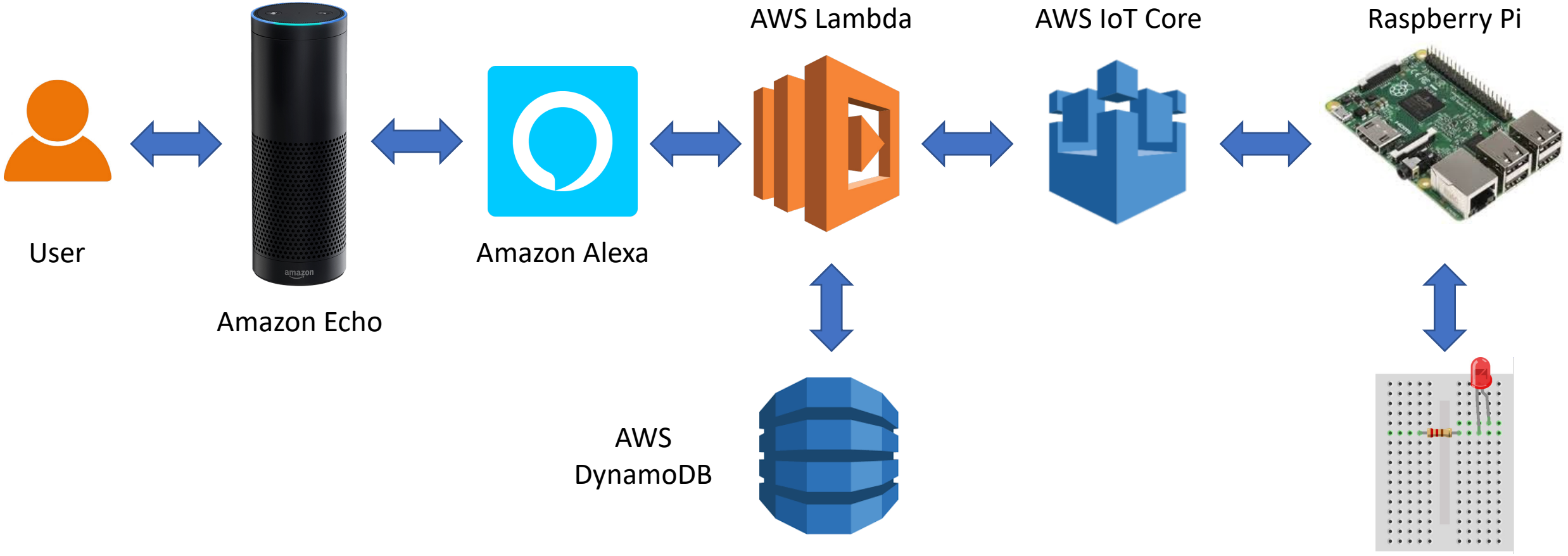
Control My Pi

An Amazon Alexa Skill to Read/Write GPIOs

Topics to Discuss…

- Alexa Introduction
- System Discussion
- Code Walkthrough
- System Setup
  - Alexa Developer Console
  - AWS IoT
  - AWS Lambda
  - AWS DynamoDB
  - Raspberry Pi

Basic Block Diagram



What do the various components do?

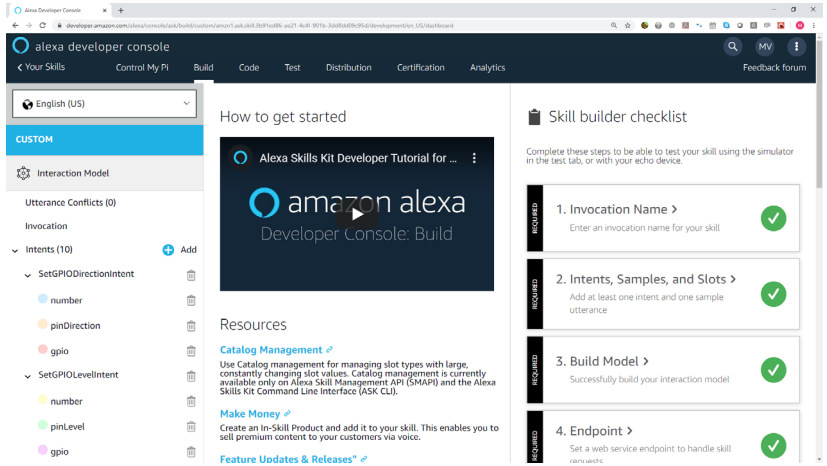
- **Amazon Echo/Alexa** – device/cloud based voice service. We create a skill that sends specific events/requests based on specific voice commands.
- **AWS Lambda** – event-driven, serverless computing platform (handles Alexa Intents)
- **AWS DynamoDB** – fully managed NoSQL database for storing Alexa user information (ie pin states)
- **AWS IoT Core** – Enables cloud/physical devices to communicate to one another via MQTT and device shadows
- **Raspberry Pi** – Single board computer with AWS IoT Core Python SDK that subsribes/publishes to MQTT Topics

