



# Python for Alexa Enabled Smart Home Programming



Ganesh Kumar T K

PyCon India 2019  
Chennai, IN



# Hello World.py




```
def Noob_user(Name,nick,Status):  
    print( "Name:", Name)  
    print ("About :", about)  
    print ("Status: ", Status)  
    return;
```

```
Noob_user(Name = "Ganesh Kumar ", about = "4  
months old Alexa Developer. 4 year old Pythonista.",  
Status = "Student @ IIIT Kanchipuram")
```

# IOT

Internet of Things





“It's not so much about the emergence of new technology, it's the convergence – the ability to use sensors for everything in the world to basically be a computer, whether it's your contact lens, your hospital bed or a railway track”

**Harriet Green**, Chief of IoT at IBM





# Woah!!

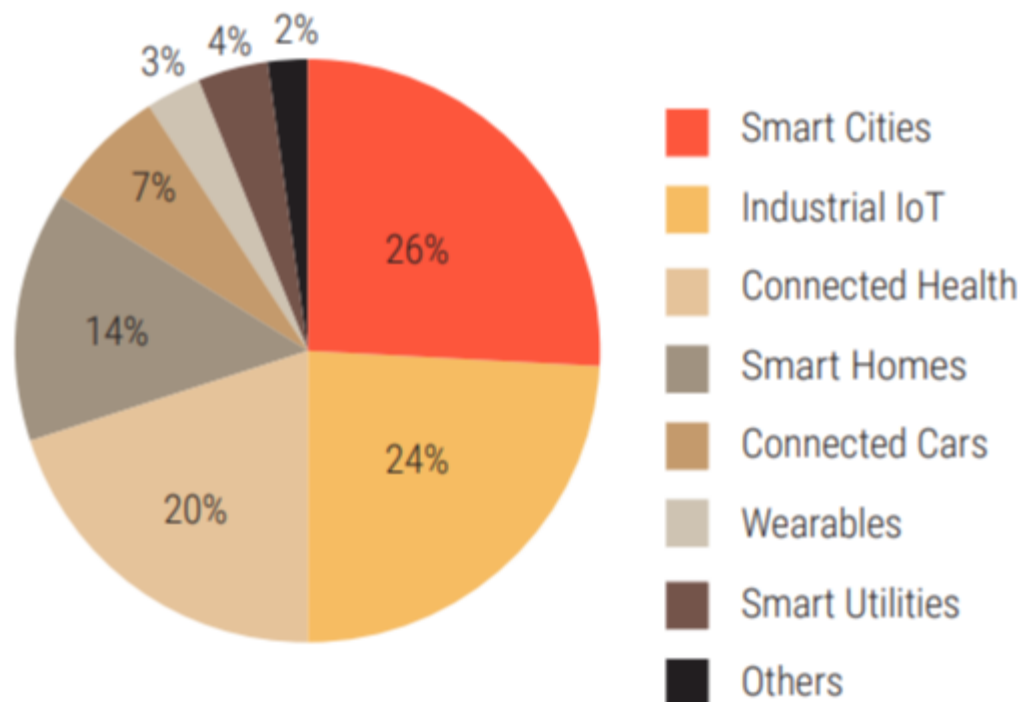
Total Adult  
Population  
**252**  
**MILLION**



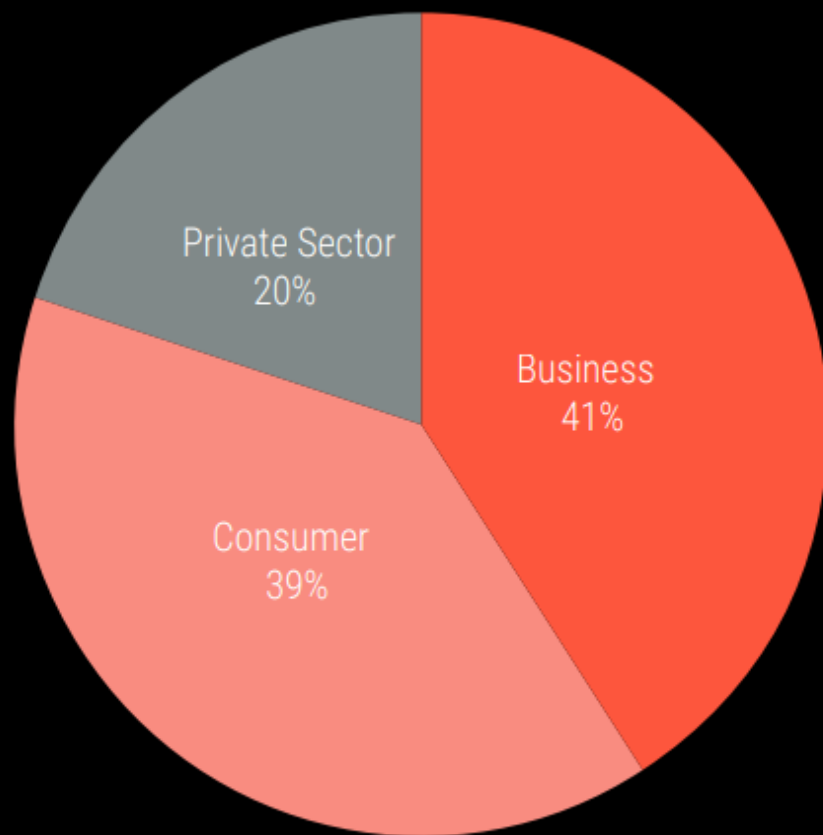
Smart Device  
Owners  
**47.3**  
**MILLION**

