Creative Integrated Design, Fall 2016

Project Kalexa

with Amazon Web Service

D조

최영진 김준혁 이상훈

윤석찬(AWS, Tech Evangelist)

Table of Contents

[1. Abstract 3](#_Toc469527264)

[2. Introduction 3](#_Toc469527265)

[A. 프로젝트의 배경 및 중요성 3](#_Toc469527266)

[B. 프로젝트 개요 4](#_Toc469527267)

[3. Background Study 4](#_Toc469527268)

[A. 관련 접근 방법 / 기술 장단점 분석 4](#_Toc469527269)

[B. 프로젝트 개발환경 7](#_Toc469527270)

[4. Goal/Problem & Requirement 8](#_Toc469527271)

[A. 제공 언어의 한계 극복 8](#_Toc469527272)

[B. 기본 기능 구현 8](#_Toc469527273)

[C. 추천 알고리즘 구현 8](#_Toc469527274)

[D. 정량적 달성 목표 9](#_Toc469527275)

[5. Project Architecture 10](#_Toc469527276)

[A. Architecture Diagram 10](#_Toc469527277)

[B. Architecture Description 10](#_Toc469527278)

[6. Implementation Spec 11](#_Toc469527279)

[A. Input/Output Interface 11](#_Toc469527280)

[B. Inter Module Communication Interface 11](#_Toc469527281)

[C. Modules 12](#_Toc469527282)

[7. Solution 14](#_Toc469527283)

[A. Implementation Details 14](#_Toc469527284)

[B. Implementation Issues 14](#_Toc469527285)

[8. Results 14](#_Toc469527286)

[A. Experiments 14](#_Toc469527287)

[B. Result Analysis and Discussion 14](#_Toc469527288)

[C. Limit 14](#_Toc469527289)

[D. Future Plan 14](#_Toc469527290)

[9. Division & Assignment of Work 14](#_Toc469527291)

[10. Demo Plan 14](#_Toc469527292)

[11. Conclusion 14](#_Toc469527293)

[[Appendix] User Manual 14](#_Toc469527294)

[[Appendix] Detailed Implementation Spec 14](#_Toc469527295)

# Abstract

본 프로젝트에서는 Amazon Alexa에서 동작하는 음성 인식 기반의 응용 프로그램을 제작하였다. Alexa는 아마존에서 개발한 음성 인식 시스템으로, 간단한 음성 비서 기능에 더하여 사용자가 다양한 응용 프로그램을 추가할 수 있도록 되어 있다.

그러나 Alexa는 한국어를 지원하지 않아 한국어로 된 질문을 하거나 응답을 받을 수가 없다. 본 프로젝트는 이 점에 착안하여 Alexa에서 한국어 관련 서비스를 제공하는 것을 목표로 하였다. 한국어 번역 기능, 한국어 학습 기능, K-POP 컨텐츠 제공 등을 주요 요소로 정하여, 어플리케이션을 디자인하고 구현하였다.

해당 프로그램을 제작하기 위하여 AWS(아마존 웹 서비스)내의 다양한 서비스를 활용하였다. 주요 로직을 실행하는 Lambda, 파일 저장소의 역할을 하는 S3, 데이터 저장소의 역할을 하는 dynamoDB 등을 융합하여 서버를 구현하였다. 그리고 네이버의 기계번역 API, 음성합성 API를 사용하여 외국인들이 한국어 음성을 들을 수 있도록 하였다. 또한 멜론의 차트, 앨범, 아티스트 등의 API를 활용하여 K-POP 관련 컨텐츠를 제공하였다. 이 과정에서 여러 프레임워크들이 쓰였으며, 정확도 및 속도를 개선하기 위해 다양한 기법을 적용하였다.

# Introduction

## 프로젝트의 배경 및 중요성

최근 K-POP 스타들의 많은 해외 진출과 소셜 미디어의 발달로 인해 해외에서의 K-POP에 대한 열기가 뜨겁다. 해외 팬들이 다양한 한국어 컨텐츠를 접하면서 한국어 학습에 대한 수요 또한 증가하고 있는 추세이다. 이러한 측면에서 외국인들에게 K-POP에 대한 정보를 제공해 줄 뿐만 아니라 한국어를 간단하게 학습할 수 있고 올바른 발음을 들려주는 기능을 제공하는 것은 충분히 필요가치가 있다.

