

The background features a dark blue gradient on the left, transitioning into a large, abstract, curved shape on the right. This shape is composed of various shades of purple and blue, with a bright orange-yellow highlight along its bottom edge. The overall design is modern and dynamic.

# aws SUMMIT

KOREA | MAY 10-11, 2022

T8S1

# Amazon Lex를 활용한 한국어 챗봇 구축하기

강정희  
솔루션즈 아키텍트  
AWS



© 2022, Amazon Web Services, Inc. or its affiliates. All rights reserved.

# Agenda

대화형 인터페이스의 사용사례

Amazon Lex : 대화형 AI

Amazon Lex 주요 개념 살펴보기

실전 Chatbot 구성하기

Demo

# 대화형 인터페이스의 사용사례

# 고객은 상황에 맞는, 다양한 상호작용을 원합니다.



## 다양한 플랫폼 요구사항

- 전화, 모바일, 앱, 웹...

## 음성 대화 vs. 채팅

- 응답 시간, 시각정보 활용 등의 선호 존재

## Covid-19 시대와 비대면 업무 선호

- 새로운 채널 혹은 기존 채널의 확대 필요

# 대화형 인터페이스의 장점



자연스러움



온 디맨드



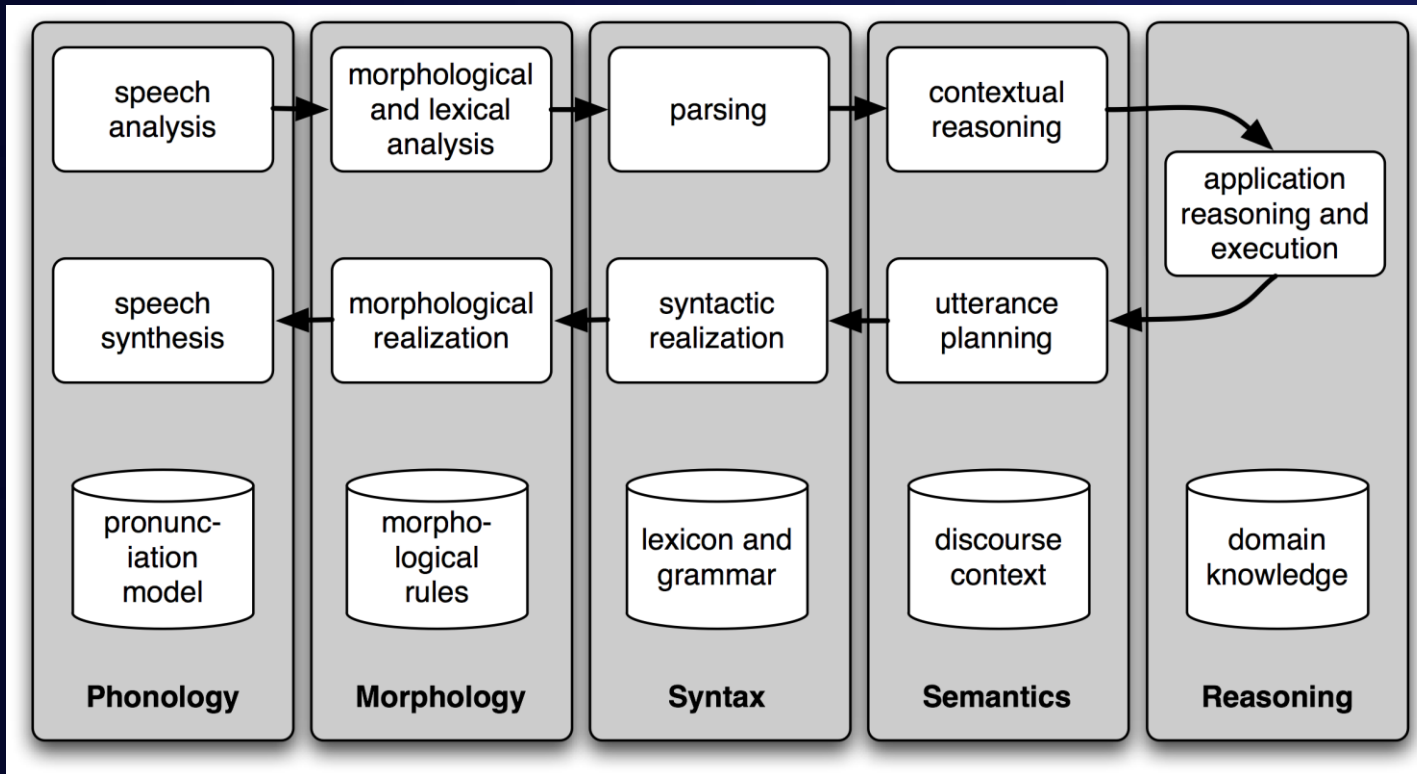
접근성



효율성

# 자연어 처리의 어려움 - 복잡한 처리 과정

자연어 처리(NLP)는 언어학의 개념과 깊은 연관성을 지니고 있으며, 다양하고 복잡한 과정으로 구성되어 있습니다.



## 음성 인식 (음운론)

- Speech-to-Text
- Text-to-speech

## 형태소 분석 (형태론)

- 동의어, 동음이의어 처리
- 중의성 해소
- 신조어, 미등록어 판별 및 등록

## 구문 분석 (통사론, 의미론)

- 문법 기반 구조 분석
- 맥락 기반 추론

Image Source: [http://magizbox.com/training/natural\\_language\\_processing/site/tasks.html](http://magizbox.com/training/natural_language_processing/site/tasks.html)

# 음성 인식의 어려움 - 하나만 놓고 보면 쉬울까요?

자연어 처리의 한 분야인 음성 인식(ASR Automatic Speech Recognition)에서  
특히 해결이 쉽지 않은 문제들로 어떤 것들이 있을까요?

- 높은 노이즈
- 대화 상의 artifacts와 뉘앙스(nuance)
- 화자가 여럿일 경우
- 처음 접하는 용어들
- 방언, 사투리 처리
- 활용 사례가 적은 언어 또는 은어 처리

•  
•  
•



# 인식 이후의 텍스트 분석 과정은 더 복잡합니다.

또, 수없이 쏟아지는 텍스트에 대해서도 빠르게 파악하고 이해해야 합니다.  
이를 위해서는 어떤 것들을 해결해야 할까요?



- 고유명사 구분 및 추출 (NER)
- 언어에 대한 자동 인식
- 핵심 키워드 파악 및 문서 요약
- 텍스트의 성향 분석
  - 감성 분석
  - 의도 분석, 오피니언 마이닝
- 맥락 기반 추론
- 시간 흐름에 따른 변화 내역 추적
- **지능형 질의 응답**

# 하지만 누구에게나 머신러닝이 쉽지만은 않습니다.

이렇게 뛰어난 기술 발전에도 불구하고  
여전히 AI/ML은 다가가기 어려운 영역일 수 있습니다. 왜 그럴까요..?



