

# Bixby 서비스 아키텍처

## 1. 배경지식

### 1-1. 음성 인식과 자연어의 이해

- ASR (Automatic Speech Recognition)
  - 발화를 문자로 변환하는 기술
    - "오늘 서울 날씨 알려줘"
    - 발화: 사람이 말하는 음성언어
- NLU (Natural Language Understanding)
  - 문장의 의미를 이해하는 기술 (의도를 파악하고 의도를 충족하기 위한 값을 찾아내는 기술)
    - [오늘][서울][날씨 알려줘]

## 2. 서비스 아키텍처 과정

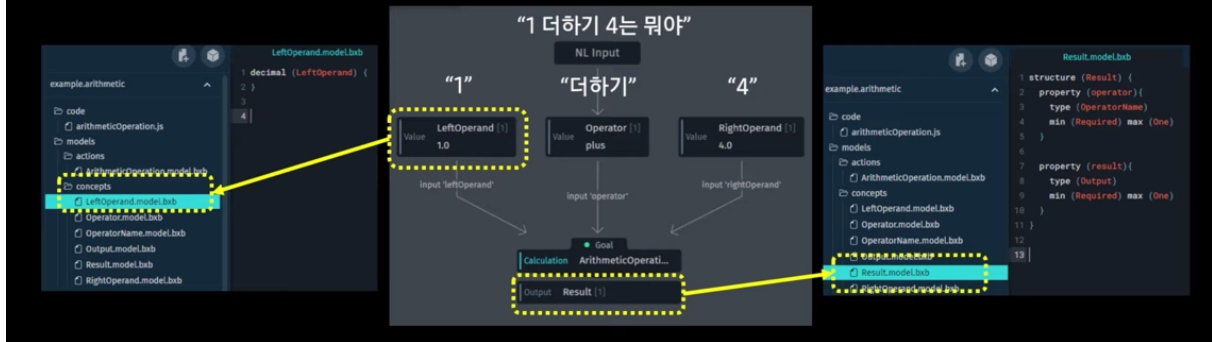


1. 사용자가 "오늘 날씨 알려줘"라고 발화한 뒤 빅스비 서버로 전송된다.
2. 빅스비 서버는 ASR 모듈을 이용해 해당 음성을 문자로 변환한다.
3. NLU 모듈을 통해 해당 문자를 어떤 캡슐로 보내야 하는지 분류한다. 해당 발화가 어떤 의도(Intent)이며, 동작을 위해 필요한 값(Value)이 어떤 것인지 분석한다.
4. 실시간으로 실행 절차를 나타내는 Plan Graph를 생성한다. Plan은 결과값을 얻기 위해 실행해야 하는 할 일을 순서도 모양으로 나타낸 것을 말한다. 할 일을 수행하기 위해 실제 개발자가 짠 JS 코드를 실행한다.
  - 빅스비 캡슐을 바탕으로 동작을 실행하기 위한 계획을 재구성한 것을 Plan이라고 한다.
5. 외부 서버 연동이 필요하면 필요한 API를 불러오거나 연산을 실행한다.