*Source: Voicebot voice shopping US Consumer Adoption and Attitudes 2018 Report*



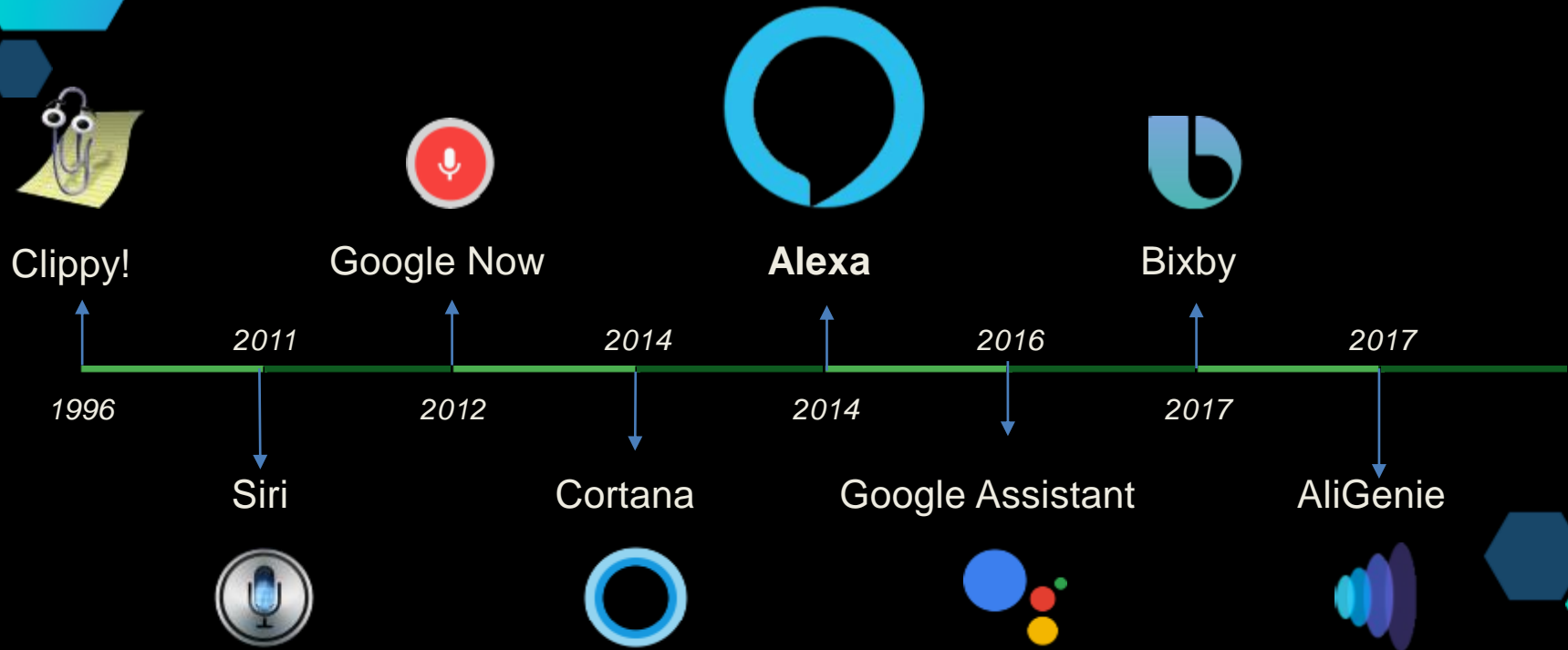


[Source: GrowthEnabler Analysis]

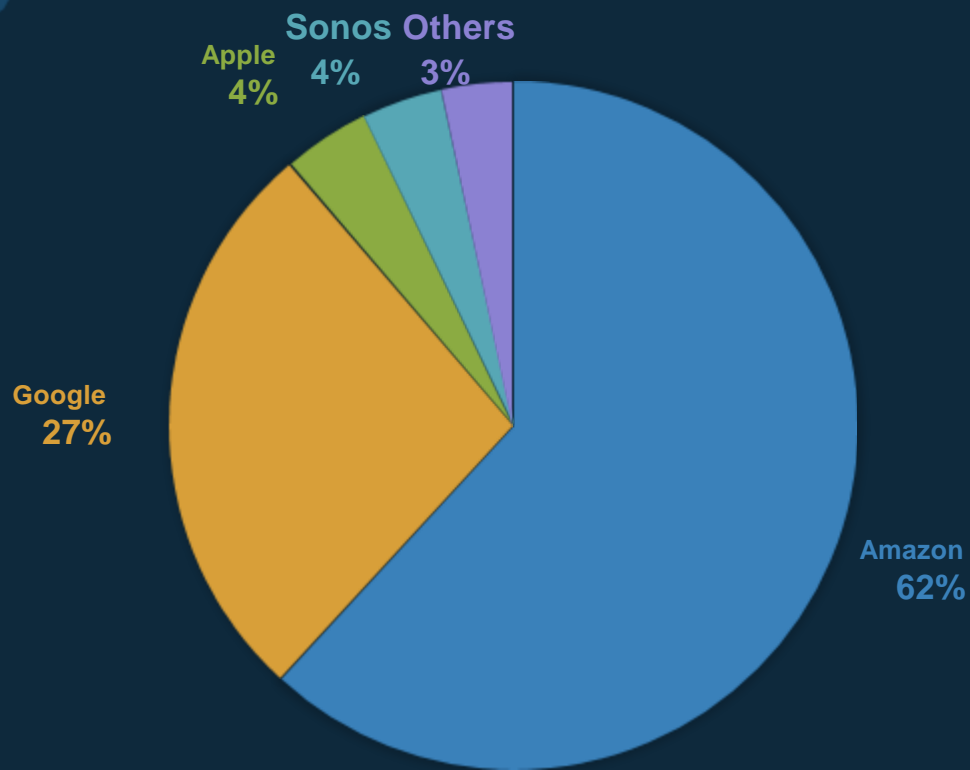


[Source: IoT.uk]