- 학습용 데이터셋이 없어요
- 데이터 과학자, 머신 러닝 전문가가 없어요.
- 컴퓨팅 리소스가 부족해요
- 머신 러닝 같은 거 잘 몰라도 그냥 바로 쓸 수 있는 건 없나요?
- 모든 걸 직접 만들려고 하니엄두가 안나요
- 지금 비즈니스에서 UX를 향상시키고 싶은데 시스템 API 개발만 가능해요

# Amazon Lex : 대화형 AI



# Amazon Lex : 대화형 AI



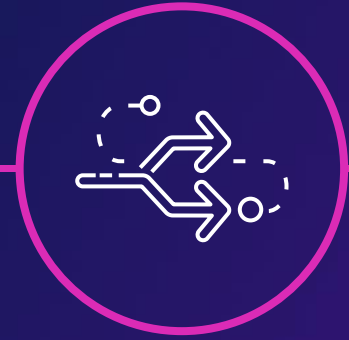
높은 품질의 텍스트,  
음성 자연어 이해(NLU)



손쉬운 사용 – 완전 관리형  
AI 서비스



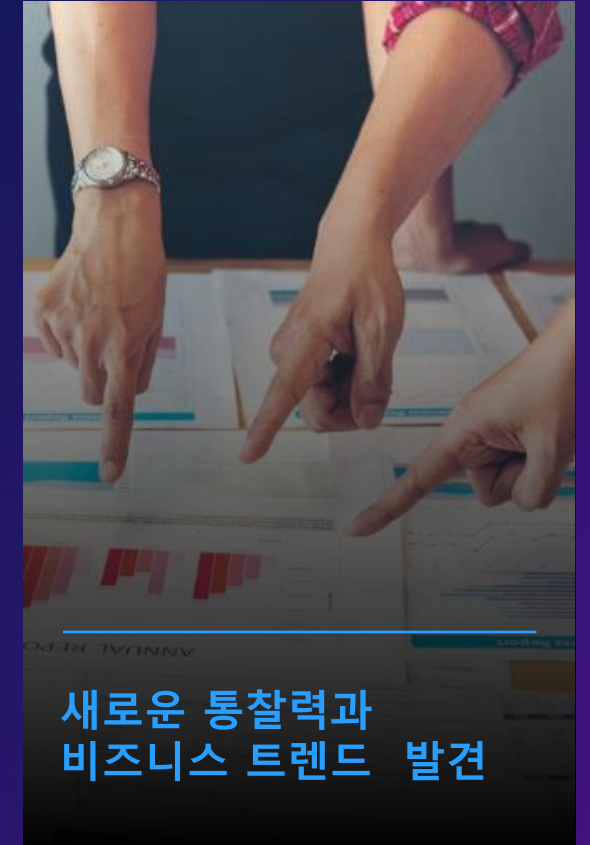
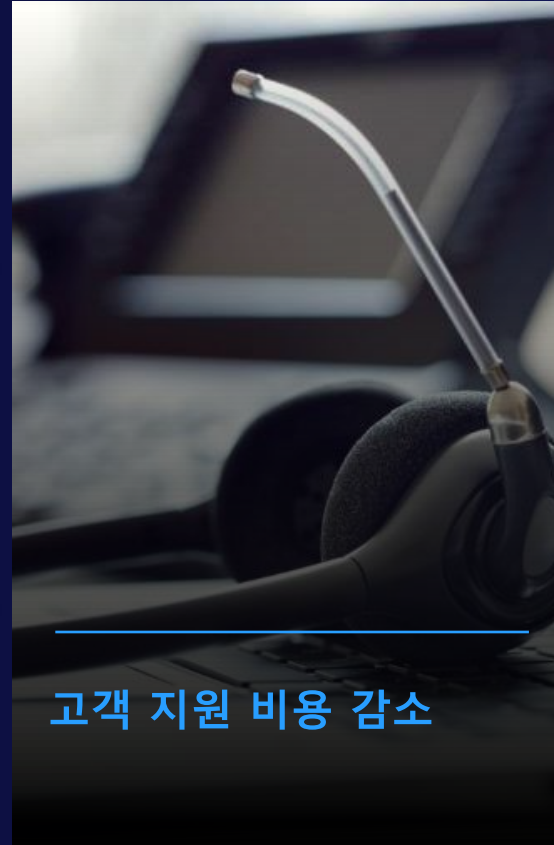
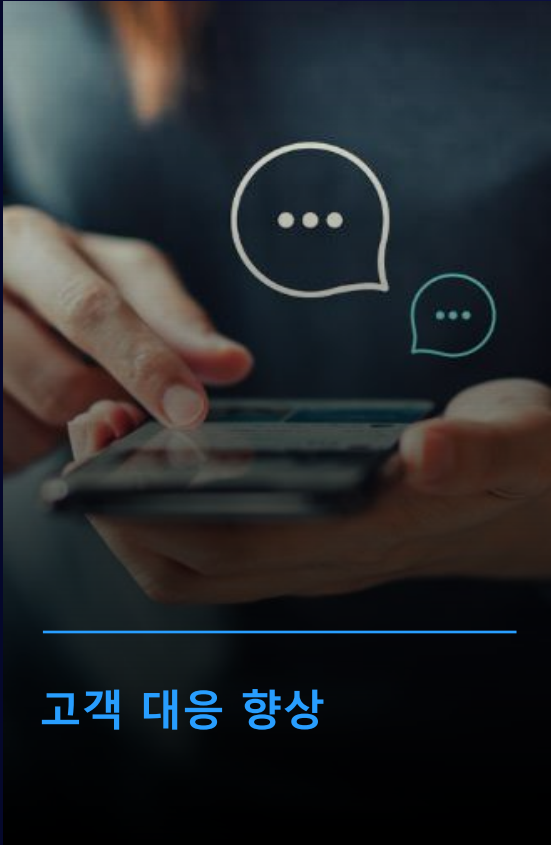
지속적으로 여러 채널과  
규모에 맞게 개발하고  
배포가능



AWS 서비스와의 통합:  
Amazon Connect, Amazon  
Kendra, and AWS Lambda

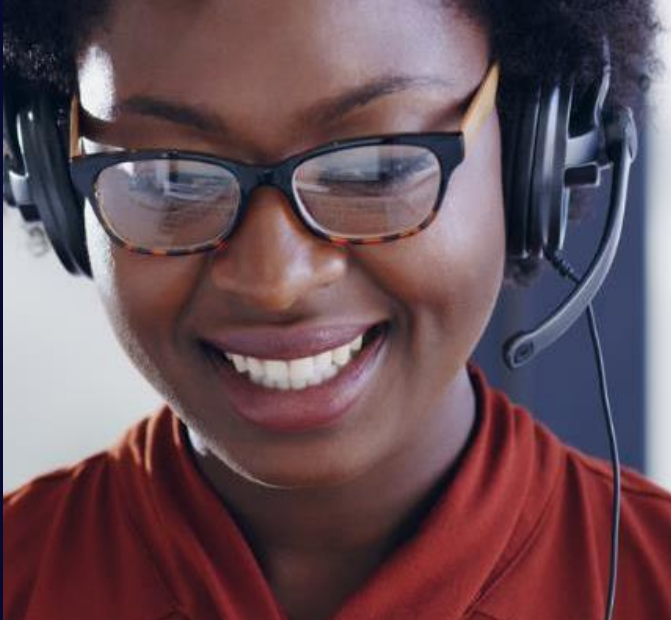
고객의 요구에 맞는 대화형 인터페이스를 디자인하고 빌드합니다.

