

Natural Language Processing

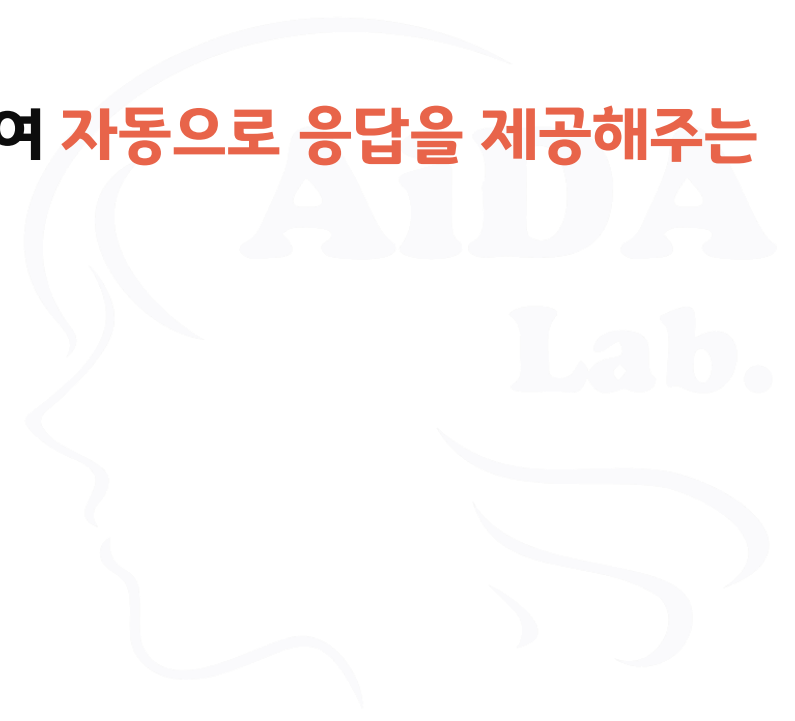
Chatbot 시스템

강사 양석환



Chatbot 시스템 개요

- Chatbot = **Chatter** + **Robot**
- 대화하는 **로봇**이라는 의미
- 사람과 텍스트나 음성으로 대화를 나눌 수 있는 프로그램
- 메신저 기반 환경에서 사용자들의 질문이나 요구사항에 대하여 **자동으로** **응답을** **제공해주는** **에이전트 서비스**



- ~수년 전

- 다양한 시나리오에 대응할 수 있는 규칙을 기반으로 개발
- 특정 키워드에 대한 정해진 답변을 제공하는 단순한 형태

- 최근

- 딥러닝 기술의 발달에 따라 자연어를 이해, 응답하는 수준으로 발전함



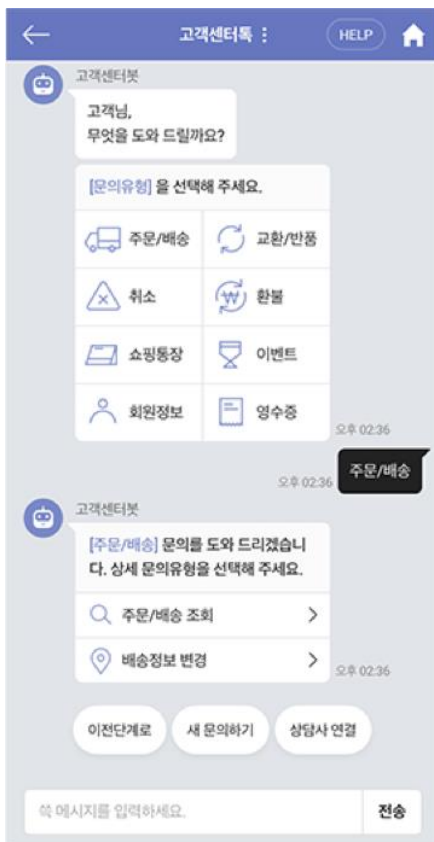
- **과거의 챗봇**

- 단순히 사용자와 잡담을 나누며 시간을 때우는 재미 요소에 불과

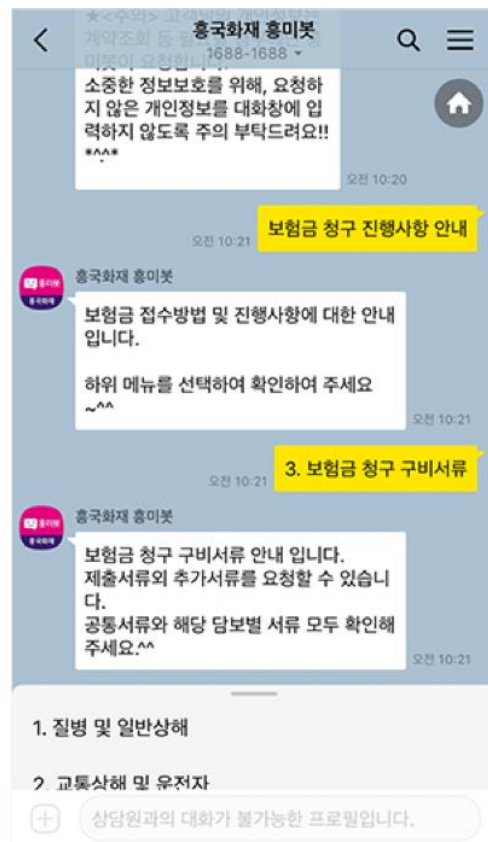
- **최근의 챗봇**

- 고객센터, 예약, 상담 시스템 등에서 활발히 도입 중
- 사람을 대신하여 자동화된 24시간 CS(Call Service) 업무 수행
- 상담 서비스에 대한 비용절감 효과로 인해 점점 사용 범위가 확장되고 있음

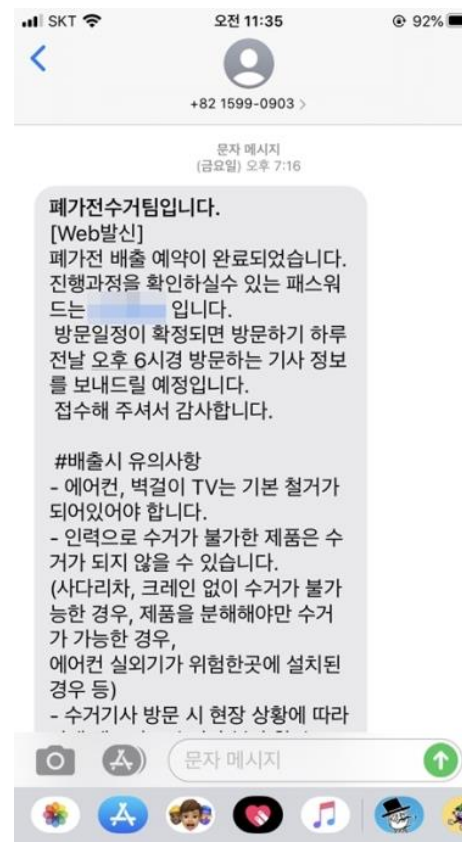




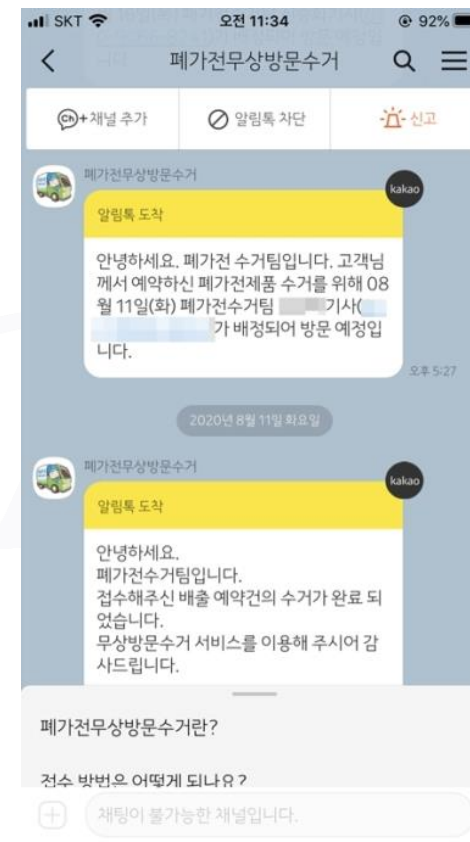
SSG닷컴 챗봇 서비스



흥국화재 챗봇 서비스



SKT 챗봇 서비스



Chatbot 시스템의 주요 기술

• 메신저를 기반으로 하는 챗봇은

- 도형, 문자, 음성 등의 패턴을 인식하여
- 인간이 쓰는 언어를 처리할 수 있어야 하며
- 논리적으로 내용, 키워드를 추론할 수 있어야 함
- 사용자의 요구 정보를 비정형 데이터에서 검색하고
- 현실의 상황을 정보화하고 활용하는
- 지능화된 서비스를 제공해야 함

도형, 문자, 음성 패턴인식 기술

자연어 처리 기술

시맨틱 웹(차세대 지능형 웹) 기술

텍스트 마이닝 기술

상황인식 컴퓨팅 기술

자료 참고: 인공지능 기반의 '챗봇(ChatBot)' 개념과 사례 소개 (http://www.snpo.kr/bbs/board.php?bo_table=npo_aca&wr_id=3201), 서울NPO지원센터

- **패턴인식(Pattern Recognition) 기술**

- 도형, 문자, 음성 등의 데이터에서 패턴을 기계적, 알고리즘 적으로 학습하여 인식, 식별 시키는 기술

- **자연어처리(Natural Language Processing) 기술**

- 인간이 사용하는 보통의 언어(자연어)를 컴퓨터가 인식할 수 있도록 하여 그 내용을 기반으로 데이터를 처리하는 기술
- 정보검색, 질의응답, 시스템 자동번역, 통역 등에 활용됨

자료 참고: 인공지능 기반의 '챗봇(ChatBot)' 개념과 사례 소개 (http://www.snpo.kr/bbs/board.php?bo_table=npo_aca&wr_id=3201) , 서울NPO지원센터

• 시맨틱 웹(Semantic Web) 기술

- 컴퓨터가 정보자원의 뜻을 이해하고, 논리적 추론까지 가능한 차세대 지능형 웹 기술
- 기존의 웹을 확장하여 컴퓨터가 이해할 수 있는 잘 정의된 의미를 기반으로 의미적인 상호운용성 (Semantic Interoperability)을 실현하여
- 다양한 정보자원의 처리 자동화, 데이터의 통합 및 재사용 등을 컴퓨터가 스스로 판단, 수행하며
- 인간과 컴퓨터 모두를 잘 이해할 수 있는 웹을 만드는 것을 목표로 함