따라서 본 프로젝트의 목표는 크게 한국어 학습 제공, K-POP 정보 제공의 두 가지로 나뉜다. 현재 Amazon Alexa가 한국어 인식/발음을 모두 지원하지 않고 있는 점이 한국어 기반의 정보를 제공하는 프로젝트에 가장 큰 한계와 문제점이다. 이 제약 조건 속에서 어떤 방식으로 문제를 해결할지 설계하고, 완성도 있는 결과물을 구현하는 것이 본 프로젝트의 일차적 목표이다. 더 나아가서는 K-POP 음악 추천과 같은 심도 있는 기능을 제공하고, 정량적으로 측정 가능한 지표들을 개선하는 작업을 하고자 한다.

## 프로젝트 개요

본 프로젝트에서 제공하려는 기능들을 정리하면 다음과 같다.

* 한국어 학습
  + 한국어 번역 및 원어민 발음 제공
  + 한국어 퀴즈
* K-POP 정보
  + K-POP 차트 정보 제공
  + 가사, 아티스트 및 그에 관련된 정보 제공
  + 음악 재생
* 음악 추천
  + 감정 기반 추천
  + 사용자 데이터 기반 추천

# Background Study

## 관련 접근 방법 / 기술 장단점 분석

### Amazon Alexa



Figure 1 Amazon Echo

아마존 알렉사(Alexa)는 2015년 출시된 일종의 음성 비서이다. 에코(Echo)라는 원통형 스피커 형태로 출시되었는데, 이 안에는 음성 인식 프로그램인 Alexa가 들어 있다. Alexa는 시각, 날씨 알림 등 기본 기능은 물론이고 주요 뉴스나 특정 정보를 브리핑하는 기능을 제공한다. 아마존은 장차 Alexa가 집 및 사무실을 자동화하는 데 핵심적인 역할을 할 것이라고 기대하고 있다.

Alexa는 현재 한국어를 지원하지 않는다. Apple 사의 Siri 등 다른 음성 인식 인공지능 역시 한국어를 지원하는 경우를 찾기 어렵다. 본 프로젝트에서는 Alexa를 타 서비스들과 연계하여 한국어 관련 기능을 제공하는 것을 목적으로 한다. 한국어를 인식하고 한국어를 구사하는 것은 가능하지도 않거니와 본 프로젝트의 범위에서 벗어난다. 그러나 영어 전용 프로그램인 Alexa에 한국어 번역 질의 응답이나 K-POP 등 인기 있는 한국 컨텐츠를 제공하는 기능을 추가하여 Alexa의 가용 범위를 늘리는 데 의의가 있다.

### Amazon Web Service

Amazon은 세계적인 전자 상거래 회사이나, 지난 2006년 Amazon Web Service(AWS)라는 이름으로 클라우드 컴퓨팅 서비스를 시작하였다. AWS는 Amazon.com에서 축적된 기술과 저렴한 가격을 바탕으로 급속도로 성장하였고, 전 세계적으로 가장 널리 이용되는 클라우드 서비스가 되었다. 현재는 Amazon.com이 AWS의 가장 큰 고객이라는 주객이 전도된 듯한 말까지 나오는 상황이다.

