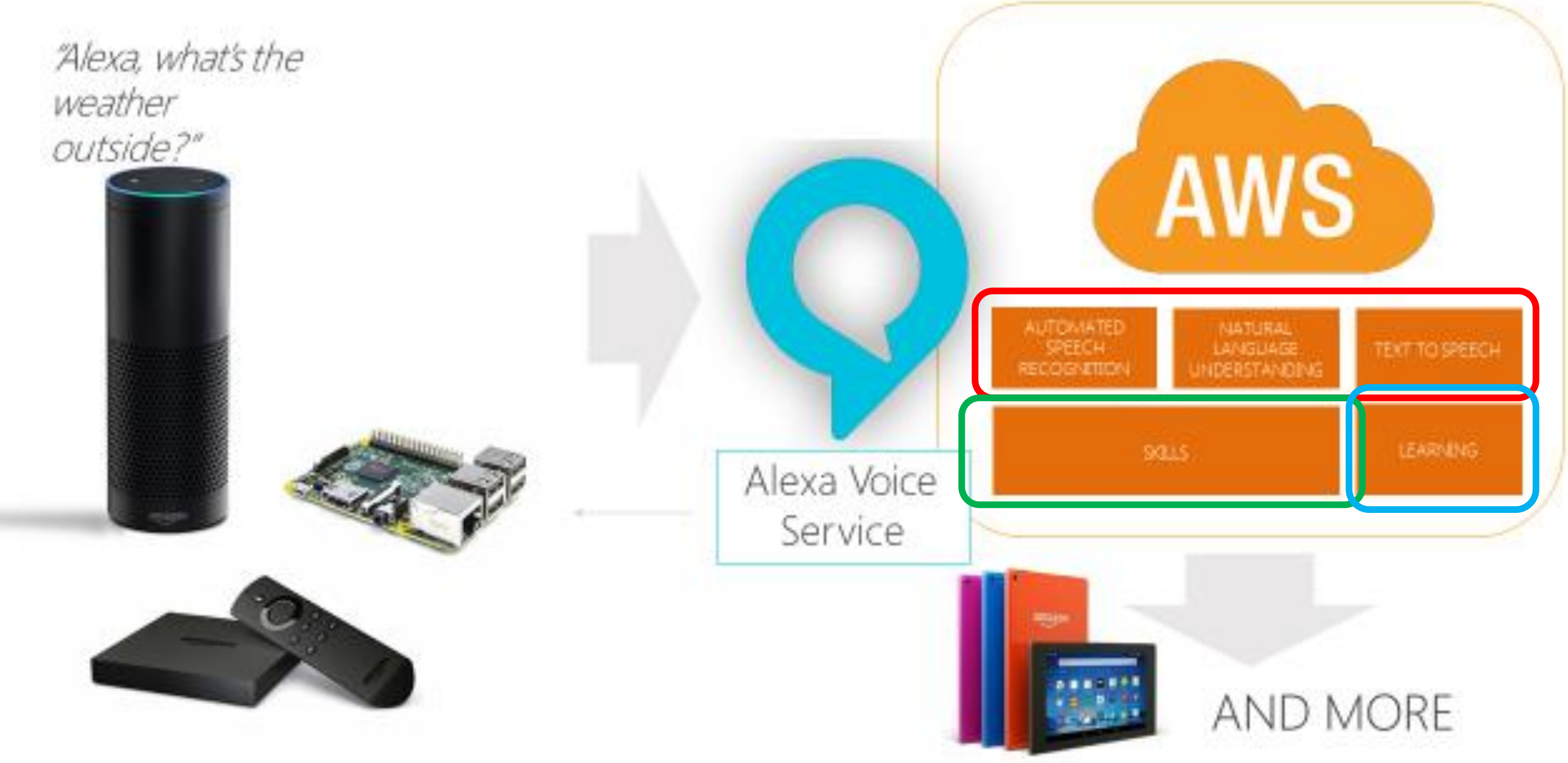
# An Introduction to Using AWS and ASK to Build Voice Driven Experiences

# What is Alexa?

Alexa, Hello.

# THE ALEXA PLATFORM



*"Alexa, what's the weather outside?"*

Alexa Voice Service

AWS

AUTOMATED SPEECH RECOGNITION

NATURAL LANGUAGE UNDERSTANDING

TEXT TO SPEECH

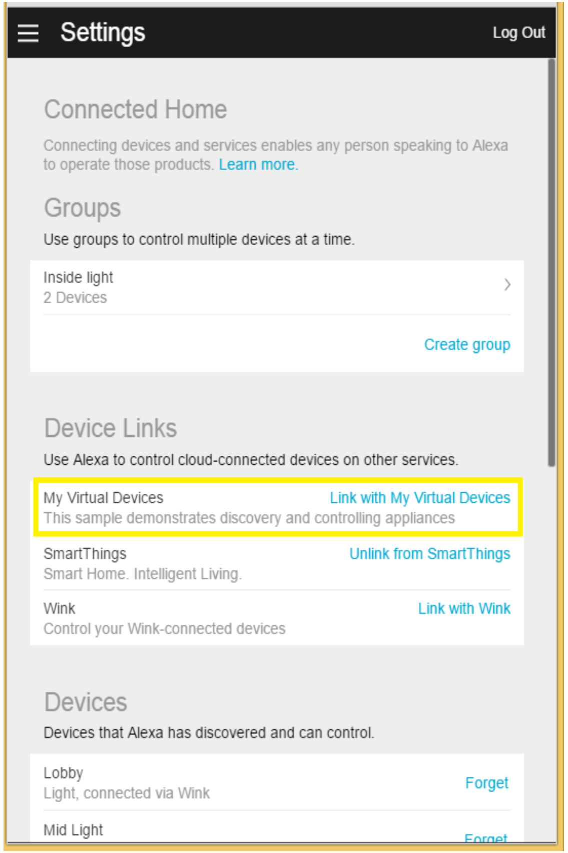
SKILLS

LEARNING

AND MORE

# Connected Home (CoHo) and Lighting API

# Alexa App



☰ Settings

Log Out

Connected Home

Connecting devices and services enables any person speaking to Alexa to operate those products. Learn more.

Groups

Use groups to control multiple devices at a time.

Inside light
2 Devices ›

Create group

Device Links

Use Alexa to control cloud-connected devices on other services.

| | |
|---|---|
| My Virtual Devices<br>This sample demonstrates discovery and controlling appliances | Link with My Virtual Devices |
| SmartThings<br>Smart Home. Intelligent Living. | Unlink from SmartThings |
| Wink<br>Control your Wink-connected devices | Link with Wink |

Devices

Devices that Alexa has discovered and can control.

| | |
|---|---|
| Lobby<br>Light, connected via Wink | Forget |
| Mid Light | Forget |

http://alexa.amazon.com

# Alexa – What Makes A Good Skill

# Key Design Principles for ALEXA SKILLS

- ✓ Users Can Speak to Your Skill Naturally and Spontaneously
- ✓ Alexa Should Understand Most Requests to Your Skill
- ✓ A Skill Should Respond in an Appropriate Way
- ✓ Skills Should Provide High Value
- ✓ Skills Should Evolve Over Time

# Users Can Speak to Your Skill Naturally and Spontaneously

# Users Can Speak to Your Skill Naturally and Spontaneously

The experience of using your Alexa skill should allow users to not have to think about what to say and allow them to not remember how to say it.

They should be able to converse with Alexa just as they would another human.

All they need is a rough idea of what Alexa can do (e.g. playing music, setting a timer, etc.), and they just ask her to do it.

# Users Can Speak to Your Skill Naturally and Spontaneously

This is the real value of voice interface, but this value can quickly erode in a skill that forces users to interact in unnatural ways.

You should try to remove artificial skill syntax and make interactions within your skill as natural as possible.

Allowing your users to make simple requests without having to think about the format those requests should be in, will create a much better experience.

# Example of a Conversation in ALEXA SKILLS

Odd Phrasing: Very odd and/or lengthy invocations that inhibit using the skill in a conversational and spontaneous way.

Alexa, ask [davefacts] for a fact when the fact is of type davefact.
*Alexa, ask [dave] for a [fact].*

Lengthy Invocations: The combination of skill name with the task is often difficult to remember the exact syntax .

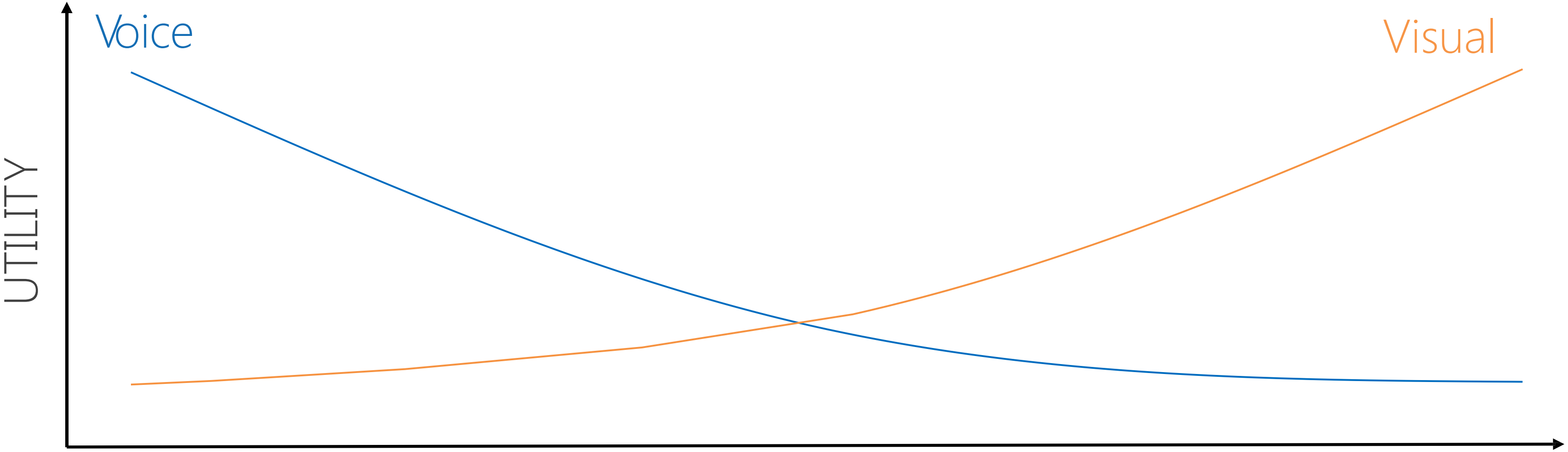
Alexa, ask [transportation service alerts] for the [current status] of [the monorail A].
*Alexa, ask [trafficbuddy] about [monorail A]*

# Having a Good Conversation in an ALEXA SKILL

- Makes It Clear that the User Needs to Respond
- Doesn’t Assume Users Know What to Do
- Clearly Presents the Options
- Keeps It Brief
- Avoids Overwhelming Users with Too Many Choices
- Offers Help for Complex Skills
- Asks Users Only Necessary Questions
- Uses Confirmations Selectively
- Obtains One Piece of Information at a Time
- Makes Sure Users Know They are in the Right Place
- Avoids Technical and Legal Jargon

*Write for the Ear, not the Eye!*

# Skills Should Provide High Value



## Doing

Performs a task

## Pinpointing

Identifies specific info

## Telling

Provides quick reference

## Browsing

Gives info on a broad subject

High Utility

Low Utility

### Doing

Performs a Task

**“Alexa, get me an Uber ride.”**

“There is a driver on his way with an ETA of 5 min.”

