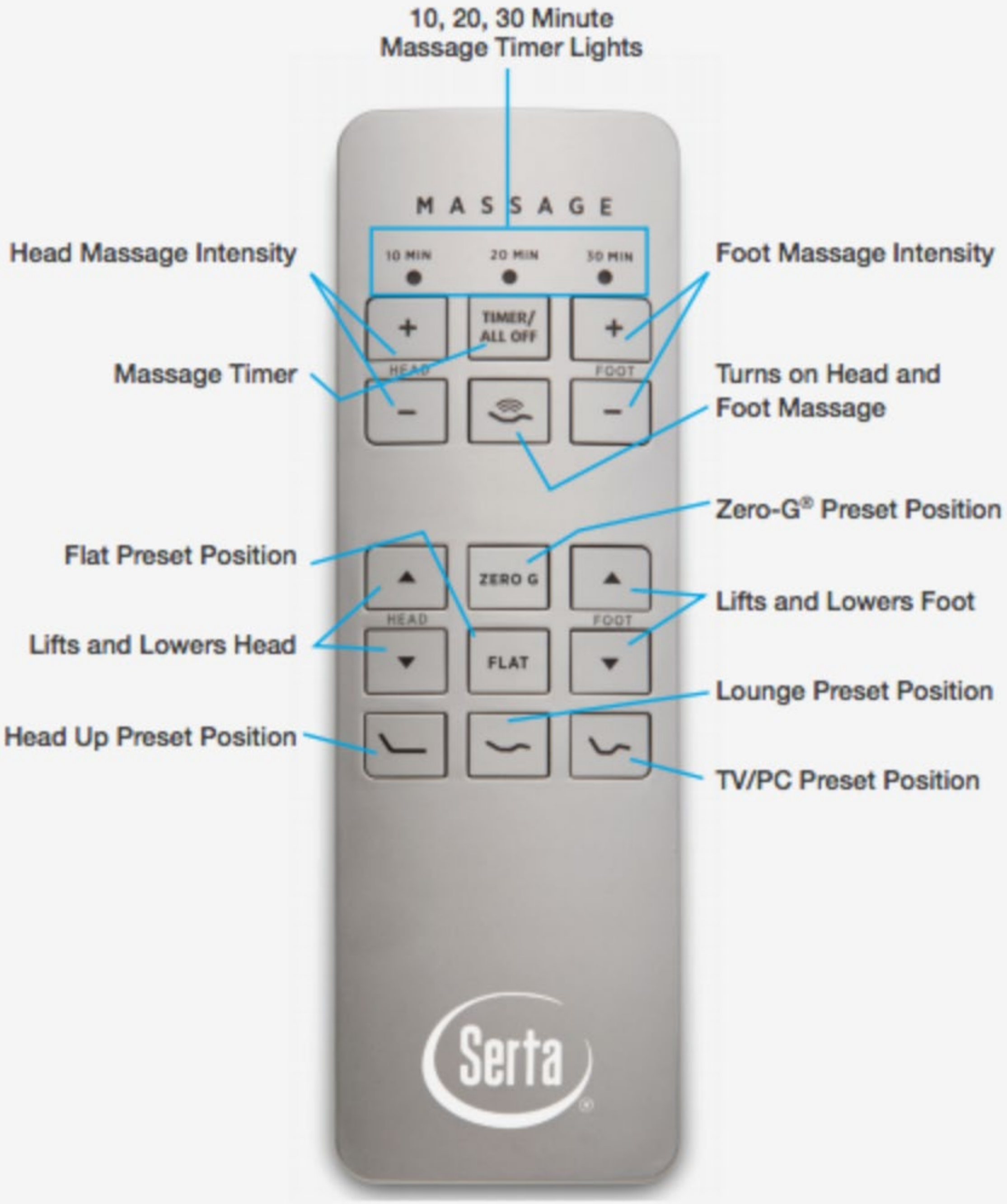
e5fe160400000002 → Lift Foot

e5fe1608000000fe → Lower Foot

e5fe160080000086 → Head Up Preset

e5fe1600200000e6 → Lounge Preset

e5fe1600400000c6 → TV/PC Preset



TECHNICAL CHAIN

End-to-End Technical Chain of this achievement | 2

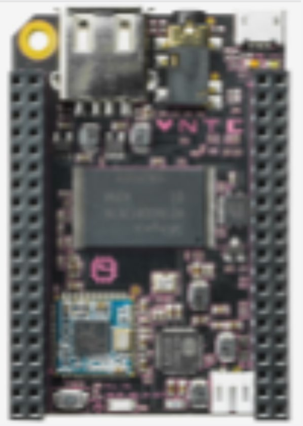
□ **Step 2: Control the Bed with BLE**

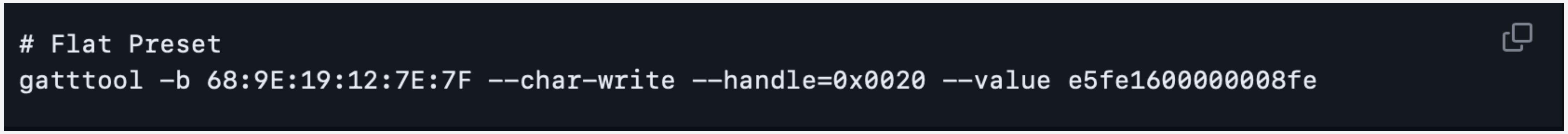
+ The **Raspberry Pi 3** and **CHIP** single board computers have an integrated Bluetooth LE module that can be used to control the base of the bed.

