Creating IoT Solutions with Serverless Architecture & Alexa

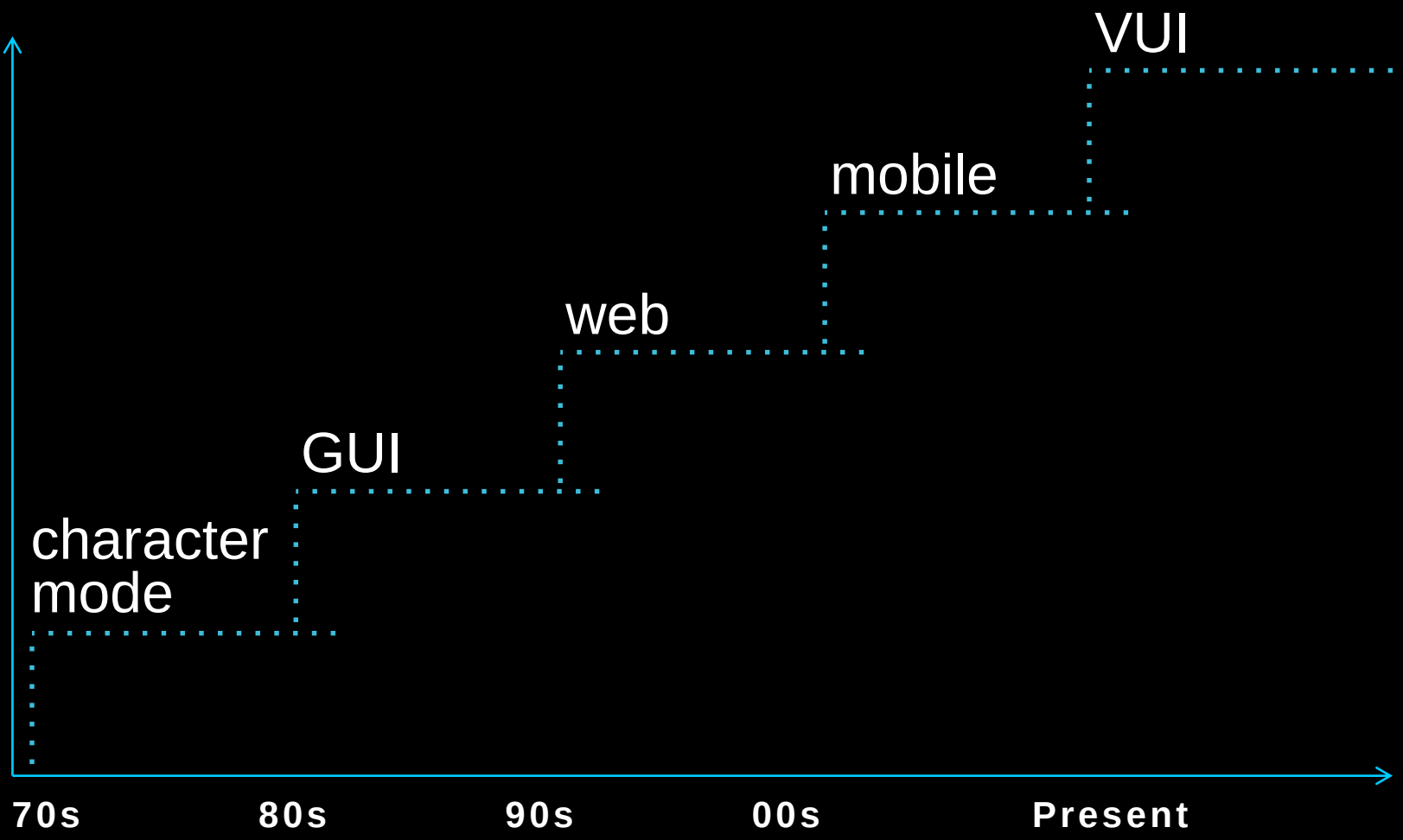
John Chang (張書源)
Technology Evangelist
johnchan@amazon.com