자료 참고: 인공지능 기반의 '챗봇(ChatBot)' 개념과 사례 소개 (http://www.snpo.kr/bbs/board.php?bo_table=npo_aca&wr_id=3201) , 서울NPO지원센터 위키백과 (https://ko.wikipedia.org/wiki/시맨틱_웹)

- **텍스트 마이닝(Text Mining) 기술**
 - 비정형 텍스트 데이터에서 새롭고 유용한 정보를 찾아내는 과정 또는 기술
 - 데이터를 요약하여 키워드를 찾아내는 기술과 함께 활용되기도 함
- **상황인식컴퓨팅(Context-Aware Computing) 기술**
 - 챗봇의 경우, Context보다 좁은 범위의 Text-Aware Computing에 해당
 - 가상공간에서의 현실의 상황을 정보화하고
 - 이를 활용하여 사용자 중심의 지능화된 서비스를 제공하는 기술

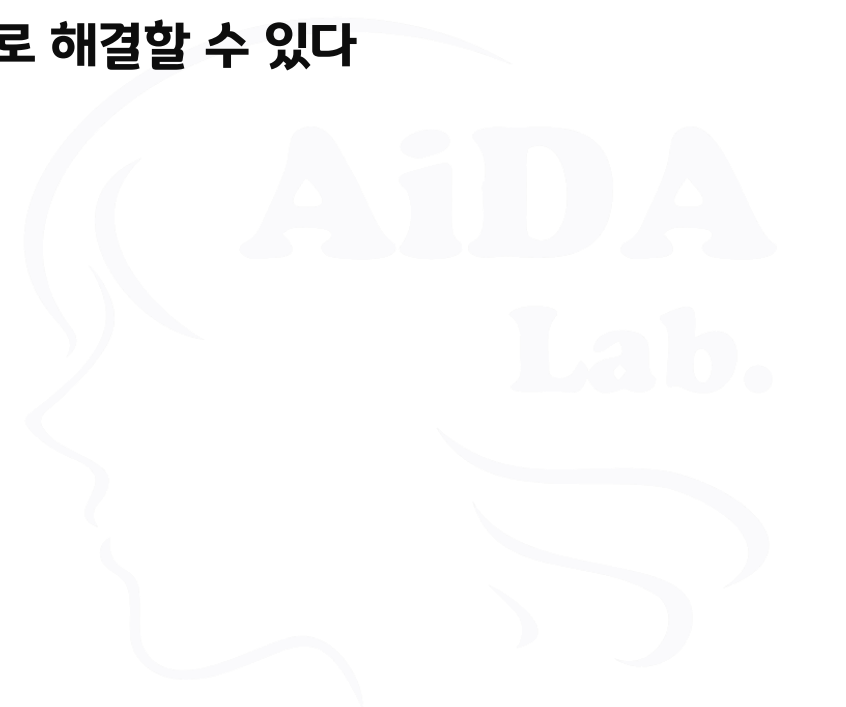
자료 참고: 인공지능 기반의 '챗봇(ChatBot)' 개념과 사례 소개 (http://www.snpo.kr/bbs/board.php?bo_table=npo_aca&wr_id=3201) , 서울NPO지원센터

- 기업이 챗봇을 선호하는 이유
 - 반복되는 고객문의를 자동화 할 수 있다.
 - 365일 24시간 고객의 문의에 빠르게 답변할 수 있다.
 - 한 번에 여러 고객을 동시응대 할 수 있다.
 - 고객상담에 드는 시간과 비용을 줄일 수 있다.
 - 전반적인 상담 경험이 좋아져 고객 만족도가 높아진다.



- **고객이 챗봇을 선호하는 이유**

- 궁금증이 생겼을 때 시간에 관계없이 빠른 답변을 받을 수 있다
- 전화나 이메일보다 부담 없이 기업에 문의를 할 수 있다
- 반품/배송 문의 등 굳이 상담원 연결이 필요 없는 고객문의를 자동으로 해결할 수 있다
- 이색적인 사이트 경험을 할 수 있다



Chatbot 시스템을 이용한 스마트 홈

• 왜 스마트 홈인가?

- 현재의 AI 교육은 영상인식, 처리 분야에 집중되어 있음
- 음성인식 관련 교육은 소수에 그침
- 음성인식 기술로 우리는 무엇을 할 수 있는가?
 - 전문지식을 요구하는 연구 분야는 일단 고려대상에서 제외한다면...
 - 생각할 수 있는 서비스는 AI 스피커를 크게 벗어나지 못함
 - 음성인식 기술로 생각할 수 있는 서비스는 말로 무엇인가를 수행하도록 지시하는 것
 - 자율 주행, 방법 및 보안, 영상인식/추적 기반 불량 검출 등 영상인식/처리 기술과는 적용 대상의 범위가 다름

• 음성인식 기술을 통해 지시할 수 있는 범위는?

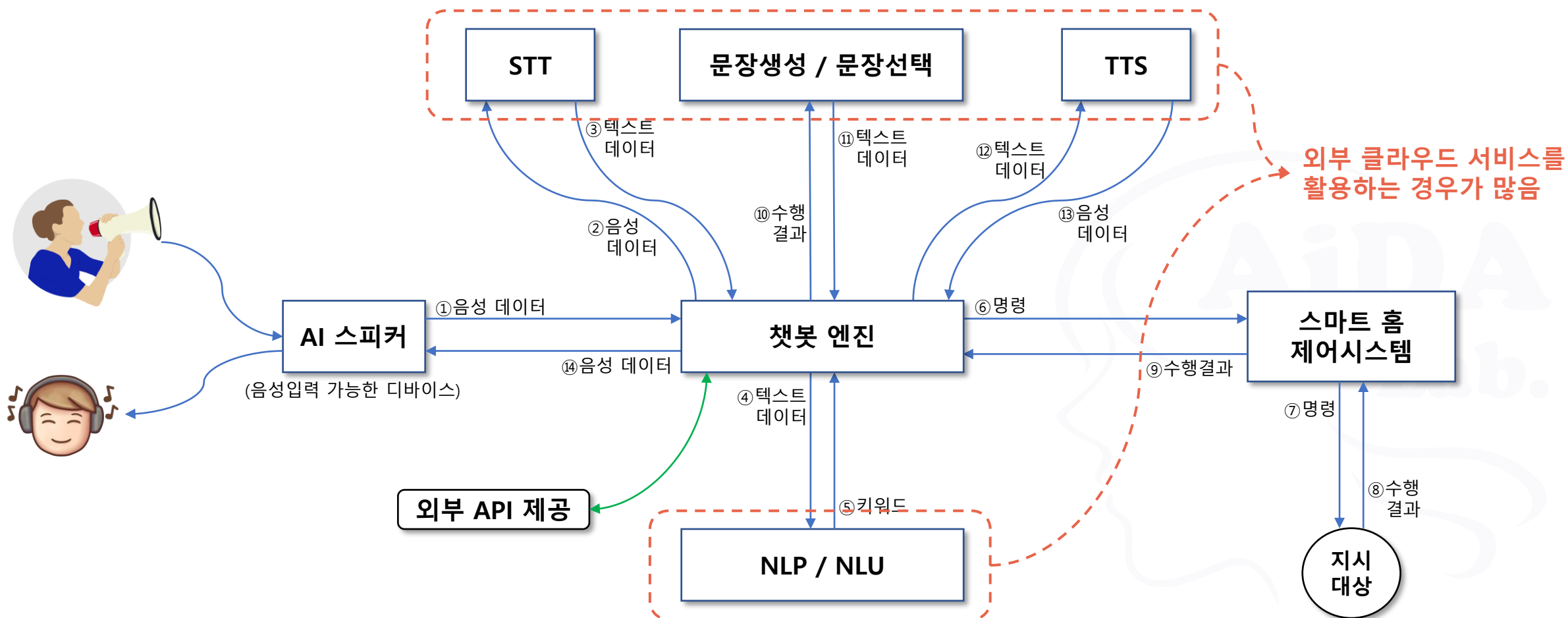
- 지금 몇 시야?
- 내일 날씨는 어때?
- ~~ 검색해줘
- 조용한 음악 골라서 틀어줘
- 방에 불 꺼줘
- 주방의 가스 밸브 잠가줘
- 자동차 시동 걸어줘

이런 것 말고 무엇이 있을까???

결국 우리가 할 수 있는 것은 대체로 이런 종류이다

챗봇 서비스, 특히 음성인식 챗봇 서비스로
가장 쉽게 접근할 수 있는 것은 스마트 홈 서비스이다

• 음성인식 챗봇 기반의 스마트 홈 구성 및 프로세스



- 왜 STT, TTS, NLP/NLU, 문장생성/문장선택 기능은 클라우드 서비스를 많이 사용하는가?
 - 개발에 투입되는 자원의 규모가 크다
 - 복합적인 기술이 적용되므로 개발이 어렵다
 - 독자적인 알고리즘을 만들 필요가 없다
 - 해당 기능은 어떤 서비스를 선택, 사용하더라도 목표가 동일하다
 - 구태여 직접 만들어서 사용하지 않아도 기능 상 문제는 없다
 - 직접 만드는 것보다 전문 업체에서 개발하는 것이 성능이 더 뛰어나다