+ Scanning for the **base** and obtaining the **MAC** address of it.

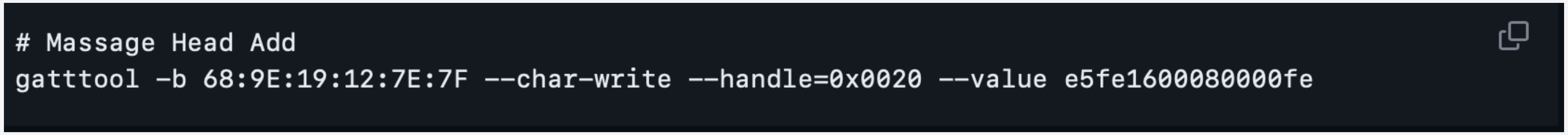
+ **Captured values** (in previous slide) should be written to handle and confirm that it controls the bed.

+ Doing some **testing** like sending some of the captured commands to the device.





```
# Flat Preset
gatttool -b 68:9E:19:12:7E:7F --char-write --handle=0x0020 --value e5fe1600000008fe
```



```
# Massage Head Add
gatttool -b 68:9E:19:12:7E:7F --char-write --handle=0x0020 --value e5fe1600080000fe
```

TECHNICAL CHAIN

End-to-End Technical Chain of this achievement | 3

□ Step 3: Registration of the Bed on AWS IoT

+ Set environment variables

+ Create the certificates

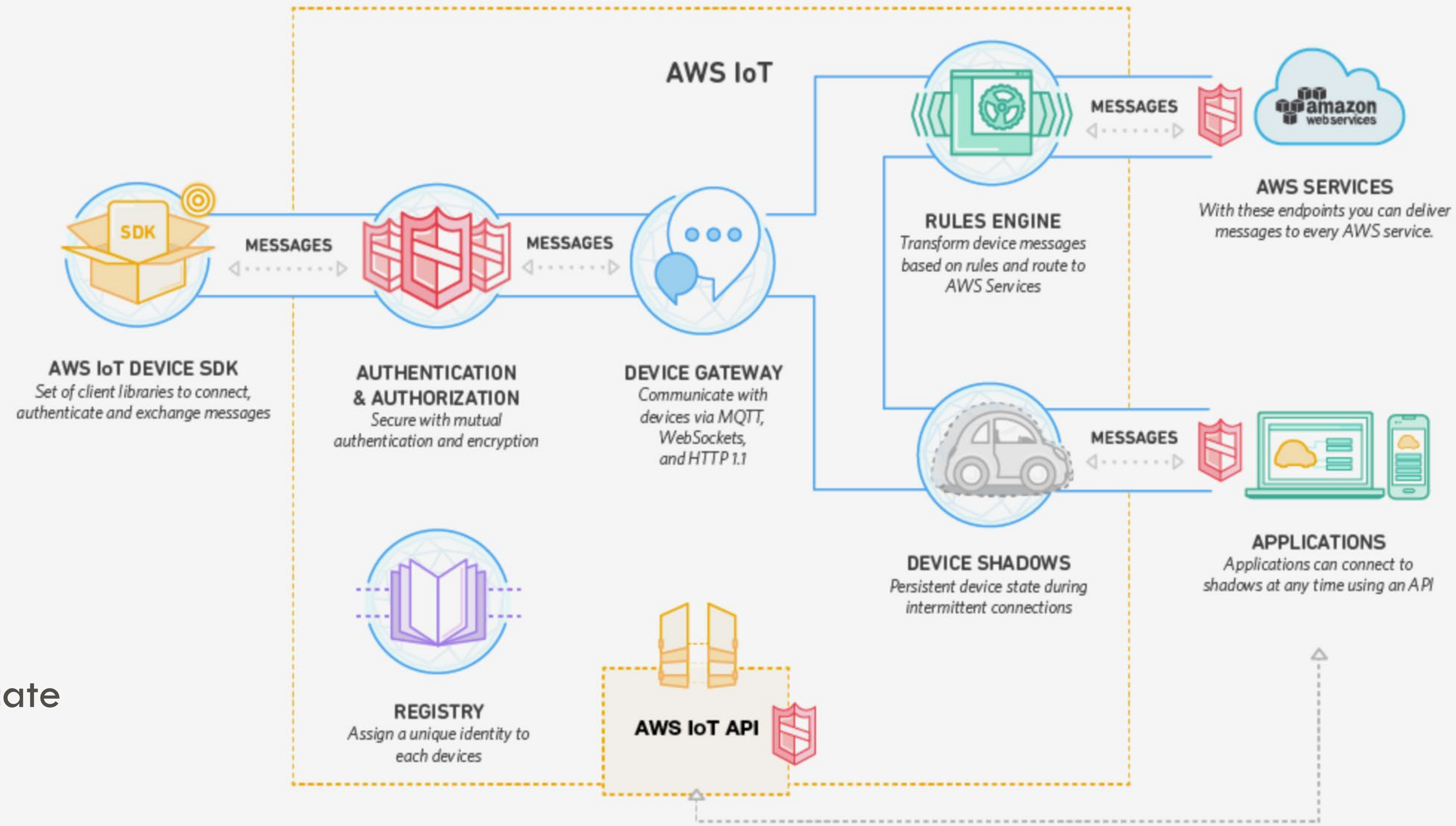
+ Create the IoT Thing

+ Connect cert. to thing

+ Create the policy

+ Attach the policy

+ Download the root certificate


AWS IoT
SDK
MESSAGES
MESSAGES
MESSAGES
MESSAGES
amazon web services
AWS IoT DEVICE SDK
Set of client libraries to connect, authenticate and exchange messages
AUTHENTICATION & AUTHORIZATION
Secure with mutual authentication and encryption
DEVICE GATEWAY