# 장점 및 기대효과





# 사용 사례



## 고객센터

고객 문의를 자동으로  
해결하거나 올바른 상담원으로  
안내하는 가상 상담원



## 정보 봇

일상적인 요청 및 자주 묻는  
질문을 처리하는 챗봇



## 기업 생산성 봇

기업 내부 업무 활동을  
간소화하여 업무효율 향상

# 주요 기능

NEW!

## 자동화된 chatbot designer



대화 기록을 이용하여  
몇 주가 아닌 몇시간  
만에 Chatbot 작성

## 자연스러운 대화



높은 품질의 ASR과 NLU  
(Amazon Alexa 와 같은  
기술 적용),  
컨텍스트 관리,  
주고받는(multi-turn) 대화

## 개발자 생산성 도구



원클릭 배포, 봇  
라이프사이클 관리 도구,  
V2 console and API,  
streaming API

## AWS 서비스 통합



Amazon Polly, Amazon  
Kendra, AWS Lambda 와  
기본적으로 통합

## Contact center 통합

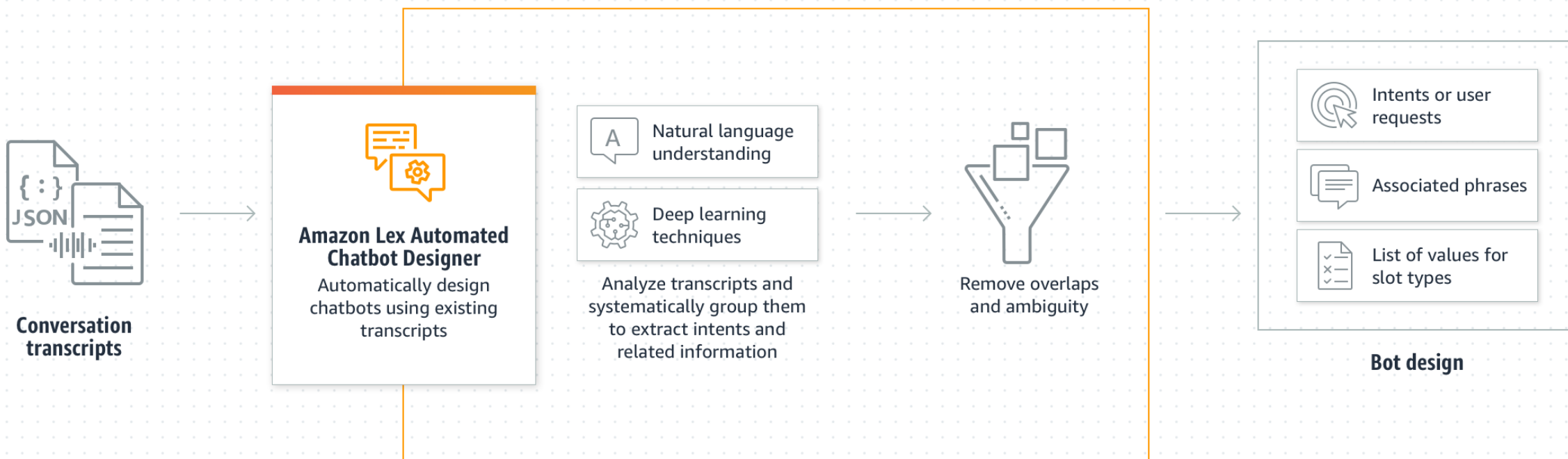


Amazon Connect,  
Genesys, 8x8 와 통합

# 자동화된 Chatbot Designer (평가판) – 작동 방식

NEW !

대화 기록을 통해 자동으로 Chatbot을 설계할 수 있습니다.



현재 영어만 지원합니다.



# AWS 통합



## Amazon Kendra

ML 기반의 사용하기 쉬운  
지능형 검색 서비스입니다.

봇이 구조화되지 않은 문서  
와 FAQ에서 가장 정확한  
답변을 찾을 수 있도록  
합니다.



## Amazon Polly

다양한 스타일과 언어의  
음성으로 사용자에게  
응답할 수 있는 TTS(텍스트  
음성 변환) 서비스입니다.



## Amazon Lambda

Amazon Lex를 기존  
엔터프라이즈  
애플리케이션 및  
데이터베이스와 쉽게  
통합할 수 있습니다.



## Amazon Connect

기본적으로 AWS의  
클라우드 기반 컨택 센터인  
Amazon Connect와  
통합되어 개발자가 전화를  
통해 고객 쿼리를 처리할  
수 있는 음성 기반 대화형  
봇을 구축할 수 있습니다.