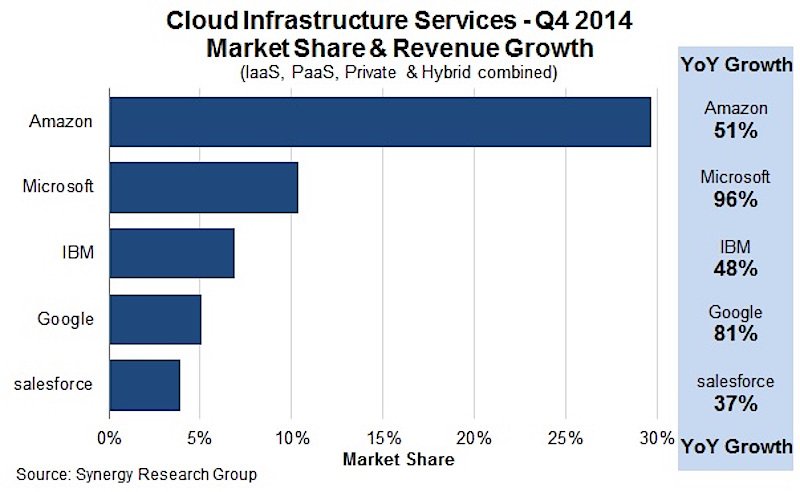


Figure 2 Cloud Service 시장 점유율

본 프로젝트는 AWS 코리아와 진행하는 것이기도 하고, Alexa가 아마존의 서비스이기 때문에 AWS와의 연동이 용이하다는 장점 때문에 AWS의 솔루션들을 사용하기로 결정하였다.

### AWS Lambda

아마존 웹 서비스의 람다(Lambda)는 컴퓨터과학에서 익명 함수(Anonymous Function)을 지칭하는 용어에서 유래하였다. 프로그래밍 언어에서 익명 함수를 복잡한 선언 과정 없이 사용하듯이, 프로그래머가 개발 환경에 신경쓰지 않고 로직에만 신경쓸 수 있도록 한다. 예를 들어, 간단한 HTTP 동작을 처리하기 위해서도 일반적으로 서버를 구축하고 기본적인 웹 개발 환경을 갖추어야 한다. 그러나 아마존 람다는 해당 기능만 구현하면 서버를 관리할 필요 없이 AWS의 환경에서 실행이 가능하다. 기능에 맞는 환경을 편리하게 선택 가능하며, 호출하는 시점만 서버 자원을 사용하므로 가격 면에서도 합리적이다.

본 프로젝트는 기능 자체는 많으나 외부 인터페이스와의 연결이 큰 부분을 차지하고 있어 코드 자체의 규모는 그리 크지 않을 것으로 예상된다. 또한 서버 머신이나 운영 체제 등에 종속될 경우도 없다고 판단되기에, 간편하게 로직을 구현하고 실행할 수 있는 람다가 적합하다고 판단하였다. EC2 등 서버를 사용하면 서버 환경 설정 및 관리에 불필요한 시간이 들어가고, 비용도 더욱 발생하는 데 반해 유리한 점이 많다고 할 수 있다.

### AWS S3

AWS S3은 Simple Storage Service의 약자로, 빠르고 안정적인 저장 공간을 제공하는 솔루션이다. 또한 서비스가 성장함에 따라 필연적으로 저장되는 데이터도 늘어나는데, 이를 자동으로 처리해 줌으로서 Scalablity에 영향을 준다. 본 서비스에서는 한국어 재생을 위한 음성 MP3 파일의 저장에 필수적이다.

### AWS DynamoDB

DynamoDB는 NoSQL 기반 데이터베이스이다. 본 프로젝트의 경우 복잡한 테이블 구조가 필요하지 않아, 빠르게 구현할 수 있는 DynamoDB를 선택하였다. 유저 정보, 곡 정보 등이 저장되게 된다.

### 네이버 기계 번역 / 음성 합성 API

국내 유명 포털인 네이버는 개발자 저변을 확대하면서 여러 공개 API들을 제공하고 있다. 본 프로젝트에서는 그 중 기계 번역 및 음성 합성 API를 사용할 것이다. 기계 번역은 사람의 손을 거치지 않아 아직까지는 번역의 질이 떨어지나, Alexa와 같은 실시간 서비스에서 즉시 응답을 줄 수 있다는 장점이 있다. 또한 네이버의 기계 번역은 한국어 품질이 수준급이라고 알려져 있다. Alexa에서 한국어 번역 질의가 들어왔을 때, 이를 한국어 응답으로 되돌려 주기 위해 사용한다.

음성 합성 API란 TTS(Text to Speech) 서비스라고도 불리는데, 텍스트를 음성으로 변환한다. Alexa는 영어 응답을 지정할 수 있으나 한국어는 발음 자체가 불가능하다. 음성 합성 사용은 이를 해결하기 위한 아이디어로, 음성 합성 API에서 반환된 MP3 파일은 언어와 관계 없이 재생이 가능하기에 Alexa가 한국어를 말하도록 할 수 있다.