Communicate with devices via MQTT, WebSockets, and HTTP 1.1
RULES ENGINE
Transform device messages based on rules and route to AWS Services
AWS SERVICES
With these endpoints you can deliver messages to every AWS service.
DEVICE SHADOWS
Persistent device state during intermittent connections
APPLICATIONS
Applications can connect to shadows at any time using an API
REGISTRY
Assign a unique identity to each devices
AWS IoT API

TECHNICAL CHAIN

End-to-End Technical Chain of this achievement | 4

□ Step 4: Create AWS Lambda Function to Update IoT Thing Shadow

This lambda function will be called by the Alexa Skills SDK after the voice intent has been processed in the AVS model. Based on the incoming intent, the function will update the device shadow desired state which will notify the controller via MQTT:

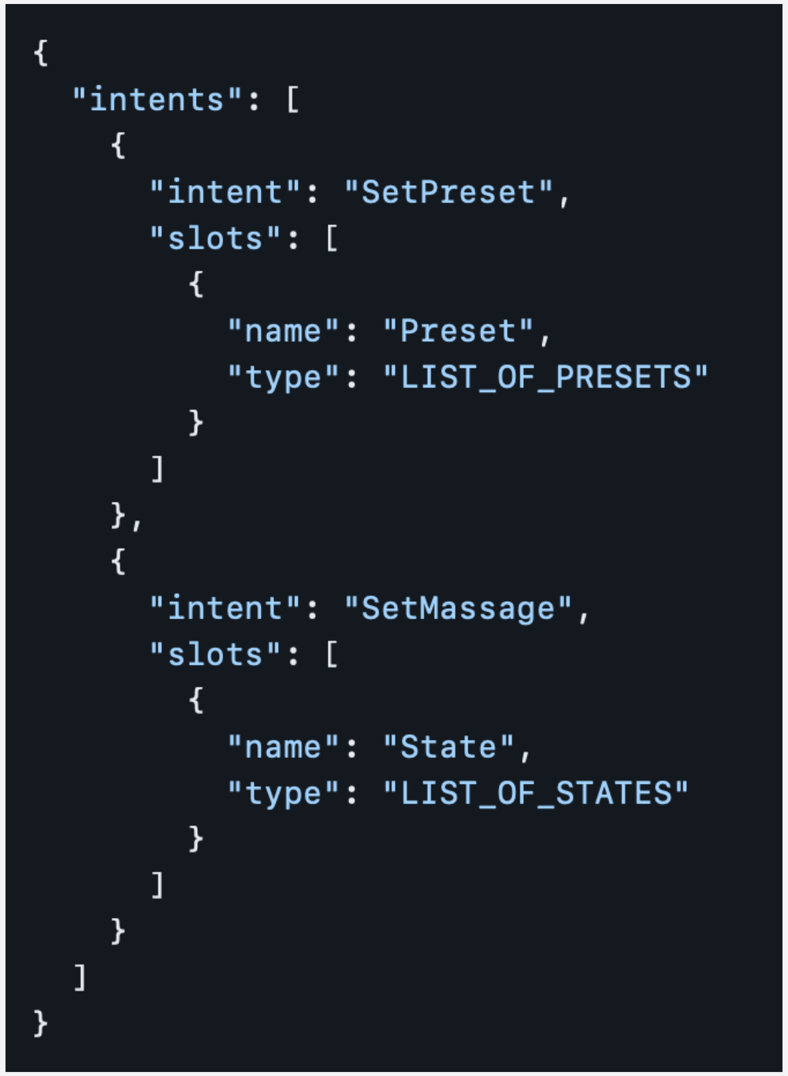
+ Set environment variables
+ Create an execution role
+ Attach policy to role that allows it to update the shadow
+ Update the source with the corresponding IoT endpoint
+ Create lambda code zip file
+ Create the lambda function
+ Getting the logs from Cloudwatch Logs