January 2017

amazon web services

Today's agenda

- Skill building fundamentals
- Voice User-Interface
- Alexa skill building

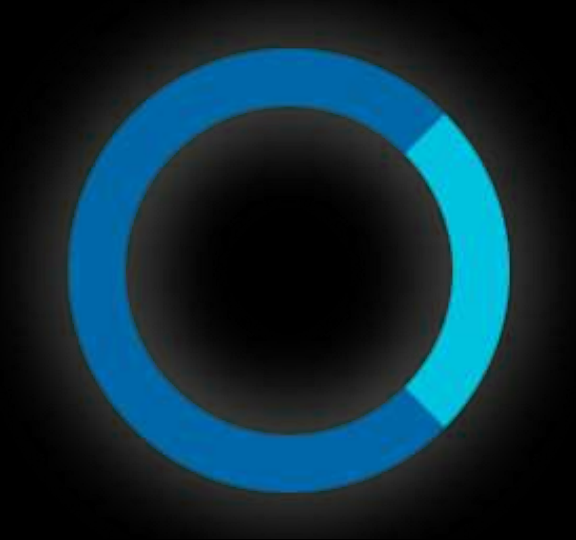

VUI
mobile
web
GUI
character mode
70s
80s
90s
00s
Present


11:42

Skill Building Fundamentals

Before we code

1. developer.amazon.com
2. aws.amazon.com



The Alexa Service

Alexa Skills Kit

amazon alexa

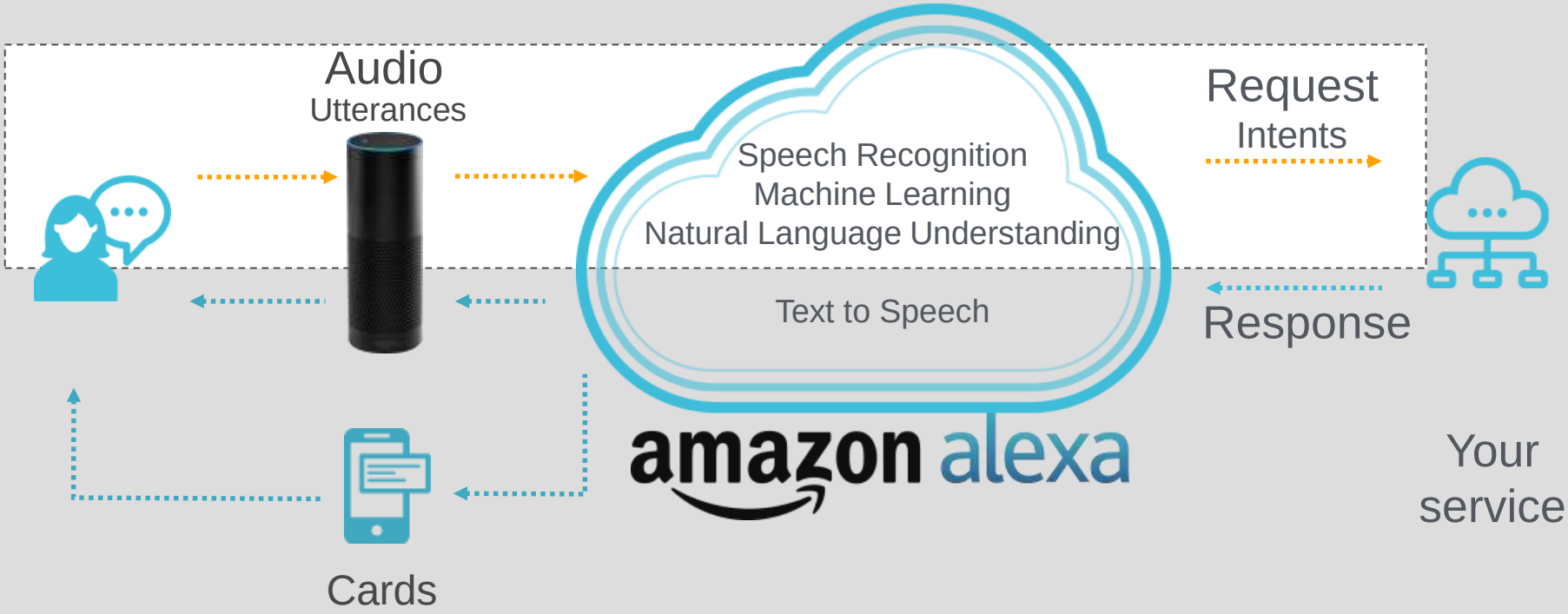
Alexa Voice Services

amazon echo

amazon fireTV

Alexa Skills Kit: Processing a request

Alexa Skills Kit: Utterances and Intents



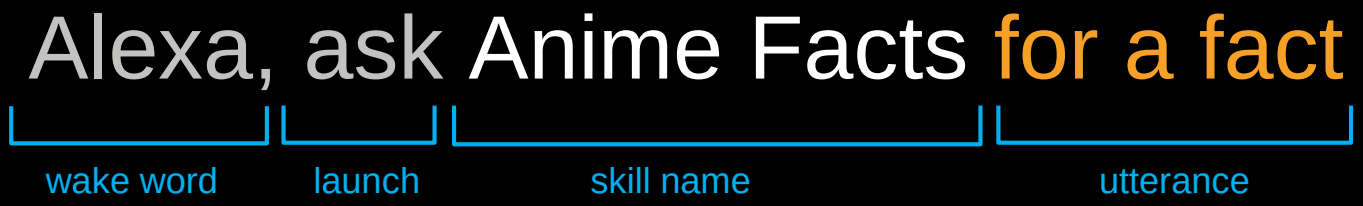
ASR – Automatic Speech Recognition

fȯr tē tīmz

- Forty Times?
- For Tea Times?

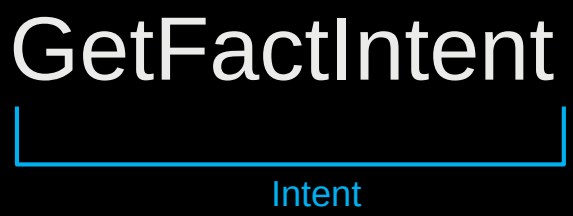
- For Tee Times?
- Four Tee Times?

Utterances and Intents



Utterances and Intents

Utterances and Intents



```
{
  "intents": [
    {
      "intent": "GetHoroscope",
      "slots": [
        {
          "name": "Sign",
          "type": "LITERAL"
        },
        {
          "name": "Date",
          "type": "DATE"
        }
      ]
    },
    {
      "intent": "GetLuckyNumbers",
      "slots": []
    }
  ]
}
```

Building an Alexa Skill

Hosted service

- You **define** interactions for your Voice App through **intent schemas**
- Each intent consists of two fields:
  - The **intent field** gives the name of the intent
  - The **slots field** lists the slots associated with that intent