### 멜론 실시간 차트 / 곡 검색 API

Melon은 국내 대형 온라인 음악서비스이다. SK 플래닛 개발자 센터에서는 Melon에 질의할 수 있는 API를 제공하여, 이를 통해 Melon의 여러 정보를 요청할 수 있다. 이 중에서 실시간 차트 정보 및 곡 검색 API를 사용할 것이다. 실시간 차트는 실시간으로 반영되는 가요 순위이다. 이를 이용하면 어디에서나 한국 가요의 실시간 순위를 그대로 받아 볼 수 있다. 차트의 순위 범위는 파라미터 값을 통해 조정할 수 있다. 곡 검색 API는 파라미터에 search keyword를 추가해서 호출하면 그 keyword를 가지고 검색된 곡의 정보를 반환한다. 본 프로젝트에서는 특정 가수의 차트 순위 곡 이외에 어떤 곡이 있는지에 대한 정보를 제공하기 위해 사용하였다.

## 프로젝트 개발환경

여러 명의 조원으로 구성된 프로젝트이니만큼 협업 도구가 필수적이다. 소스 버전 관리 도구로 git을 사용하였는데, 본 프로젝트의 경우 소스 파일 외에도 관리해야 할 텍스트 정보 파일들이 여럿 존재하여 이 역시 git을 통해 기록이 남도록 하였다. 원활한 커뮤니케이션을 위해서는 현재 전 세계적으로 널리 쓰이는 업무용 메신저인 slack을 사용했다. 또한 git과 slack의 연동을 통해 실시간으로 상대의 작업내역에 대한 알림을 받을 수 있도록 하였다.



Figure 3 개발 Platform & Tools

서버로는 AWS의 Lambda를 사용하였고, 개발 언어는 NodeJS(버전 4.3)이다.  DB로는 AWS에서 제공하는 noSQL DB인 DynamoDB를 사용한다. 데이터 저장소로는 S3를, 추후 데이터 캐싱에는 ElastiCache 를 사용할 예정이다. Alexa는 개발 플랫폼인 Alexa Skill Set을 제공한다. 이를 통해 본 프로젝트에서 Alexa가 인식해야 하는 문자열 및 변수들을 설정할 수 있다. Alexa의 테스트는 문자열 입력 방식 및 웹 상에서의 음성 인식 방식이 모두 가능하다. 또한 네이버의 음성 합성 및 기계 번역 API를 사용한다. 이들의 개발은 주로 각자의 로컬 환경에서 이루어졌으나, AWS Console에서 테스트하는 경우도 많았다.

Amazon Web Service는 상용 서비스이다. Lambda의 경우 사용자가 서버에 접근할 수 없으나, 내부적으로는 서버 자원을 사용하므로 이를 누적하여 과금하고, DynamoDB나 S3의 경우도 저장 공간과 이용 시간에 따라 요금이 청구된다. 본 프로젝트에서는 한국 AWS의 개발용 계정을 받는 것을 검토하였으나, 상용화 전에는 AWS Free tier 및 학생 지원 프로그램으로 충분하다고 판단되어 이를 이용하였다. 네이버 API의 경우도 보통 10,000글자/일에 대하여 무료로 제공되고 있다.

# Goal/Problem & Requirement

## 제공 언어의 한계 극복

Alexa가 지원하는 언어는 영어와 독일어이다. 따라서 한국어 인식은 불가능하며, 한국어 발음을 위하여 여러 방안을 강구하였다. 우선 한국어 발음을 알파벳으로 표현하여 Alexa가 그대로 읽게 하는 방법이 있다. 그러나 이 방법은 Alexa 응답의 발음이 매우 부자연스러우며 유저 입장에서 올바른 음성을 들을 수 없다. 따라서 이 문제점의 해결을 위해 음성 합성을 사용하기로 하였다. 네이버 등에서 제공하는 음성 합성 API는 한국어 문자열을 전달하면 매끄러운 발음으로 변환한 MP3 File을 반환한다. 이 방법으로 부자연스러움을 최소화하고 올바른 한국어 발음을 출력할 수 있다.