# Smart Assistants







Source: *Voicebot voice shopping US Consumer Adoption and Attitudes 2018 Report*



Source: Google





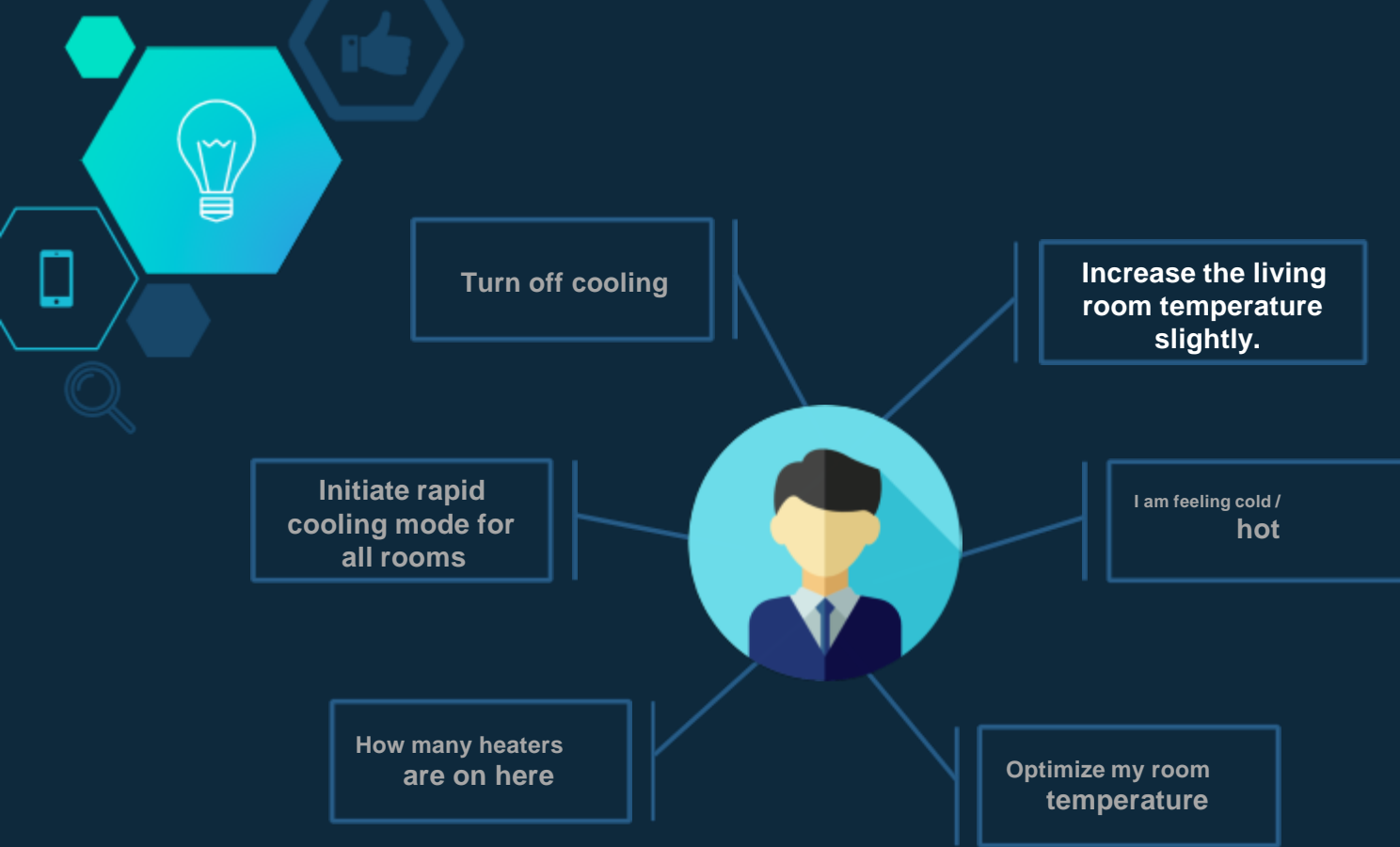
# Commands





# ~~Commands~~ Conversations





# Interactions



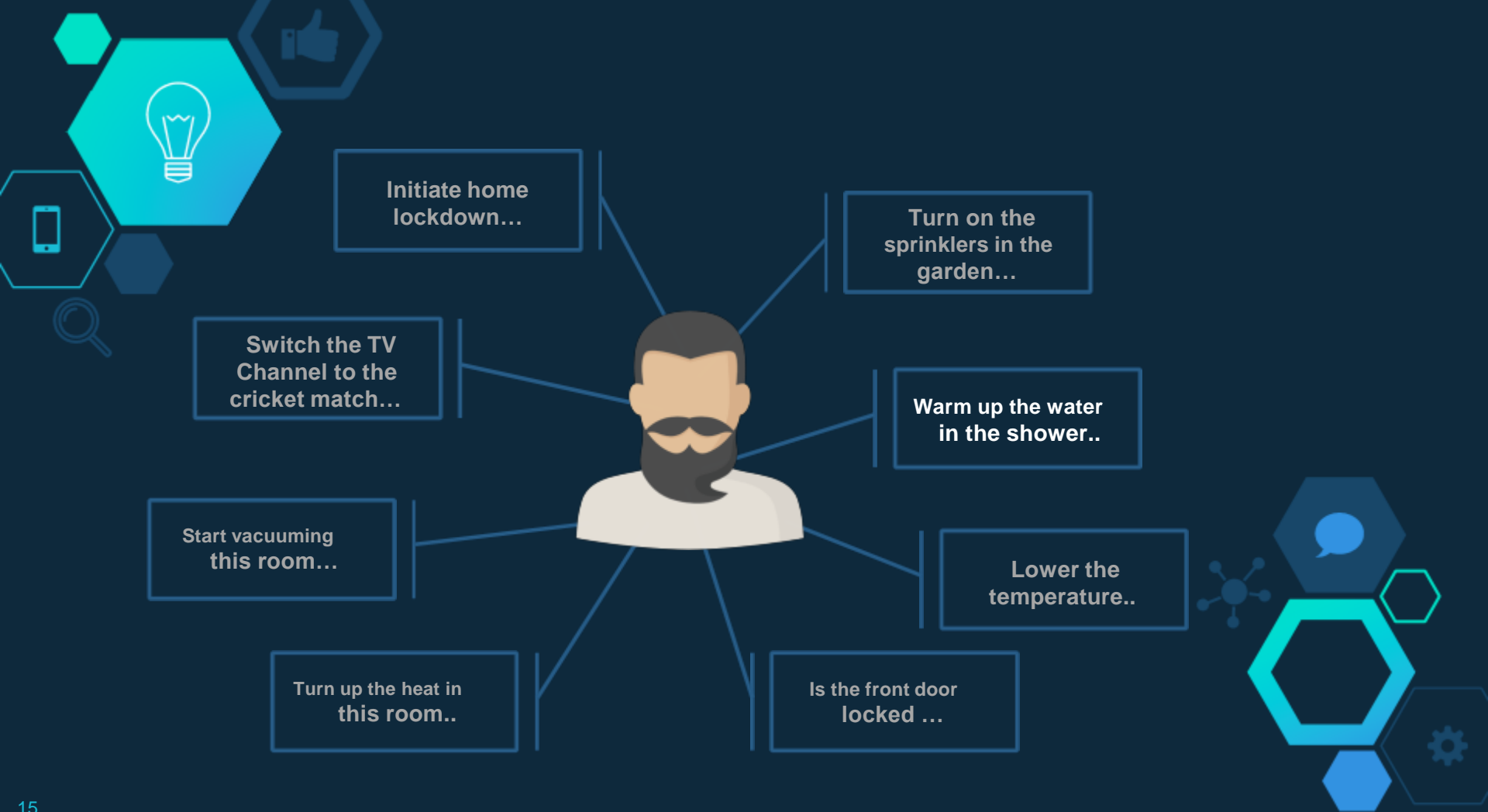
Light



Name  
On/Off  
Color  
Brightness  
State  
Location



Alexa, initiate  
night mode in  
living room

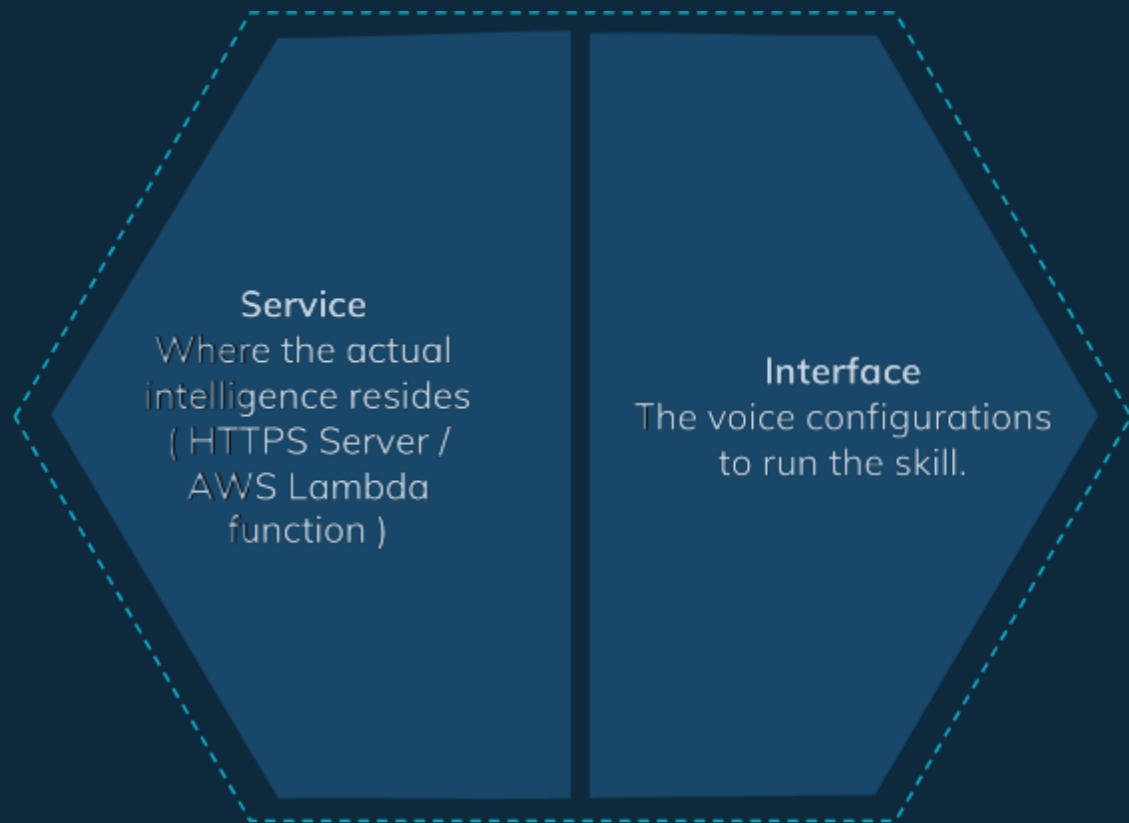


A decorative graphic on the left side of the slide consists of a large cyan hexagon in the center, surrounded by several smaller hexagons of varying shades of blue and cyan. These smaller hexagons contain white icons: a lightbulb, a thumbs-up, a smartphone, a magnifying glass, a gear, and a speech bubble. There is also a small network-like icon with a central node and several connecting lines.