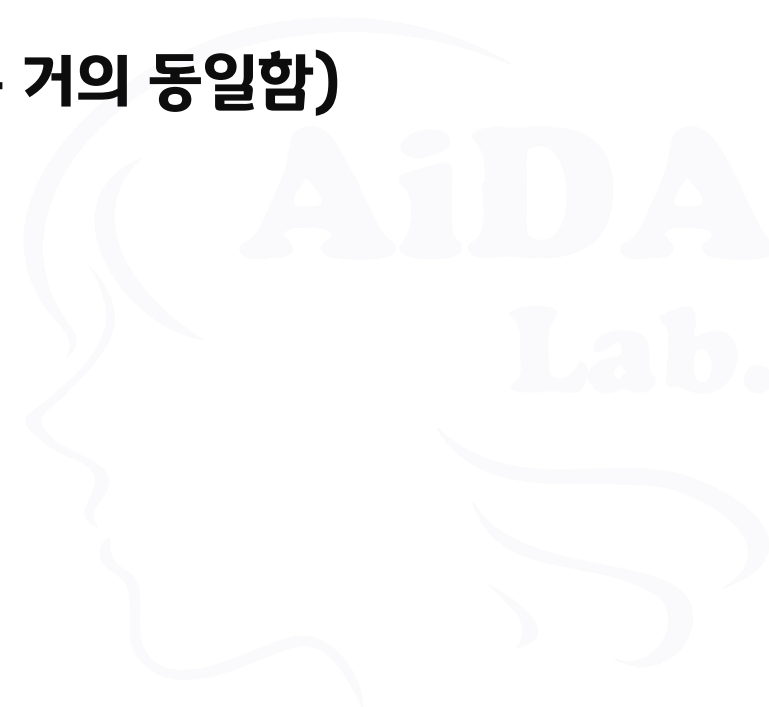
- **AI 클라우드 서비스의 종류**
 - Amazon AWS AI
 - IBM Cloud(구, Bluemix)의 Watson
 - Microsoft Azure
 - Google Cloud AI
 - Naver Cloud Platform, HyperCova
 - 등 다양한 서비스가 존재함

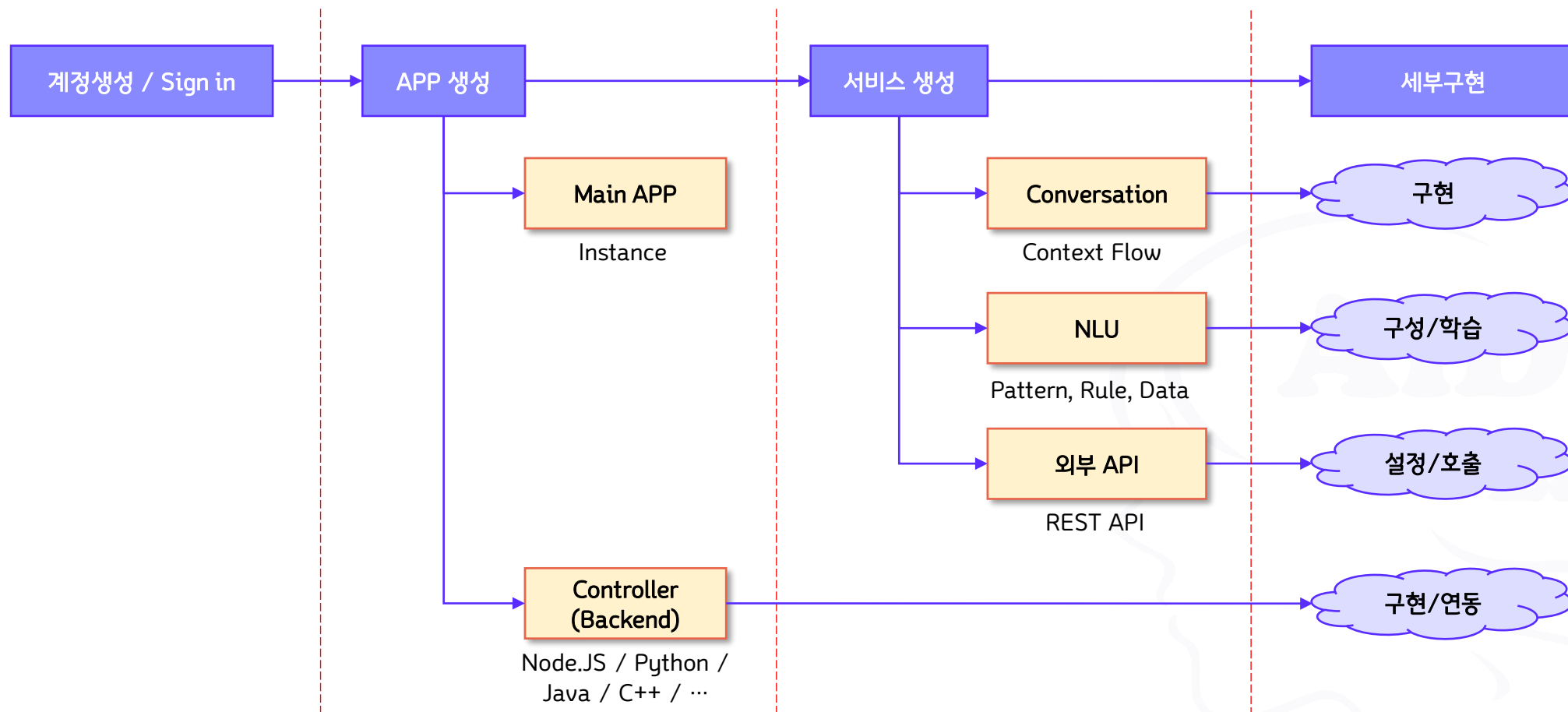


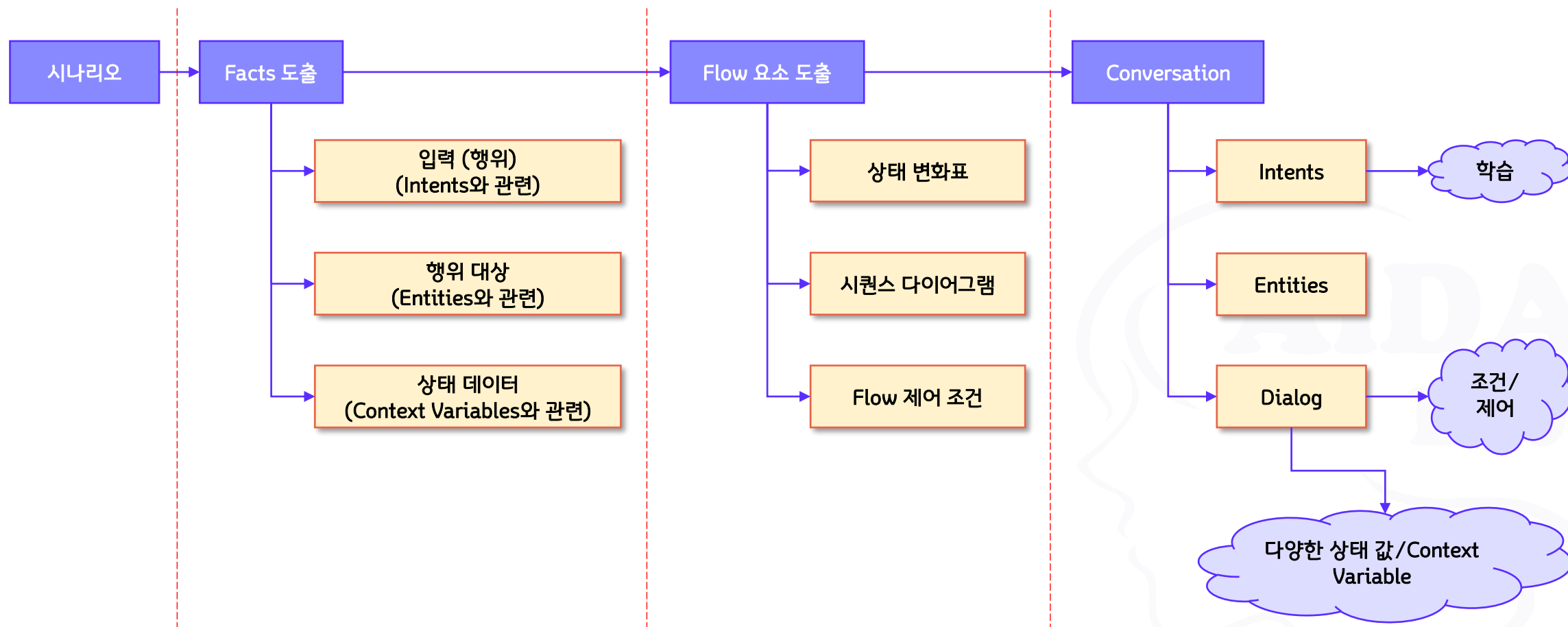
- 다양한 클라우드 서비스 중 음성인식 챗봇 개발 시 사용하는 기능은?
 - STT (Speech To Text)
 - TTS (Text To Speech)
 - NLP (Natural Language Processing)
 - NLU (Natural Language Understanding)
 - Conversation / Context Tools
 - Learning(Training) Tools



- 음성인식 챗봇 개발을 위한 클라우드 서비스의 사용방법은 어떤 서비스를 이용하더라도 거의 같은 개념으로 접근 가능함
- 직접 개발한다고 하더라도 기존 서비스의 진행 방식을 이해한다면 더욱 접근이 용이함
(용어 등은 각 클라우드 서비스에 따라 다를 수 있으나 개념은 거의 동일함)







- 하윤
 - “새로운 장치를 구입했어. 등록해줘”
- Watson
 - “장치를 검색하겠습니다. 장치에 등록 버튼을 누르세요.”
 - “스마트 전구 2개를 검색했습니다. 먼저 깜빡이는 새로운 전구를 등록합니다. 어느 방에 설치 됩니까?”
- 하윤
 - “침실에 설치할건데, 이름을 ‘무드등’ 으로 해줘”
- Watson
 - “ ‘무드등’으로 이름을 변경했으며, ‘침실’에 등록했습니다.”
 - “깜빡이는 새로운 전구를 등록합니다. 어느 방에 설치 됩니까?”
- 하윤
 - “거실에 설치할거야”
- Watson
 - “이 장치의 새로운 이름을 알려 주세요.”
- 하윤
 - “ ‘독서등’ 이라고 해줘.”
- Watson
 - “ ‘독서등’으로 이름을 변경했으며, ‘거실’ 에 등록했습니다.”

