

# Contents

|                                       |      |
|---------------------------------------|------|
| 1. Introduction .....                 | 1 -1 |
| 2. Overview .....                     | 2 -1 |
| 3. Reference Architecture.....        | 3--1 |
| 4. Conceptual Design .....            | 4-2  |
| 5. Example Use Case .....             | 5 -2 |
| 6. Development Phases .....           | 6-3  |
| 7. Current/Future Functionality ..... | 7 -3 |



# 1. Introduction

The following document provides an overview of the ██████ Voice Assistant product. This document covers a product that is currently in development and is subject to change without prior notice. This document is intended for internal use only.

# 2. Overview

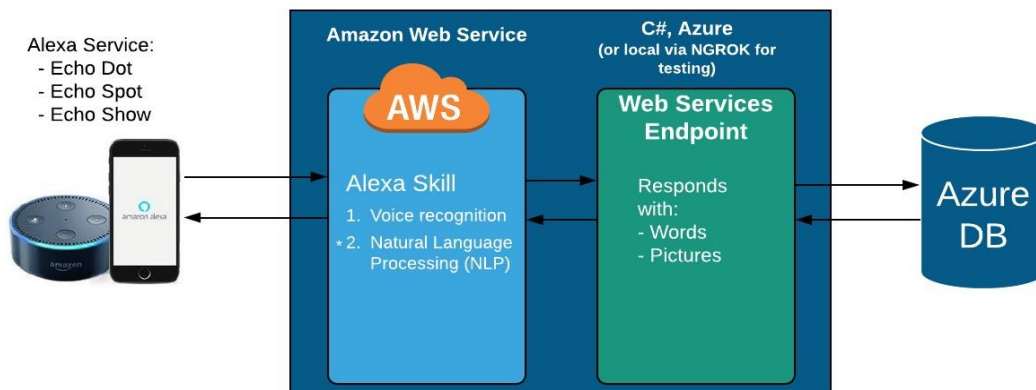
As a natural technological evolution of ██████ group's functions, a recommendation was submitted to develop a voice-assistant platform that would allow users to use their Amazon Alexa devices (Echo Dot, Echo Spot and Echo Show) to receive bills, EOBs, etc. and have them read to them, and in the case of bills, to pay those bills, if so desired. A demo of this product was presented to leadership in February of 2018 and they would like us to expand on this demo and hand it over to ██████ team in the hope of developing it into a full-fledged product.

# 3. Reference Architecture

A basic logical architecture has been developed featuring three main components:

1. the database
2. the Alexa service, consisting of:
  - The Alexa Skill
  - Web service endpoint
3. the Alexa Echo devices

See figure below for a logical diagram of the Dara architecture:



\* To define the voice interface, you map users' *spoken input* to the *intents* your cloud-based service can handle. To declare this mapping, you supply the following inputs:

1. **Intents:** An *intent* represents an action that fulfills a user's spoken request. Intents can optionally have arguments called *slots*.
2. **Sample utterances:** A set of likely spoken phrases mapped to the intents. This should include as many representative phrases as possible.
3. **Custom slot types:** A representative list of possible values for a slot. Custom slot types are used for lists of items that are not covered by one of Amazon's built-in slot types.

## 4. Conceptual Design

The end-user makes a request through the Echo device, for example, “Alexa, read my mail.” The request is then sent to Amazon’s web service platform. Alexa’s voice recognition software, identifies the user. Alexa’s Natural Language Processing (NLP) skill then turns the request into Intents. An Intent represents an action that fulfills the user’s request. So in the request above, “read mail” may represent a request, for Alexa to go out and retrieve your emails. This is how it knows what to do.

The NLP is important, because there may be many ways for a user to request the same action, for example, “Alexa, read my mail,” or “Alexa pull up my email,” etc. They all mean the same thing, but unless those are entered in the Intent, Alexa will not recognize them. Intents can also have Slots. Slots are arguments or variables within the Intent. Let’s say Alexa is set up to check both a user’s Gmail account and their work email account. In this case, the Intent “read mail” may have Slots for “work email” and “Gmail”. In this case, Alexa may ask the user to confirm which account he or she wishes to access.

Once the call is made (and clarified if necessary), the request is passed to the Azure web service endpoint. This is the component that queries the database to retrieve the information requested. For testing purposes, this component will use NGROK instead of an Azure endpoint to allow it to work locally instead of going out to query an actual database. This will change once the product becomes available. The web service endpoint then passes the information back to the Amazon Web Service, and this can be either in the form of verbal or written text, or in the case of case of Echo Show, visual. The requested information is then displayed on the user’s device.

## 5. Example Use Case

The following use case reflects a sample call between Alexa and a user.