### Searching/Pinpointing

Identifies specific info

**“Alexa, ask Vendor if there are Madonna tickets available for this weekend.”**

“There are a limited amount of tickets, ranging from $49 to $279.”

### Telling

Provides a quick reference point

**“Alexa, tell me a cat fact.”**

“It is well known that dogs are superior to cats.”

### Browsing

Gives info on a broad subject

**“Alexa, ask Amazon what’s on sale.”**

“The following items are on sale right now…”

# Skills Should Provide High Value

Voice is conversational. Very different than touch driven experiences. Less is  more.

Aim for skills that perform tasks on behalf of the user.

# A Skill Should Evolve Over Time

# A Skill Should Evolve Over Time

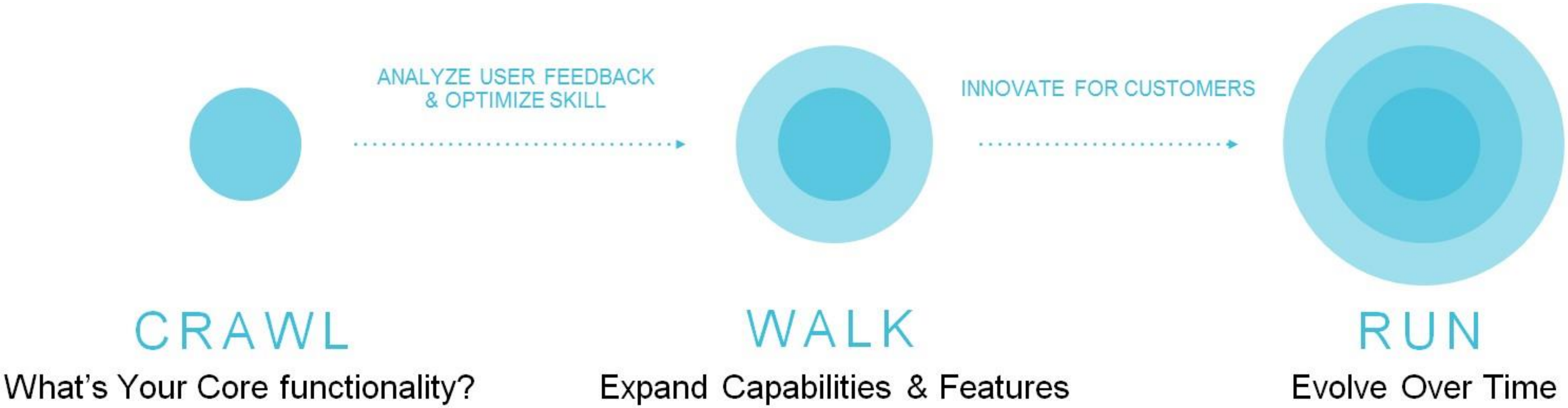
Voice user interfaces work well when they are focused, and give quick responses.

Start with a primary use case that both communicates your business case, but is also a clear winner for a voice user interface.

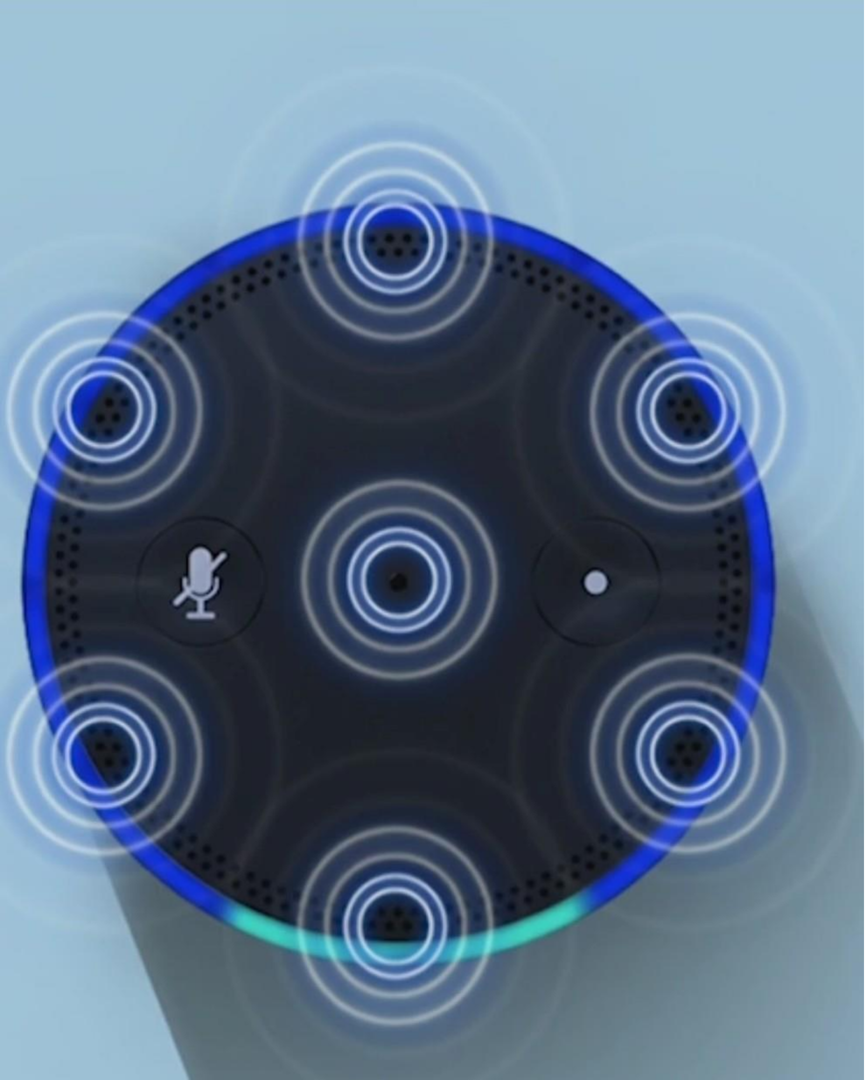
Do one thing well, and add in capabilities allowing it to get smarter over time.

# BUILD YOUR PRIMARY USE CASE FIRST.

Then expand to lesser use cases.


ANALYZE USER FEEDBACK & OPTIMIZE SKILL
INNOVATE FOR CUSTOMERS
CRAWL
What’s Your Core functionality?
WALK
Expand Capabilities & Features
RUN
Evolve Over Time

# Alexa Skills Kit (ASK) Overview



## Creating your own
# ALEXA SKILLS

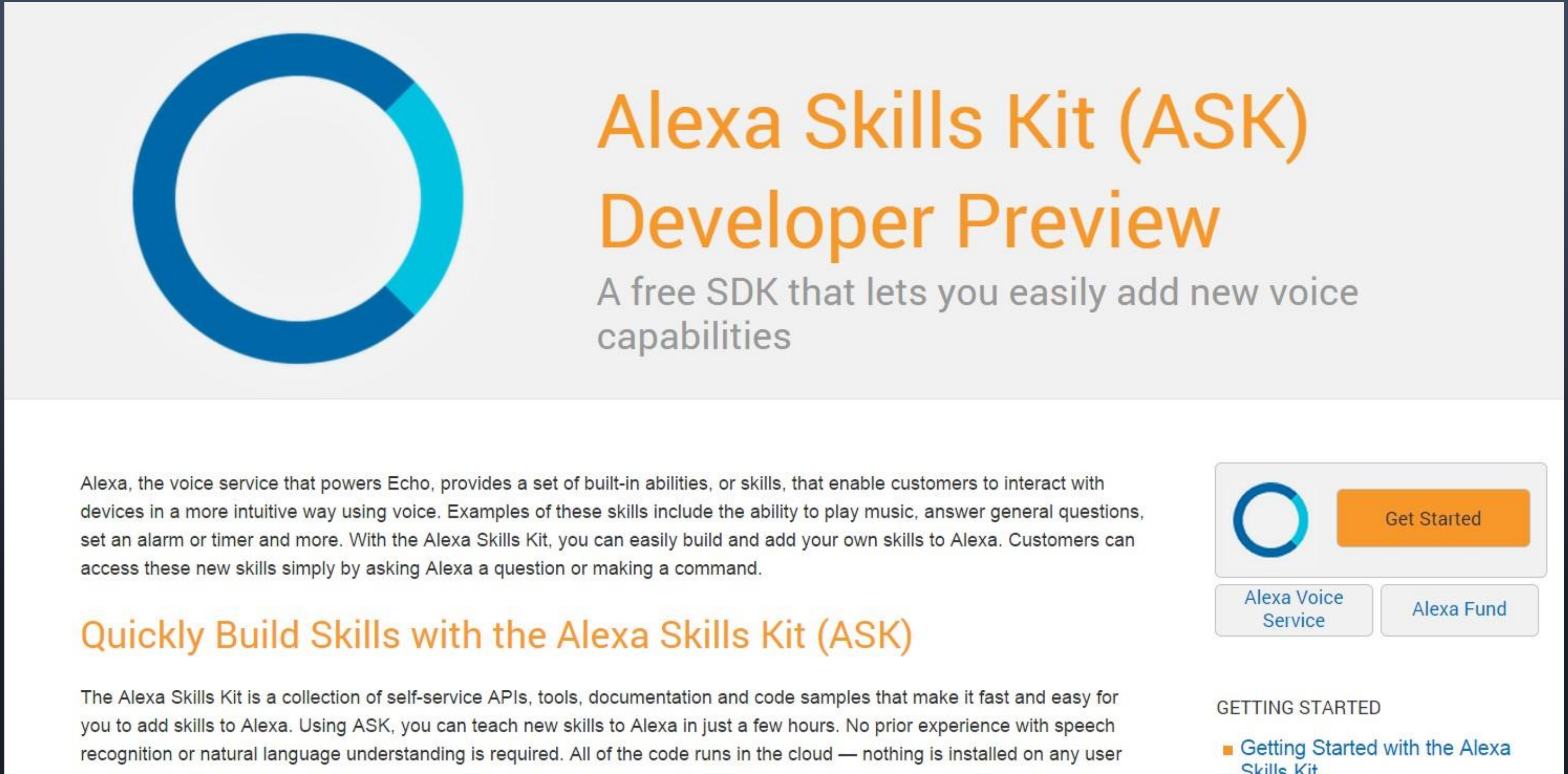
Alexa Skills have two parts:

Configuration data in Amazon Developer Portal

Hosted Service responding to user requests

# ALEXA SKILLS KIT (ASK)

https://developer.amazon.com/ask



## Alexa Skills Kit (ASK) Developer Preview

A free SDK that lets you easily add new voice capabilities

Alexa, the voice service that powers Echo, provides a set of built-in abilities, or skills, that enable customers to interact with devices in a more intuitive way using voice. Examples of these skills include the ability to play music, answer general questions, set an alarm or timer and more. With the Alexa Skills Kit, you can easily build and add your own skills to Alexa. Customers can access these new skills simply by asking Alexa a question or making a command.

