Hack the UK with Amazon Alexa

posted by Amazon Alexa on MindSumo.com

Submission by **Vignesh Rajagopal**

Full name:	Vignesh Rajagopal
School-affiliated email address:	vrajago6@hawk.iit.edu
School name:	Illinois Institute of Technology
Current major:	Electrical and Computer Engineering
Graduation date:	December 2017
Name of skill:	Office Assistant (UK) (prototype)
SKILL ID:	
amzn1.ask.skill.ab745af3-5a71-4cf3-97f2-26a0dd7ee10d	
Link to demo video (via Youtube):	http://youtu.be/MqwlS2_eUx4
Link to source code (via GitHub):	
https://github.com/vigr13/HackUKAlexaMindSumoChallenge	

Contents

Identifying the problem

I have always wanted to get to know the United Kingdom better – unfortunately, the only firsthand experience I have consisted of a rainy afternoon layover in London. Luckily attending middle school in Stockholm, Sweden meant that many of my former classmates currently work in the UK and other neighboring countries. Knowing a catchup with some of them was long overdue, I sent out a few pleasantries and waited (time difference).

Hearing back from them was intriguing because as I asked them about their day to day lives, one common trend I noted was their exasperation when it came to talking about their day job. Curious, I dug further to try and see why this was the case and what could be done.

As it turns out, my friends (mostly fresh college graduates) who worked in corporate offices were often given lowly jobs that didn't allow them to explore their full capabilities and contribute in the impactful manner they were expecting to. The majority of them have to do secretarial-level work like take notes at meetings, send out reminders, and coordinate between different calendars for scheduling events. While this type of work is by no means lowly, it failed to stimulate my friends who had earned an expensive degree and felt like their entry-level positions was similar to that of an intern. In a similar thread, I presume that interns at such corporations feel underwhelmed and ultimately unprepared for the so-called real world when they do get out there.

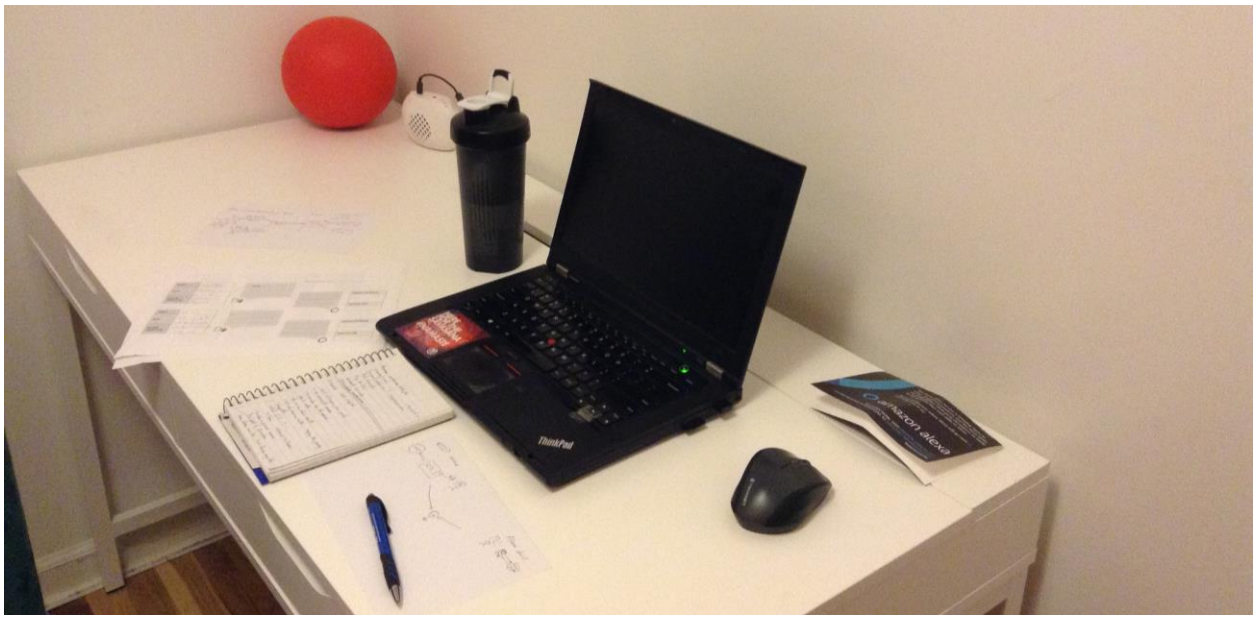
Reading further into corporate environments in the UK, I learned that a large majority of employees in the UK work long, intense hours. The UK has opted out of the European Working Time Directive (see link: http://www.flexibility.co.uk/flexwork/offices/facilities2.htm) which means that work weeks longer than 48 hours are common. Having a personal assistant to take care of the more mundane task so that employees are able to be more efficient and productive seems sensible.