- Slots can also included types such as **LITERAL**, **NUMBER**, **DATE**, etc.

- “what is…”
- “what’s…”
- “tell me…”
- “give…”
- “give me…”
- “get…”
- “get me…”
- “find…”
- “find me…”

Building an Alexa Skill

Hosted service

- The mappings between intents and the typical utterances that invoke those intents are provided in a tab-separated text document of sample utterances.
- Each possible phrase is assigned to one of the defined intents.
- GetHoroscope what is the horoscope for {pisces|Sign}
- GetHoroscope what will the horoscope for {leo|Sign} be {next tuesday|Date}

Alexa Skills Kit: Requests and Responses

HTTP Header

```
POST / HTTP/1.1
Content-Type : application/json;charset=UTF-8
Host : your.application.endpoint
Content-Length :
Accept : application/json
Accept-Charset : utf-8
Signature:
SignatureCertChainUrl: https://s3.amazonaws.com/echo.api/echo-api-cert.pem
```

Request Body Syntax

The request body sent to an Alexa app is in JSON format.

```
{
  "version": "string",
  "session": {
    "new": boolean,
    "sessionId": "string",
    "application": {
      "applicationId": "string"
    },
    "attributes": {
      "string": object
    },
    "user": {
      "userId": "string"
    }
  },
  "request": object
}
```

```
{
  "type": "IntentRequest",
  "requestId": "string",
  "timestamp": "string",
  "intent": {
    "name": "string",
    "slots": {
      "string": {
        "name": "string",
        "value": "string"
      }
    }
  }
}
```