## 기본 기능 구현

본 어플리케이션에서 제공할 기본적인 기능들을 구현한다. 첫째로, 한국어 학습 관련 기능이 있다. 유저는 영어로 번역 질의를 하여 그에 해당하는 한국어 문장을 들을 수 있어야 한다. 또한 간단한 한국어 퀴즈를 만들어 학습의 재미를 늘릴 수 있도록 한다. 둘째로, K-POP 정보 제공 기능이 있다. 유저는 K-POP 차트에 대한 질문을 할 수 있으며, 특정 곡의 가사나 아티스트 정보 등을 얻을 수 있다. K-POP 음악의 재생 역시 가능하다. 궁극적으로는 유저 정보에 따라 음악을 추천할 것이다.

자세한 스펙의 경우 6. Implementation Spec에서 다루기로 한다.

## 추천 알고리즘 구현

해외의 K-POP 팬은 본인이 좋아하는, 혹은 해당 언어권에서 유명한 가수나 곡 이외의 다른 한국 음악에 대한 접근성이 떨어질 것으로 사료된다. 그런 점을 고려할 때 본인의 선택 외에 시스템 상에서 자동으로 추천해주는 기능이 있다면 사용자 입장에서 매우 유용할 것이라고 생각한다. 보다 취향에 맞는 곡을 추천해주기 위해 유저 기반의 데이터를 수집, 활용하여 추천하는 Collaborative Filtering을 적용하고자 한다. 예를 들면 현재 노래를 들은 사용자가 직전에 들은 노래, 현재 노래를 들은 사용자들이 많이 들었던 다른 노래, 나와 가장 많이 들은 곡이 겹치는 사용자의 플레이리스트 중 랜덤 추천 등이 가능하다. 또한 현재의 기분에 따라 미리 정해진 음악을 재생하는 방식의 추천을 할 수 있다. 이러한 추천 알고리즘을 극대화하고자 한다.

## 정량적 달성 목표

### 정확성(Accuracy)

Prototype 구현 단계에서 Alexa 테스트 결과, 엉뚱한 답을 내놓거나 아예 답을 하지 않는 경우가 많았다. 초기에는 영어 음성 인식의 문제라고 판단하여 본 프로젝트에서 개선할 여지가 없다고 보았으나, 리서치 결과 정확한 응답에 미치는 요소들이 몇 가지 있었다. 이 부분들을 개선하여 거의 대부분의 경우에 정확한 응답을 얻을 수 있도록 목표를 세웠다.

주요 이슈들과 개선 결과는 7. Solution 및 8. Results에서 서술할 것이다.

### 응답 시간(Response time, Performance)

일반 유저 상대로 하는 실시간 어플리케이션인 만큼 응답 시간이 매우 중요한 요소이다. 본 프로젝트는 기능이 다양하고 그 중 일부는 여러 단계의 로직을 필요로 한다. 또한 여러 가지 솔루션들을 같이 사용하여 Performance에 영향을 미치는 요인들이 다양하다. 따라서 기능 구현과 동시에 늘 속도의 개선을 염두에 두어야 했다. Prototype 단계에서부터 최종 결과물까지 각 기능들의 속도를 최대한 증가시키려고 하였다.

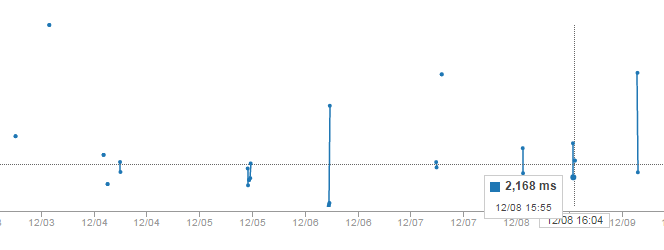


Figure 4 CloudWatch의 응답 시간 그래프

### 처리량(Throughput, Scalability)

본 프로젝트에서 또 하나의 목표를 세운 것은 확장 가능성이다. 개발, 테스트 및 데모 시연에서는 유저가 많지 않지만, 유저가 기하급수적으로 늘어났을 때의 대응을 하나의 도전 과제로 삼았다. 수많은 Alexa에서 동시에 요청이 들어오더라도 속도 저하 없이 처리할 수 있도록 노력하였다. 다만 처리량의 경우 대부분 사용 솔루션의 선택 및 설계에 의존하기 때문에, 특별히 측정하여 비교하지는 않았다. 이론적으로 시뮬레이션을 하기로 했다.