Furthermore, there seems to be a larger emphasis on the work-life balance across the pond than here in the United States. The drawback is that only a small percentages of companies have policies in place to help increase the work-life balance (see link: http://businessculture.org/northern-europe/uk-business-culture/ukwork-life-balance/). A scheduling/personal assistant that knows details about the employee's personal schedule would help to avoid potential scheduling conflicts and help maintain that employee's satisfaction and productivity.

I have seen Alexa in a plethora of places – yet, it is only ever portrayed as a "personal" voice assistant. Why not take that a step further, harness existing capabilities and turn Alexa into a "business" voice assistant?

Developing an Alexa-based solution

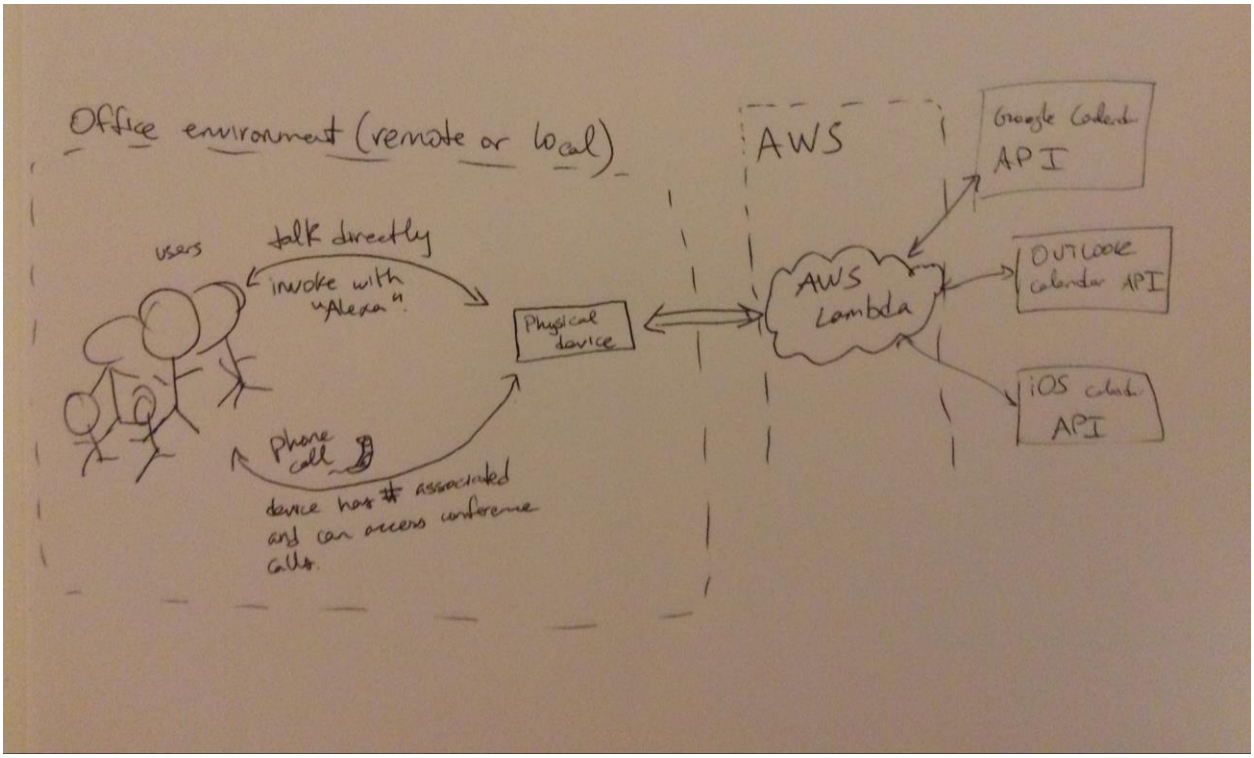
I recently attended an Alexa dev days event here in Chicago, Illinois and was inspired to try and create a workout-assistant skill. This provided me with enough familiarization that I had no major technical hurdles.