Handling Amazon Alexa service Requests

- You need to handle **POST** requests to your service over port 443 and parse the JSON
- You need to check the **session variable** to see if the user started a new session or if request is from existing one
- Requests always Include a **type**, **requestId** and **timestamp**
- requestId maps directly to **LaunchRequest**, **IntentRequest** and **SessionEndedRequest**

Alexa Skills Kit: Request and Response

"Help"

Lambda Request

```
{
  "session": {
    "sessionId": "SessionId.6fd62a42-8032-4570-b54
    "application": {
      "applicationId": "amzn1.ask.skill.4d1da9f0-f
    },
    "attributes": {
      "STATE": "_STARTMODE"
    },
    "user": {
      "userId": "amzn1.ask.account.AFP3ZWPOS2BGJR7
    },
    "new": false
  },
  "request": {
    "type": "IntentRequest",
    "requestId": "EdwRequestId.ba9cb888-abc3-44bd-
    "locale": "en-US",
    "timestamp": "2016-11-05T17:51:17Z",
    "intent": {
      "name": "AMAZON.HelpIntent",
      "slots": {}
    }
  },
  "version": "1.0"
}
```

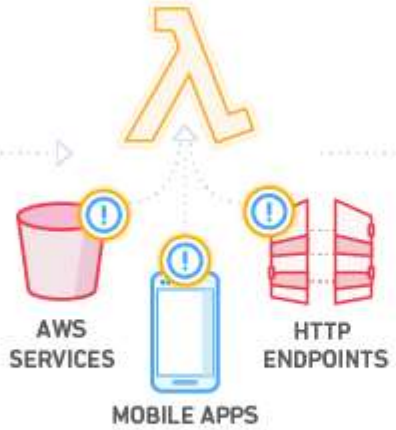
Lambda Response

```
{
  "version": "1.0",
  "response": {
    "outputSpeech": {
      "type": "SSML",
      "ssml": "<speak> I will ask you some quest
    },
    "reprompt": {
      "outputSpeech": {
        "type": "SSML",
        "ssml": "<speak> I will ask you some que
      }
    },
    "shouldEndSession": false
  },
  "sessionAttributes": {
    "STATE": "_STARTMODE"
  }
}
```

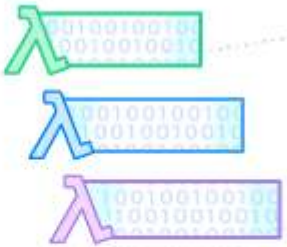
Listen

Using Lambda for Processing the Request & generating the response

Upload your code to AWS Lambda



Set up your code to trigger from other AWS services, HTTP endpoints, or in-app activity



Lambda runs your code only when triggered, using only the compute resources needed

Pay just for the compute time you use

Demo

Slots in the utterances

The answer is one

slot value

one is the answer

slot value

one

slot value

AnswerIntent {value: “one”}

Intent

slot value

Utterances and Intent Schema

Utterances

```
AnswerIntent the answer is {Answer}
AnswerIntent my answer is {Answer}
AnswerIntent is it {Answer}
AnswerIntent {Answer} is my answer
AnswerIntent {Answer}
```

Intent Schema

```
{
  "intents": [
    {
      "intent": "AMAZON.HelpIntent"
    },
    {
      "intent": "AnswerIntent",
      "slots": [
        {
          "name": "Answer",
          "type": "AMAZON.NUMBER"
        }
      ]
```

Utterances and Intent Schema

Utterances

```
AnswerIntent the answer is {Answer}
AnswerIntent my answer is {Answer}
AnswerIntent is it {Answer}
AnswerIntent {Answer} is my answer
AnswerIntent {Answer}
```

Intent Schema

```
{
  "intents": [
    {
      "intent": "AMAZON.HelpIntent"
    },
    {
      "intent": "AnswerIntent",
      "slots": [
        {
          "name": "Answer",
          "type": "AMAZON.NUMBER"
        }
      ]
```