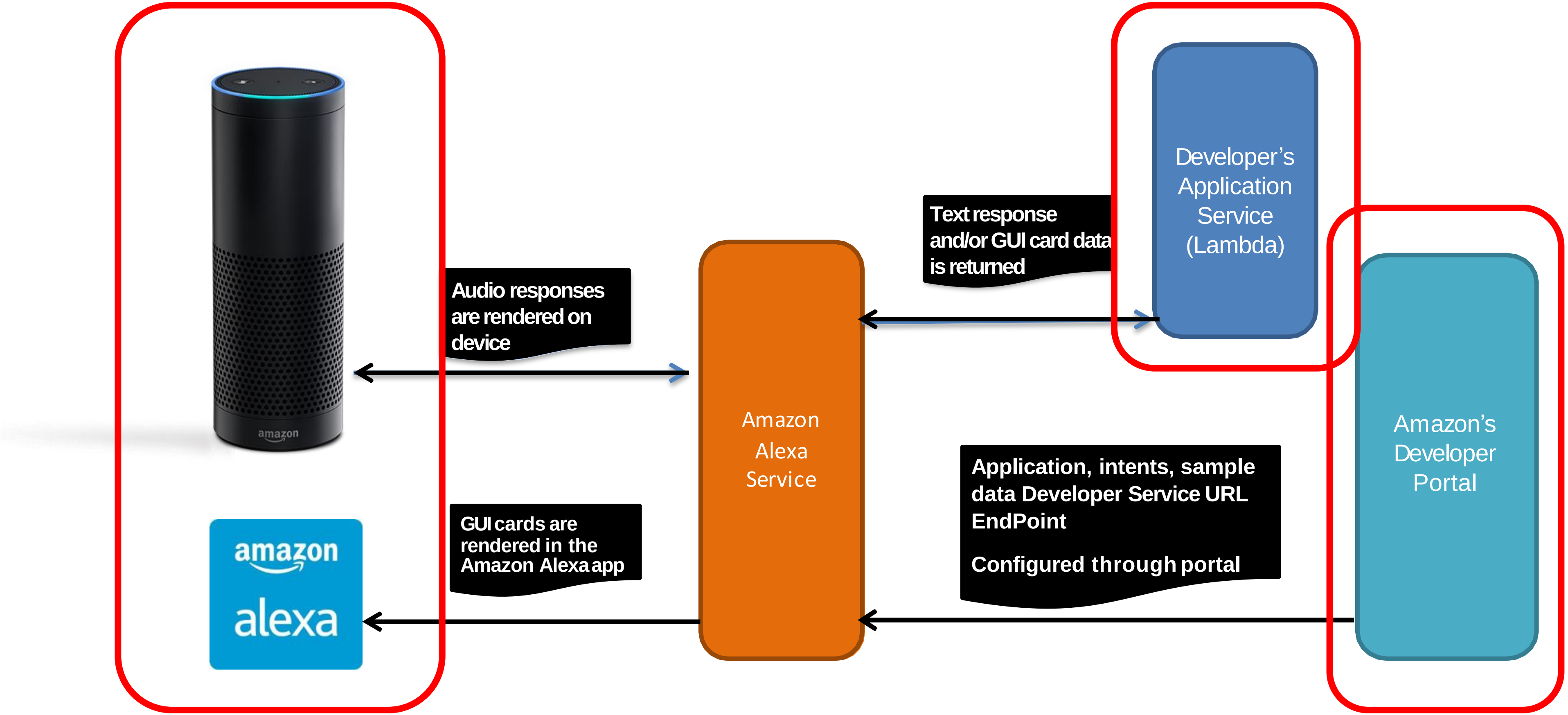
### Quickly Build Skills with the Alexa Skills Kit (ASK)

The Alexa Skills Kit is a collection of self-service APIs, tools, documentation and code samples that make it fast and easy for you to add skills to Alexa. Using ASK, you can teach new skills to Alexa in just a few hours. No prior experience with speech recognition or natural language understanding is required. All of the code runs in the cloud — nothing is installed on any user

GETTING STARTED

- Getting Started with the Alexa Skills Kit

# Alexa Skills Kit Architecture



Audio responses are rendered on device

GUI cards are rendered in the Amazon Alexa app

Amazon Alexa Service

Text response and/or GUI card data is returned

Developer's Application Service (Lambda)

Application, intents, sample data Developer Service URL EndPoint

Configured through portal

Amazon's Developer Portal

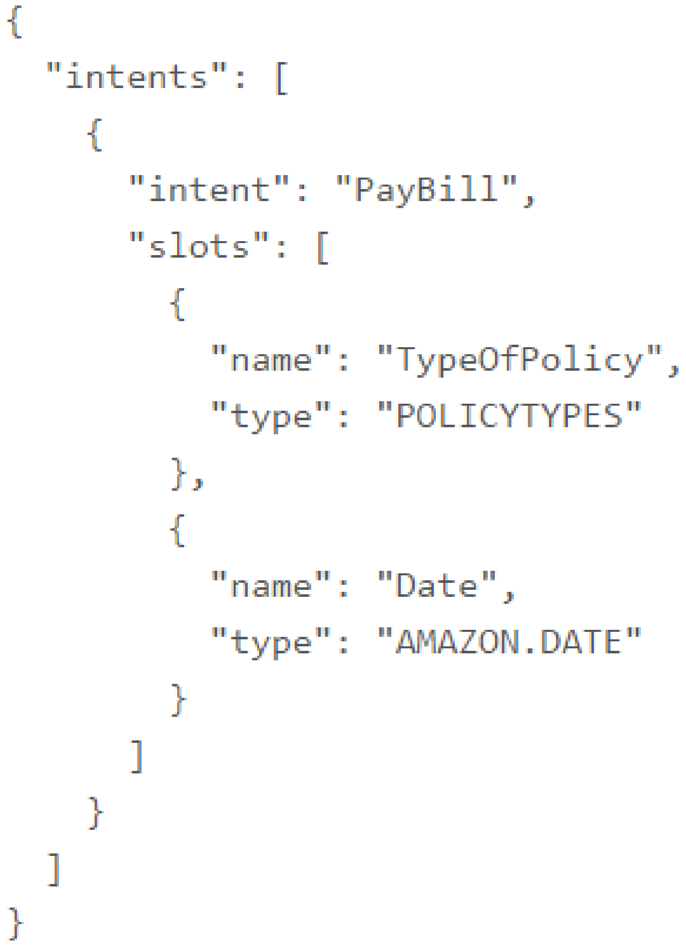
Configuring a new Alexa Skill

# Demo

# Building an Alexa Skill

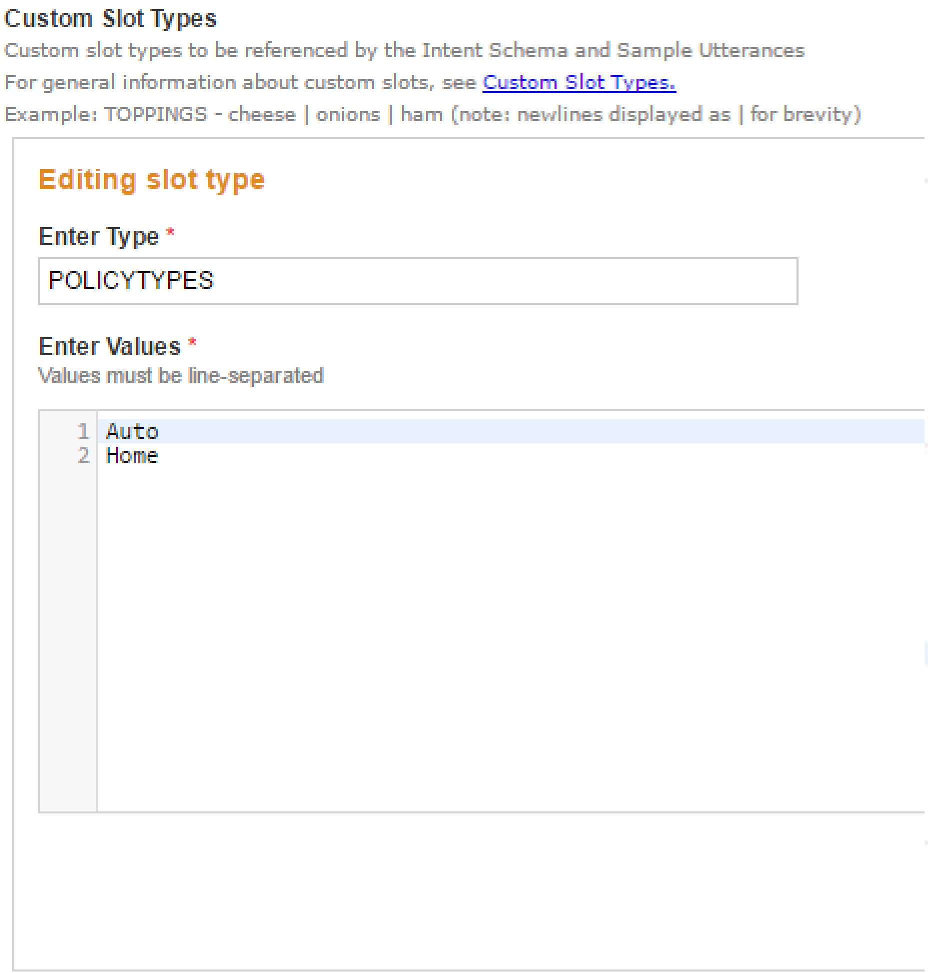
# Amazon Developer Portal

## Step #1 - Intents



```
{
  "intents": [
    {
      "intent": "PayBill",
      "slots": [
        {
          "name": "TypeOfPolicy",
          "type": "POLICYTYPES"
        },
        {
          "name": "Date",
          "type": "AMAZON.DATE"
        }
      ]
    }
  ]
}
```

- 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 be any internal types such as AMAZON.LITERAL, AMAZON.NUMBER, AMAZON.DATE, etc. or they can be ones you create.