# Project Architecture

## Architecture Diagram

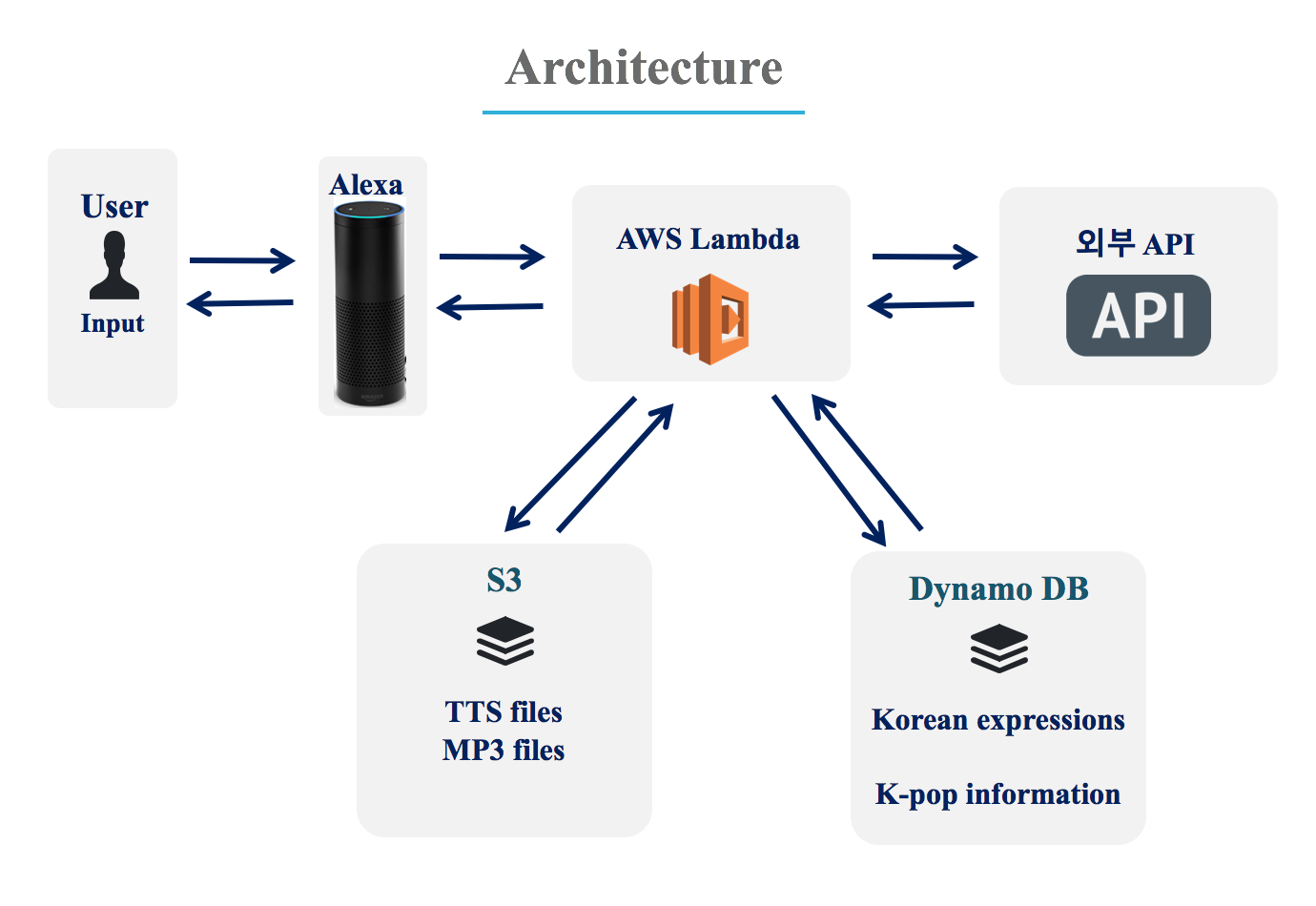


Figure 5 Architecture Diagram

## Architecture Description

위의 그림은 본 프로젝트의 기본 구조를 보여준다.