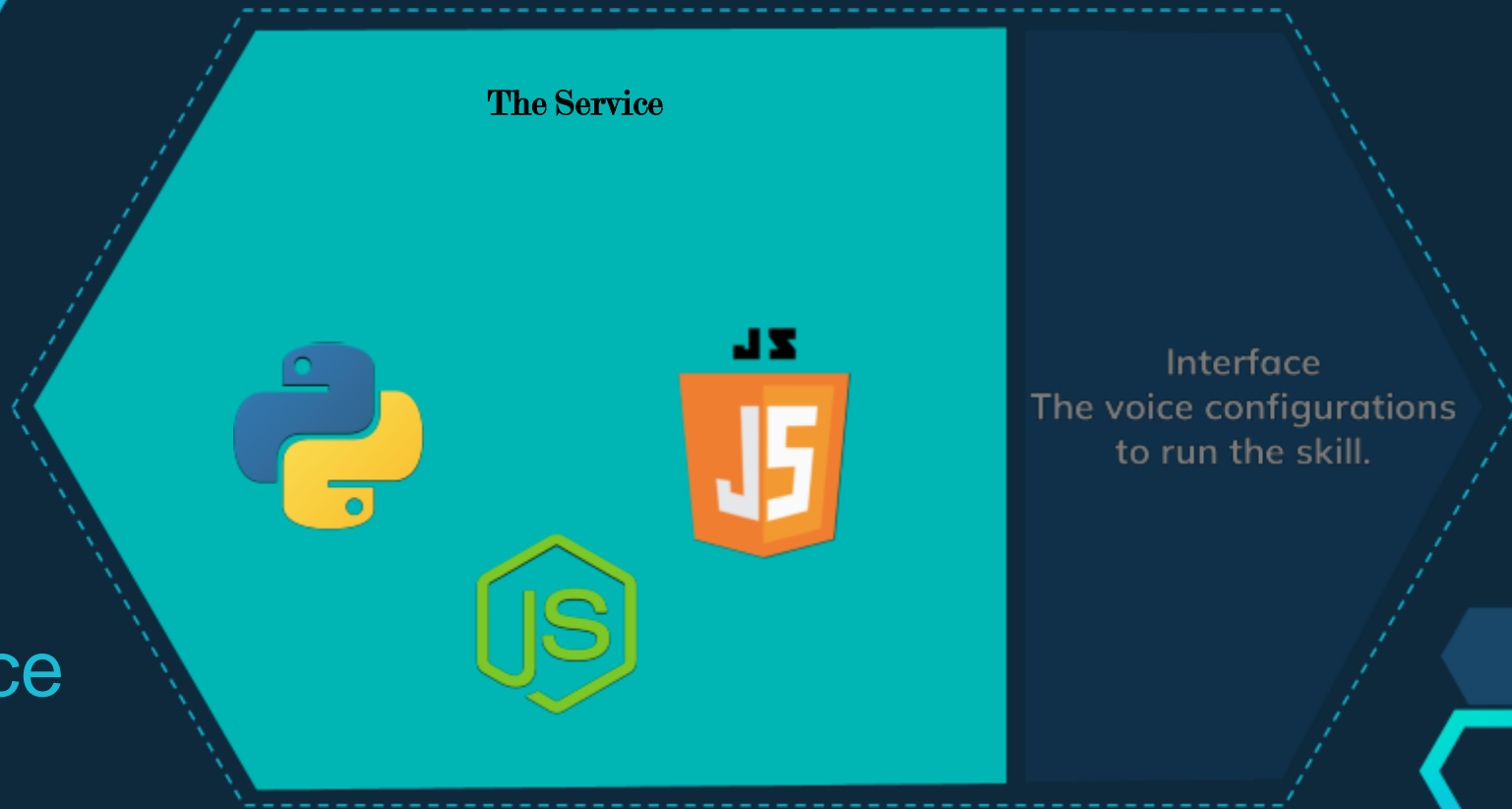
# The Alexa Framework



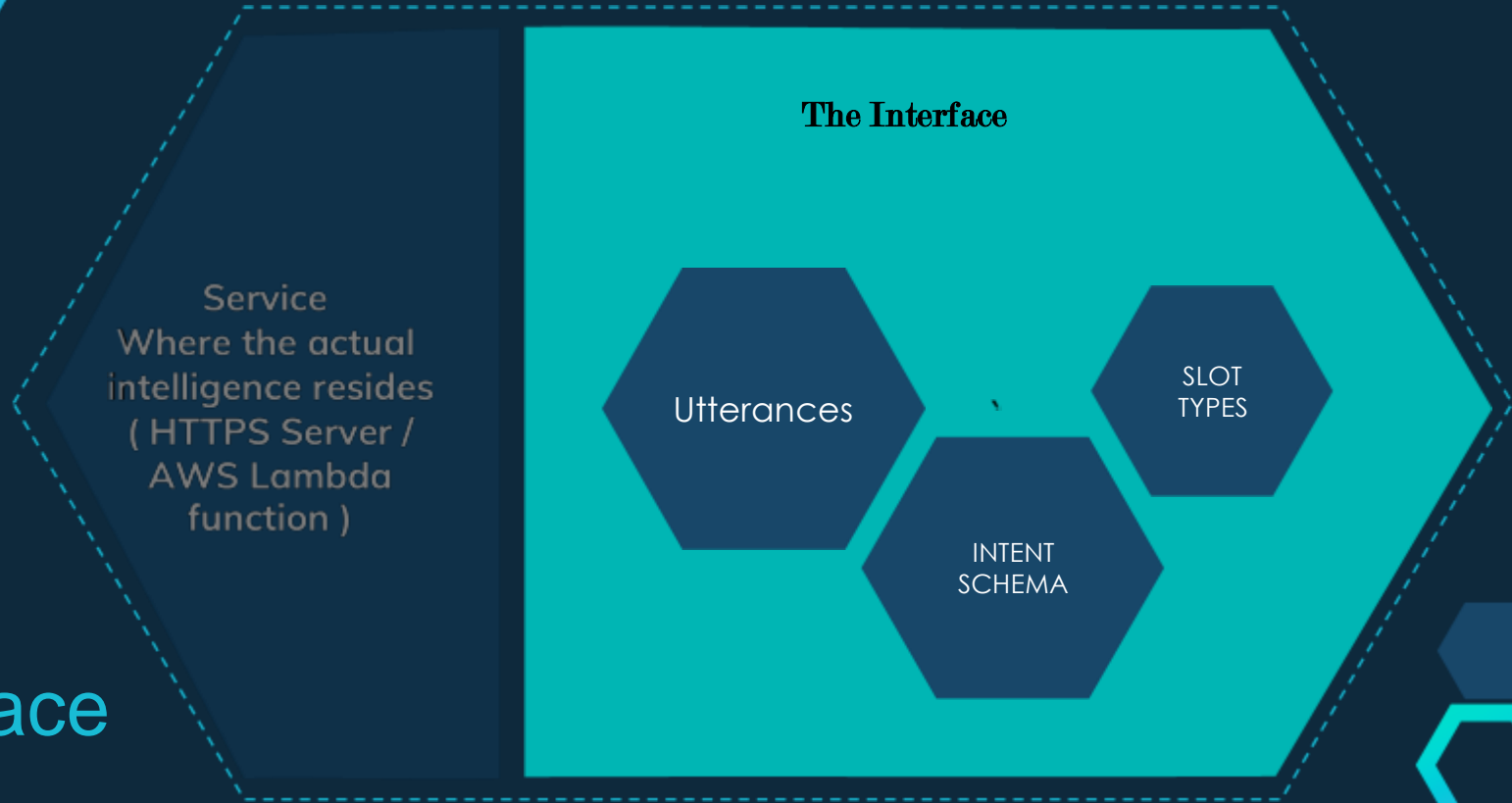
# Skill



# Skill Service



# Skill Interface





# Terminologies

Building a Skill



Slot

“Alexa, ask Zomato for the nearest [restaurantType]”

Invocation Name

Utterance

maps to

Intent





# Invocation Name

What a user must say to “start” the skill. Each skill has a unique Invocation Name





Slot

“Alexa, ask Zomato for the nearest [restaurantType]”

Invocation Name

Utterance

maps to

Intent





# Utterance

A spoken phrase that maps to an intent. A skill can have many utterances.







Slot

“Alexa, ask Zomato for the nearest [restaurantType]”

Invocation Name

Utterance

maps to

Intent





# Slot

A variable within an utterance. Variable possibilities must be defined.





Slot

“Alexa, ask Zomato for the nearest [restaurantType]”

Invocation Name

Utterance

maps to

Intent





# Intent

Whenever you speak to an Alexa Skill, your message is mapped to an intent which decides what code to run within your skill.





Slot

**“Alexa, ask Zomato for the nearest [restaurantType]”**

Invocation Name

Utterance

maps to

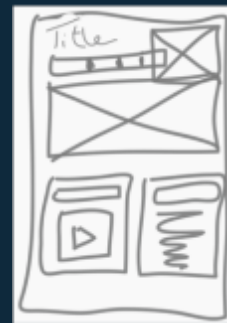
Intent





**Service**

Utterances



Interaction Model



**Interface**





# Flask-Ask

A Flask extension that has wrappers for the common constructs for the Alexa skills API.

Package	Version	
ask-sdk-runtime	<a href="#">pypi</a> <a href="#">v1.9.0</a>	<a href="#">downloads</a> <a href="#">37k</a>
ask-sdk-core	<a href="#">pypi</a> <a href="#">v1.9.0</a>	<a href="#">downloads</a> <a href="#">50k</a>
ask-sdk-dynamodb-persistence-adapter	<a href="#">pypi</a> <a href="#">v1.9.0</a>	<a href="#">downloads</a> <a href="#">37k</a>
ask-sdk	<a href="#">pypi</a> <a href="#">v1.9.0</a>	<a href="#">downloads</a> <a href="#">36k</a>
ask-sdk-webservice-support (Beta)	<a href="#">pypi</a> <a href="#">v0.1.1</a>	<a href="#">downloads</a> <a href="#">352</a>
flask-ask-sdk (Beta)	<a href="#">pypi</a> <a href="#">v0.1.0</a>	<a href="#">downloads</a> <a href="#">366</a>
django-ask-sdk (Beta)	<a href="#">pypi</a> <a href="#">v0.1.0</a>	<a href="#">downloads</a> <a href="#">326</a>



# Flask-Ask

A Flask extension that has wrappers for the common constructs for the Alexa skills API.