Built-in and Custom Slot Types

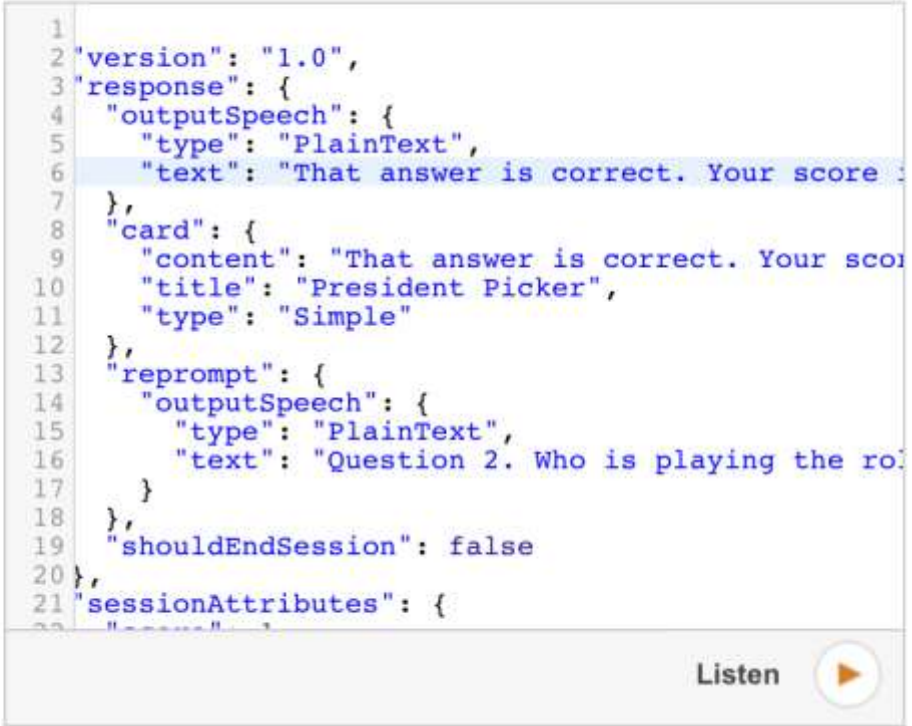
```
"type": "AMAZON.NUMBER"
"type": "ListOfAnimals"
```

Session Attributes

Lambda Request

```
{
  "session": {
    "sessionId": "SessionId.e0174f7b-2249-4932-93f
    "application": {
      "applicationId": "amzn1.echo-sdk-ams.app.f0a
    },
    "attributes": {
      "score": 0,
      "currentQuestionIndex": 0,
      "speechOutput": "Question 1. Is this year gc
      "correctAnswerText": "no. there will be an a
      "repromptText": "Question 1. Is this year gc
      "questions": [
        2,
        21,
        14,
        11,
        10
      ],
      "correctAnswerIndex": 1
    },
    "user": {
      "userId": "amzn1.ask.account.AFP3ZWPOS2BGJR7
    },
    "new": false
```

Lambda Response



```
"version": "1.0",
"response": {
  "outputSpeech": {
    "type": "PlainText",
    "text": "That answer is correct. Your score
  },
  "card": {
    "content": "That answer is correct. Your scor
    "title": "President Picker",
    "type": "Simple"
  },
  "reprompt": {
    "outputSpeech": {
      "type": "PlainText",
      "text": "Question 2. Who is playing the ro
    }
  },
  "shouldEndSession": false
},
"sessionAttributes": {
```

Listen

Session Persistence

Add dynamoDB table name in your index.js

```
exports.handler = function (event, context, callback) {
    var alexa = Alexa.handler(event, context);
    alexa.appId = appId;
    alexa.dynamoDBTableName = 'YourTableName'; // That's it!
    alexa.registerHandlers(State1Handlers, State2Handlers);
    alexa.execute();
};
```

put

```
this.attributes['yourAttribute'] = 'value';
```

get

```
var yourVariable = this.attributes['yourAttribute']
```

Thank you!