TECHNICAL CHAIN

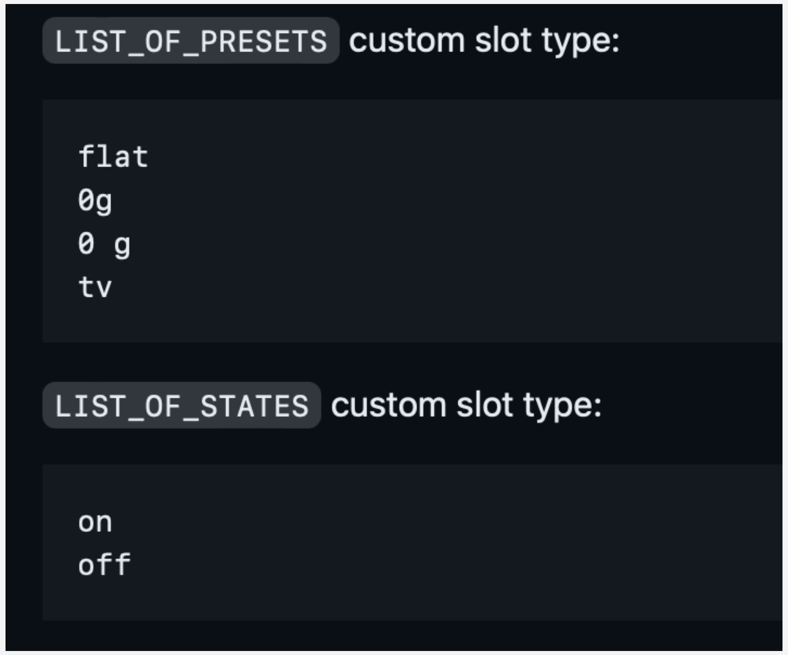
End-to-End Technical Chain of this achievement | 5

□ **Step 5: Create Alexa Custom Skill to Understand Bed Commands**

+ Create Intent Mapping Schema & Sample Utterances Files



```
{
  "intents": [
    {
      "intent": "SetPreset",
      "slots": [
        {
          "name": "Preset",
          "type": "LIST_OF_PRESETS"
        }
      ]
    },
    {
      "intent": "SetMassage",
      "slots": [
        {
          "name": "State",
          "type": "LIST_OF_STATES"
        }
      ]
    }
  ]
}
```

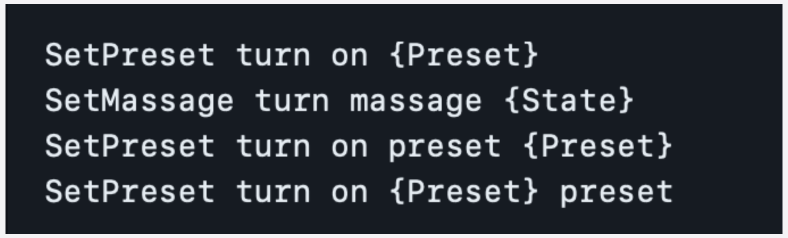


`LIST_OF_PRESETS` custom slot type:

```
flat
0g
0 g
tv
```

`LIST_OF_STATES` custom slot type:

```
on
off
```



```
SetPreset turn on {Preset}
SetMassage turn massage {State}
SetPreset turn on preset {Preset}
SetPreset turn on {Preset} preset
```