- Has decorators to map Alexa requests and intent slots to view functions
- Helps construct ask and tell responses, re-prompts and cards
- Makes session management easy
- Allows for the separation of code and speech through Jinja templates
- Verifies Alexa request signatures





# [ Developer Experience ]

Let's see how the code looks like.

# ASK CLI

Initialization, clone and deploy of skill and all related resources

`ask new`

Allows you to create a skill from a template, with all files needed to deploy.

`ask clone`

Clones a skill present in the environment of development to a local folder.

`ask deploy`

Loads a skill and all its resources in the development environment.

## **skill project folder**

```
|  
|-- .ask/  
    |-- config  
|-- hooks/  
    |-- post_new_hook.ps1  
    |-- pre_deploy_hook.ps1  
|-- lambda/  
    |-- lambda_function.py  
|-- models/  
    |-- en-IN.json  
    |-- en-US.json  
|-- skill.json
```

Dyna

Saving and retrieving

attr  
dyna



Saving and retirement

# Custom Code

```
sb = SkillBuilder()

class RequestLogger(AbstractRequestInterceptor):
    """ Logs the alexa request """
    def process(self, handler_input):
        logger.debug("ALEXA REQUEST: {}".format(handler_input.request_envelope.request))

sb.add_request_handler(LaunchRequestHandler())
sb.add_request_handler(HelpIntentHandler())
sb.add_request_handler(ExitIntentHandler())

sb.add_request_handler(KpiHandler())
sb.add_global_request_interceptor(RequestLogger())

lambda_handler = sb.lambda_handler()
```

# Localization and Translation

It is possible to manage the translation of a skill for the different languages supported by Alexa. The addition of supported languages is done by the Developer Console. This will automatically create language templates (JSON files) for each language.

Within the texts you can insert SSML meta tags too, to unlock additional features of language in Alexa

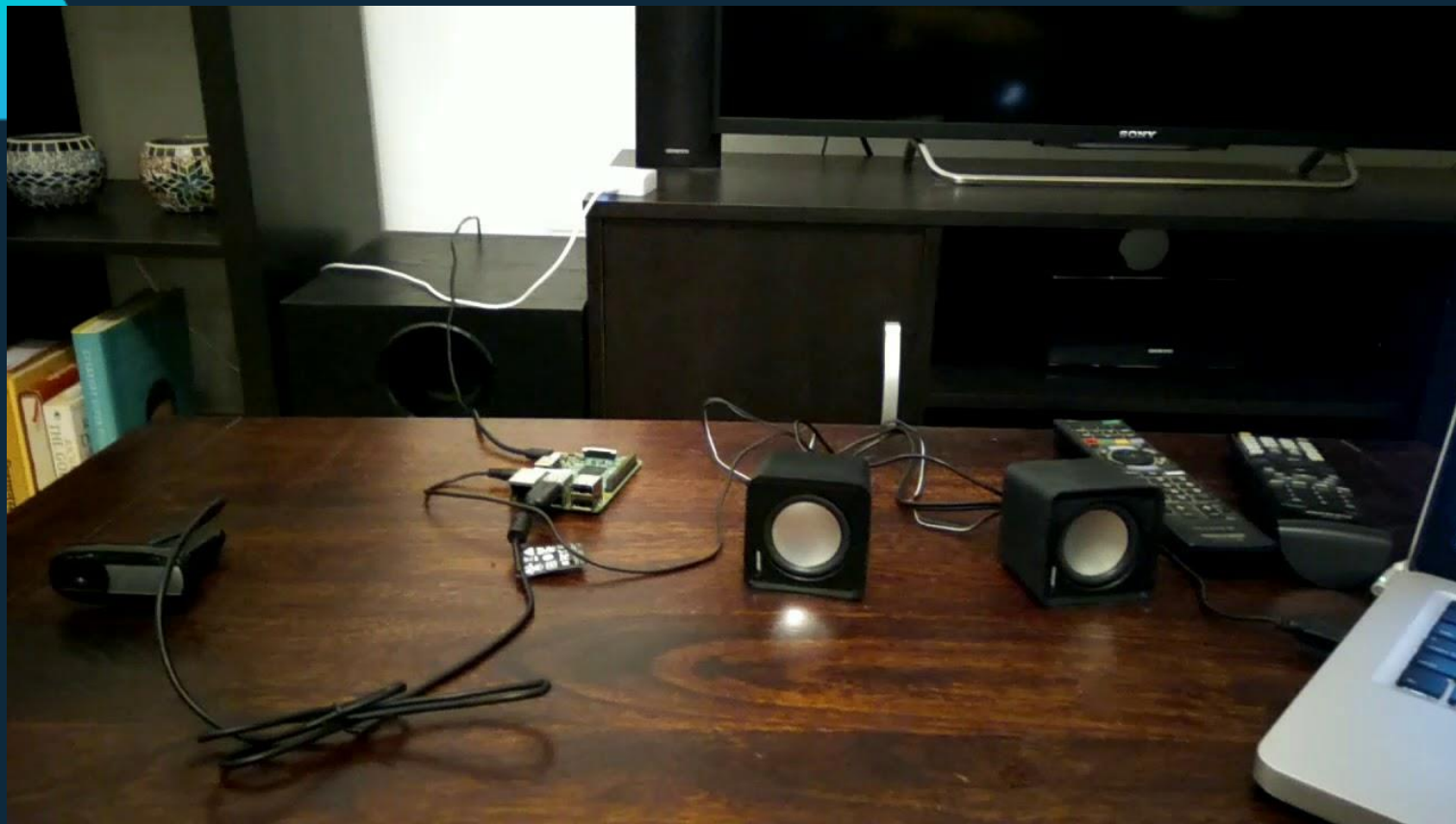
File .py

```
#. NOTE: skill name
msgid "SKILL_NAME"
msgstr "<lang xml:lang='en-US'>Web Analytics</lang>"
#. NOTE: whisper response for main intent
msgid "WHISPER_RESPONSE"
msgstr "<amazon:effect name='whispered'> Welcome
      tutti</amazon:effect>"
```



# Live Deploy & Testing :







# Testing a Skill



A decorative graphic on the left side of the slide consists of a large cyan hexagon in the center. Surrounding it are several smaller hexagons of varying shades of blue and cyan. Some of these hexagons contain white icons: a lightbulb, a thumbs-up, a smartphone, a magnifying glass, a gear, and a speech bubble. There is also a small network diagram icon with a central node and several connected peripheral nodes.