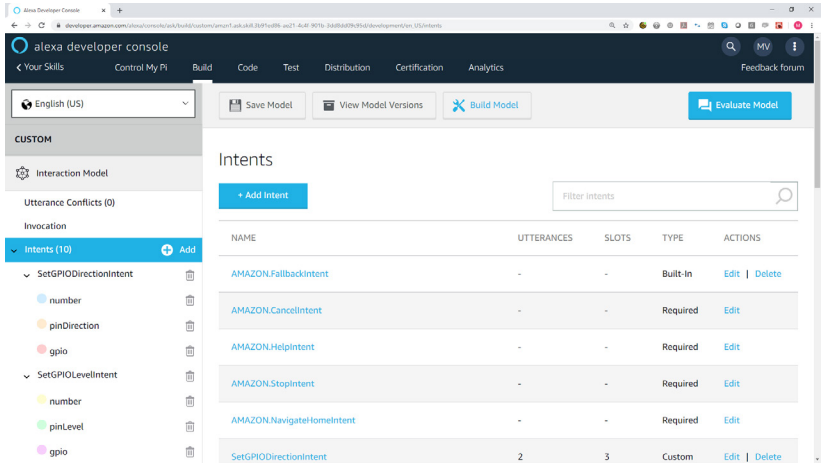
Amazon Echo / Alexa

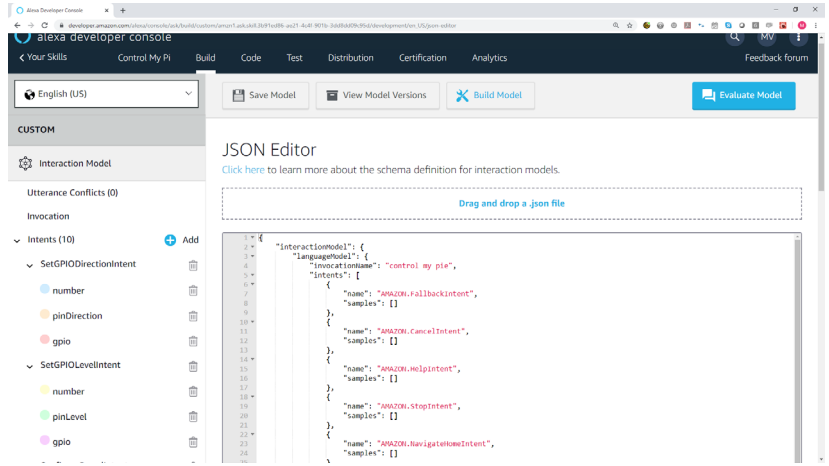
Create voice model that interprets user voice commands and sends JSON based events

- Interaction Model Consists of the following:
  - **Invocation** – name to begin interaction with custom skill
  - **Intents** – represent actions that fulfill user's spoken requests
    - Sample Utterances – what words/phrases trigger a specific event
    - Slots – words/phrases that represent variable information (commands/pins)
    - Slot Types - determines how the user input is handled and passed on to your skill
  - **Endpoint** – ARN/HTTPS location where events are set to

Amazon Echo / Alexa Developer Console







Control My Pi Custom Intents

- **SetGPIODirectionIntent** – configures the RPi pin as input or output

  slots: number, pinDirection, gpio
- **SetGPIOLevelIntent** – sets the RPi pin as High/Low

  slots: number, pinLevel, gpio
- **ConfigureBoardIntent** – configures what type of RPi is used (not enabled)
- **ReadGPIOLevelIntent** – reads the level from the RPi pin

  slots: number, gpio
- **ReadGPIODirectionIntent** – Reads the direction of the RPi pin (not enabled)

Control My Pi Utterances / slots

- **SetGPIOLevelIntent**

  set pin 4 to output → set {gpio} {number} to {pinLevel}

- **ReadGPIOLevelIntent**

  what is pin 5 set to? → what is {gpio} {number} set to?

  what is the level of GPIO 7? → what is the level of {gpio} {number}?