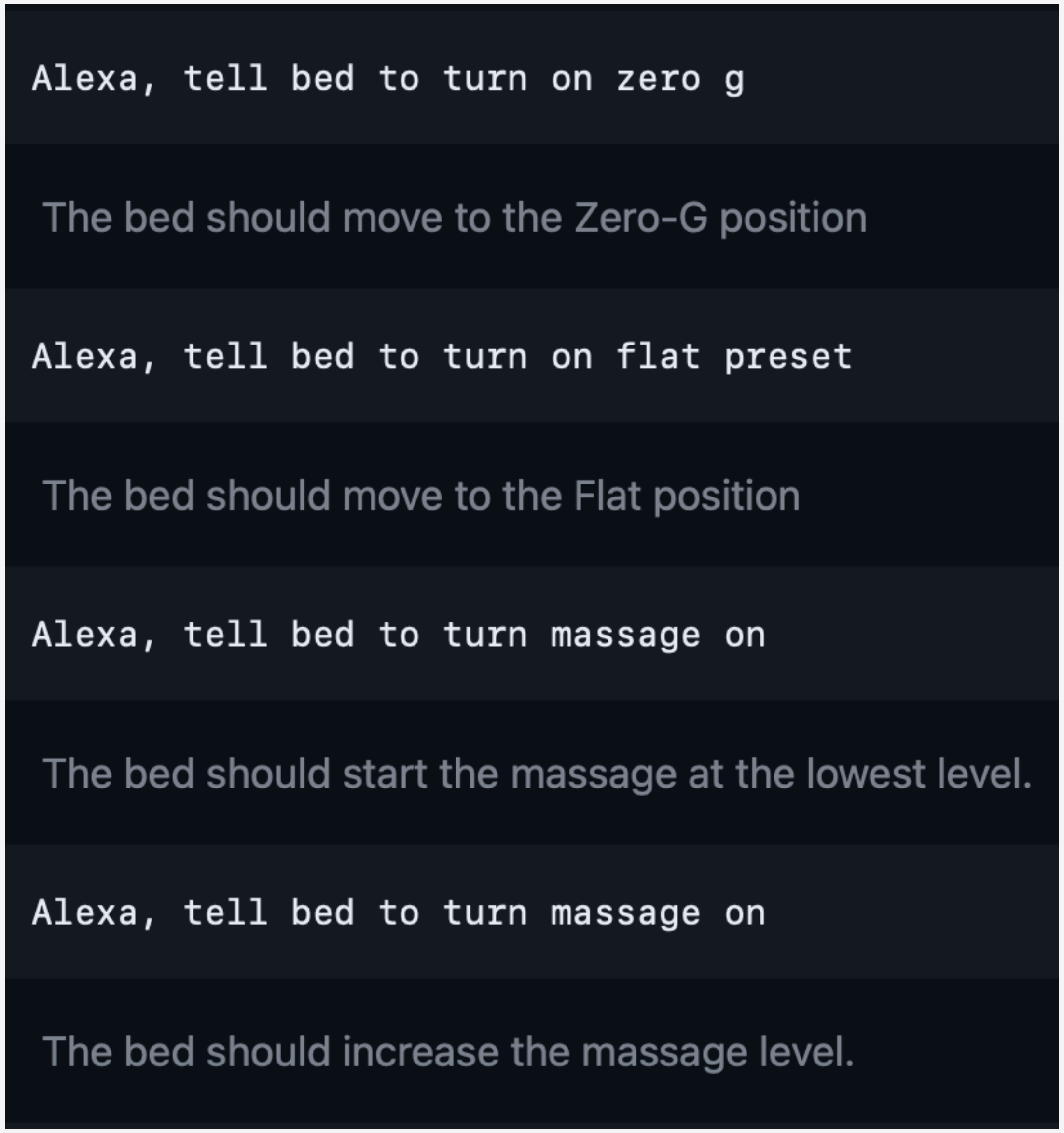
TECHNICAL CHAIN

End-to-End Technical Chain of this achievement | 5 (Cont'd)

□ **Step 5: Create Alexa Custom Skill to Understand Bed Commands (Cont'd)**

+ Create Skill

+ Testing



```
Alexa, tell bed to turn on zero g
```

The bed should move to the Zero-G position

```
Alexa, tell bed to turn on flat preset
```

The bed should move to the Flat position

```
Alexa, tell bed to turn massage on
```

The bed should start the massage at the lowest level.

```
Alexa, tell bed to turn massage on
```

The bed should increase the massage level.