# ngrok

exposes local servers behind NATs and firewalls to the public internet over secure tunnels.



# ngrok

Local http to public https!!!

- Demoing web sites without deploying
- Building webhook consumers on your dev machine
- Testing mobile apps connected to your locally running backend
- Stable addresses for your connected devices that are deployed in the field
- Running personal cloud services from your home

A decorative graphic on the left side of the slide. It features a large, solid cyan hexagon in the center. Surrounding it are several smaller hexagons of varying shades of blue and cyan. Some of these hexagons contain white icons: a lightbulb, a thumbs-up, a smartphone, a magnifying glass, a gear, and a speech bubble. There is also a small network diagram icon with a central node and several connecting lines.

# AWS Lambda

AWS Lambda is a compute service that lets you run code without provisioning or managing servers.



# AWS Lambda

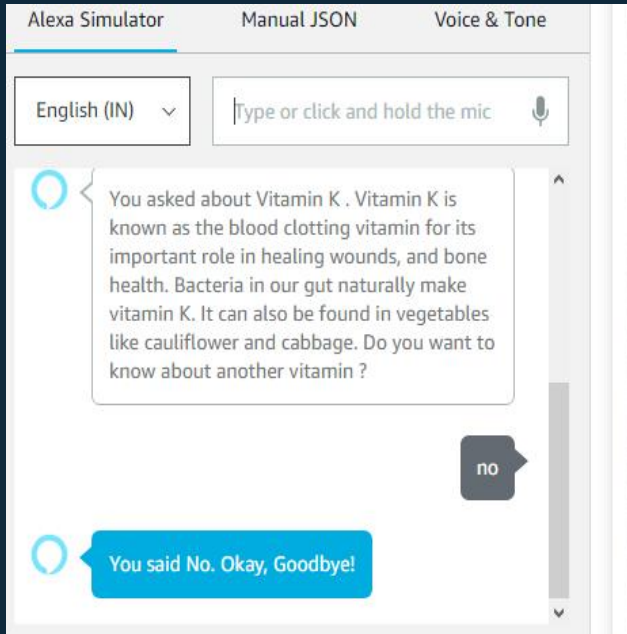
A compute service that lets you run code without provisioning or managing servers.

- Executes your code only when needed
- Scales automatically, from a few requests per day to thousands per second.
- Charged for compute, not downtime
- Any kind of application backend
- build serverless applications composed of functions that are triggered by events

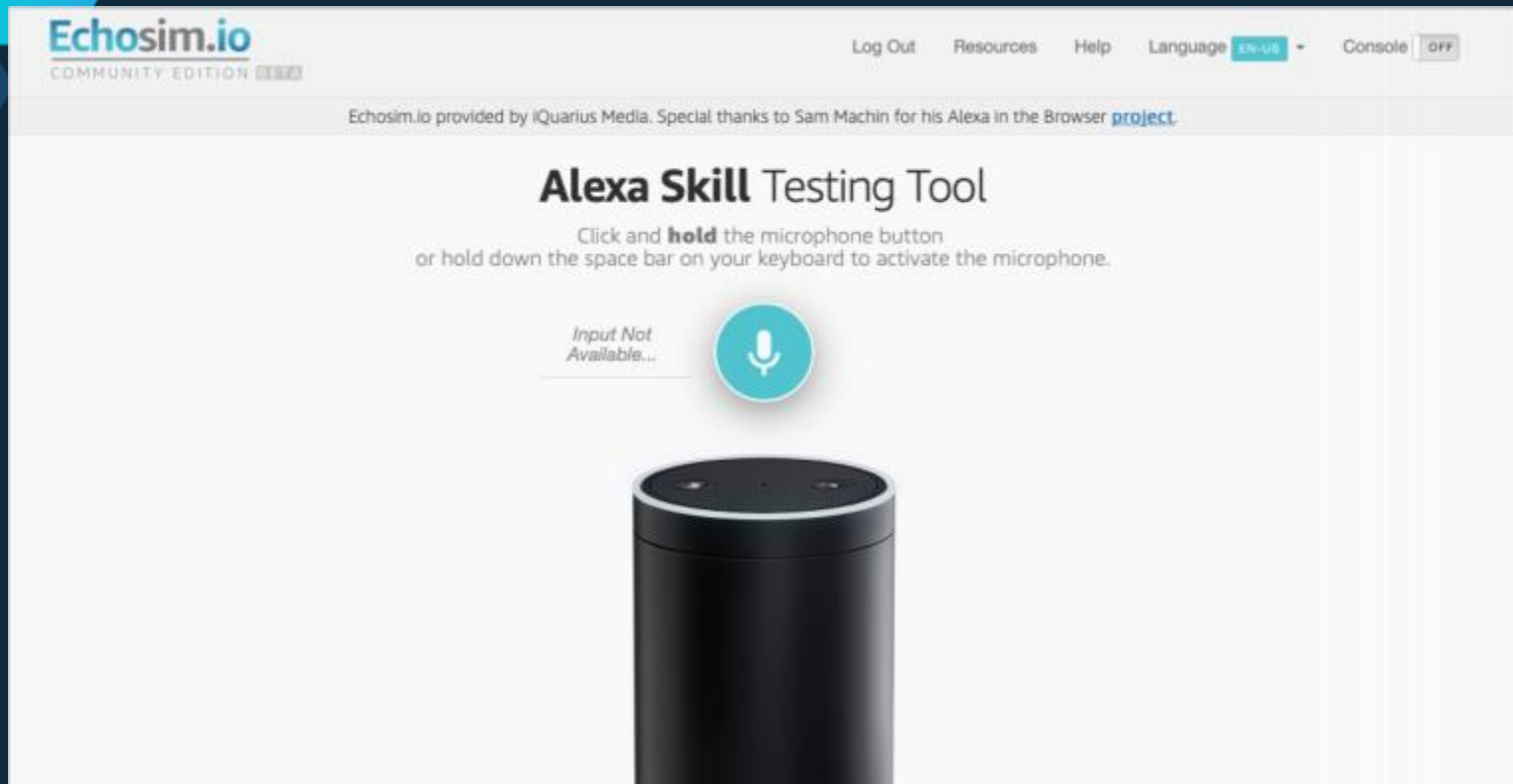


# Alexa Skill Test

# Alexa Simulator



# Echosim.io



Source: Google Images

## Getting started



\*Fields required for certification



amzn1.echo-sdk-ams.app

## Airport Info

airport info

☐ HTTPS ☒ Lambda ARN (Amazon Resource Name) 

```
arn:aws:lambda:us-east-1:██████████:function:airportInfoService
```