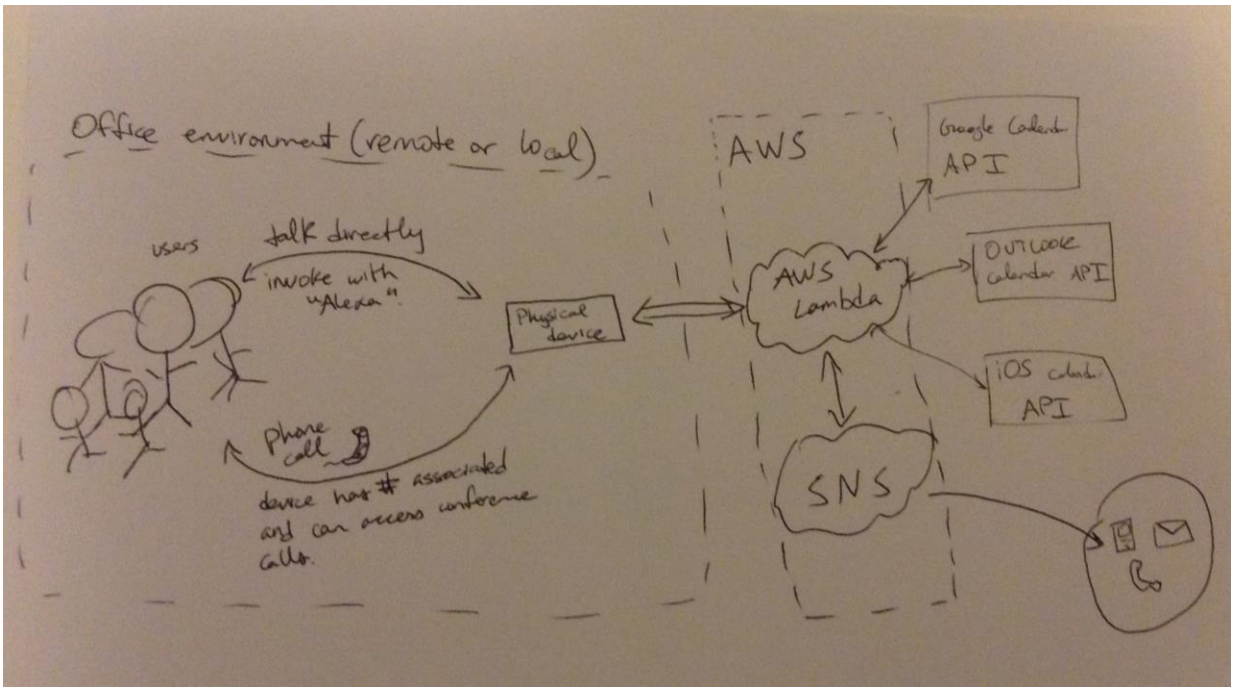
The workstation – including one dodgeball to help me think

Paper

I love putting pen to paper so that is where I began. Below is my image of what I initially came up with.