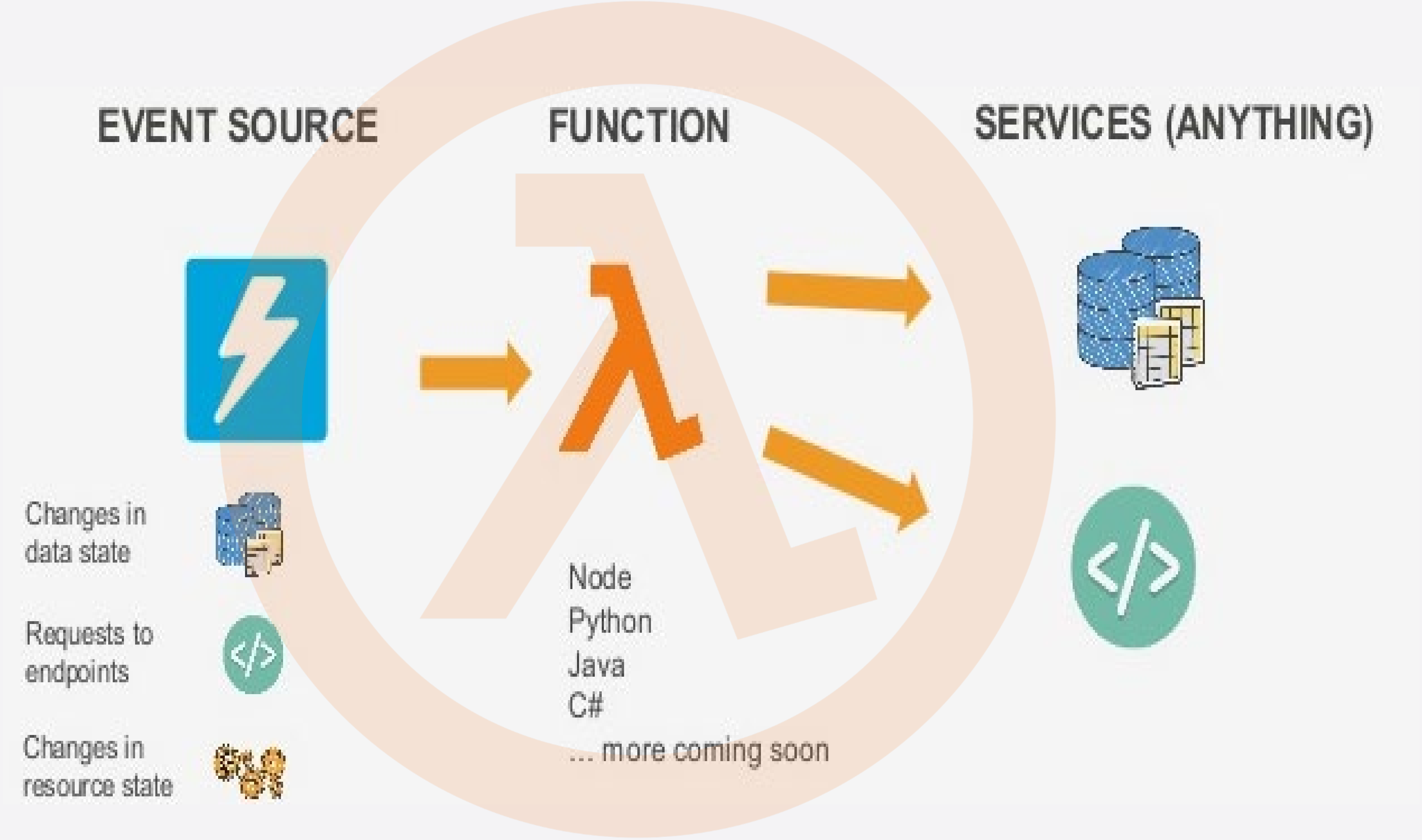
TECHNO COMPONENT

AWS Lambda Function | Definition

- AWS Lambda is a compute service that lets you run code without provisioning or managing servers.
- Lambda runs your code on a high-availability compute infrastructure and performs all the administration of the compute resources, including server and operating system maintenance, capacity provisioning and automatic scaling, and logging. With Lambda, all you need to do is supply your code in one of the language runtimes that Lambda supports.
- You organize your code into Lambda functions. The Lambda service runs your function only when needed and scales automatically. You only pay for the compute time that you consume.

TECHNO COMPONENT

AWS Lambda Function | Definition (Cont'd)



TECHNO COMPONENT

AWS Lambda Function | Main Features

□ Event-driven compute : Lambda is event-driven. It executes code in response to events, such as HTTP requests, change in data, or messages from other AWS services.

□ Multiple language support: lambda supports various programming languages like python, Node.js, java, C# and more.

□ Statelessness: functions are stateless, meaning no server affinity is maintained between invocation.

□ Automatic scaling: AWS Lambda automatically scales your applications based on the number of incoming events.

□ Pay-as-you go pricing: you are billed based on the number of requests and the execution time of your code.

TECHNO COMPONENT

AWS Lambda Function | Use Cases

□ Real-time data processing: Lambda can process and analyze data in real-time, making it ideal for IoT application, log processing, and data transformation.

□ Backend for mobile and web apps: Server as a backend for applications by handling authentication, database queries, and more.

□ Automation: automate tasks and workflows by triggering Lambda functions based on various events.