# Contact center 구축

## 가상 상담원



사람과 유사한 가상  
상담원과 IVR을 통해  
통화량을 분담

---

상담원 통화량 감소

## 지능형 라우팅



발신자의 의도를 파악하여  
적절한 상담원 대기열로  
라우팅

---

고객 경험 향상

## 정보 수집

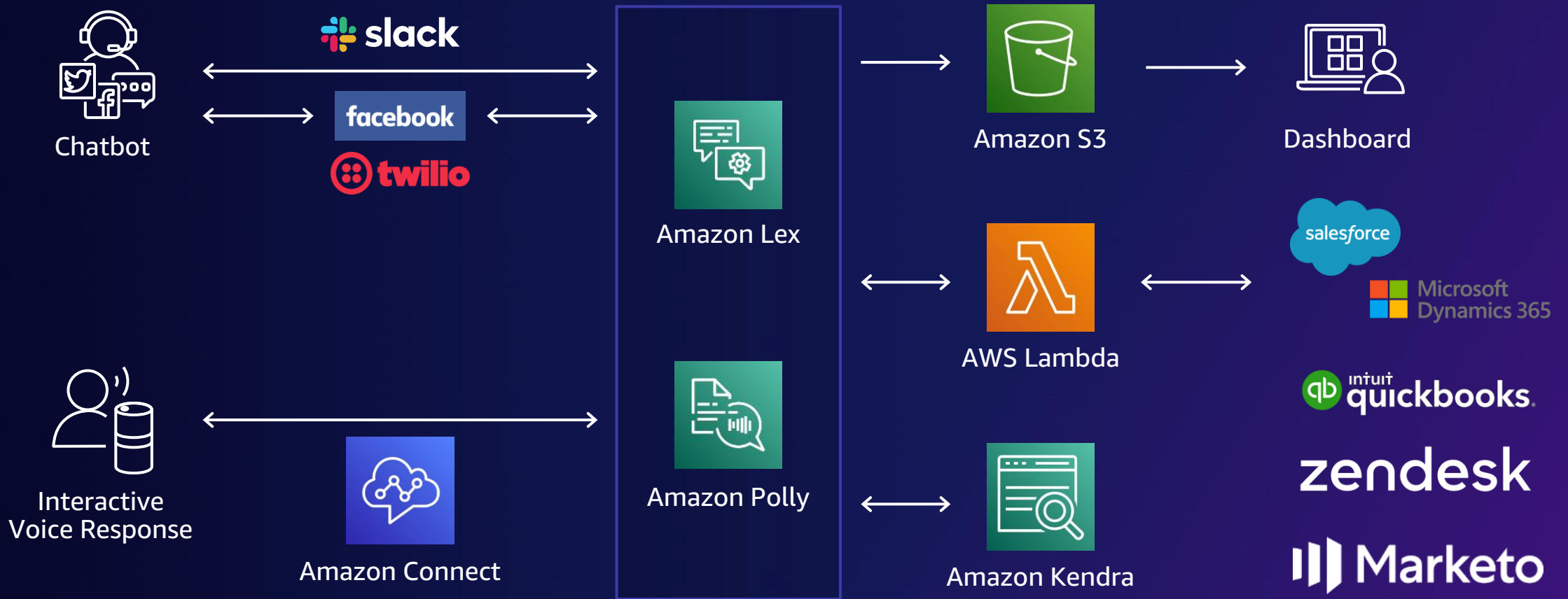


Collect key information to  
assist agents in resolving  
questions efficiently

---

통화 처리 시간 단축

# AWS 통합 - 참조 아키텍처



# Amazon Lex 주요 개념 살펴보기

# Amazon Lex – 적용 기술 및 서비스

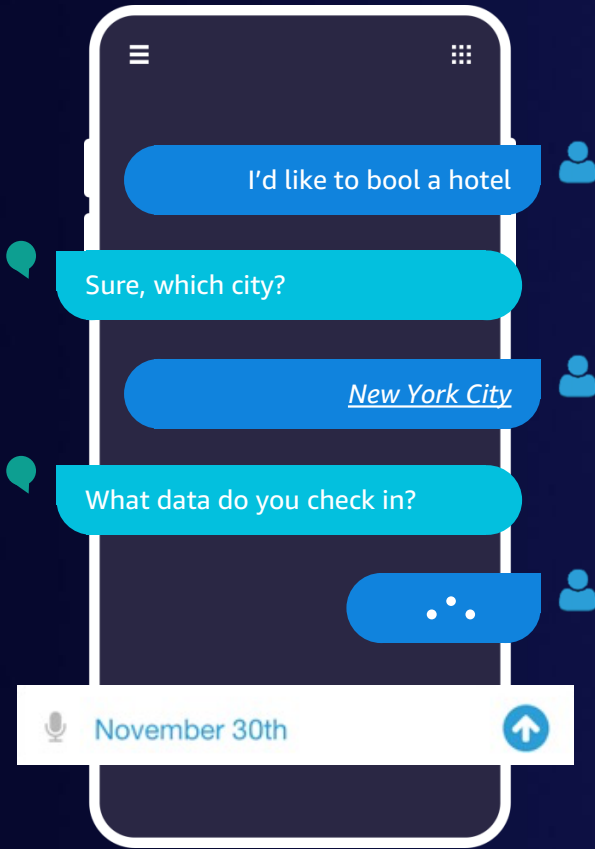


# Amazon Lex 구성요소

BookHotel

의도(Intent)

사용자의 자연어 입력을 통해서  
수행하고자 하는 의도



# Amazon Lex 구성요소

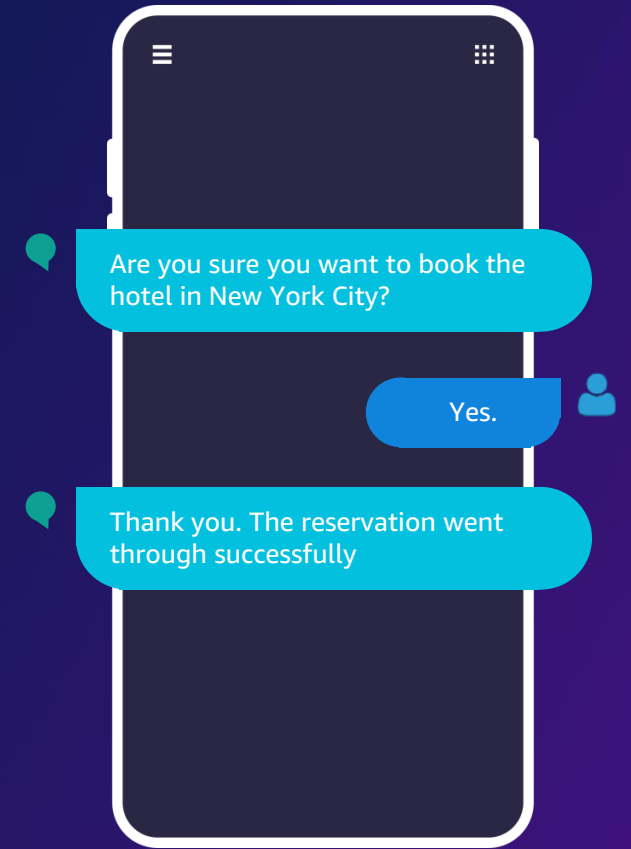
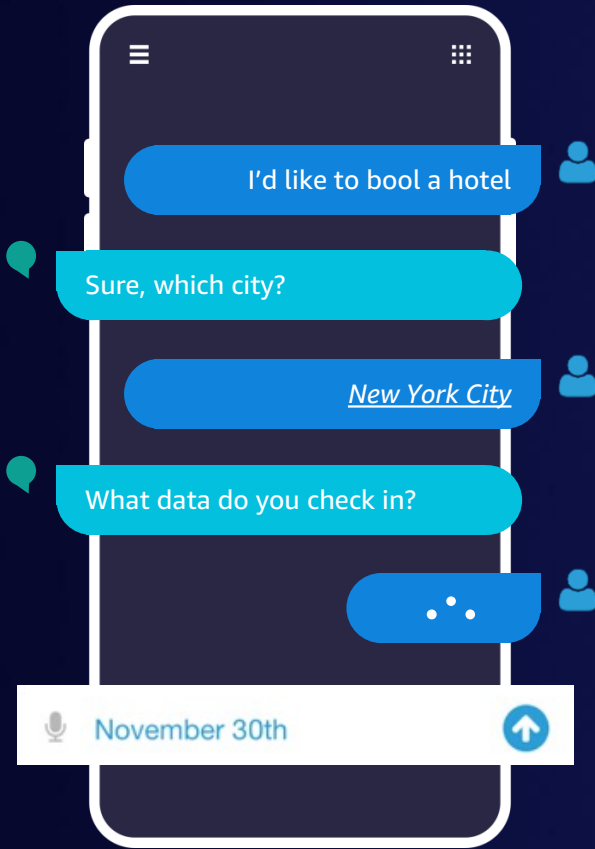
**BookHotel**