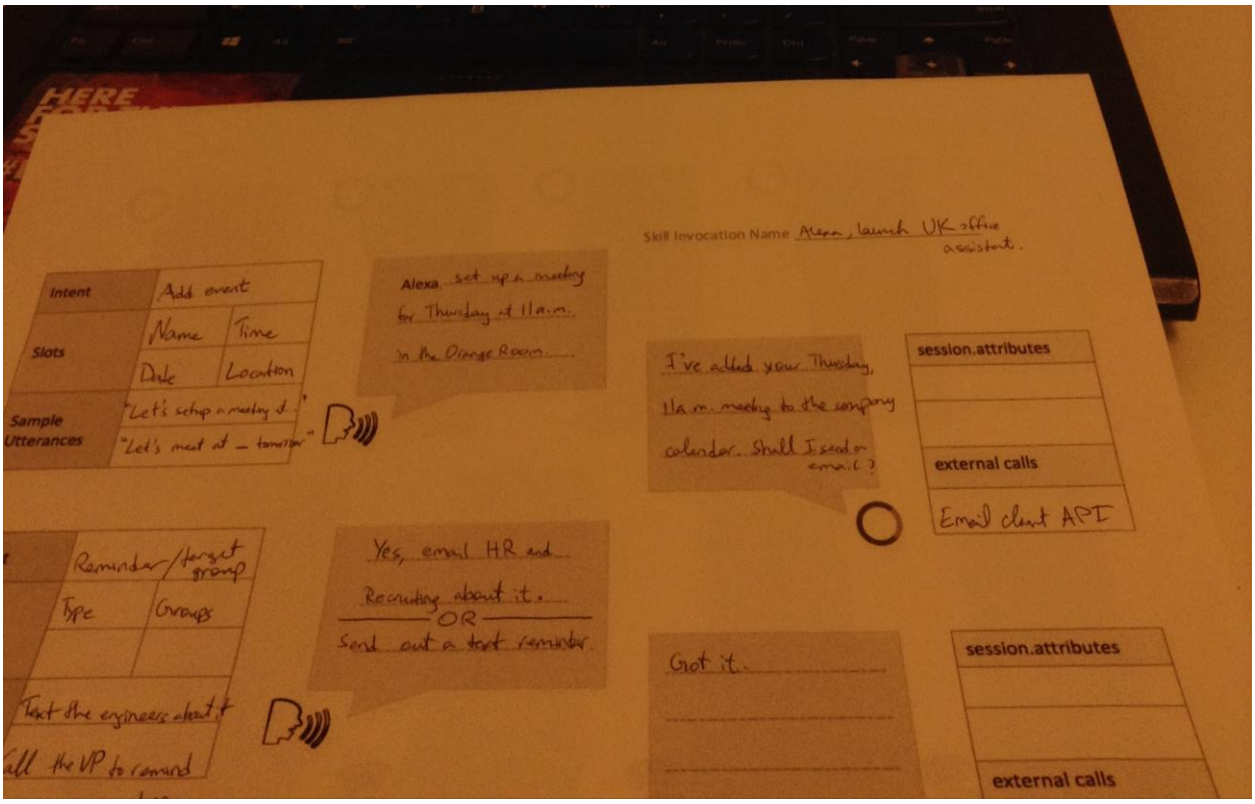


This seemed a little basic, so I decided to explore the different services offered by AWS. The "AWS in plain English" webpage proved useful for this. While thinking about my application, the concern at the forefront of my mind was the tasks my UK friends had described to me. I decided to incorporate the use of SNS in order to create topics. My improved and final block diagram looked like this:



Voice User Interfaces (VUI)

The Amazon Alexa event came in extremely useful here as there was a whole section on good VUI practices. This is undoubtedly important – developers have always known the importance of user interfaces and keeping this at the forefront of their minds usually serves to create a long-lasting product. I used the following blueprints to ensure that the voice interface for my skill would be intuitive and require very little thinking or learning on the users' part:



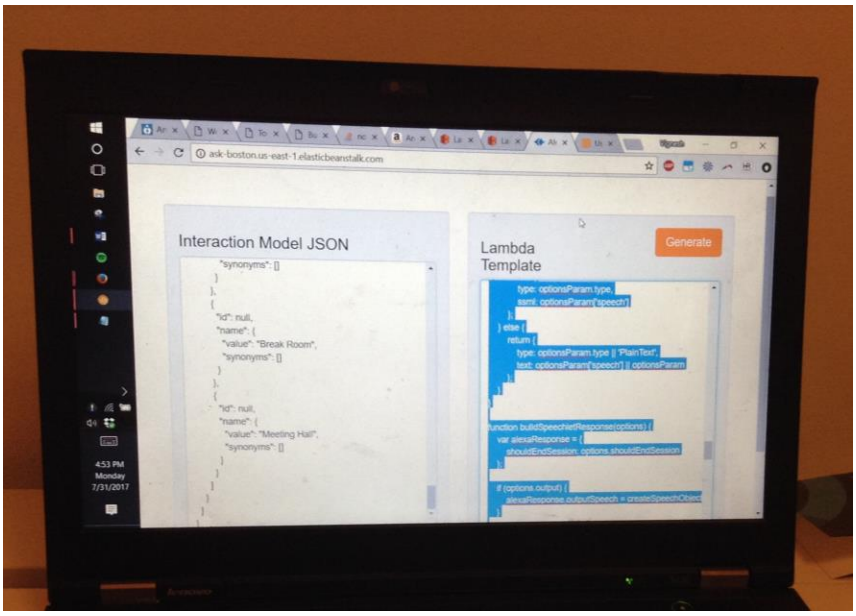
AWS environment

While I am familiar with sending a "card" from an Alexa device to the Alexa app on a smartphone, this has the minor inconvenience in a group setting of requiring every member to have the app installed. A more universal solution would be to connect this skill to AWS's Simple Notification Service, where a topic can be opened for each session and then once everyone involved subscribes to this topic via either SMS or email, subsequent notifications can simply be published to the topic and then update all subscribed. In addition, SNS offers HTTP subscription which can be useful if the office has a