TECHNO COMPONENT

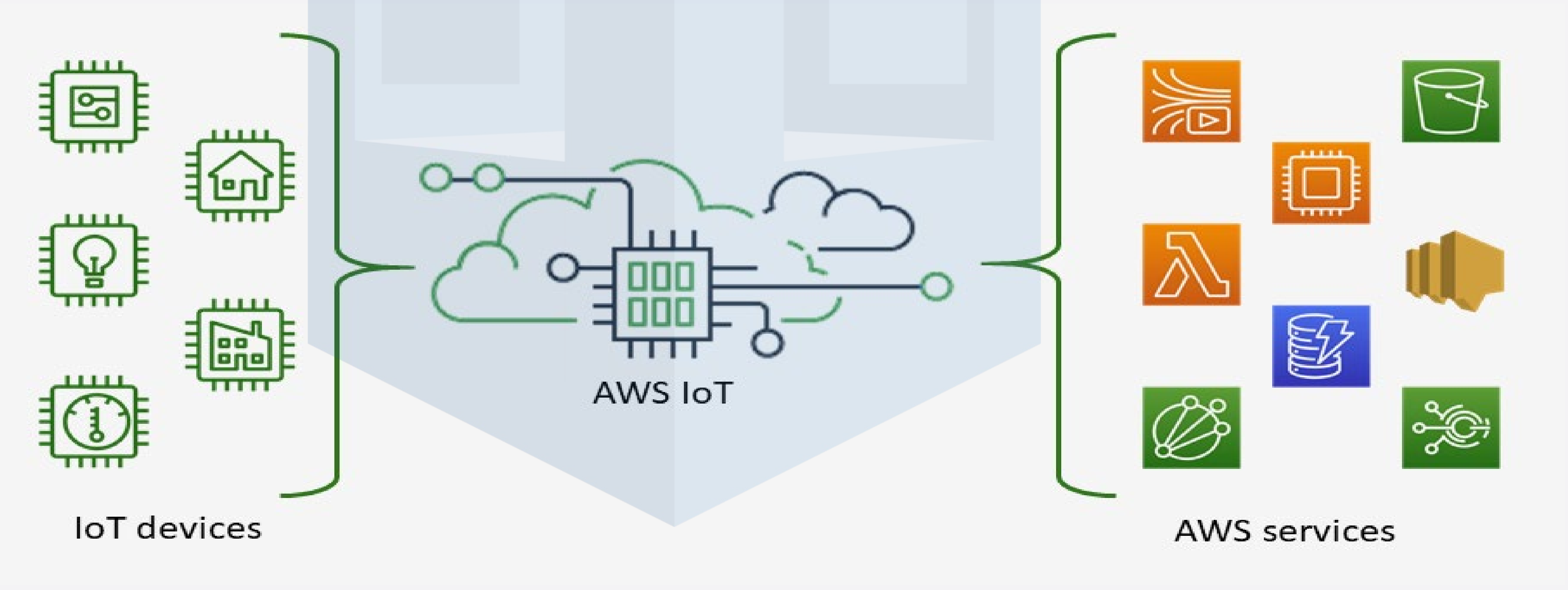
AWS Lambda Function | Technical Aspects

- Execution environment: Aws Lambda provides a runtime environment for your code. You package your code and its dependencies into a deployment package.
- Event sources: Events can originate from various sources like Amazon S3, API Gateway, AWS IoT, and more, Lambda responds to these events by invoking your function.
- Concurrent execution: multiple instance of your function can run simultaneously, based on the number of incoming events. AWS manage the execution environment.
- Integration: Lambda can integrate with other AWS services and third-party APIs. It can also used in AWS step Functions for complex workflows.
- Security: Lambda functions run within an execution role, which determines the permissions and access to AWS resources.

TECHNO COMPONENT

AWS IoT | Definition

□ AWS IoT provides the cloud services that connect your IoT devices to other devices and AWS cloud services. AWS IoT provides device software that can help you integrate your IoT devices into AWS IoT-based solutions. If your devices can connect to AWS IoT, AWS IoT can connect them to the cloud services that AWS provides.



TECHNO COMPONENT

AWS IoT | Main Features

- Device registry: AWS IoT maintains a registry of all connected devices, managing metadata and device identities.
- Secure communication: Supports secure device communication using protocols like MQTT and HTTP.
- Rules engine: AWS IoT Rules engine allows for routing and processing device data to other AWS services.
- Device shadows: Provides virtual representations of physical devices, allowing for control and synchronization of device states.

TECHNO COMPONENT

AWS IoT | Use Cases

□ Industrial IoT: Monitor and control industrial equipment, optimize operations, and enable predictive maintenance.

□ Smart home and building: connect and manage smart devices in homes, building, and cities.

□ Healthcare: enable remote patient monitoring, asset tracking, and medication management.

TECHNO COMPONENT

AWS IoT | Technical Aspects

- □ Device SDKs: AWS IoT device SDKs are available for various programming languages to facilitate device integration.
- □ Device certificates: secure device communication is achieved using X.509 certificates and policies.
- □ MQTT protocol: devices communicate with AWS IoT using the MQTT (message querying telemetry transport) protocol.
- □ Integration: IoT data can integrated with AWS services like AWS Lambda, AWS kinesis, and Amazon S3 for data analysis and storage.
- □ Device management: AWS IoT provides tools for device onboarding, authentication, and over-the-air (OTA) updates.

SERVICES USED

ALEXA VOICE SERVICE

Deployment: Public Cloud because it is owned, hosted, managed and operated by AWS, a public cloud service provider. Moreover, it can be used by public users.

Service: PaaS because it is like a backend API on AWS that the developers can use to translate voice to text.

SERVICES USED

ALEXA SKILL KIT

Deployment: Public Cloud because it is owned, hosted, managed and operated by AWS, a public cloud service provider. Moreover, it can be used by public users.

Service: PaaS because it is a software development kit (SDK) on AWS that the developers can use to build conversational applications (skills).