입력

- 새로운 장치를 구입했어
- 등록해줘
- 침실에 설치할건데
- 이름을 ‘무드등’ 으로 해줘
- 침실에 설치할건데 이름을 ‘무드등’ 으로 해줘
- 거실에 설치할거야
- ‘독서등’ 이라고 해줘

상태데이터

- 장치 수량
- 장치 명
- 장소 명
- 장치 상태 (On/Off 등)

행위대상

- 장치
- 등록 버튼
- 스마트 전구
- 새로운 전구
- 어느 방
- 침실
- 무드등
- 거실
- 독서등

- 하윤
 - “새로운 장치를 구입했어. 등록해줘”
- Watson
 - “장치를 검색하겠습니다. 장치에 등록 버튼을 누르세요.”
 - “스마트 전구 2개를 검색했습니다. 먼저 깜빡이는 새로운 전구를 등록합니다. 어느 방에 설치 됩니까?”
- 하윤
 - “침실에 설치할건데, 이름을 ‘무드등’ 으로 해줘”
- Watson
 - “ ‘무드등’으로 이름을 변경했으며, ‘침실’에 등록했습니다.”
 - “깜빡이는 새로운 전구를 등록합니다. 어느 방에 설치 됩니까?”
- 하윤
 - “거실에 설치할거야”
- Watson
 - “이 장치의 새로운 이름을 알려 주세요.”
- 하윤
 - “ ‘독서등’ 이라고 해줘.”
- Watson
 - “ ‘독서등’으로 이름을 변경했으며, ‘거실’ 에 등록했습니다.”

입력

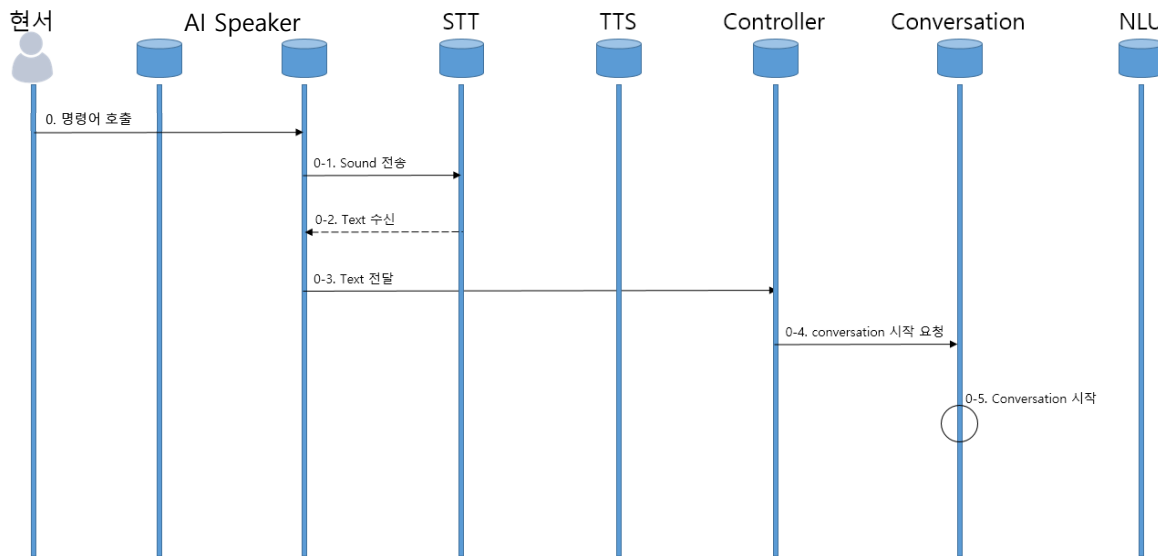
- 새로운 장치를 구입했어. 등록해줘
- 침실에 설치할건데
- 이름을 ‘무드등’ 으로 해줘
- 침실에 설치할건데, 이름을 ‘무드등’ 으로 해줘
- 거실에 설치할거야
- ‘독서등’ 이라고 해줘

행위

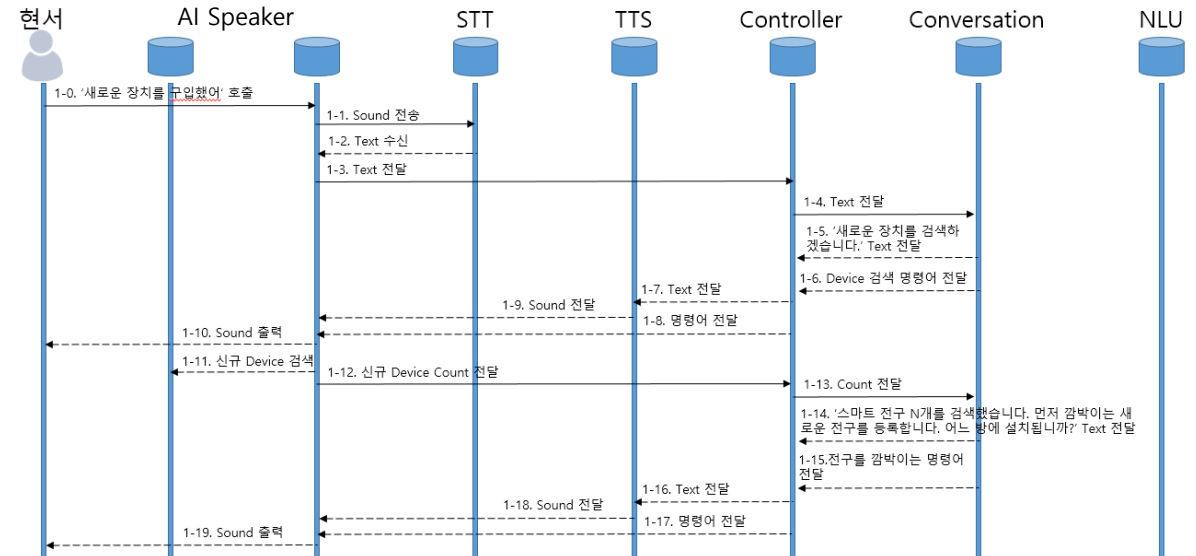
- 장치(스마트 전구) 검색
- 장치(스마트 전구) 깜빡이기
- 장치(스마트 전구) 설치(=등록)
- 장치(스마트 전구) 설치 위치 파악
- 장치(스마트 전구) 설치 이름 파악

시나리오, Facts & Component Diagram → Sequence Diagram

Activity Diagram – Scenario #1 (호출)

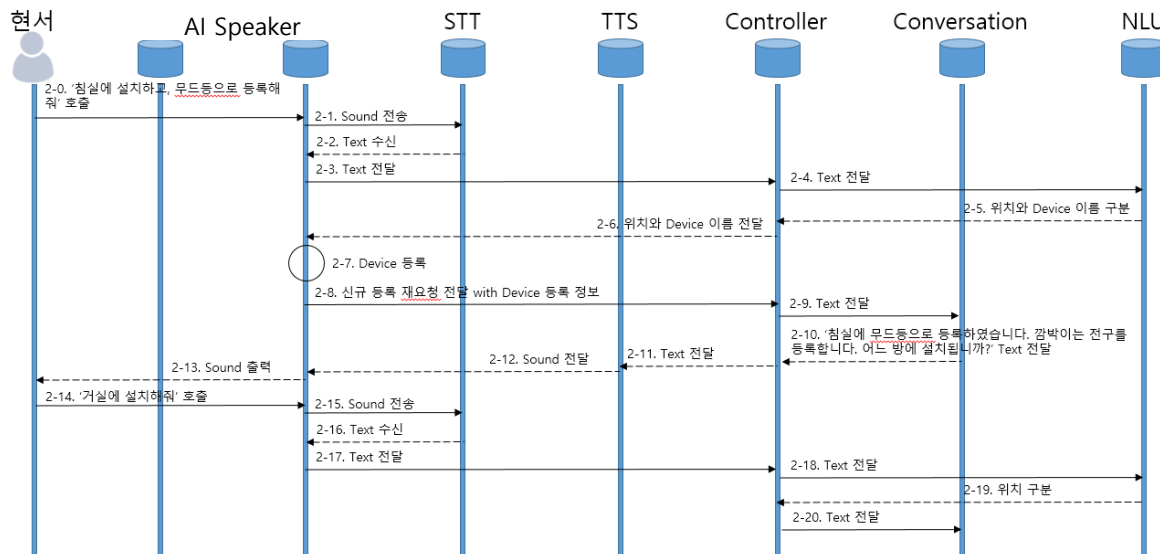


Activity Diagram – Scenario #1 (장치검색)