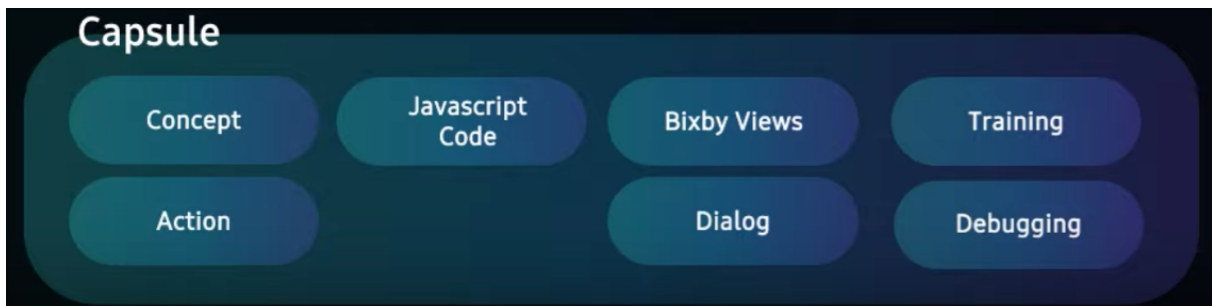
## Plan Graph 예시

# Plan Graph 예시

계산기를 예로 들어봅시다

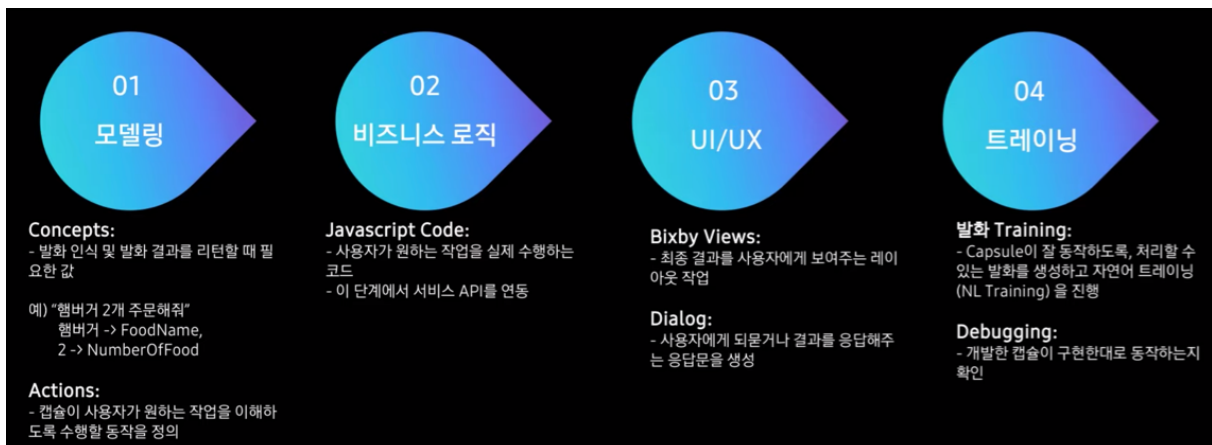


- 가운데 : 계산기 캡슐 Plan Graph
- 왼쪽, 오른쪽 : 계산기 캡슐 빅스비 실제 코드
- Plan Graph에 LeftOperand 인풋값 이름과 실제 값인 1.0이 나와있고, concepts 폴더 안에도 LeftOperand라는 파일명이 있다. 즉, Plan Graph의 LeftOperand라는 노드와 대응되는 컨셉이라는 의미다.
- 1 + 4의 결과값인 Result라는 아웃풋이 나와있다. 이 역시도 concepts 폴더 안에 대응되는 Result라는 파일명을 찾을 수 있다.
- 왼쪽, 오른쪽의 코드만 잘 짜면 Plan Graph가 자동으로 생성된다. 자연어 처리와 같은 머신러닝은 따로 신경쓰지 않아도 빅스비가 알아서 처리해준다.



빅스비 캡슐의 컴포넌트들이다. 개발자는 이 컴포넌트들을 잘 개발하고 테스트한 뒤 Submit 하면 된다.

## 3. Bixby Capsule 구조



## 1. 모델링

- Concepts : 발화를 인식하고 발화 결과를 리턴할 때 필요한 값
- Actions : 컨셉에 들어있는 값을 활용하는 동작

## 2. 비즈니스 로직

- JavaScript Code : 사용자가 원하는 작업을 실제 수행하는 코드, 서비스 API 연동

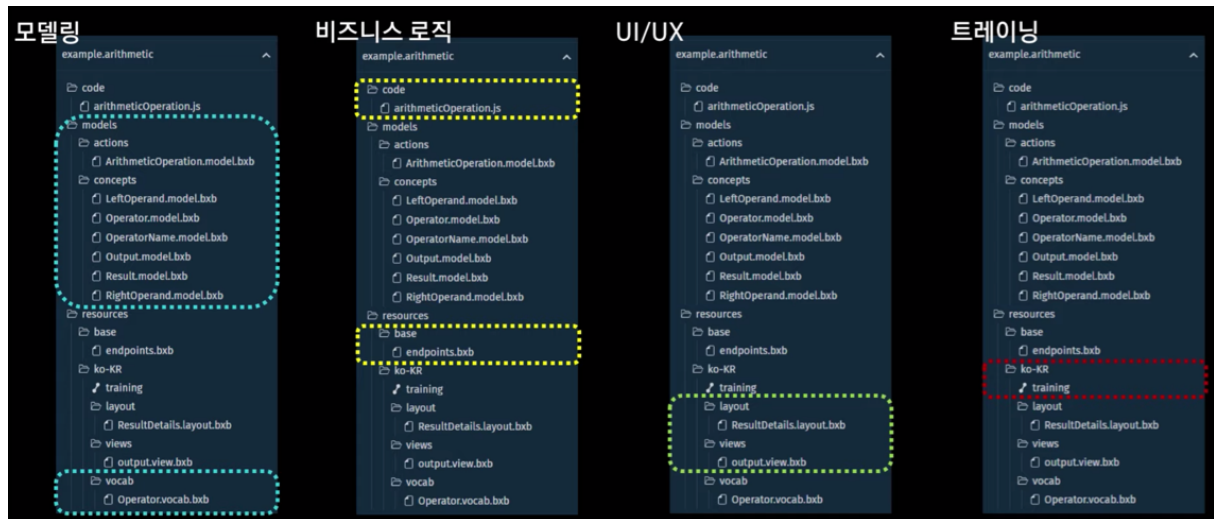
## 3. UI/UX

- Bixby Views
- Dialog

## 4. 트레이닝

- 발화 Training
- Debugging

### 계산기 코드와 맵핑



## 1. 모델링

- 계산이라는 동작을 나타내는 Actions
- 피연산자, 연산 종류 등 계산에 필요한 값을 나타내는 Concepts
- Concepts 값에 대한 예상 발화를 모아놓은 Vocab

## 2. 비즈니스 로직

- 실제 동작하는 코드를 모아둔 JavaScript 파일
- JavaScript 파일과 Concepts 파일을 맵핑시켜주는 endpoints 파일

## 3. UI/UX

- 계산 결과를 예쁘게 표현하는 layout 과 views 파일

## 4. 트레이닝

- 발화를 학습시키는 training 파일
- 언어별로 학습시킬 수 있게 언어별로 폴더가 존재한다.