centralized TV bulletin board that posts updates and decisions made. Posting to this would then allow for further cohesiveness and allow the various departments of the corporation to function together like a well-oiled machine.

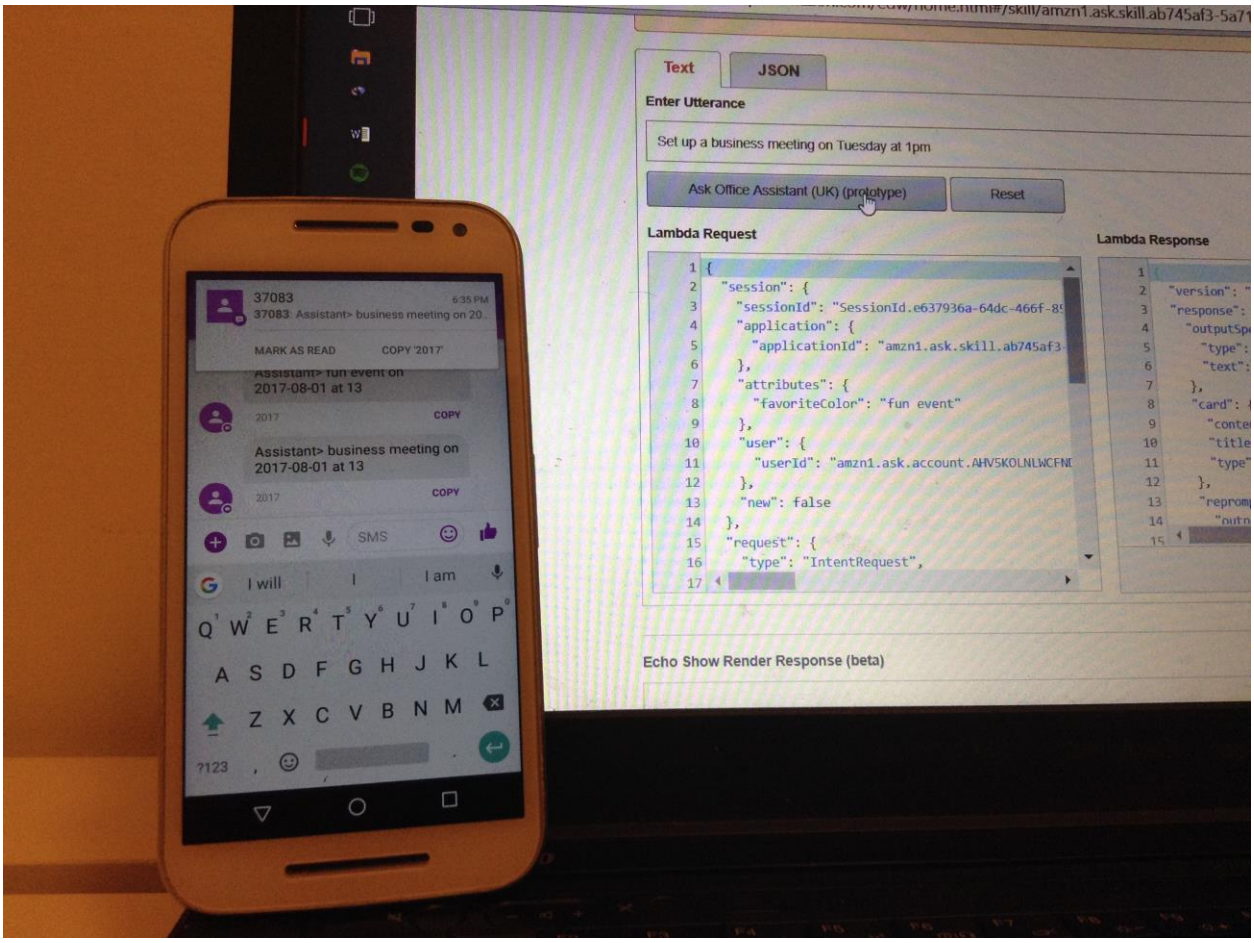
DynamoDB can further be used in order to increase the capabilities of this skill. While I did not explore this option in this implementation, I can see how it can be used to keep a record of the various events scheduled as well as allow for authentication. This way, unauthorized event requests can be rejected and logged in order to maintain the security and privacy of business operations.