**의도(Intent)**

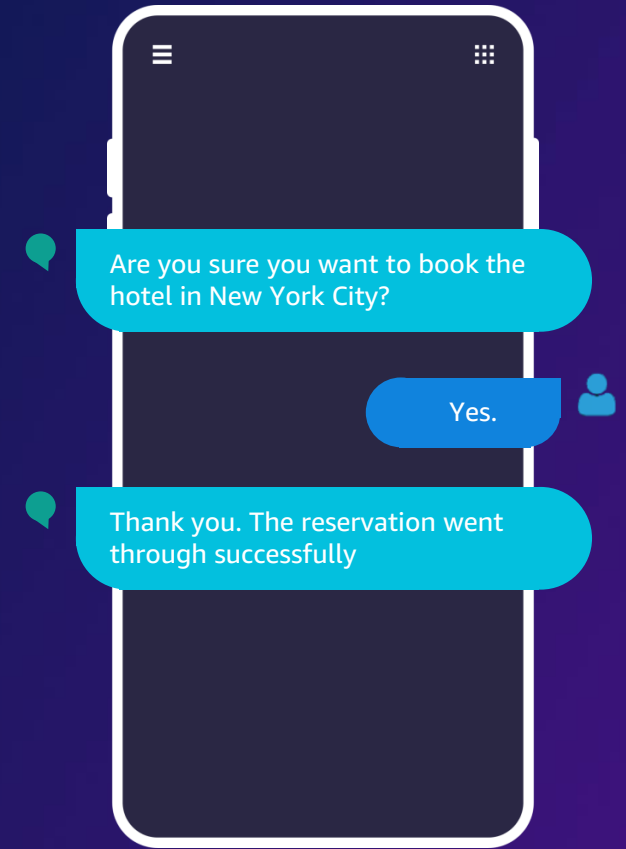
사용자의 자연어 입력을 통해서  
수행하고자 하는 의도

**Utterances**

의도를 동작시키는 문장이나 구문

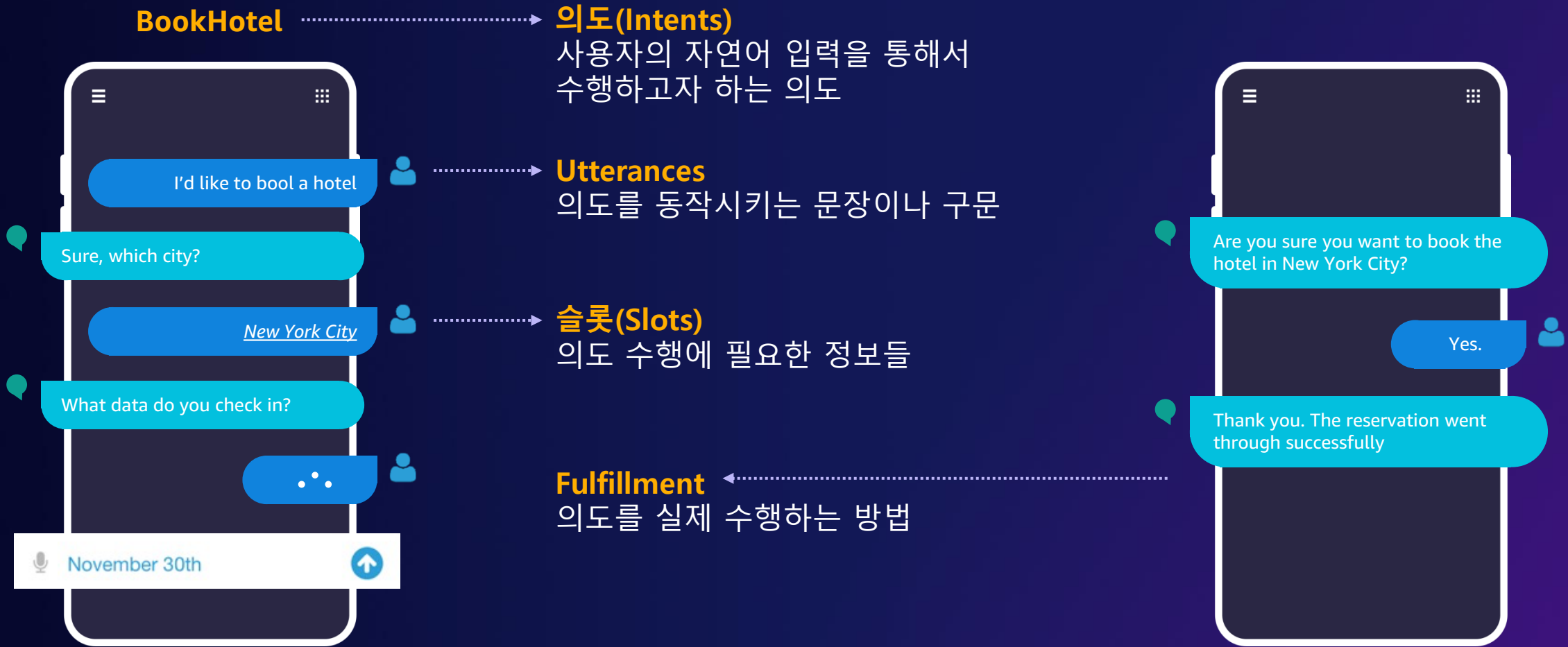


# Amazon Lex 구성요소

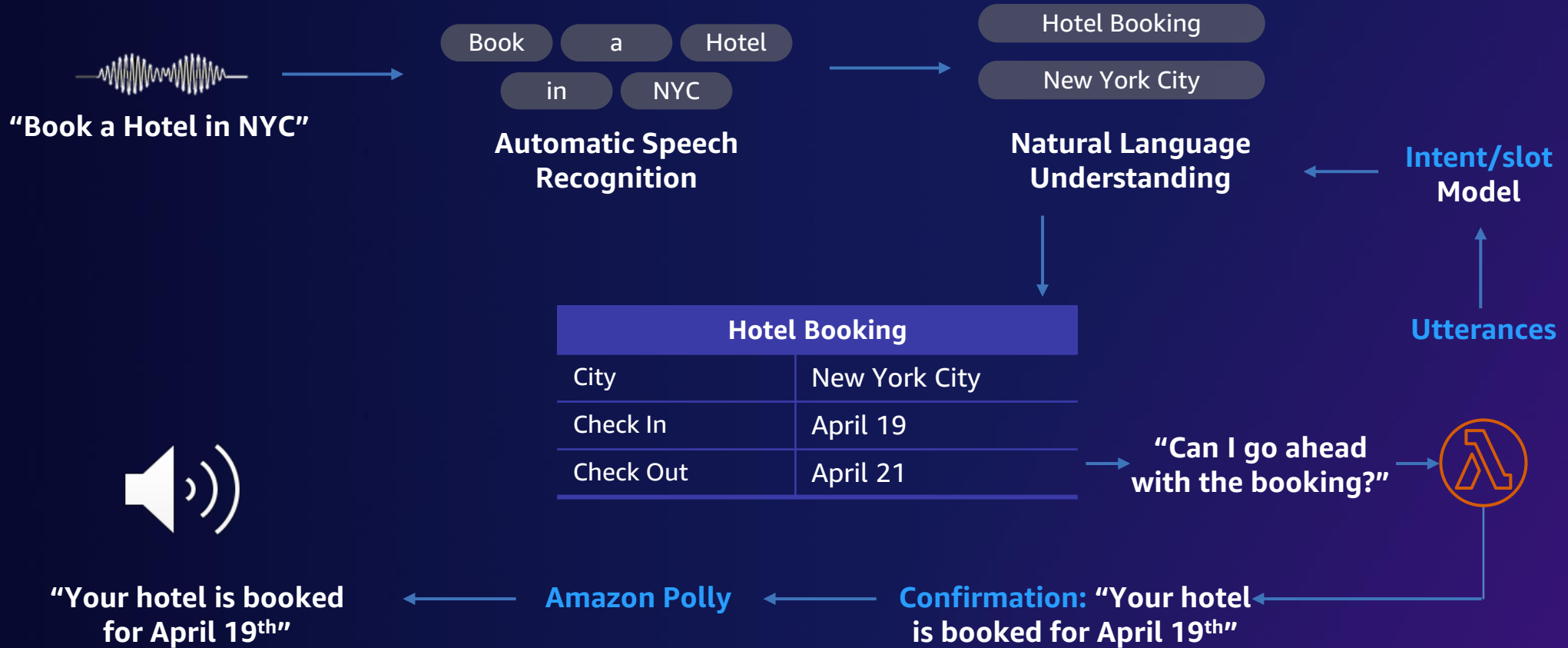