Custom Slot Types

Custom slot types to be referenced by the Intent Schema and Sample Utterances

For general information about custom slots, see Custom Slot Types.

Example: TOPPINGS - cheese | onions | ham (note: newlines displayed as | for brevity)

Editing slot type

Enter Type *

POLICYTYPES

Enter Values *

Values must be line-separated

```
Auto
Home
```

# Building an Alexa Skill

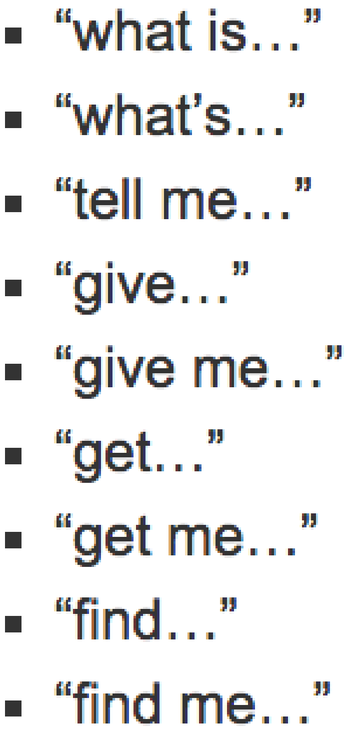
# Amazon Developer Portal

## Step #2 Increasing Accuracy with CUSTOM SLOTS

- Created inside Interaction Model page once in the Developer Portal
- Greaty reduces the number of spoken utterances required
- Can define as many as you need with values line seperated
- Can be combined with existing AMAZON internal types

## Building an Alexa Skill

# Amazon Developer Portal



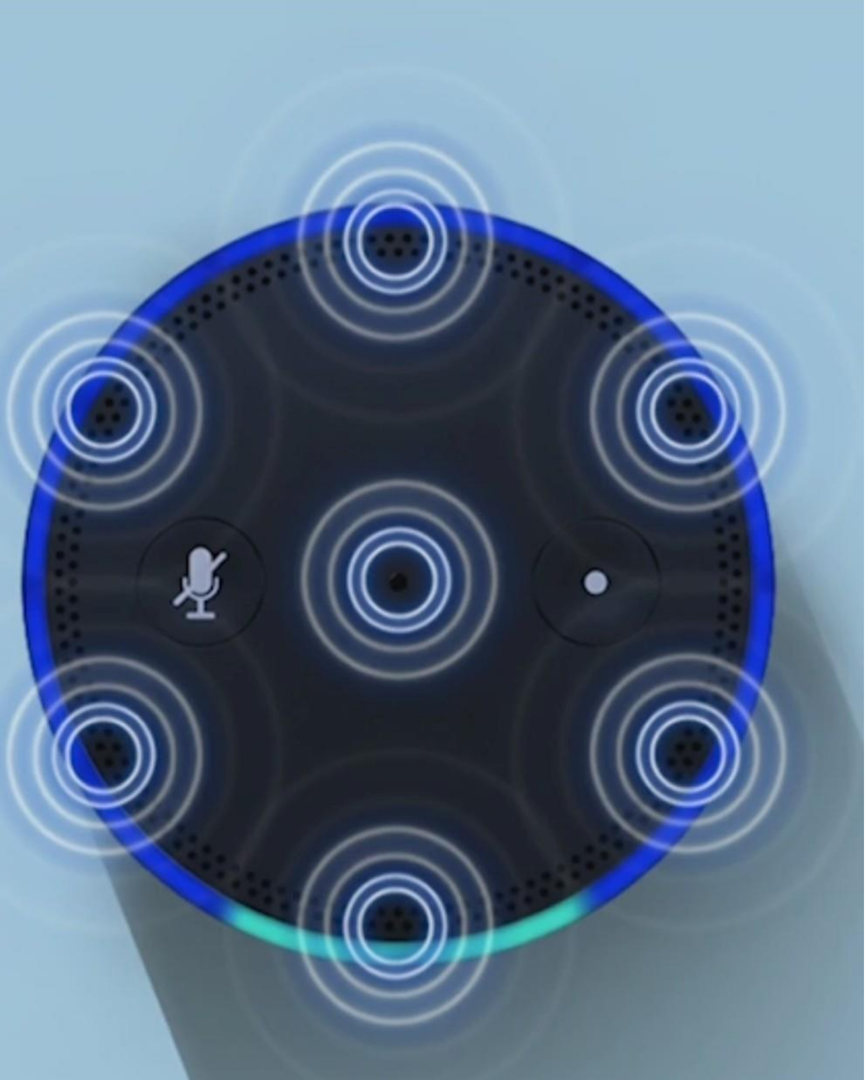
- “what is…”
- “what’s…”
- “tell me…”
- “give…”
- “give me…”
- “get…”
- “get me…”
- “find…”
- “find me…”

## Step #3 - Utterances

- The typical utterances that invoke those intents are entered
- Action words provide the best natural conversation
- Each possible phrase is assigned to one of the defined intents.

BillPay pay my {TypeOfPolicy} bill

BillPay pay my {TypeOfPolicy} bill on {Date}



## The Amazon Alexa Service

# WILL POST EVENTS

LaunchRequest

> Maps to onLaunch() and occurs when the user launches the app without specifying what they want

IntentRequest

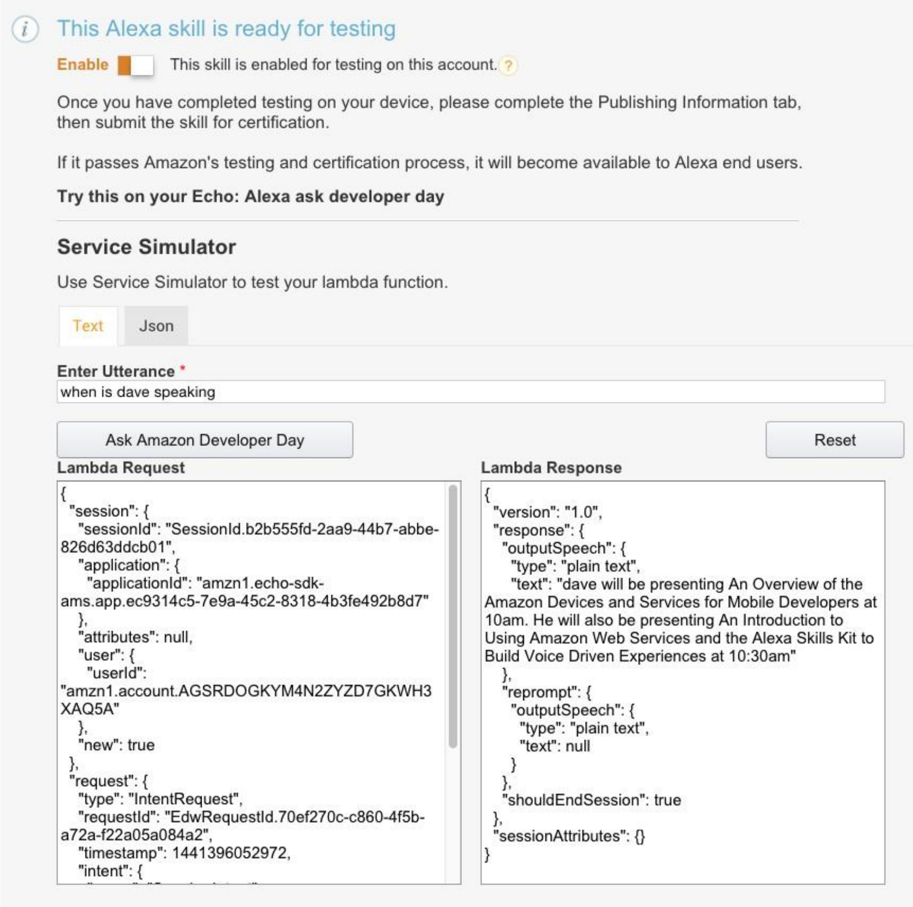
> Maps to onIntent() and occurs when when the user specifies an intent

SessionEndedRequest

> Maps to OnSessionEnded() and when the user ends the session

My Colors Alexa Skill – JavaScript and Node JS

# Demo



This Alexa skill is ready for testing

Enable This skill is enabled for testing on this account.

Once you have completed testing on your device, please complete the Publishing Information tab, then submit the skill for certification.

If it passes Amazon's testing and certification process, it will become available to Alexa end users.

**Try this on your Echo: Alexa ask developer day**

**Service Simulator**

Use Service Simulator to test your lambda function.