Slots

gpio: "pin" or "g.p.i.o."

number: AMZN.number

pinLevel: "high" or "low"

AWS Lambda Intent Handlers

Lambda responds to events sent from Alexa

- **LaunchRequestHandler** – called when the skill opens
- **SetGPIODirectionIntentHandler** – responds to requests to set direction
- **SetGPIOLevelIntentHandler** – responds to requests to set level
- **ReadGPIOLevelIntentHandler** – responds to requests to read GPIO level
- **YesIntentHandler**
- **NoIntentHandler**
- **HelpIntentHandler**
- **CancelAndStopIntentHandler**
- **SessionEndedRequestHandler**
- **ErrorHandler** – responds to errors

AWS DynamoDB

- Stores user state variables such as pin state
- Uses `ask-sdk-dynamodb-persistence-adapter` for easy integration into lambda code

AWS IoT Core

- Create `ControlMyPi` Thing that enables MQTT communication and stores shadow information
- Lambda publishes MQTT messages while Raspberry Pi Subscribes*
- Topic structure set up as: `controlmypi/command/pin`
- Commands are: `setgpiolevel`, `setgpiodirection`, `readgpiolevel`
- Raspberry Pi publishes to reserved message, `$aws/things/ControlMyPi/shadow/update`

AWS IoT Core – Thing Policy

- Need to make sure device has appropriate permissions…
- Eventually need to constrain the iot:UpdateThingShadow a bit further from just the wildcard…

```
{
  "Version": "2012-10-17",
  "Statement": [
    {
      "Effect": "Allow",
      "Action": [
        "iot:Publish",
        "iot:Receive"
      ],
      "Resource": [
        "arn:aws:iot:region:account:topic/sdk/test/java",
        "arn:aws:iot:region:account:topic/sdk/test/Python",
        "arn:aws:iot:region:account:topic/$aws/things/ControlMyPi/shadow/update",
        "arn:aws:iot:region:account:topic/controlmypi/*",
        "arn:aws:iot:region:account:topic/frommypi/*"
      ]
    },
    {
      "Effect": "Allow",
      "Action": "iot:UpdateThingShadow",
      "Resource": [
        "*"
      ]
    },
    {
      "Effect": "Allow",
      "Action": [
        "iot:Subscribe"
      ],
      "Resource": [
        "arn:aws:iot:region:account:topicfilter/sdk/test/java",
        "arn:aws:iot:region:account:topicfilter/sdk/test/Python",
        "arn:aws:iot:region:account:topicfilter/topic_1",
        "arn:aws:iot:region:account:topicfilter/controlmypi/*",
        "arn:aws:iot:region:account:topicfilter/frommypi/*"
      ]
    },
    {
      "Effect": "Allow",
      "Action": [
        "iot:Connect"
      ],
      "Resource": [
        "arn:aws:iot:region:account:client/sdk-java",
        "arn:aws:iot:region:account:client/controlmypi*",
        "arn:aws:iot:region:account:client/sdk-nodejs-*"
      ]
    }
  ]
}
```

Raspberry Pi

- **Written in Python** – Using AWS IoT Python SDK
- **Connects to AWS IoT** – device/cloud based voice service. We create a skill that sends specific events/requests based on specific voice commands.
- **Subscribes to MQTT Topics** – Lambda publishes commands to MQTT Topic that the RPi responds to.
- **Publishes to Reserved MQTT Topics** – Since Lambda cannot subscribe to IoT MQTT Topics, Rpi will publish to reserved MQTT topic that sets the devices thing shadow which Lambda can read.
- **Interfaces with GPIOs** – Based upon received commands, Rpi will set the pin direction and output as well as read the pin level.
- **Loop Forever** – Simple loop that sleeps → responds to events

Code/System Walkthrough/Demo

https://github.com/mvartani76/alexa-controlmypi-skill

AWS Alexa – What do we need to do?

- Make sure we have an Alexa developer account
- https://developer.amazon.com/en-US/alexa
- Navigate to Developer Console…
- https://developer.amazon.com/alexa/console/ask
- Create Skill
  - Invocation Name
  - Create/Build Model (Intents, Samples, and Slots)
  - Configure Endpoint (I used Lambda but could try Alexa Hosted)
  - Test

AWS IoT Core – What do we need to do?

- Make sure we have an AWS account
- Create a thing…
- Transfer start.sh script and appropriate keys/certificates to Raspberry Pi
- Make sure code uses correct thing name and topics
- Configure thing policy

AWS Lambda – What do we need to do?

- Make sure we have an AWS account
- Create a function (Node.js 10.x)
- Configure environment variable `AWS_IOT_ENDPOINT` to your aws iot endpoint referenced in lambda code
- Copy lambda ARN for Alexa

AWS DynamoDB – What do we need to do?