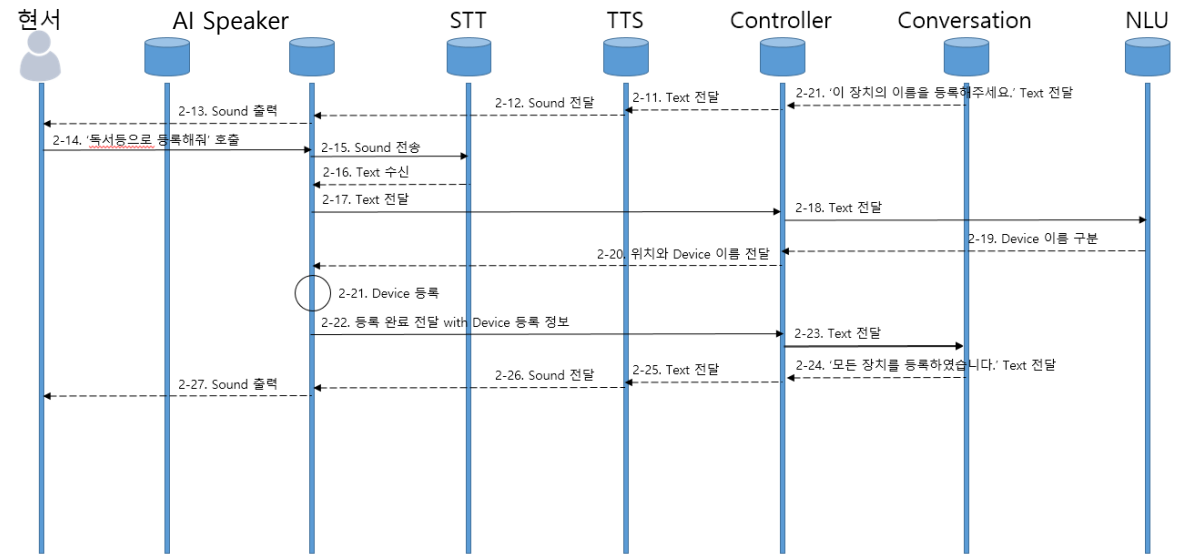


시나리오, Facts & Component Diagram → Sequence Diagram

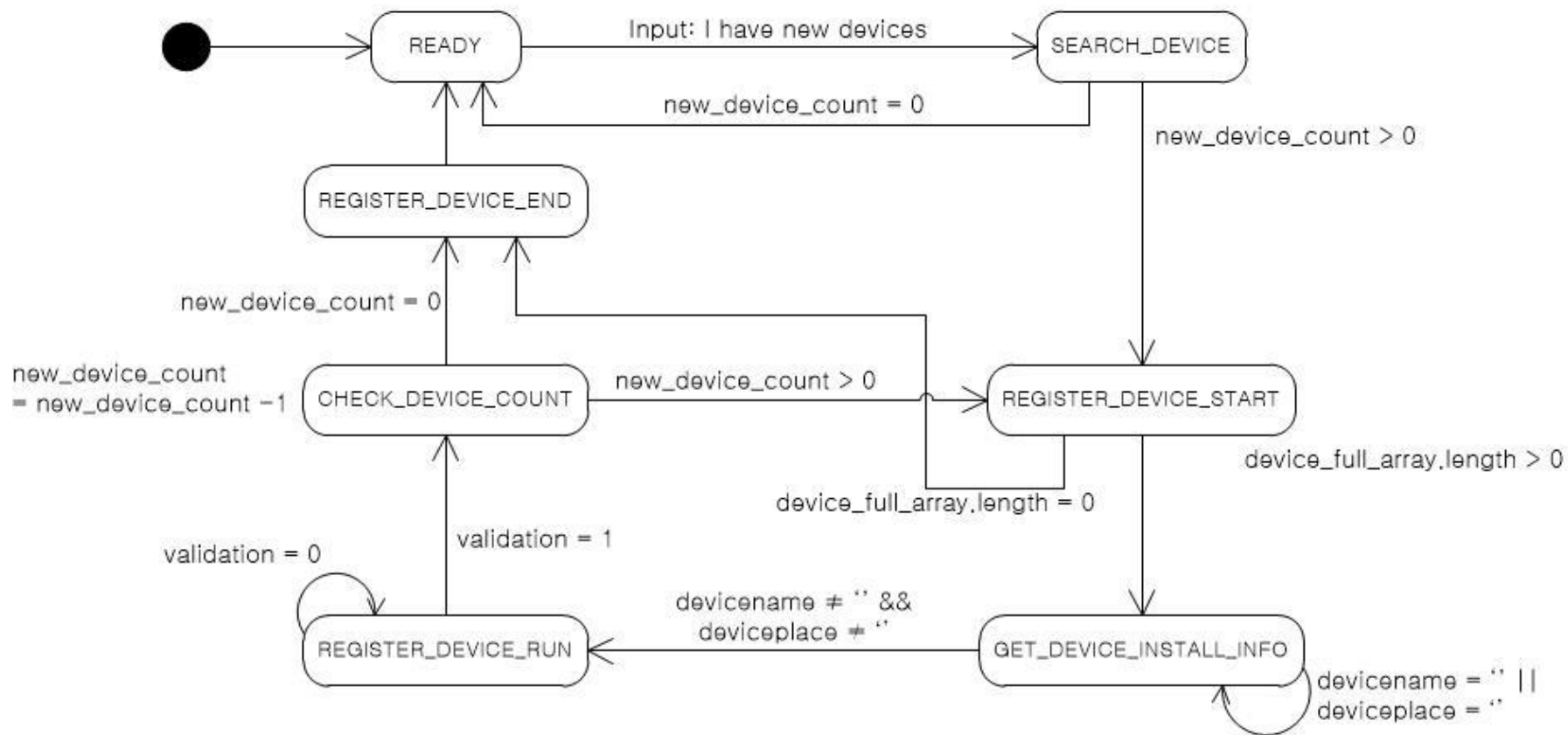
Activity Diagram – Scenario #1 (장치등록)

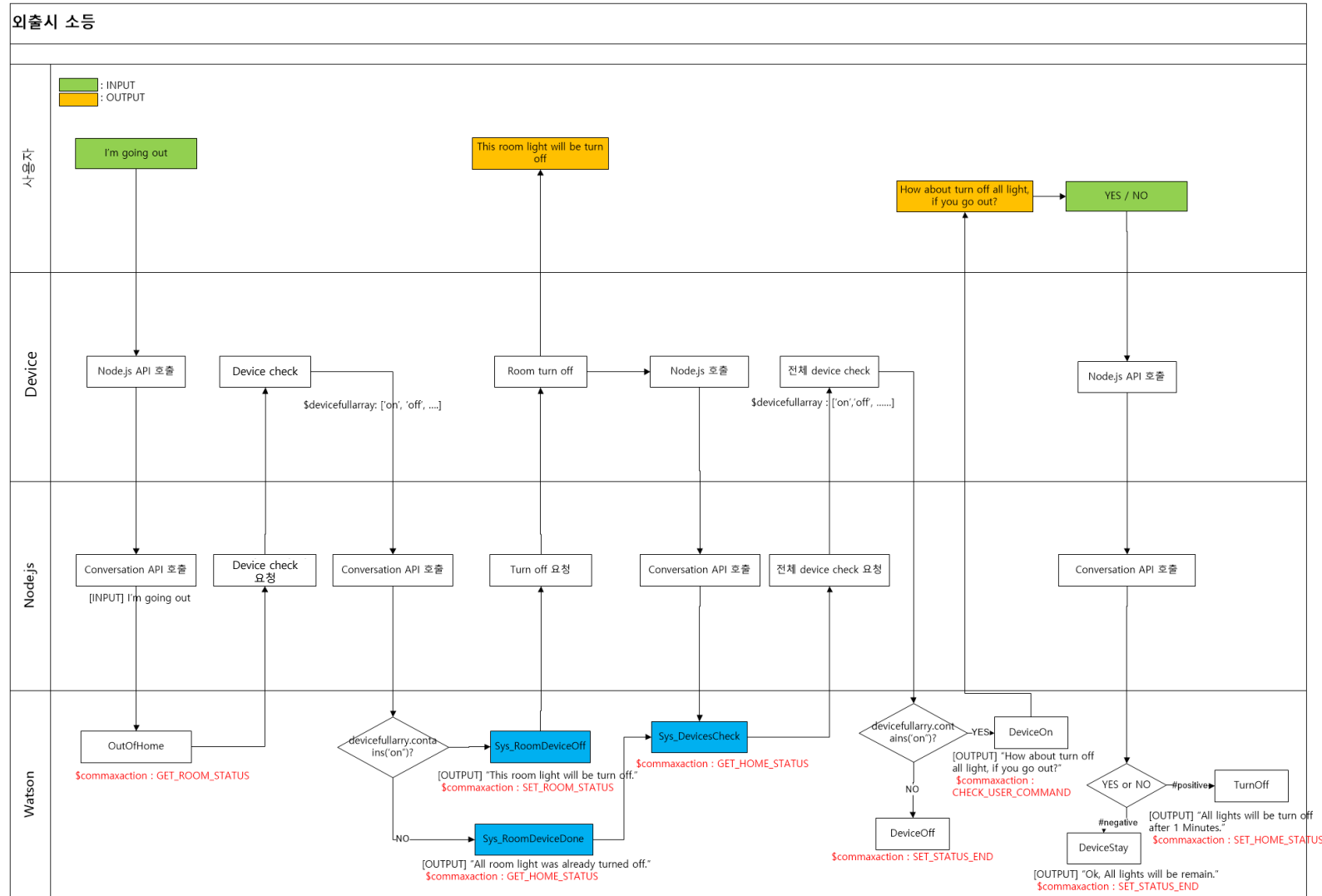


Activity Diagram – Scenario #1 (장치등록)