사용자가 Alexa에 말을 하면, Alexa는 미리 지정된 Utterance에 따라 이를 해석한다. 해석된 문장을 분류한 뒤 그에 맞는 Parameter와 함께 서버 역할을 하는 AWS Lambda에 전달한다. Lambda 함수는 모든 로직을 처리하는 역할을 한다. 기능에 따라 세부 흐름은 상이하지만, 기본적으로 HTTP 통신, DB 접근, MP3 및 Speaking 문자열 전달 등의 역할을 한다. Lambda가 다른 Lambda 혹은 외부의 API를 호출할 수도 있다.

S3는 파일들을 저장하는 Storage이다. 본 프로젝트에서는 한국어 문장을 TTS를 이용하여 만든 MP3 파일, K-POP 음악 MP3 파일들을 저장한다. Alexa에 이를 URL 기반으로 전달하면 음성 파일을 재생할 수 있다. 다만 임의의 MP3 파일을 그대로 재생할 수는 없으며, Alexa에 맞는 포맷으로 변환해주어야 했다.

DynamoDB는 noSQL 형태의 데이터 저장소이다. 가장 최근에 요청된 차트 정보 혹은 곡 정보 등이 필요에 따라 저장된다. 또한 유저 상태를 저장하여 유용하게 쓰거나, 중복된 API 호출을 방지한다. 이 때 API 호출을 하지 않음으로써 속도의 개선효과도 기대할 수 있다.

외부 API 또한 빠질 수 없는 요소이다. 본 프로젝트에서 제공되는 컨텐츠인 한국어 학습 컨텐츠에서 네이버의 기계 번역 API 및 음성합성 API가 사용된다. 그리고 K-POP 관련 정보는 Melon API를 통해 가져온다.

# Implementation Spec

## Input/Output Interface

입력과 출력은 모두 마이크와 스피커가 내장되어 있는 아마존 에코에 의해 이루어진다. 음성으로 입력이 들어오면 에코 내의 음성 인식 모듈이 Text로 변환하여 미리 정의된 Utterance를 통해 Parsing한다. 이 정보를 인자로 하여 Lambda API를 호출한다. 출력은 두 가지 형식이 가능한데, 일반 평문을 읽도록 할 수 있고, SSML 태그를 이용한 음성 파일 재생이 가능하다. 실제 음성의 출력은 아마존 에코 내의 TTS 엔진에 의해 동작한다.