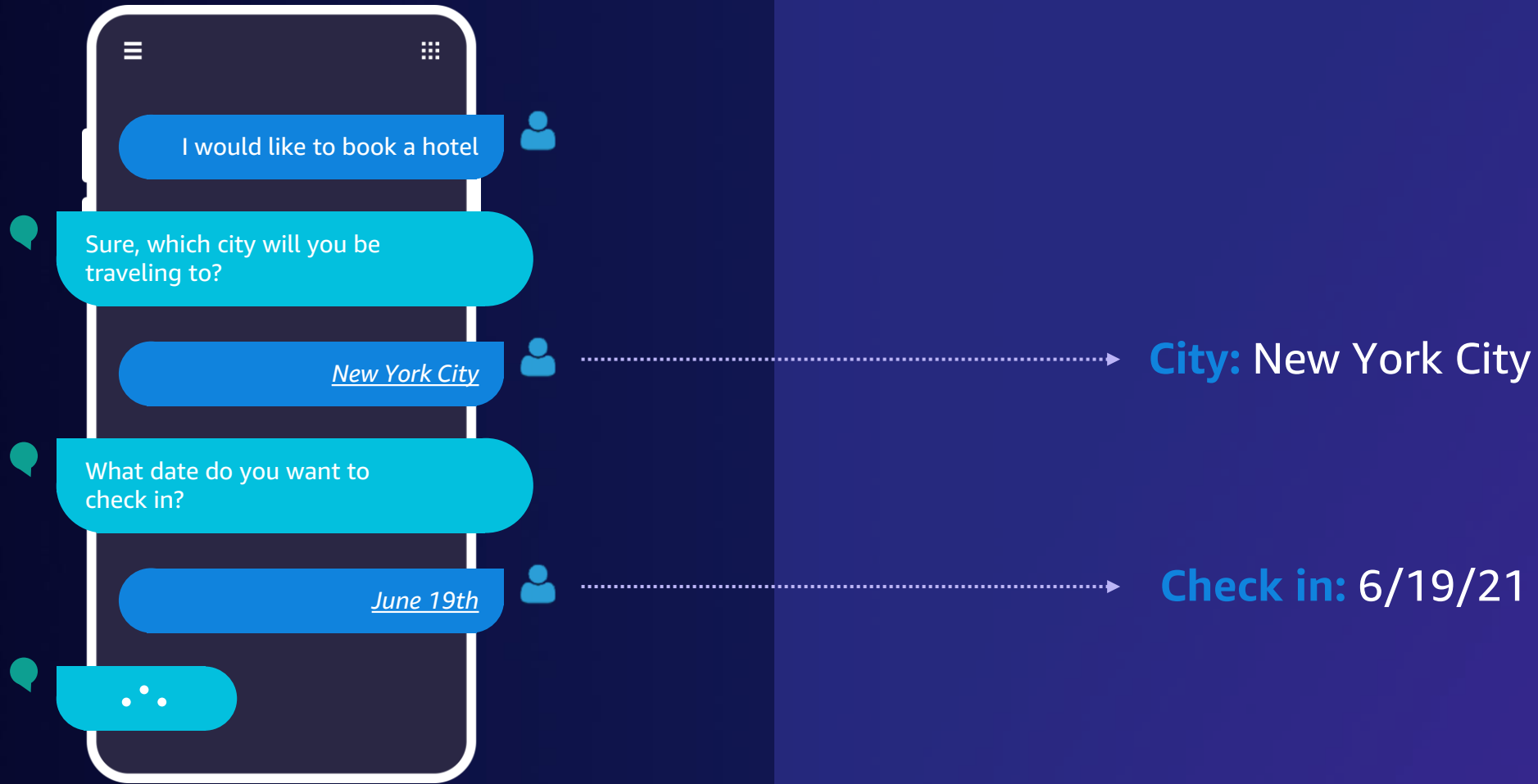
# Amazon Lex 구성요소



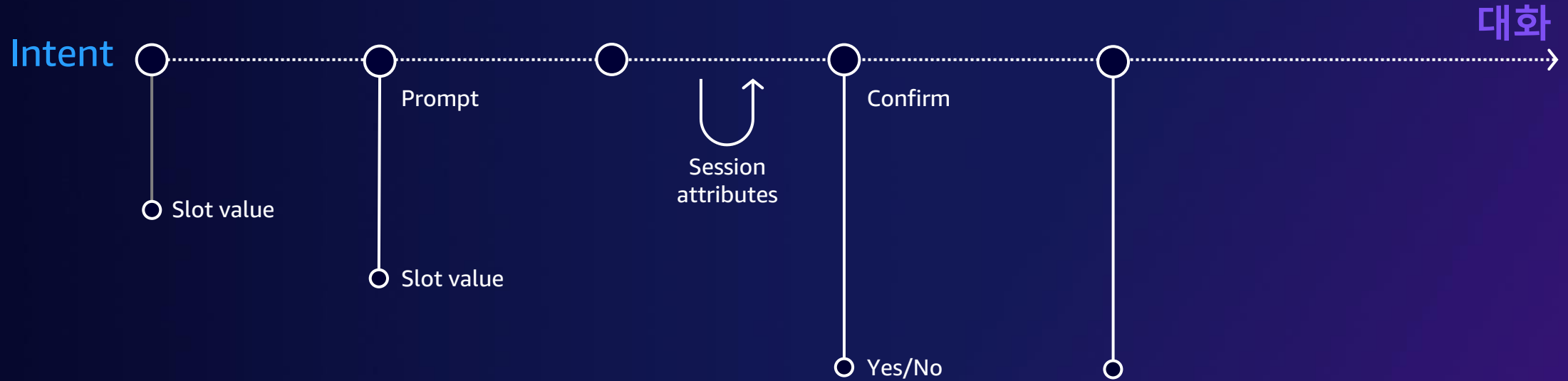
# 예시 : “Book a hotel” 의도를 음성으로 처리하는 과정



# 슬롯 정보 요청 (slot elicitation)



# 대화의 문맥(Context) 유지



Lex는 대화 전반에서, 데이터를 저장하여 컨텍스트를 유지합니다.