Save

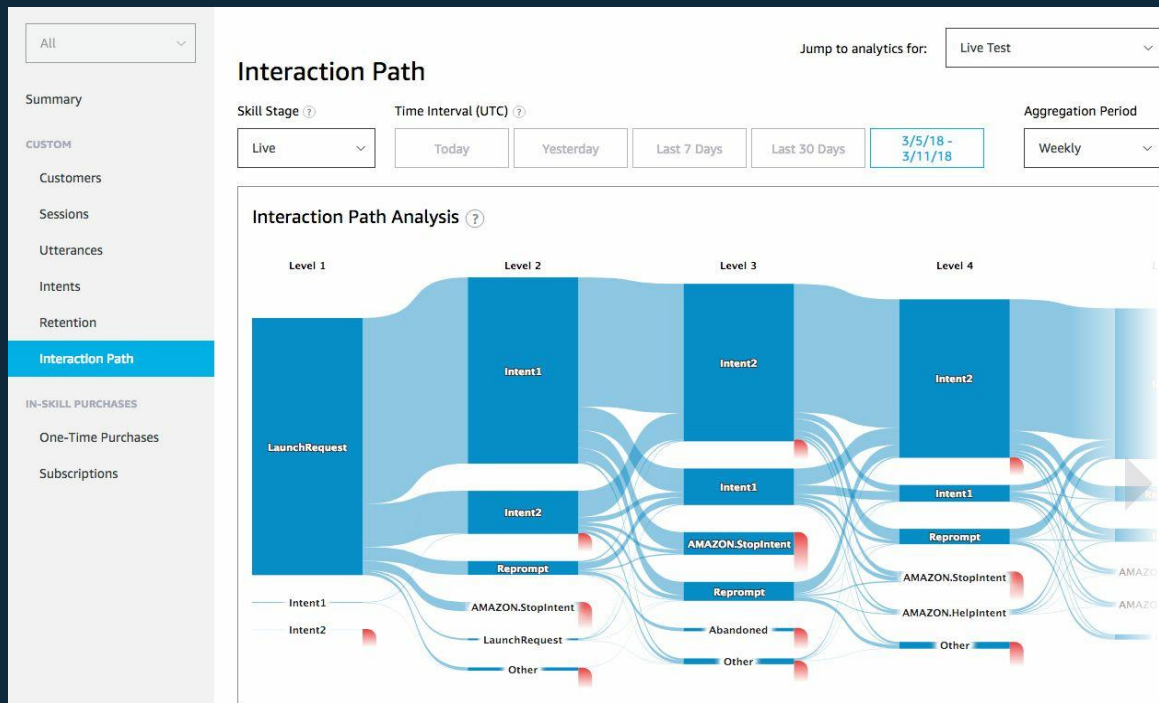
Next



# Web Analytics Skill



Google  
Analytics



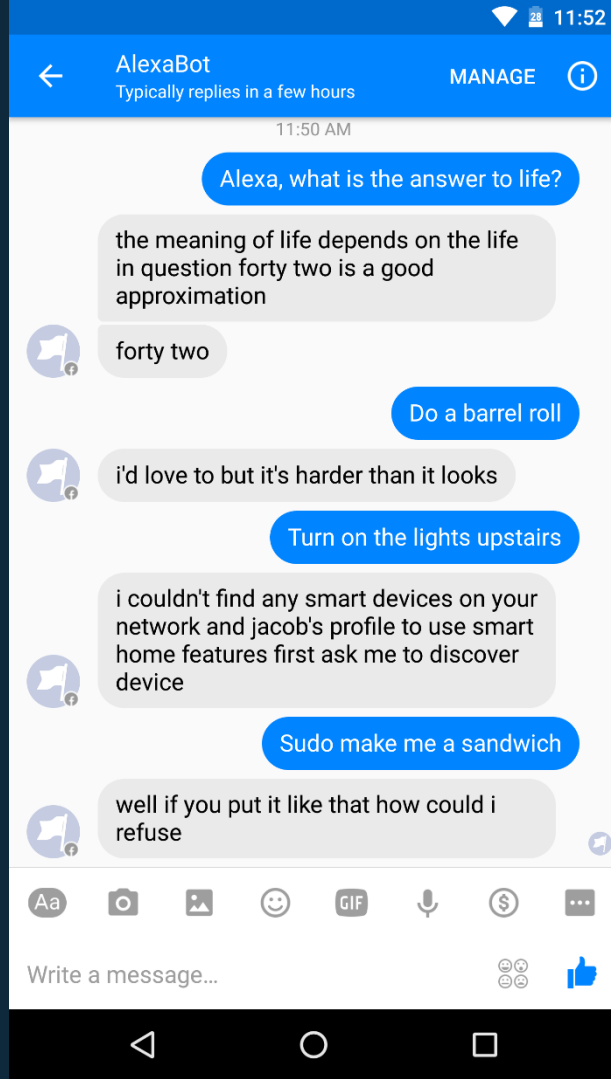


# The Future

- Interactive Social Networks
- Interactive Idea boards
- Voice Gaming
- Crisis Alerting
- Office Integration – Meeting room bookings, etc.
- Always aware state!! ( Creepy!! )



# AlexaBot





# Sources

- <https://www.theverge.com/2019/1/4/18168565/amazon-alexa-devices-how-many-sold-number-100-million-dave-limp>
- <https://voicebot.ai/2018/12/24/rbc-analyst-says-52-million-google-home-devices-sold-to-date-and-generating-3-4-billion-in-2018-revenue/>
- <https://developer.amazon.com/alexaprize/challenges/current-challenge/faqs>
- <https://github.com/alexa/alexa-skills-kit-sdk-for-python>
- <https://medium.freecodecamp.org/how-to-create-an-alexa-skill-that-manages-to-do-lists-11c4bab29ea5>
- <https://github.com/alexa/skill-sample-nodejs-petmatch/tree/master/analytics>
- <https://developer.amazon.com/it/docs/account-linking/understand-account-linking.html>





# Thanks!

## Questions??

You can find me at:

- q [ganeshkumartk@outlook.com](mailto:ganeshkumartk@outlook.com)
- q [coderganesh@github](https://github.com/coderganesh)
- q [GaneshKumarTK@LinkedIn](#)
- q [GaneshKumarTK's Portfolio](#)