- Create a new table
- Table name can be whatever but you will need to reference this table name in the Lambda code
- Primary key will be `id`

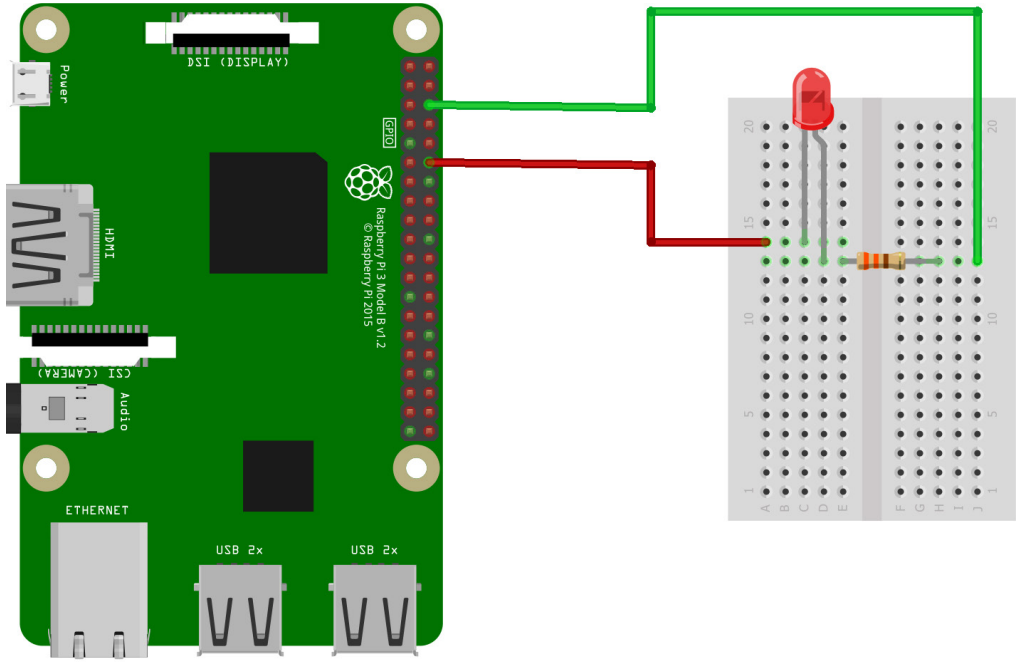
Raspberry Pi – What do we need to do?

- Make sure you have latest code
- Upload AWS IoT Core Certificates/Keys
- Build AWS SDK
- Ensure that MQTT reserved message matches your thing name
  - `$aws/things/ThingName/shadow/update`

- Connect LED Breadboard circuit to Raspberry Pi

How do we connect the LED to the Rpi?

Simple LED Circuit Connected to RPi



Make sure long end of LED connects to positive

Rpi 3 Model B GPIO Pinout Diagram

		Pin no.			
DC Power	3.3V	1	2	5V	DC Power
SDA1, I²C	GPIO 2	3	4	5V	DC Power
SCL1, I²C	GPIO 3	5	6	GND	
GPIO_GCLK	GPIO 4	7	8	GPIO 14	TXD0
	GND	9	10	GPIO 15	RXD0
GPIO_GEN0	GPIO 17	11	12	GPIO 18	GPIO_GEN1
GPIO_GEN2	GPIO 27	13	14	GND	
GPIO_GEN3	GPIO 22	15	16	GPIO 23	GPIO_GEN4
DC Power	3.3V	17	18	GPIO 24	GPIO_GEN5
SPI_MOSI	GPIO 10	19	20	GND	
SPI_MISO	GPIO 9	21	22	GPIO 25	GPIO_GEN6
SPI_CLK	GPIO 11	23	24	GPIO 8	SPI_CE0_N
	GND	25	26	GPIO 7	SPI_CE1_N
I²C ID EEPROM	DNC	27	28	DNC	I²C ID EEPROM
	GPIO 5	29	30	GND	
	GPIO 6	31	32	GPIO 12	
	GPIO 13	33	34	GND	
	GPIO 19	35	36	GPIO 16	
	GPIO 26	37	38	GPIO 20	
	GND	39	40	GPIO 21	

Note the difference between GPIO.BOARD and GPIO.BRCM pin numbering

Remaining Items / Future Work?

- Add board configuration that maps PINs
  - GPIO.BOARD (pins on board) vs GPIO.BCM (pins by Broadcom SoC Channel)
  - Broadcom numbering changes between versions of Raspberry Pi
- Further error validation
- AWS GreenGrass

- Others?