Developer portal

The developer portal for building a skill is very intuitive and guides you through the steps in a cohesive manner. In addition, certain third party tools such as Echosim.io and Alexa2Lambda (pictured) are useful for voice tests of your skill and for generating Lambda code given your list of intents, respectively.



Along with providing tips every step of the way, there is also a very useful testing guide where you can test how well your skill responds to a typical series of user voice commands. The following image encapsulates one such test of my skill, where I input "Set up a business meeting on Tuesday at 1pm". This contains three slots – name of meeting, date, and time. SNS is then used to send my phone a text message (pictured) detailing this upcoming event.



Third-party API integrations

Calendar clients

Third-party calendar APIs can also be setup through the helper app for the skill in order to customize the Lambda code for the specific business or organization. For instance, during setup, the names, emails, and designations of the various employees can be entered into the database in order to be able to effectively use natural language – such as "inform marketing about these notes" – and have it reach out to all those employees tagged with 'marketing'. This integration would vary from business to business as some may be using GSuite or Outlook or any other of the number of email clients out there. This was not explored here in order to focus on the hardware idea (see following section) but would be done on a case-by-case basis.

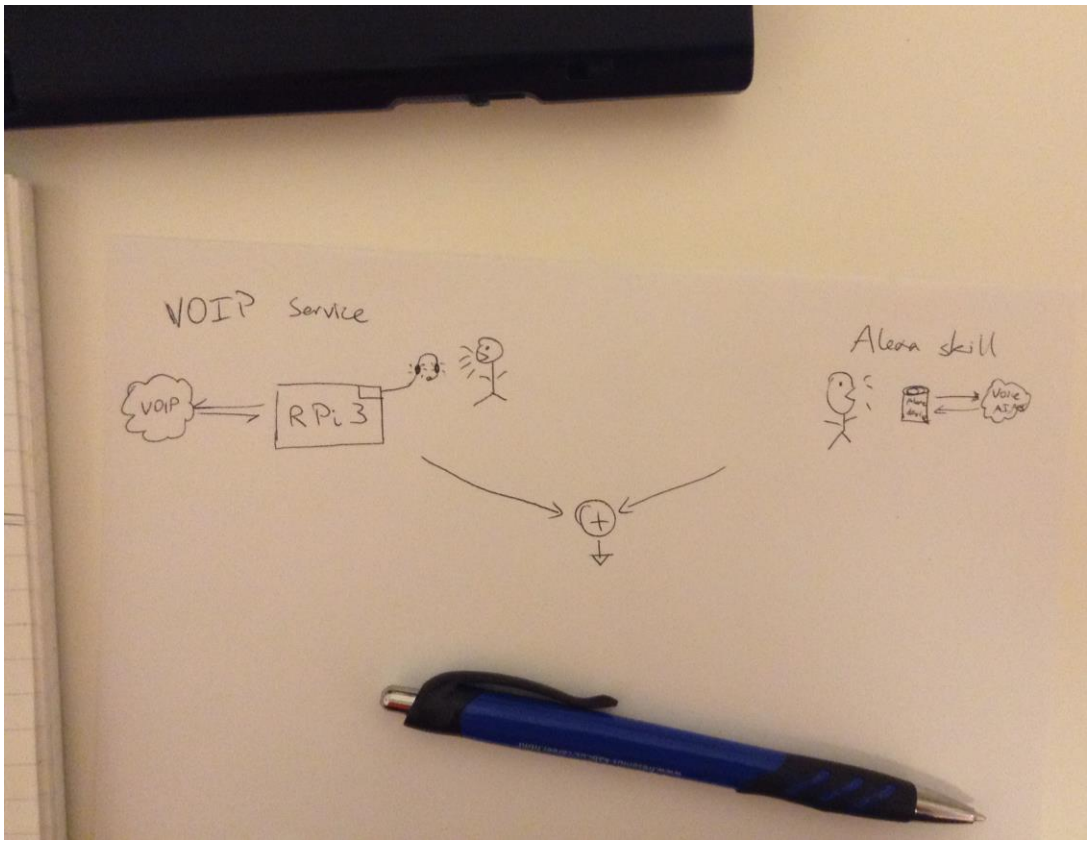
Extendable hardware (Raspberry Pi prototype)

The Idea

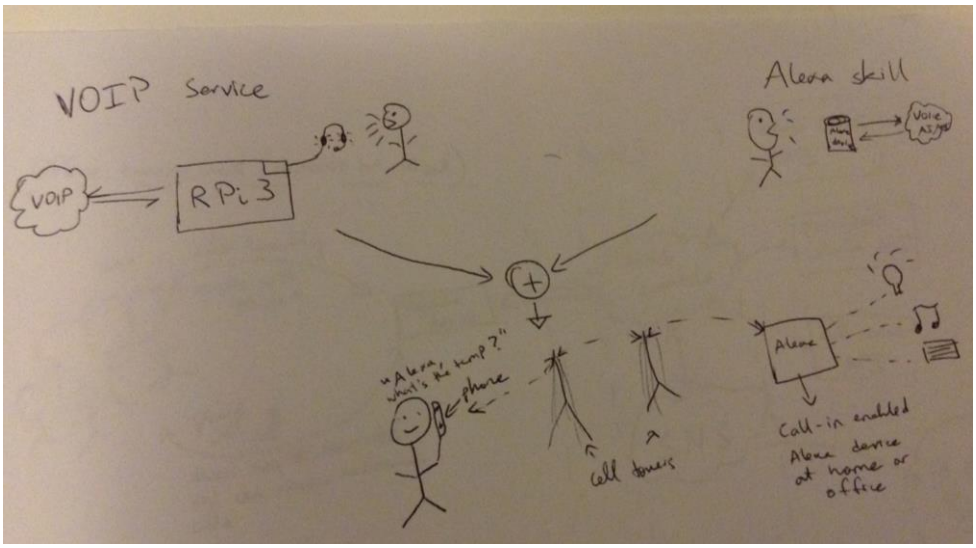
One thing I am currently looking into is extending this Alexa capability in order to allow for remote voice assistance in a conference call environment. As I understand it, there is currently no way for an Alexa owner to "call" their device when they are away from it. This stuck out to me like a valuable feature that could potentially be an easy implementation. Why is physical presence paramount to being able to take advantage of a voice assistant?