**Enter Utterance** *

when is dave speaking

**Lambda Request**

```
{
  "session": {
    "sessionId": "SessionId.b2b555fd-2aa9-44b7-abbe-
826d63ddcb01",
    "application": {
      "applicationId": "amzn1.echo-sdk-
ams.app.ec9314c5-7e9a-45c2-8318-4b3fe492b8d7"
    },
    "attributes": null,
    "user": {
      "userId":
"amzn1.account.AGSRDOGKYM4N2ZYZD7GKWH3
XAQ5A"
    },
    "new": true
  },
  "request": {
    "type": "IntentRequest",
    "requestId": "EdwRequestId.70ef270c-c860-4f5b-
a72a-f22a05a084a2",
    "timestamp": 1441396052972,
    "intent": {
```

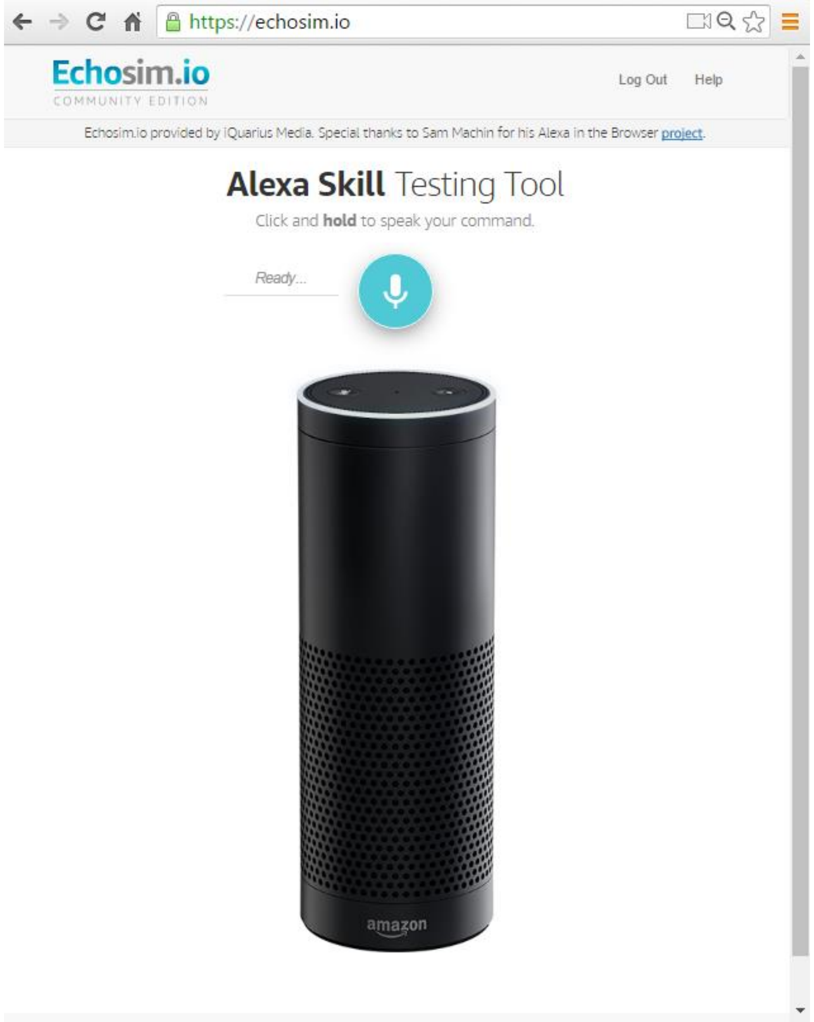
**Lambda Response**

```
{
  "version": "1.0",
  "response": {
    "outputSpeech": {
      "type": "plain text",
      "text": "dave will be presenting An Overview of the
Amazon Devices and Services for Mobile Developers at
10am. He will also be presenting An Introduction to
Using Amazon Web Services and the Alexa Skills Kit to
Build Voice Driven Experiences at 10:30am"
    },
    "reprompt": {
      "outputSpeech": {
        "type": "plain text",
        "text": null
      }
    },
    "shouldEndSession": true
  },
  "sessionAttributes": {}
}
```

# Testing Your Skill

## SERVICE SIMULATOR

- Enabled once a Skill has been configured in the Developer Portal
- Use spoken utterances to generate ad hoc results
- Use JSON to verify requests
- Combine with AWS Lambda Unit Tests to verify both client and service side Alexa end points



https://echosim.io

Echosim.io

COMMUNITY EDITION

Log Out Help

Echosim.io provided by iQuarius Media. Special thanks to Sam Machin for his Alexa in the Browser project.

**Alexa Skill** Testing Tool

Click and **hold** to speak your command.

Ready...

# Testing Your Skill

# New Emulator Available

- Runs in a browser
- https://echosim.io/

http://developer.amazon.com/ask

http://developer.amazon.com/blog

# Digging Deeper into Voice Design

Alexa Skills Kit Voice Design Best Practices - http://bit.ly/voicedesign

Alexa Skills Kit Voice Design Handbook - http://bit.ly/voicehandbook

Wired for Speech: How Voice Activates and Advances the Human-Computer Relationship, by Nass and Brave

The Elements of VUI Style: A Practical Guide to Voice User Interface Design, by Bouzid and Ma

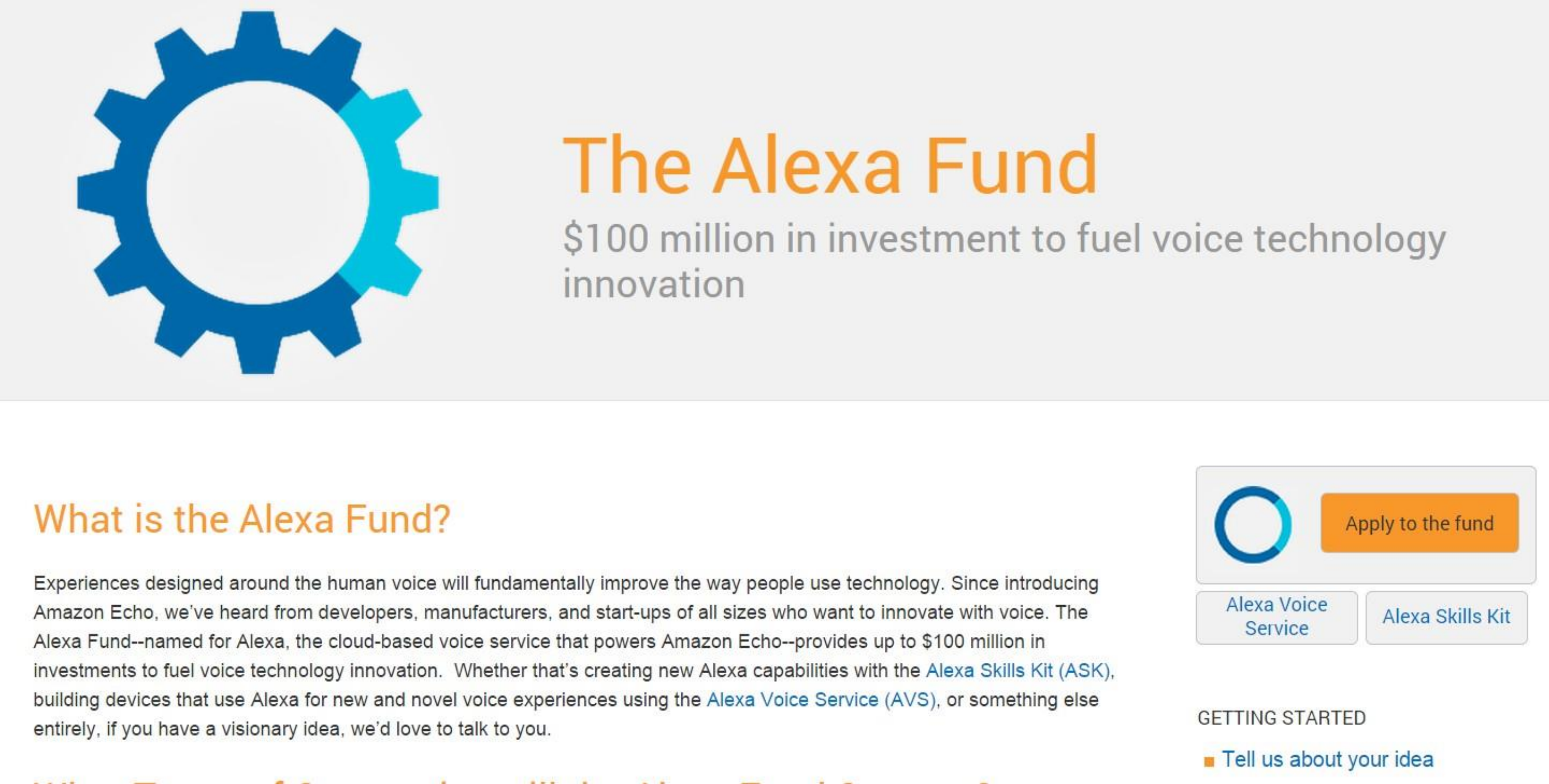
Don’t Make Me Tap!: A Common Sense Approach to Voice Usability, by Bouzid and Ma

The Voice in the Machine: Building Computers That Understand Speech, by Pieraccini

Voice User Interface Design, by Cohen, Giangola, and Balogh

# THE ALEXA FUND

https://developer.amazon.com/alexafund



## The Alexa Fund

$100 million in investment to fuel voice technology innovation

### What is the Alexa Fund?

Experiences designed around the human voice will fundamentally improve the way people use technology. Since introducing Amazon Echo, we've heard from developers, manufacturers, and start-ups of all sizes who want to innovate with voice. The Alexa Fund--named for Alexa, the cloud-based voice service that powers Amazon Echo--provides up to $100 million in investments to fuel voice technology innovation.  Whether that's creating new Alexa capabilities with the Alexa Skills Kit (ASK), building devices that use Alexa for new and novel voice experiences using the Alexa Voice Service (AVS), or something else entirely, if you have a visionary idea, we'd love to talk to you.