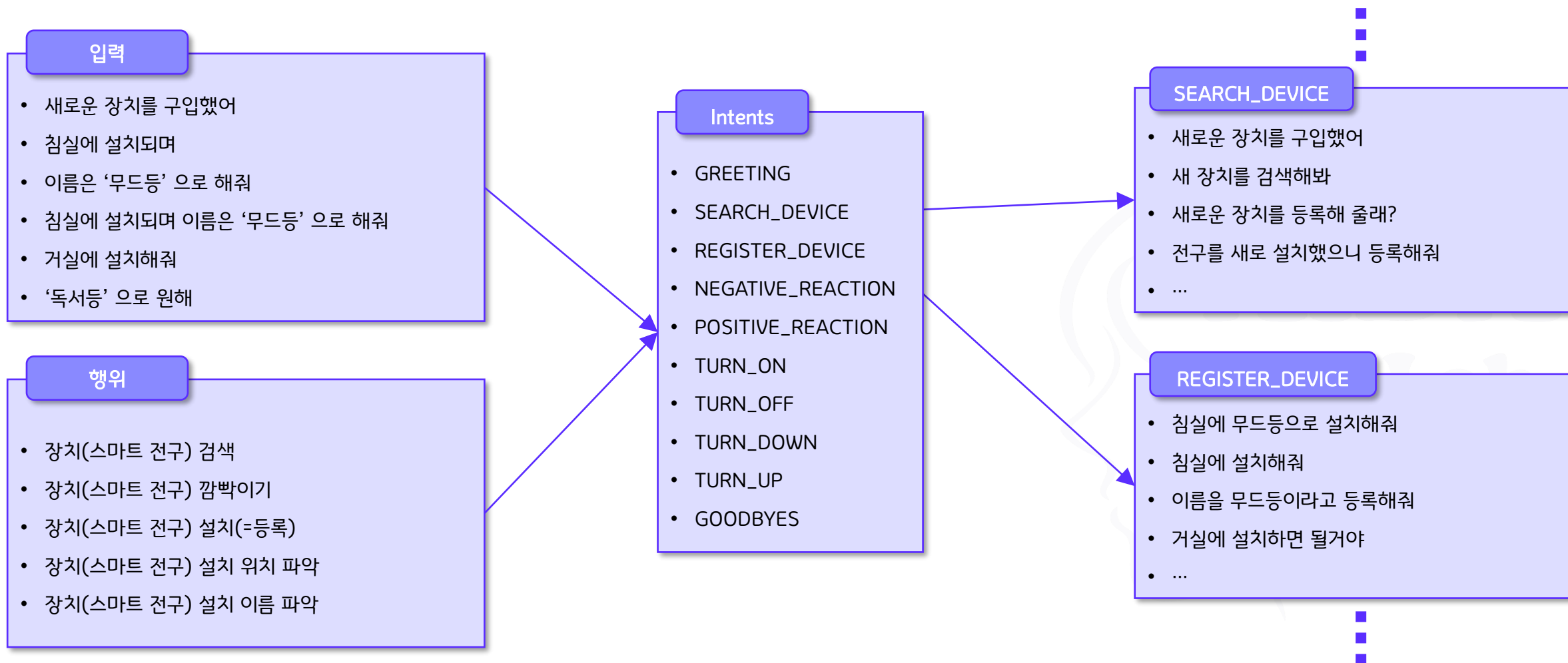


SCENARIO #1

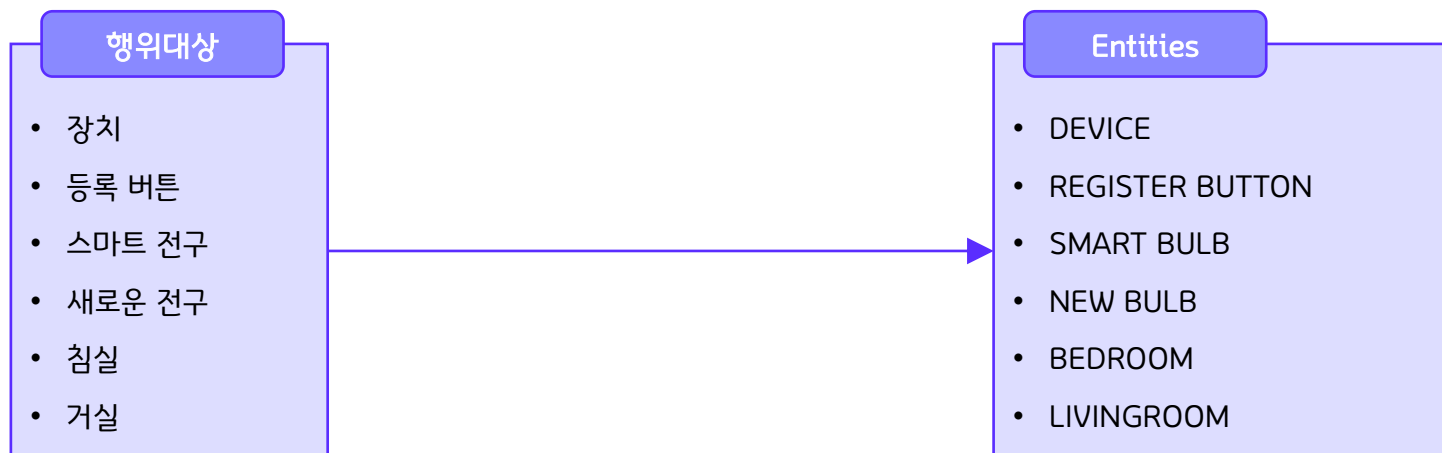




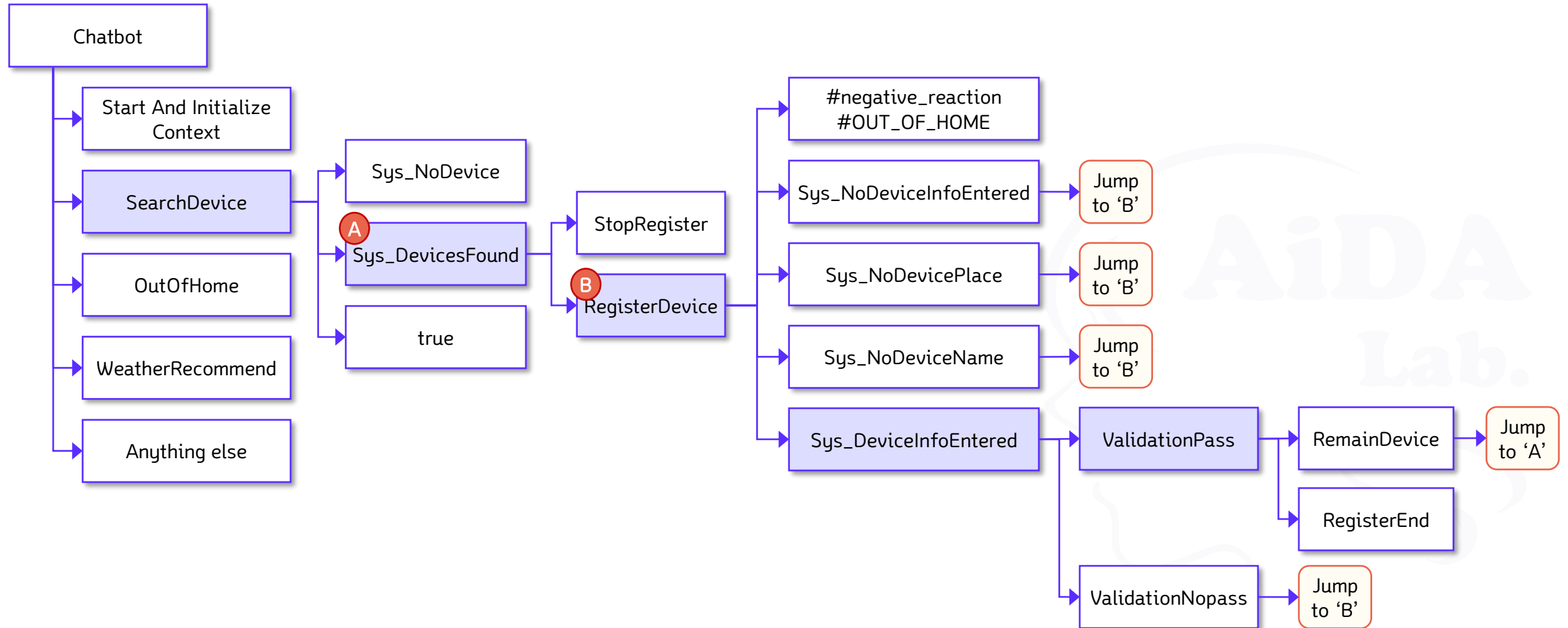
입력, 행위 → Intents



행위대상 → Entities



Dialog 구현



Dialog 구현

The screenshot displays the Dialog Editor interface. On the left, a flowchart shows a sequence of nodes: 'Sys_DeviceInfoEntered' (highlighted in green) and 'ValidationPass'. Arrows indicate the flow between them, with labels like 'Jump to RegisterDevice (Evaluate condition)' and 'Jump to ValidationPass (Evaluate condition)'. A yellow box labeled '노드' (Node) points to the 'Sys_DeviceInfoEntered' node in the flowchart.

On the right, the configuration for the 'Sys_DeviceInfoEntered' node is shown. A yellow box labeled '노드 이름' (Node Name) points to the node title. The configuration includes a condition section labeled 'If bot recognizes:' with a yellow box '제어 조건' (Control Condition) pointing to it. The condition is '\$newdevicename != "" and \$newdeviceplace != ""'. A purple box contains the text: 'Context 변수를 사용 시 \$ 기호 사용' (Use \$ symbol when using Context variables). Below this is the 'Then respond with:' section, which contains a JSON object. A yellow box labeled '전달 데이터(JSON)' (Transmitted Data(JSON)) points to this section. The JSON object is:

```
1 {
2   "context": {
3     "nlid": 0,
4     "commaxaction": "REGISTER_DEVICE_RUN"
5   },
6   "output": {
7     "text": {
8       "values": [],
9       "selection_policy": "sequential"
10    }
11  }
```

A purple box at the bottom contains the text: 'input', 'output', 'context' 등으로 구성됨 (Composed of 'input', 'output', 'context', etc.) and 'context' 내용은 REQ/RES 모두에서 갱신되면서 계속 따라다님 (The 'context' content is updated in both REQ/RES and continues to follow).

Dialog 구현

The image displays the Watson Conversation interface, illustrating the dialog flow and context management. The main dialog flow is shown on the left, with nodes for **SearchDevice**, **Sys_NoDevice**, **Sys_DevicesFound**, and **StopRegister**. The **SearchDevice** node is highlighted, showing its configuration: **#SERACH_DEVICE** (1 Response / 2 Context set). The **Sys_NoDevice** node is configured with **\$newdevicecount==0** (1 Response / 1 Context set). The **Sys_DevicesFound** node is configured with **\$newdevicecount > 0** (2 Responses / 1 Context set). The **StopRegister** node is also visible.

The right side of the interface shows the **Context variables** panel, which lists the current context variables and their values:

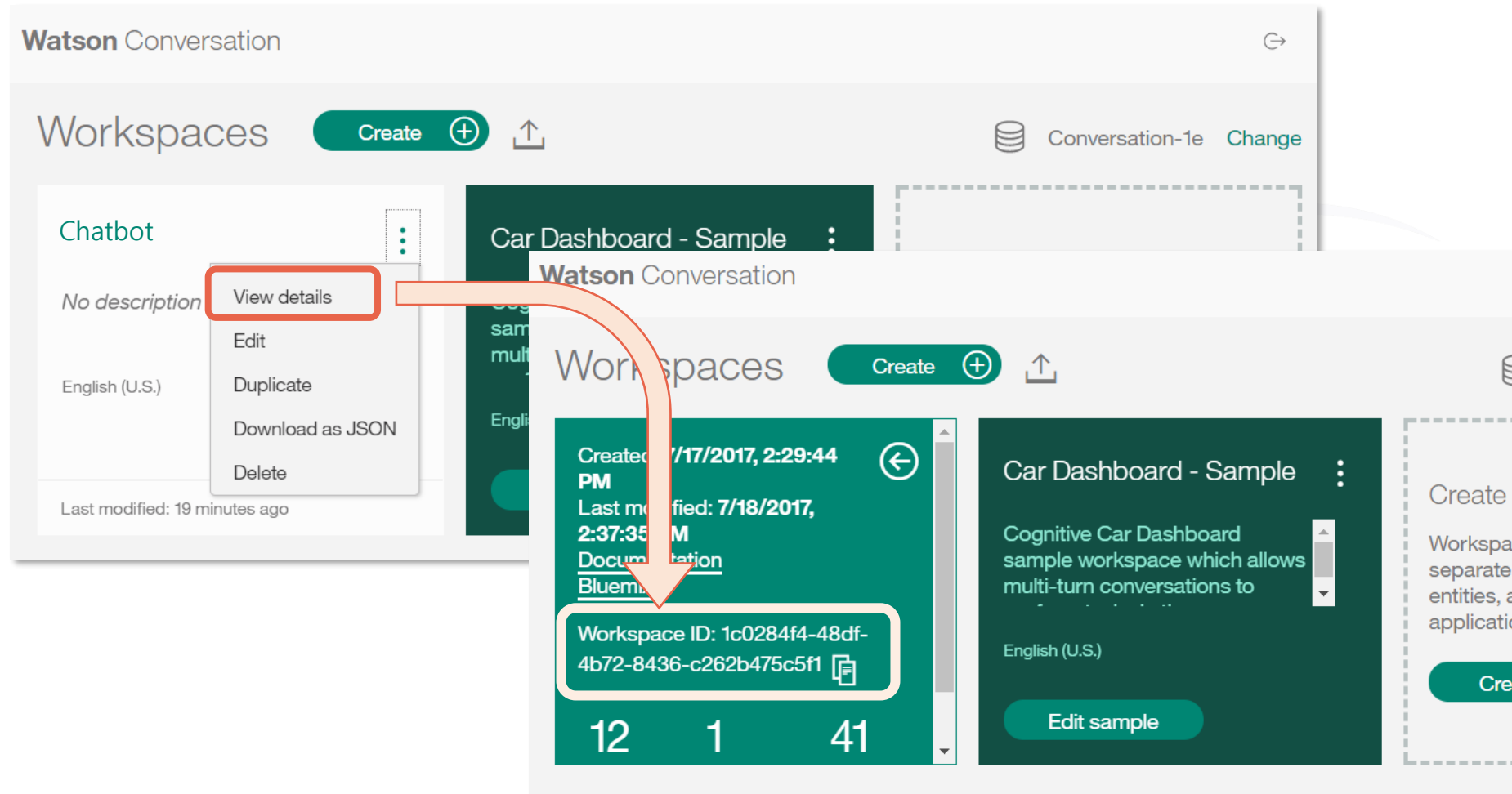
- \$timezone**: "Asia/Seoul"
- \$nlu**: 0
- \$validation**: 0
- \$currenttime**: "2017-07-18 14:37:40"