Slot values

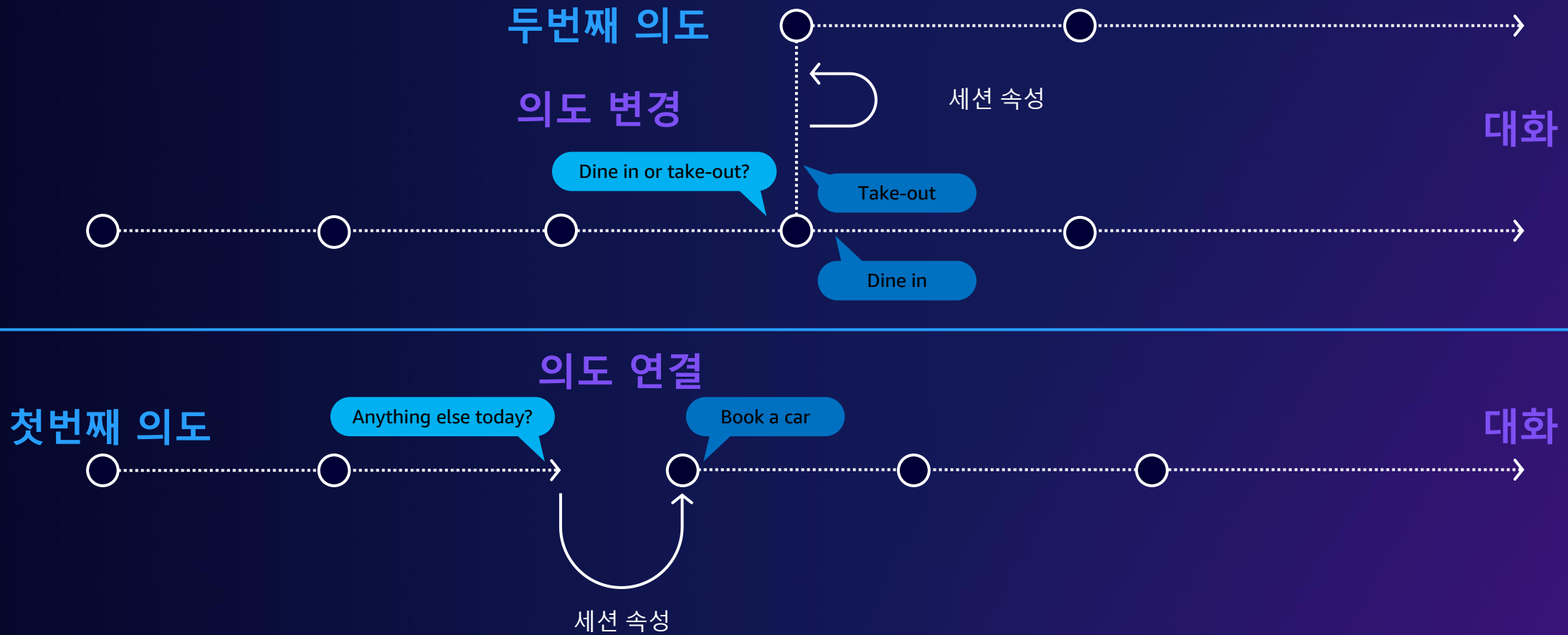
Intents

Prompts

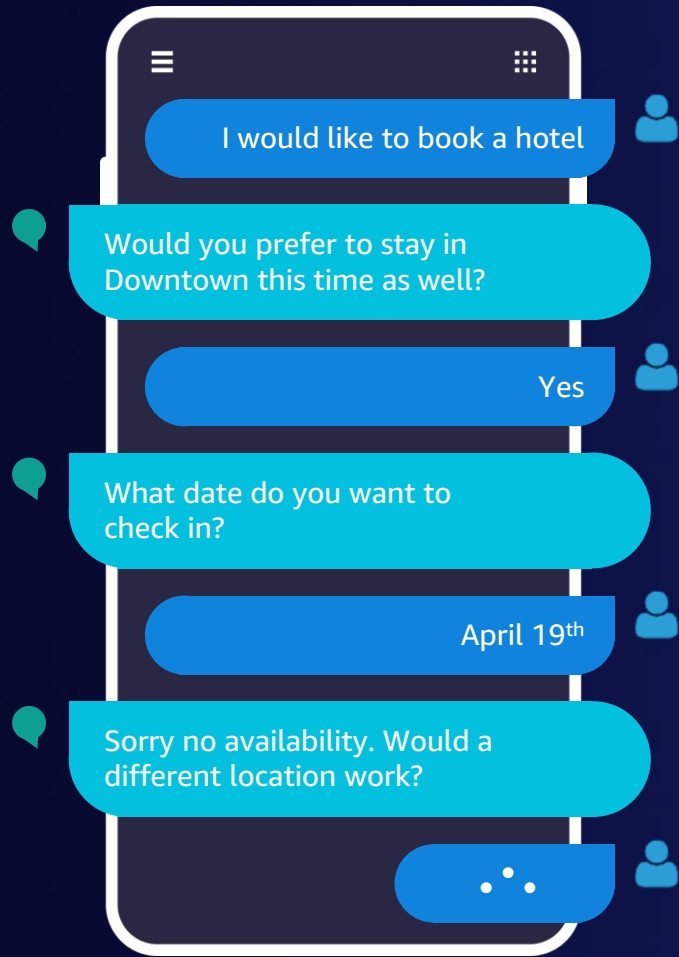
Session attributes

Confirmations

# 동적인 대화 흐름 (분기, 연결)



# 맞춤형 대화

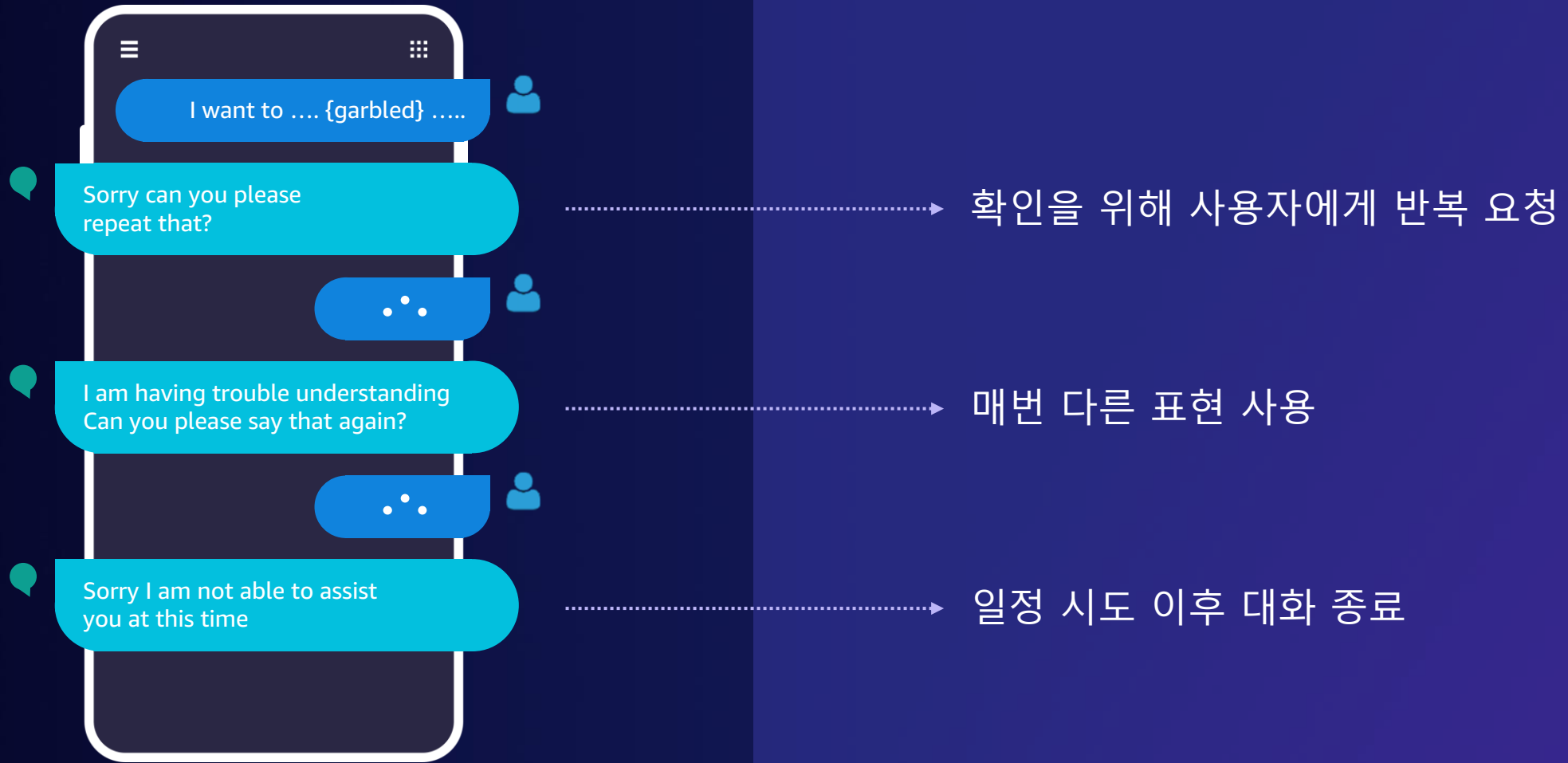