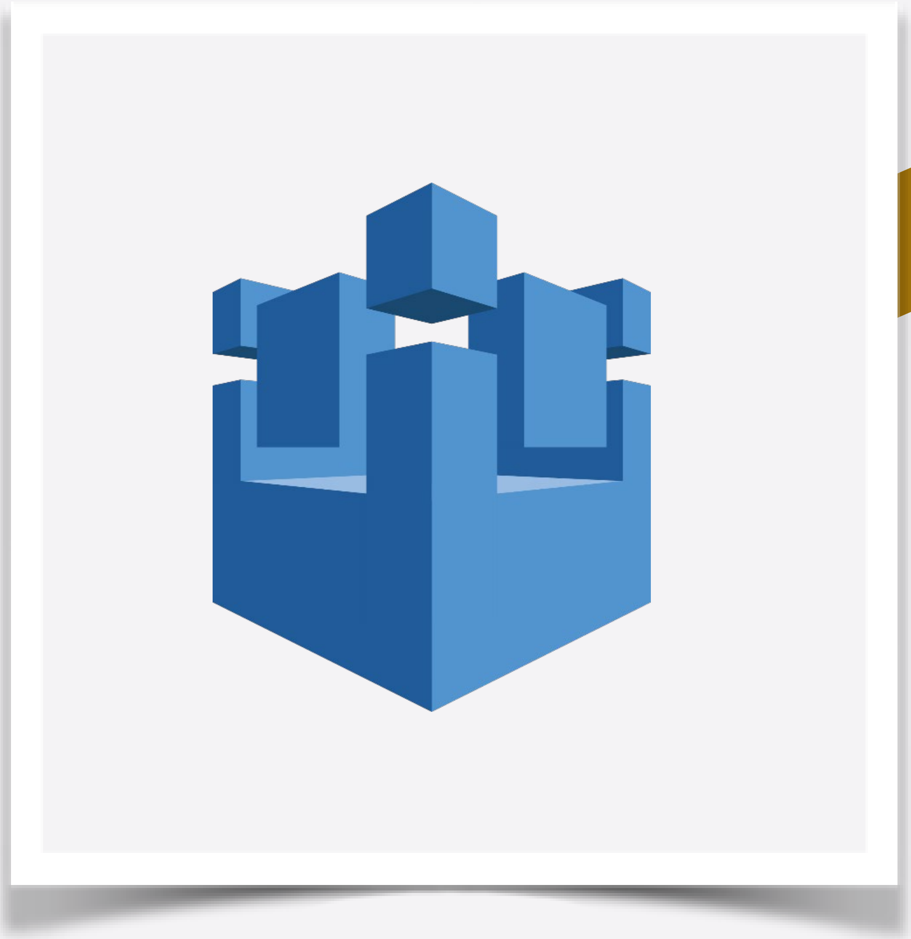
SERVICES USED

AWS LAMBDA

Deployment: Public Cloud because it is owned, hosted, managed and operated by AWS, a public cloud service provider. Moreover, it can be used by public users.

Service: PaaS because it is a platform in which users can run their code on a compute infrastructure and performs the administration of the compute resources.

SERVICES USED



AWS IOT

Deployment: Public Cloud because it is owned, hosted, managed and operated by AWS, a public cloud service provider. Moreover, it can be used by public users.

Service: PaaS because it is a platform that developers can use to create devices as things and connect them easily and securely with cloud applications and with each other.

CONCLUSION & CRITIQUE

The capabilities of AWS IoT is magical. The developers can ***connect any device that can be controlled through a chip to the IoT cloud****, register it, define its possible* ***actions*** *and then activate it with Amazon Alexa voice commands (or any other method). In the use case, it was an example of a bed but if we think about the possible devices to control from the cloud, those are numerous.*

However, we think that a use case of controlling ***only one bed at a time*** *from one Echo device is a limitation. The reason will be explained in the coming section. Therefore, one possible architecture enhancement would be to define more than one bed as a "Thing" into AWS IoT, taking the example of a children room of 2 beds and one Alexa device. Each one of the kids should be able to ask Echo to adjust his own bed by naming it.*

USE CASES FOR ALEXA

Wider Use Cases | 1 | Elderly House Care

□ **Voice Control on Chairs in Elderly House Care**

+ One room can have one or many persons on chairs

+ Nurses are responsible of taking care of elderly people in a batch and sometimes they will have dirty hands or full hands when feeding a person

+ Nurses should be able to control the position of the chair for elderly people for a better feeding experience or when moving from one to another

+ Nurses can tell Alexa Echo to adjust the chair of a certain patient to a specific configuration without the need to click any button

USE CASES FOR ALEXA

Wider Use Cases | 2 | Neonatal Intensive Care Unit

□ **Sending urgency alarm when a baby has difficulties in swallowing**

+ In NICUs, nurses take care of premature babies

+ Their tasks are very time-sensitive and they need to take immediate action when a baby mis-swallow milk

+ Invoking Alexa Echo into NICU to send an immediate call to the head of the NICU when a nurse asks Echo to do so while she take immediate action in helping the baby.

THANK YOU