Below the list is a button to **Remove all context variables**.

Annotations in the image highlight key features:

- Context 변수 테스트** (Context variable test) points to the **Manage Context** button in the top right of the dialog flow editor.
- Input Data** points to the input field in the **Try it out** window, which contains the text "i have new devices".

- **Workspace ID**



• Intents

Intents Entities Dialog

✓ #100.새_장치_등록하기

+ Add a new user example...

- ☐ 새 장치를 등록해 줄래?
- ☐ 새 장치를 설치했어
- ☐ 새 장치를 구입했어
- ☐ 새로 전등을 설치했는데 등록해줘
- ☐ 장치 검색해봐
- ☐ 전등을 새로 달았는데 찾아봐
- ☐ 전등 등록해야겠다
- ☐ 장치 등록할거야
- ☐ 전등 달았어
- ☐ 전등 사왔어 등록해줘

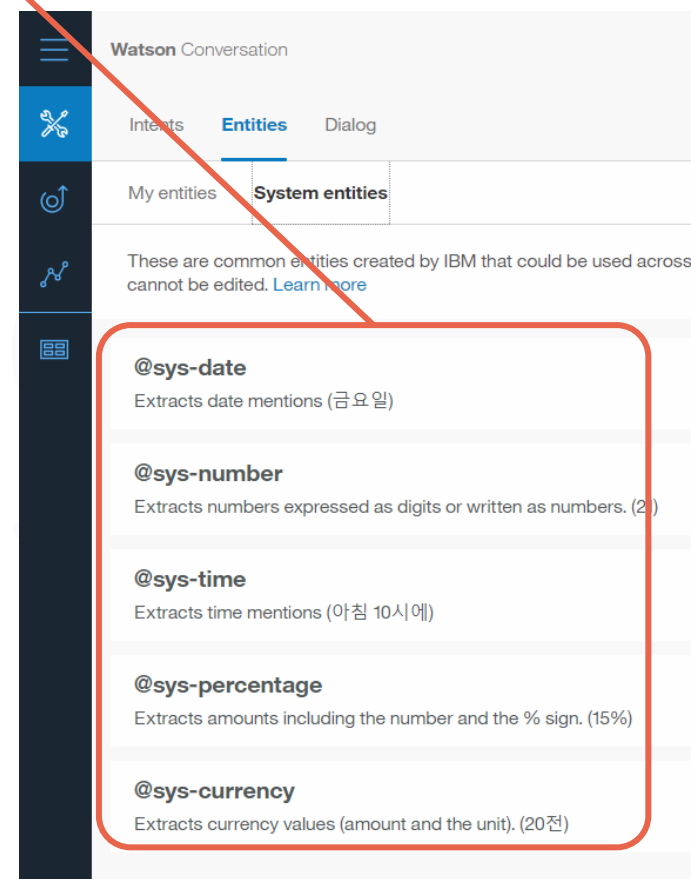
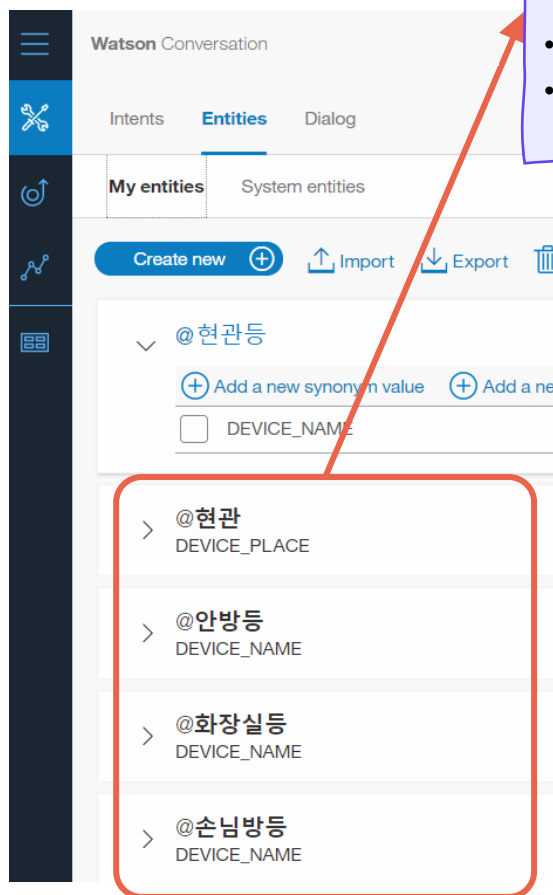
- ☐ I have new devices.
- ☐ I have some new devices.
- ☐ I bought new devices.
- ☐ I bought some new devices.

학습용 문장

- 문장 등록 후 자동으로 학습이 진행 됨
- 여러 언어를 사용해서 등록하면 각 언어별로 인식 가능
- 등록된 문장만 인식하는 것이 아니라 유사도 값을 이용하여 문장 선택
 - 다른 인텐트에 유사한 표현이 있을 경우 잘못된 선택 가능
 - 각 인텐트는 명확하게 구분 가능한 표현을 쓰는 것이 이상적
- 기본적으로 IBM에서 제공하는 인터페이스에서는 한 번에 한 문장만을 인식 가능함
 - 두 가지 이상의 명령을 함께 이어서 내릴 경우 제대로 인식 못함
 - 현재는 두 가지 이상의 명령이 포함된 학습용 문장을 입력하여 해결
- Backend에서 NLU를 통해 다수의 인텐츠를 처리 가능하도록 하는 방법을 모색 중

• Entities

- 사용자가 등록한 엔티티와 시스템 엔티티로 구성됨
- 입력된 문장에서 엔티티에 해당하는 값이 있을 경우 entities 필드에 데이터가 저장됨
- 배열 형식으로 저장되므로 동시에 다수의 값 처리 가능
- 인텐트에서 처리가 어려운 다수의 내용은 엔티티와 함께 사용할 때 일부 해결 가능함

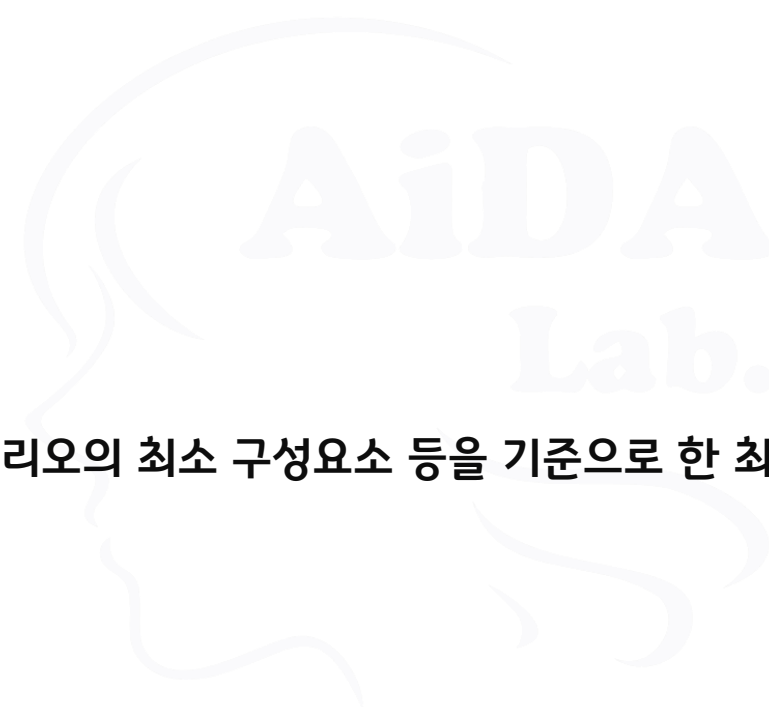


• Dialog

- 현재의 구성: 시나리오에 따라 작성되어 유연성이 부족함
- 노드 트리의 형태, 구성에 따라 하위 노드 → 상위 노드 이동이 불가능한 경우도 있음

• 해결방안

- 각 노드의 종속성을 최대한 제거, 독립적인 노드를 구성할 필요가 있음
- 지나치게 독립된 노드는 전체 프로세스의 운영을 어렵게 할 수 있음
- 각 노드와 노드의 계층은 처리해야 하는 기능 단위, 프로세스 단위, 시나리오의 최소 구성요소 등을 기준으로 한 최적화가 필요



IBM Watson Knowledge Studio

View Details

Replace

Concordance

Attribute View

Ready

Save

Close

Alpha...