With this in mind, I began my research. The Raspberry Pi is my prototyping device of choice, and Linux would allow for the type of customization I have in mind. The plan is to use a VOIP service (many are freely available) – preferably one which is compatible with conference calling – and play around with the Linux drivers such that the audio stream being input to the Alexa service is the stream being output from the VOIP service onto the device. Similarly, the audio stream being output from the Alexa service is being input to the VOIP service. While getting into the implementation of this was a little out of scope, one option here in getting streaming files would be to use the AWS service Polly, which allows for Alexa's words to be saved (as well as changing voices, etc.). If it doesn't work directly, then Polly could be used in order to save buffered audio samples and use them as input/output in the system.

Initially, it would work like this:



The two separate working entities would then come together to create the solution we are looking for. Ultimately, I came up with the following block diagram for how this device will work:



The possibilities of such a feature are endless. A few applications of being able to call Alexa are outlined in the following section.

Applications

Conference calls

If you're able to call Alexa, and Alexa is able to call you, then Alexa can be called into a conference call remotely – whether it be Skype or a phone call. In such a setting, Alexa would simply be listening in, similar to an Echo device, until invoked by "Alexa". This can be used to schedule meetings, set up email reminders, and more all without leaving the group call environment. This way, everybody is aware of the appointment being set up. This could even be taken a step further if – instead of sitting idle – Alexa keeps a buffer of say, the last thirty seconds of conversation. That way, a simple "Alexa, make a note of that last part" would allow the persons to create a live summary of the decisions made, questions brought up, and issues raised in the meeting. On the back end, these notes made could be used to send out a call summary to those part of the organization – reiterating the points of the call to participants as well as informing the other teams in the corporation about any major inter-departmental decisions made in the call.

Smart Voicemail

If you miss a call from a friend, instead of the outdated "Please leave a message after the beep", Alexa could pique up with something like "Hey, John is currently at a meeting. Ask me what you would like to know!" To this, the friend could ask, for instance "What time will he be able to call me back?" or "Just let him know to call me when he gets a chance. It's about the game tonight."

Smart voice navigation

Phone menu systems at large corporations are slowly moving away from the boring "Press 1 for Help, and 2 if you're tired of fumbling with your keypad" to a voice-based recognition system where

you can verbally ask for the service that you require, for instance. Adding intelligence to this recognition system with Alexa would be the logical next step, allowing for a more fluid user experience. The correctly designed VUI would allow for a potential customer to feel like they were receiving the same amount of care and attention as they would if they were speaking to a company representative.

Call Alexa from any location

This could be useful for several things. Smart home systems that are compatible with Alexa, for example, would benefit from such a capability. If you're away on an extended stay and would like to check on your house security system, or check if your mailbox has received any mail, or ensure that the dog-sitter has performed the required visit/walk – you can now do this by simply calling Alexa.

Closing remarks

One of the things I picked up on during the Amazon Alexa dev conference was that each time a developer designs a skill, and assigns "synonyms" within the skill developer, Alexa learns a little bit more. I am sure there are other ways that Alexa continues learning in order to refine both its voice recognition and its natural language processing ability. In order to truly perfect this, a large scale is need, and I feel that providing the ability to access Alexa from any location with a cellular connection is a step in the right direction to achieve this goal. Being able to implement this will allow for the office environment applications previously discussed, and I truly believe that it is something that can be achieved with today's existing technologies. The reason for my submission so close to the deadline is because I spent a great deal of time attempting to implement this on the Raspberry Pi – I hope to be able to do so within the next month to get a working prototype/proof of concept. Ultimately, I believe that this feature can be integrated into newer releases of the existing hardware – Echo, Echo Dot, etc. – by adding a GSM slot or even just using VOIP and updating the firmware accordingly.

Thank you for your time and consideration in reading this document.

Relieving stress as the deadline grew closer

--Vignesh Rajagopal

August 3, 2017