GETTING STARTED

- Tell us about your idea

# AGENDA

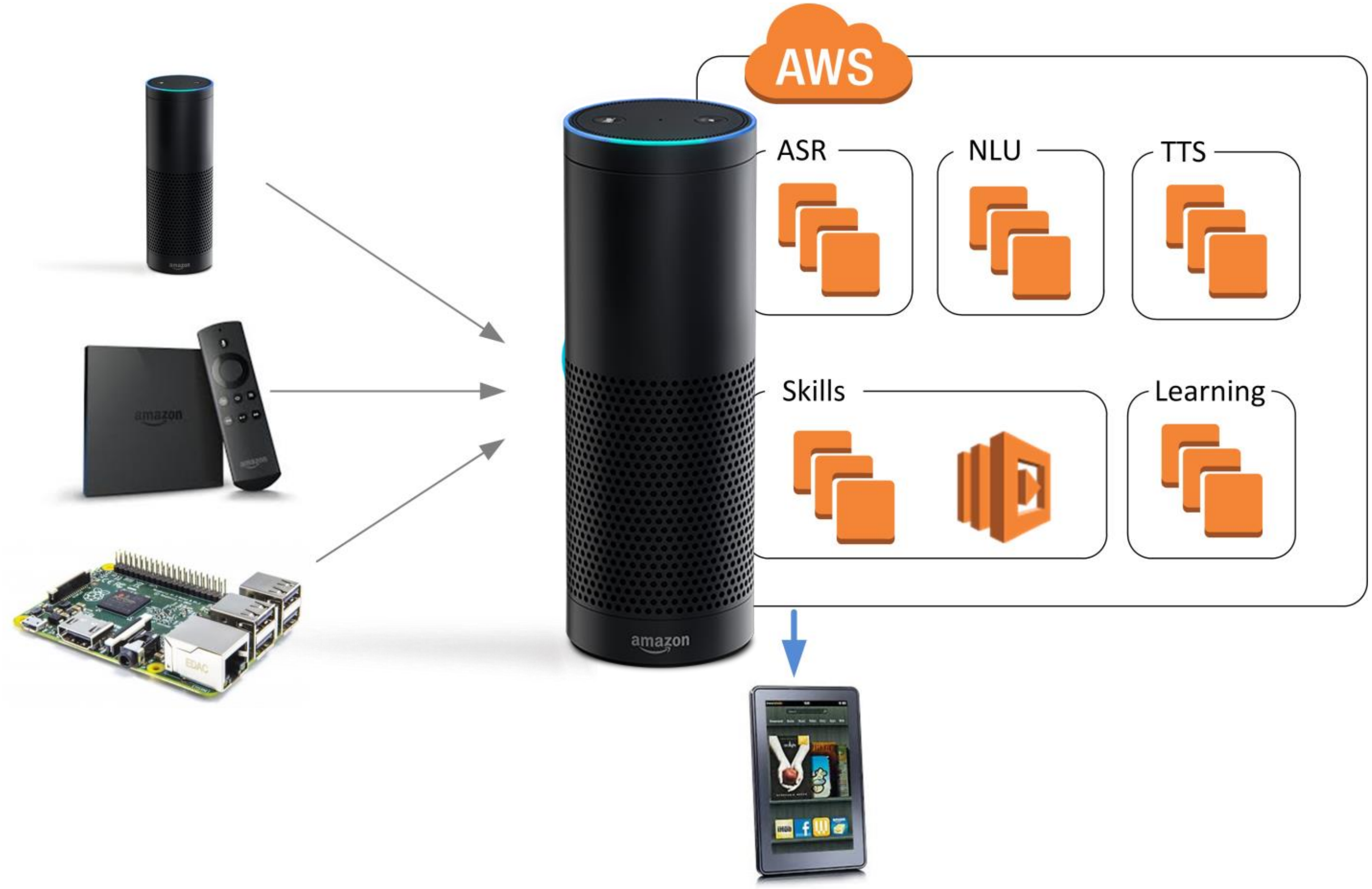
***Dev Portal = [http://developer.amazon.com/ask](http://developer.amazon.com/ask)***
***AWS portal = [http://aws.amazon.com](http://aws.amazon.com)***

# AWS Lambda

Run code without thinking about servers.

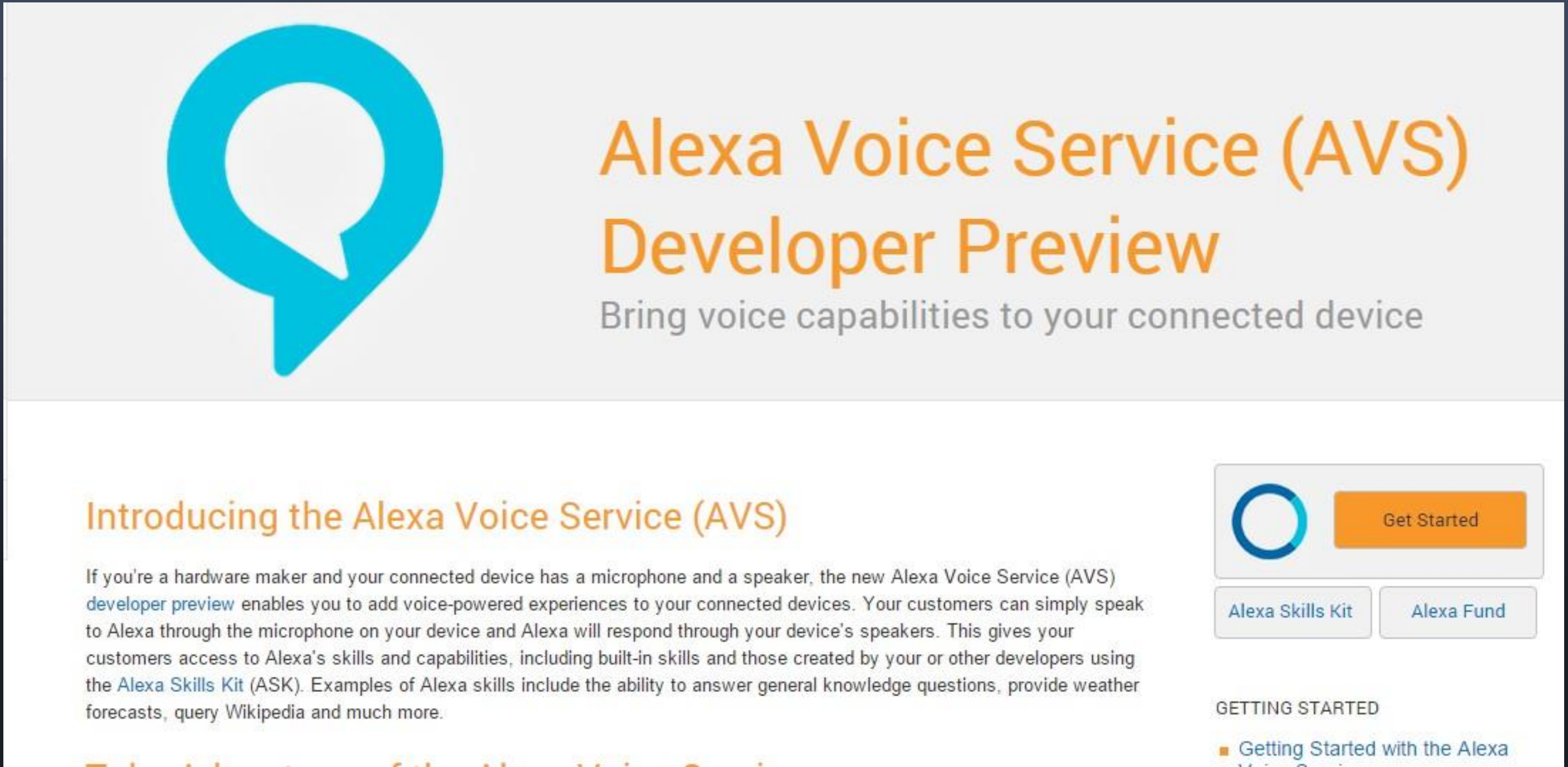
Pay for only the compute time you consume.

# Alexa platform



# ALEXA VOICE SERVICE (AVS)

https://developer.amazon.com/avs



## Alexa Voice Service (AVS) Developer Preview

Bring voice capabilities to your connected device

## Introducing the Alexa Voice Service (AVS)

If you're a hardware maker and your connected device has a microphone and a speaker, the new Alexa Voice Service (AVS) developer preview enables you to add voice-powered experiences to your connected devices. Your customers can simply speak to Alexa through the microphone on your device and Alexa will respond through your device's speakers. This gives your customers access to Alexa's skills and capabilities, including built-in skills and those created by your or other developers using the Alexa Skills Kit (ASK). Examples of Alexa skills include the ability to answer general knowledge questions, provide weather forecasts, query Wikipedia and much more.

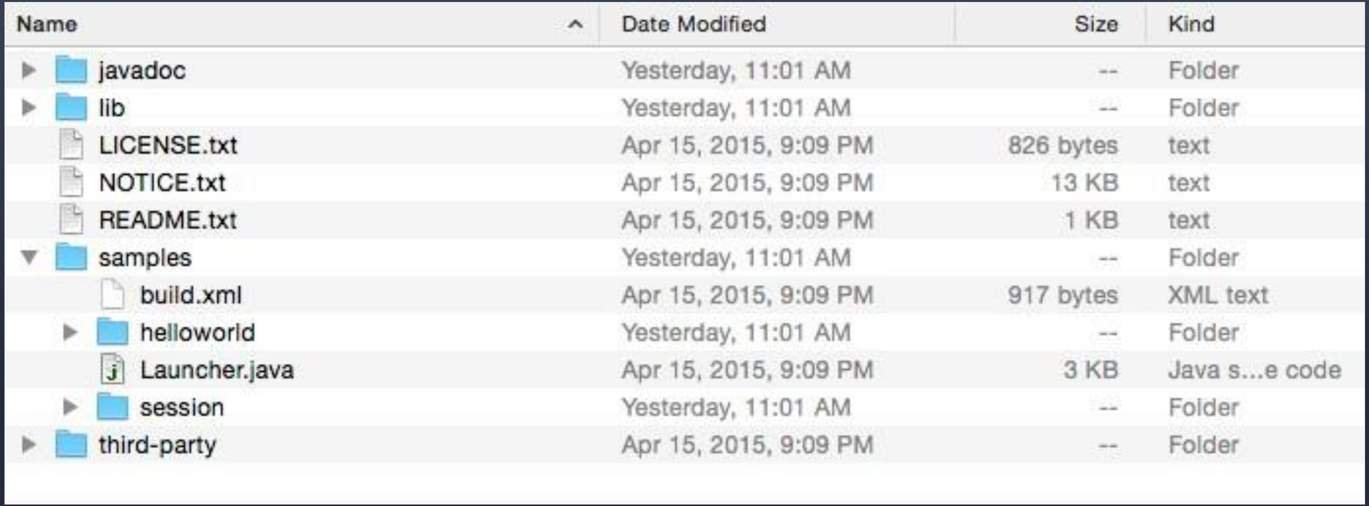
GETTING STARTED

- Getting Started with the Alexa

# ALEXA SKILLS KIT

Alexa is the cloud service used by Amazon Echo

The Alexa Skills Kit allows developers to build new Skills (new voice experiences) for Echo



| Name | Date Modified | Size | Kind |
|---|---|---|---|
| ▶ javadoc | Yesterday, 11:01 AM | -- | Folder |
| ▶ lib | Yesterday, 11:01 AM | -- | Folder |
| LICENSE.txt | Apr 15, 2015, 9:09 PM | 826 bytes | text |
| NOTICE.txt | Apr 15, 2015, 9:09 PM | 13 KB | text |
| README.txt | Apr 15, 2015, 9:09 PM | 1 KB | text |
| ▼ samples | Yesterday, 11:01 AM | -- | Folder |
| build.xml | Apr 15, 2015, 9:09 PM | 917 bytes | XML text |
| ▶ helloworld | Yesterday, 11:01 AM | -- | Folder |
| Launcher.java | Apr 15, 2015, 9:09 PM | 3 KB | Java s...e code |
| ▶ session | Yesterday, 11:01 AM | -- | Folder |
| ▶ third-party | Apr 15, 2015, 9:09 PM | -- | Folder |