14pt

1

Mention

Relation

Coreference

learning_document1.txt

1

I get frustrated on a daily basis when I ask my students the most basic of questions, such as, "How was your day?

2

" or "What did you eat for lunch?

3

" As is usual in **Korea**, the students know a decent amount of vocabulary, but it's still impossible to have even a simple conversation.

4

Since I only teach each class once a week, I can't hold myself responsible, naturally.

5

Therefore, I have decided to occasionally look into what the **Korean** teachers are teaching.

6

A page from the story book, Prank Calls are Bad, which the Korean teacher's at my school were forced to teach last week.

7

All of these stories books, full of random vocabulary and pointless topics, are made by the company that owns our Hagwon and the students' parents are forced to buy them - a pretty good business idea considering there's a whopping 597 franchises.

8

This scene comes after the boy's mother found out her son made a prank call to the police and left the phone off the hook so that grandfather couldn't get a hold of her.

9

I guess the lesson is if you make prank **phone** calls, your mother will make you sit on your knees with your hands in the air while beating you senseless.

10

After reading the book, the students wrote a book report.

Entity

Mention

Type

Subtype

Role

-

DEVICE

-

DEVICE_PLACE

• 학습용 문구 작성

DEVICE_PLACE

DEVICE_NAME

설치장소는 **침실**로 하고 이름은 **무드등**으로 해줘.
설치장소는 **침실**로 등록하고 **장치 이름**은 **무드등**으로 해줘.
 설치하는 공부방에 하고 이름은 공부등으로 해줘.
 서재에 설치하고 이름은 독서등으로 해줘.
 아버지방으로 등록하고 이름은 아버지방 등으로 해줘.
장치이름은 거실등으로 하고 **설치장소**는 거실로 하자.
 장치명은 주방등이고 장소는 주방으로 해.
 이름은 베란다등으로 하고 장소는 베란다로 등록해줘.
 민지 방으로 등록하고 이름은 초록등으로 해줘.
 민서 방에 설치하고 민서 등이라고 등록해줘.
 1층 계단에 설치하면 돼.
 민지 방에 설치해줘.
 민서 방에 등록해줘.
 장소는 어머니 방으로 해줘.
 설치장소는 누나 방으로 하면 돼.
 설치하는 침실에 해줘.
 등록은 거실로 하면 돼.
 공부방에 설치해.

- **동일한 단어**가 있으므로 해당 단어는 키워드, 나머지 요소는 패턴으로 인식
- 패턴으로 추정되는 **동일한 표현**이 상이한 단어와 함께 쓰이므로 상이한 단어의 내용은 앞으로 변화할 값으로 인식
- 변화할 값으로 인식된 내용이 DEVICE_NAME, DEVICE_PLACE로 Annotation 되었으므로 인식 패턴 확정

작은 방에 설치하고 **작은 등**으로 이름지어줘.
작은 방에 등록하고 이름은 **작은 등**으로 하자.
 이름은 작은 등으로 해줘.
 이름은 초록등이면 되겠네.
 민수 방에 등록하고 이름은 파란 등으로 해줘.
 이름은 파란 등으로 하고 설치하는 민지 방으로 해줘.
 게스트룸에 번개등이라는 이름으로 설치해줘.

주방에 불 켜줘.
 민지 방에 불 켜줘.
 안방에 불켜줘.
 큰 방에 불 켜라.
 민수 방에 불 켜라.
 공부방에 불 켜줄래?
 아버지 방에 불 켜줄래?
 어머니 방에 불 켜.

• 학습용 문구 작성

Install into **bedroom** and device name is **moodlight**.

Install into **livingroom** and name is **livinglight**.

I will install in the **bedroom** as **moodlight**.

I will install in the studyroom and the name is smartlight.

Please install it in the livingroom as livinglight.

Register in the kitchen and the name is foodlight.

Please register in the restroom.

Please register in the toilet.

Let the place be a kitchen.

Install in a small room and name is small light.

The name is bluelight and install into minji room.

The name is moodlight and install in the bedroom.

Please install it in the guestroom under the name guestroom light.

Please install it in the bedroom under the name of moodlight.

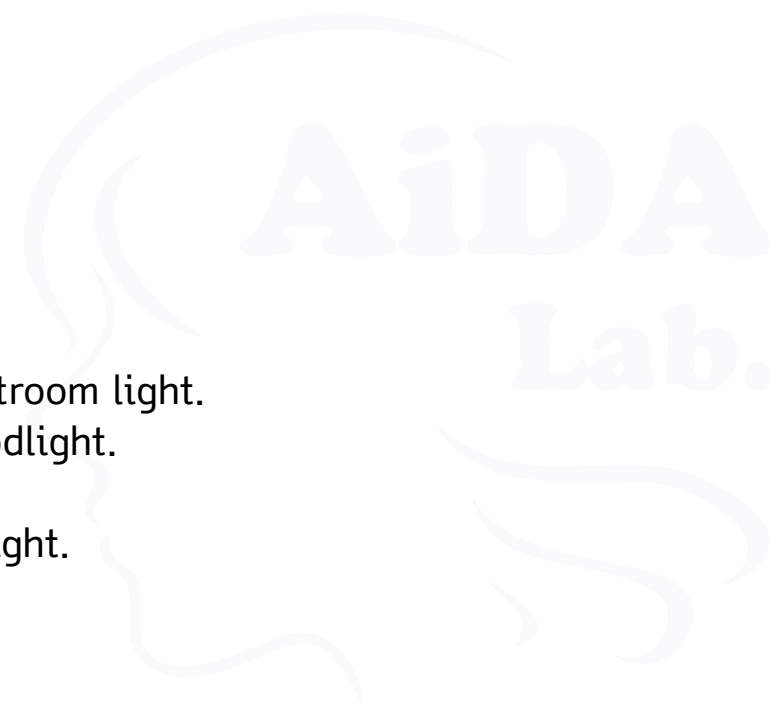
Please install it as a smartlight.

Please install it in the restroom and register it as restlight.

The name is bluelight.

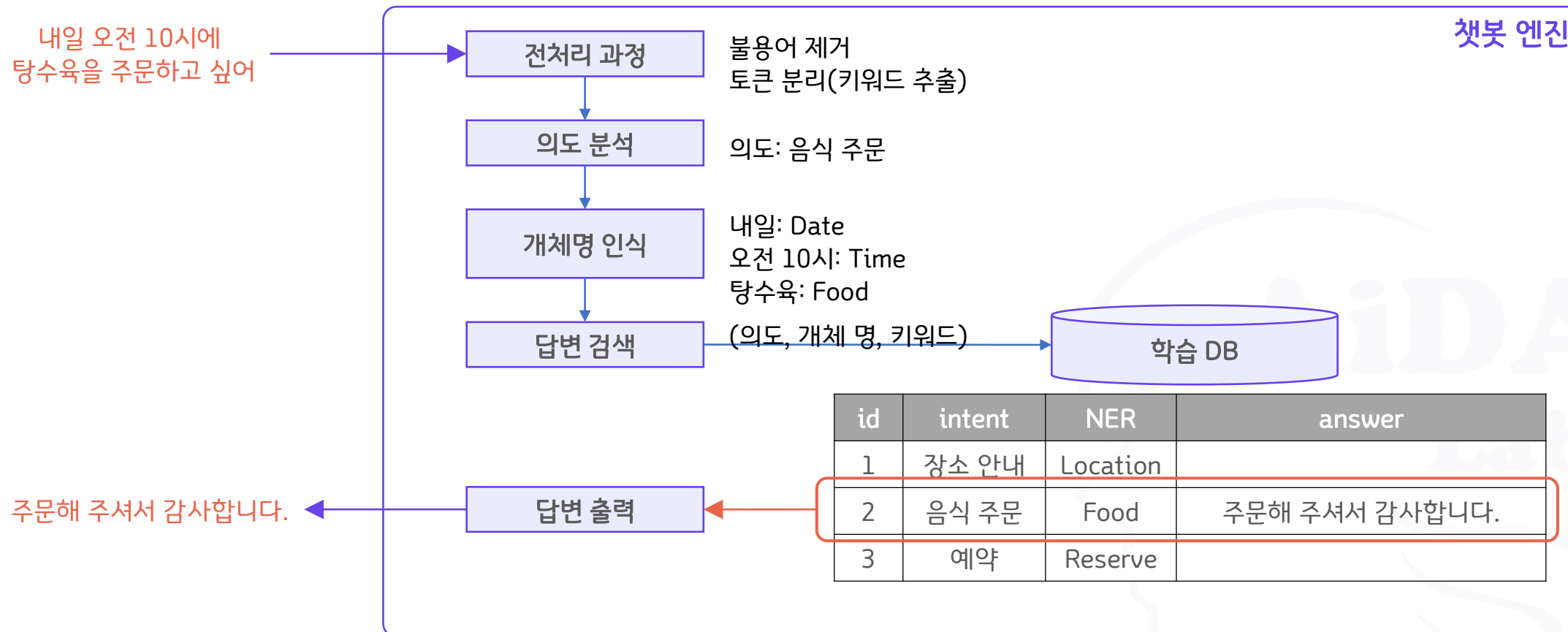
The name is greenlight.

Register in the kitchen.



Chatbot Engine

기능	설명
질문 의도 분류	화자의 질문 의도 파악 → 해당 질문을 의도 분류 모델을 이용해 의도 클래스를 예측하는 문제
개체 명 인식	화자의 질문에서 단어 토큰 별 개체 명 인식 → 단어 토큰에 맞는 개체명을 예측하는 문제
키워드 추출	화자의 질문 의미에서 핵심이 될 만한 단어 토큰 추출 → 형태소 분석기를 이용해 핵심 키워드가 되는 명사, 동사 추출
답변 검색	해당 질문의 의도, 개체 명, 키워드 등을 기반으로 답변을 학습DB에서 검색
소켓 서버	다양한 챗봇 클라이언트에서 요청하는 질문을 처리하기 위한 소켓 서버 프로그램



기능	설명
STT 호출	입력한 음성 데이터를 Cloud 서비스의 STT API로 전달, 텍스트 데이터 확보
키워드 추출	화자의 명령 내용(텍스트)에서 키워드 추출 형태소 분석기를 이용해 핵심 키워드가 되는 명사, 동사 추출
명령 수행	키워드의 내용에서 대상(명사), 동작(동사)를 이용하여 기기제어 명령 수행

THANK
YOU