1. A user wants to check his/her email, so he/she says, “Alexa check my mail”.
2. Alexa: “You have a total of 4 unread emails. Two bills, one correspondence and one EOB. To get more details say, ‘bills’, ‘correspondence’ or ‘EOB’.”
3. The user can either touch the screen (if the device is Echo Show) or say what he/she wants read.
4. User: says, or touches, “Bills”.
5. Alexa: “Bill 1 from <company 1> is for x dollars, received on <date> and due on <date>. Bill 2 from <company 2> is for x dollars, received on <date> and due on <date>. What would you like me to do next?”
6. User: “Pay bill from <company 1>.”
7. Alexa: “I’m ready to pay the bill from <company 1> for x dollars, due on <date> with the payment method we have on file. Are you sure you want me to make this payment?”
8. User: “Yes.”
9. Alexa: “The payment has been recorded and a receipt has been emailed to the payment address we have on file. What would you like me to do next?”
10. User: “Nothing.”

11. Alexa: "See you next time."

## 6. Development Phases

The [REDACTED] Voice Assistant development will initially consist of two phases.

1. Demo/POC – this is the product testing phase and is designed to be rebuilt multiple times to achieve the desired result. It does not update the database.
2. Pilot Customer – Fully functional platform for testing with real data and database connectivity.

## 7. Current/Future Functionality

Nine features were originally designed for the platform. Some are already included in the demo while others are currently under development.

- **Check Physical Mail / emails** – Identifies the type of mail / email (EOB, bill, correspondence, etc.) and differentiates it by unread and read items. This is currently in process. The current [REDACTED] skill that reads the "mail" reads them all. Adjustments have been made to read only those that are "unread" (or New in the database). In order to do anything with those emails that have already been "read", a new set of skills needs to be introduced that will do something with those "old" mail items. Perhaps something could be added like "What is the status of all my mail" or something specific to do with those mail items. It is recommended that three states (for now) "New", "In Process" and "Archived" be created. Functionality would be needed to move an item from "In Process" to "Archived", but maybe that is a new skill or an option on the "Read" skills – that is to say "Archive this item" and it would move to an archive state. Current data is static – it never really changes because this is a demo and we don't want to have to reset the data each time someone wants to run the demo. But perhaps some wording changes could at least highlight the concept of moving from "New" to "In Process" and on to "Archived". Like when you say "Pay the bill" while reading the bills, we could include in the confirmation verbiage that comes back "The bill has been paid and successfully archived" or something similar.
- **Read Physical Mail / emails** – Identifies and reads the key part of the mail / email and the actionable items. Currently there is not a separate "Read" skill, but rather the reading of the emails is integrated with the check mail function. For example, when you ask [REDACTED] to "check my mail", it provides a summary of mail items, and then allows you to 'drill down' into more detail, which includes reading the actual mail item itself, including correspondence. This is already included in the functionality.
- **Review Payment History** – Ability to retrieve payment history and when the payment was made. The current [REDACTED] skill "Review my payment history with {CompanyName}" provides first a summary of payments, and then allows you to 'drill down' to a specific payment and get the details of the payment made, including when. This is already included in the functionality.
- **Make Payments** – Ability to fetch the preferred mode of payment and make a secured payment. This functionality is built into the check/read skill. If the mail item is a bill, you are asked what you like to do, and one of the responses can be "Pay the bill". As I mentioned

earlier, I think the verbiage that comes back as a confirmation should be adjusted to reference the archiving of the item. This is still in progress.

- **Send Receipts** – Send receipts to the email once payment is made. The current [REDACTED] skill where you can make payments is designed to actually send an email; however, it has been disabled because we don't have a dedicated email address for [REDACTED], and the 'relay' method that we can use requires that the [REDACTED] service be within the [REDACTED] firewall. Until this issue can be resolved, the actual email process will remain disabled, even though the verbiage that is returned references a sent email.
- **Analytics** – Ability to analyze the data on past usage and payment history – to discover, interpret, and communicate meaningful patterns in data. This may not be something that belongs in the Alexa skill arena, but if anyone has any specific requests on something we could do the team could brainstorm about how we might make that part of the [REDACTED] Alexa skill.
- **Connect to Custom Care** – Seamlessly connects with the customer care on exceptions and unresolved problem. Alexa custom skills do not allow the Echo microphone to be used in a pass-thru mode that would allow us to connect the person to an agent through their Echo. The closest plan we could come up with was to allow them to "Request" a "live" person, and then we would respond with "A customer service representative will contact you on the phone number you have on file shortly". That will successfully simulate what they will experience from the Alexa/Echo perspective. This is currently in progress.
- **Set Reminders** – Remind pending items and payment actions to avoid missing deadlines. We have verified that there is no way to have Alexa "wake up" at a specific time or even interval to implement some sort of reminder system; therefore, what we plan to do is allow through a new skill the ability to set a reminder, as well as pull payment due dates from mail items. Then we will run a separate service (new – to be created) on Azure that "wakes up" at regular intervals and checks to see if any reminders are due. If they are we will attempt to send a text with the reminder. Also, we will enhance the existing "Check my mail" skill to check and see if there are any reminders that have not yet been notified and notify them verbally back to the users.
- **Instruct to Print / email** – Ability to take instructions and trigger action to print or email the physical mail. This will be a new set of Alexa skills and/or enhancements to the existing skills. Once the firewall issue is resolved, we can actually have it send an email – until then it will just say it's going to. As far as the physical email obviously for this demo it's only going to be saying "Your mail will be sent to the address on file" or something similar.