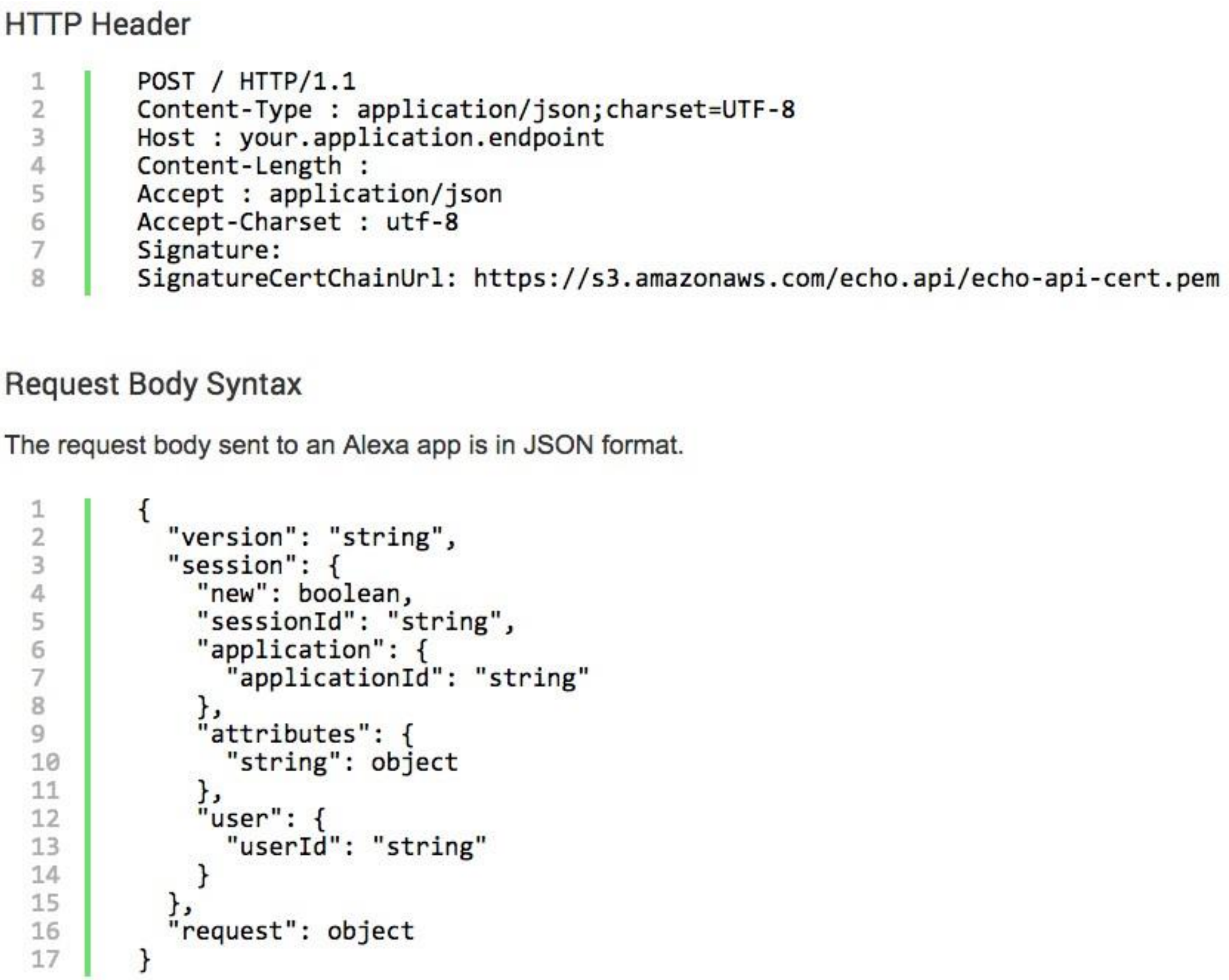
## Building an Alexa Skill

# HOSTED SERVICE

- Adheres to ASK service interface
- Uses HTTP over SSL/TLS on port 443
- Must be Internet-accessible
- Presents a trusted certificate matching domain name
  - Can use self-signed certificate for development
  - Trusted certificate required for certification

Examining the JSON Requests from

# Alexa Service



## 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
```

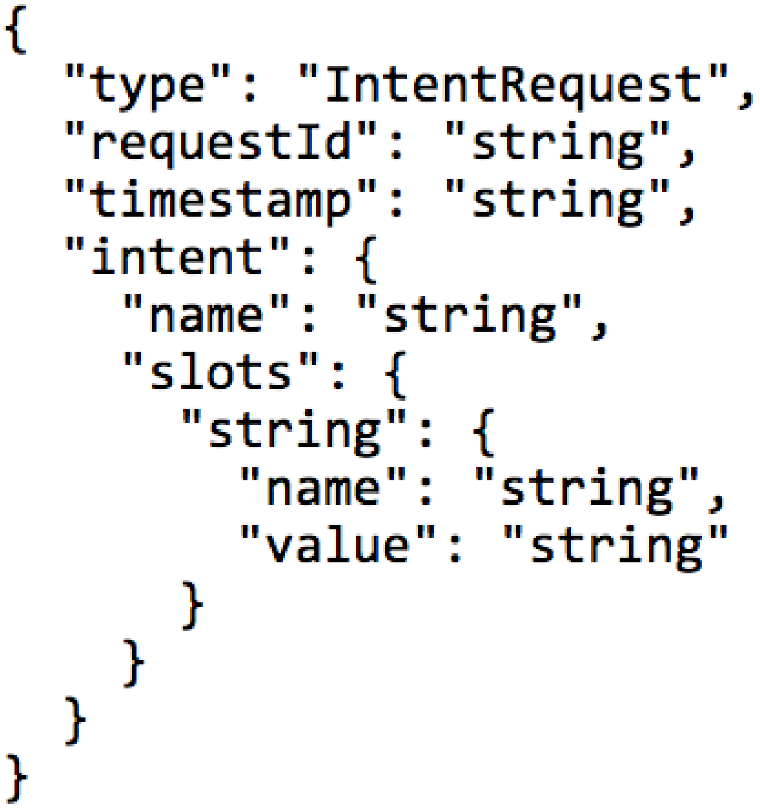
## Request Body Syntax

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
}
```

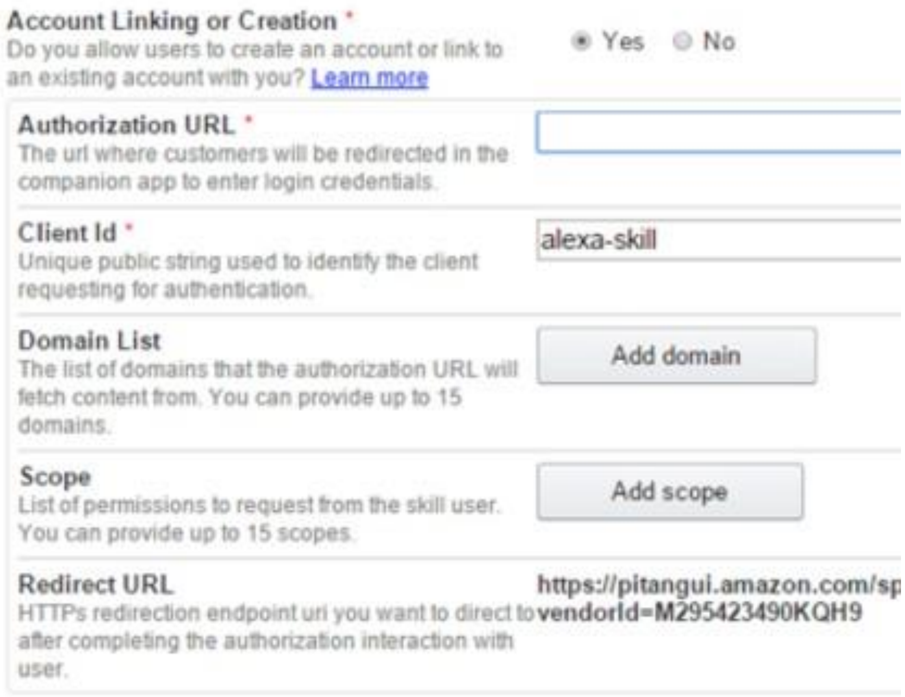
# Handling Amazon Alexa Service

## REQUESTS



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

- You will 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



Account Linking or Creation *
Do you allow users to create an account or link to an existing account with you? Learn more
● Yes ○ No

Authorization URL *
The url where customers will be redirected in the companion app to enter login credentials.

Client Id *
Unique public string used to identify the client requesting for authentication.
alexa-skill

Domain List
The list of domains that the authorization URL will fetch content from. You can provide up to 15 domains.

Scope
List of permissions to request from the skill user. You can provide up to 15 scopes.

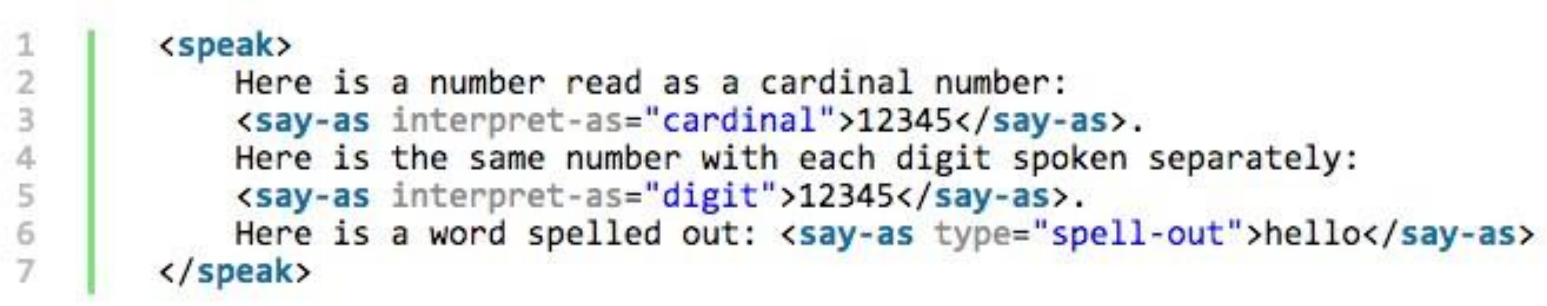
Redirect URL
HTTPs redirection endpoint url you want to direct to after completing the authorization interaction with user.
https://pitangui.amazon.com/sp
vendorId=M295423490KQH9

## Existing Customer with
# ACCOUNT LINKING

- Allow your customers to link their existing accounts with you, to Alexa.
- Customers are prompted to log in to your site using their normal credentials with webview url you provide.
- You authenticate the customer and generate an access token that uniquely identifies the customer and link the accounts.