## Inter Module Communication Interface

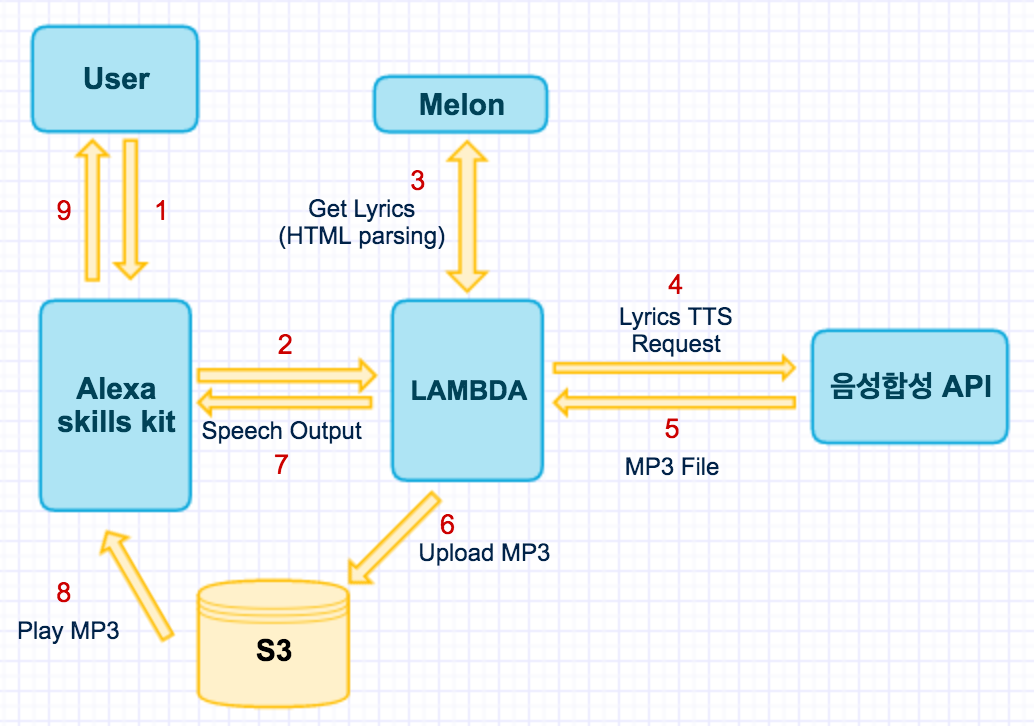


Figure 6 Module간 통신 Diagram

본 프로젝트를 구성하는 다양한 모듈은 기능에 따라 사용되기도 하고, 사용되지 않기도 한다. 또한 순서도 조금씩 다를 수 있는데, 대부분의 모듈을 사용하는 하나의 기능을 예로 들어 설명하고자 한다. 위의 그림은 가사를 재생하는 기능의 순서도를 나타낸다.

위의 그림에서 1번은 사용자가 Amazon Echo에 말을 하여 입력이 들어오는 부분이다. Alexa는 음성을 텍스트로 변환하여 텍스트를 람다 함수에 가사를 들려달라고 요청한다(순서 2). 람다는 서버의 역할을 하는 모듈로 대부분의 로직을 처리한다. 우선 람다는 해당 곡의 가사를 Melon으로부터 받아온다(순서 3). 이렇게 가져온 가사는 대부분이 한글로 이루어져 있을것이다. 하지만 alexa는 한국어를 지원하지 않기 때문에 한국어 텍스트를 음성으로 변환하는 작업이 필요하다. 이 역할은 NAVER의 음성합성 API가 담당한다. 람다는 음성합성 API에 요청을 보내고(순서 4) 응답으로 MP3 파일을 받는다(순서 5). 람다는 다시 응답으로 반환받은 MP3 파일을 S3에 업로드한다(순서 6). 람다는 S3에 파일이 올라간 것을 확인하고 alexa에게 그 파일을 읽어주라고 응답을 보낸다(순서 7). alexa는 건네 받은 MP3 파일 주소를 통해 그 파일을 로드하여(순서 8) 사용자에게 들려준다(순서 9).

## Modules

|  |  |
| --- | --- |
| Name | Description |
| RequestHandler | AlexaSkillsKit과의 통신에서 모든 입력과 출력을 관리, 간단한 로직 즉시 처리 |
| PlayLyrics | 특정 곡 ID를 인자로 받아 해당 곡의 가사를 재생 |
| EmotionBasedRecommendation | 감정 상태를 인자로 받아 해당 감정에 따라 적절한 곡을 추천 |
| Translate | 영문 텍스트를 인자로 받아 네이버 기계 번역 API 혹은 구글 Translate API를 호출하여 번역된 한국어 텍스트를 반환 |
| TTS | 한국어 텍스트를 인자로 받아 네이버 음성 합성 API를 호출, 반환된 MP3 파일을 Alexa Playable Format으로 변환, 변환된 파일을 S3에 업로드 후 URL 리턴 |
| TranslateWrapper | Translate 및 TTS 기능을 Wrapping  Translate, TTS는 여러 기능에서 복합적으로 쓰이기 때문에 분리하여 구현 |
| KPopChart | Melon API를 호출하여 1-10위 까지의 차트 정보를 가져와서 DB에 저장 |
| SpeakKPopChart | 차트 정보를 기반으로 유저에게 전달할 정보 문자열을 만들어 TTS Lambda를 호출, 해당 MP3의 URL을 반환 |
| GetOtherSongs | 가수 이름을 인자로 받아 Melon Search API 호출, 다른 곡 정보 반환 |
| ArtistBasedRecommendation | 가수 이름을 인자로 받아 해당 가수의 다른 곡 재생 |
| Quiz | 한국어 퀴즈 요청 및 각 퀴즈에 대하여 정답 체크 |
| \_QuizInserter | 퀴즈 리스트를 삽입하는 One Time 로직  (Invisible to User) |

Table 1 Module List

# Solution

## Implementation Details

## Implementation Issues

# Results

## Experiments

## Result Analysis and Discussion

## Limit

## Future Plan

# Division & Assignment of Work

# Demo Plan

# Conclusion

# [Appendix] User Manual

# [Appendix] Detailed Implementation Spec