# Changing Alexa’s inflection with **SSML**

- Alexa automatically handles normal punctuation, such as pausing after a period, or speaking a sentence ending in a question mark as a question.
- Speech Synthesis Markup Language (SSML) is a markup language that provides a standard way to mark up text for the generation of synthetic speech.
- Tags supported include: speak, p, s, break, say-as, phoneme, and w.



```
<speak>
    Here is a number read as a cardinal number:
    <say-as interpret-as="cardinal">12345</say-as>.
    Here is the same number with each digit spoken separately:
    <say-as interpret-as="digit">12345</say-as>.
    Here is a word spelled out: <say-as type="spell-out">hello</say-as>
</speak>
```

# Example of a Conversation in
# ALEXA SKILLS

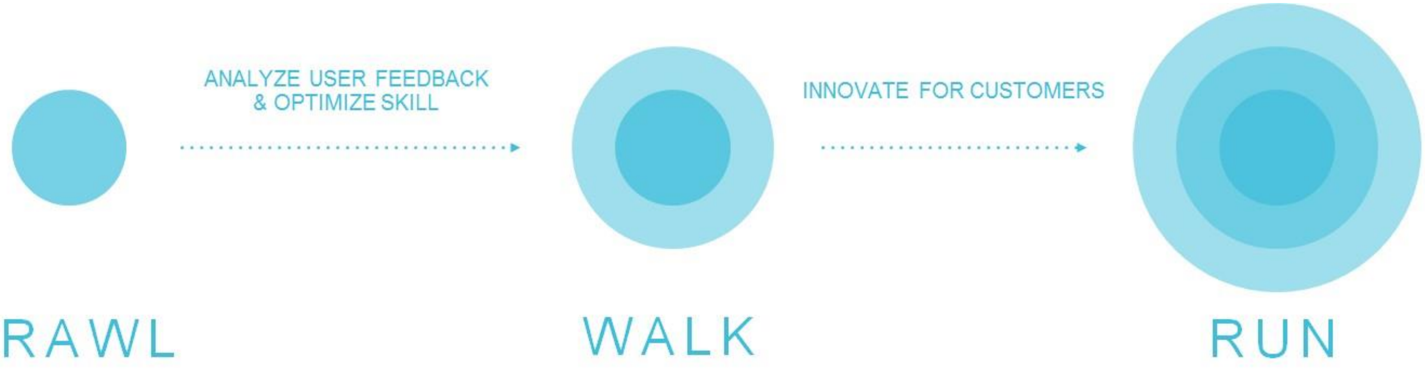
Repetitive Invocations: Some invocations (particularly those that would not necessarily need an intent) are not optimized for the “ask” model and may result in repetitive phrasing.

Alexa, ask [developerinfo] for a [developer info].
Alexa, ask [developerinfo].

# A Skill Should Evolve Over Time

# BUILD YOUR PRIMARY USE CASE FIRST.

Then expand to lesser use cases.



ANALYZE USER FEEDBACK & OPTIMIZE SKILL

INNOVATE FOR CUSTOMERS

CRAWL

What's Your Core functionality?

WALK

Expand Capabilities & Features

RUN

Evolve Over Time

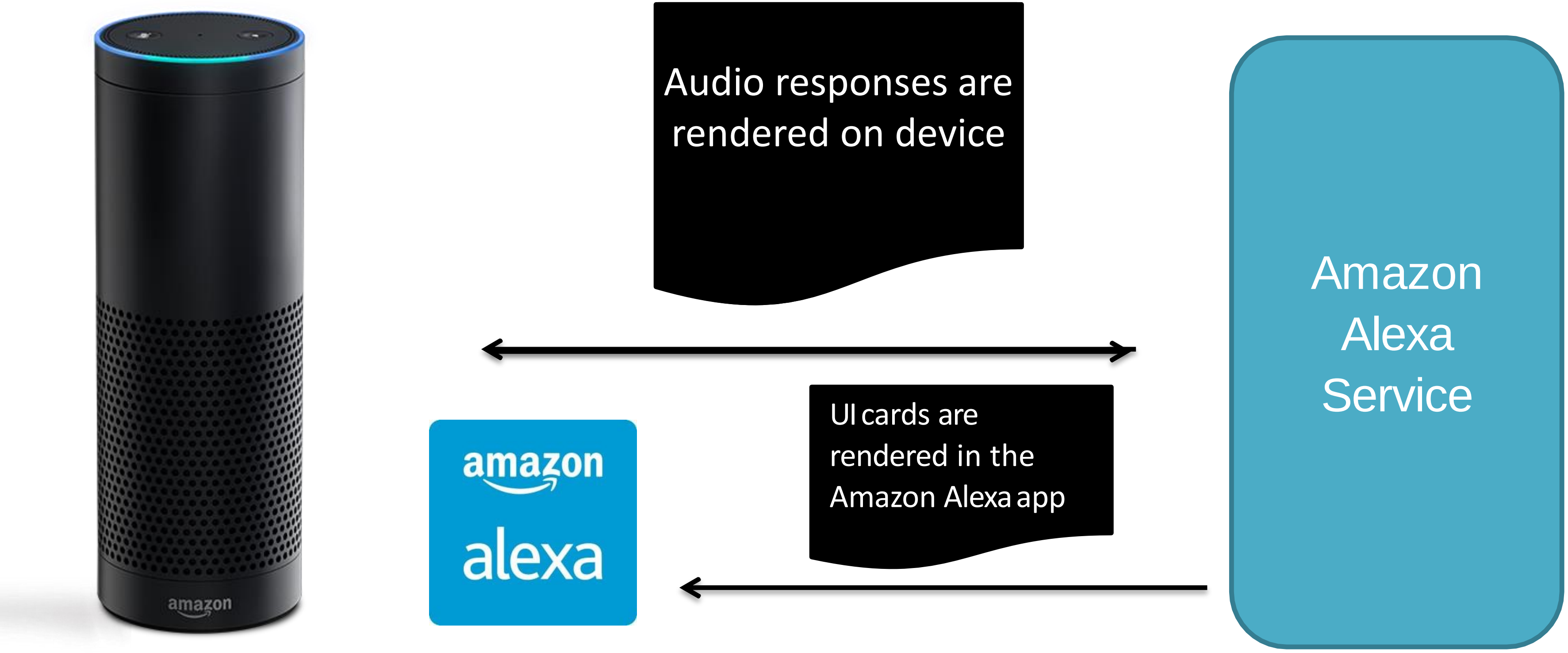
# A Skill Should Evolve Over Time

Voice user interfaces work well when they are focused, and give quick responses.

Start with a primary use case that both communicates your business case, but is also a clear winner for a voice user interface.

Do one thing well, and add in capabilities allowing it to get smarter over time.

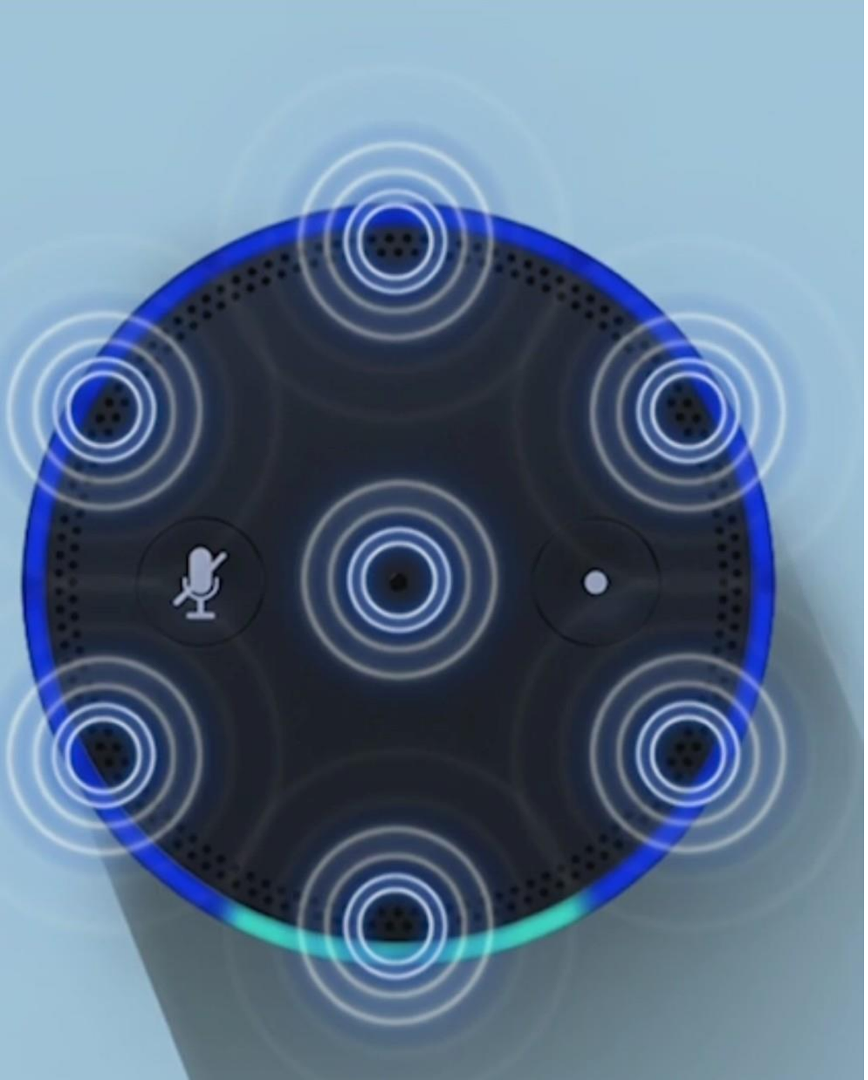
# Alexa Architecture



Audio responses are rendered on device

UI cards are rendered in the Amazon Alexa app

Amazon Alexa Service



## Sample Interactions of
# ALEXA SKILLS

One and done

“Alexa, start Astrology and get the Pisces horoscope.”

[Horoscope Provided]

Conversation

“Alexa, start Astrology”

[“What’s your sign?”]

“Pisces”

[Horoscope Provided]