사용자 설정에 따라 맞춤형 대화를 제공

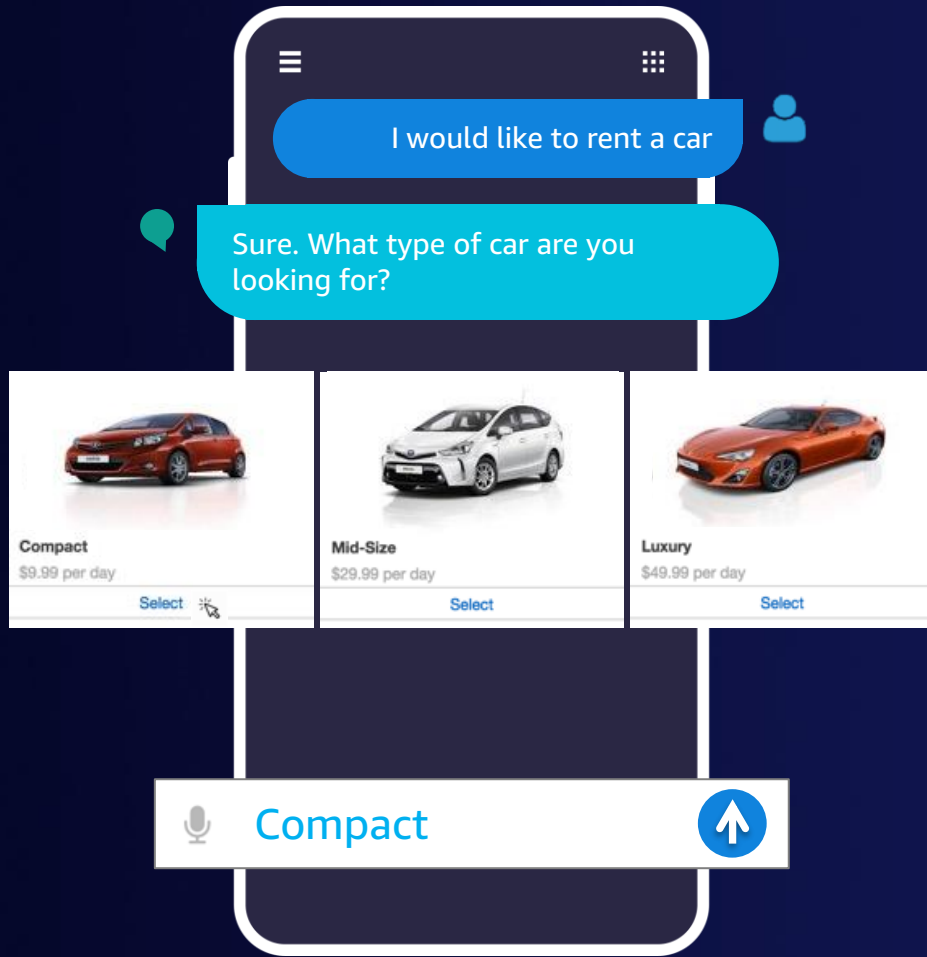


필요에 따라, 사용자 입력값을 검증하여 적절한 메시지 출력

# 에러 처리



# Rich 메시지 포맷



- 메시징 플랫폼에 알맞은 포맷을 제공
- 복수의 카드 지원
- 프리뷰 기능 제공
- 콘솔에서 테스트 가능



# 이행(Fulfillment)



## AWS Lambda 통합

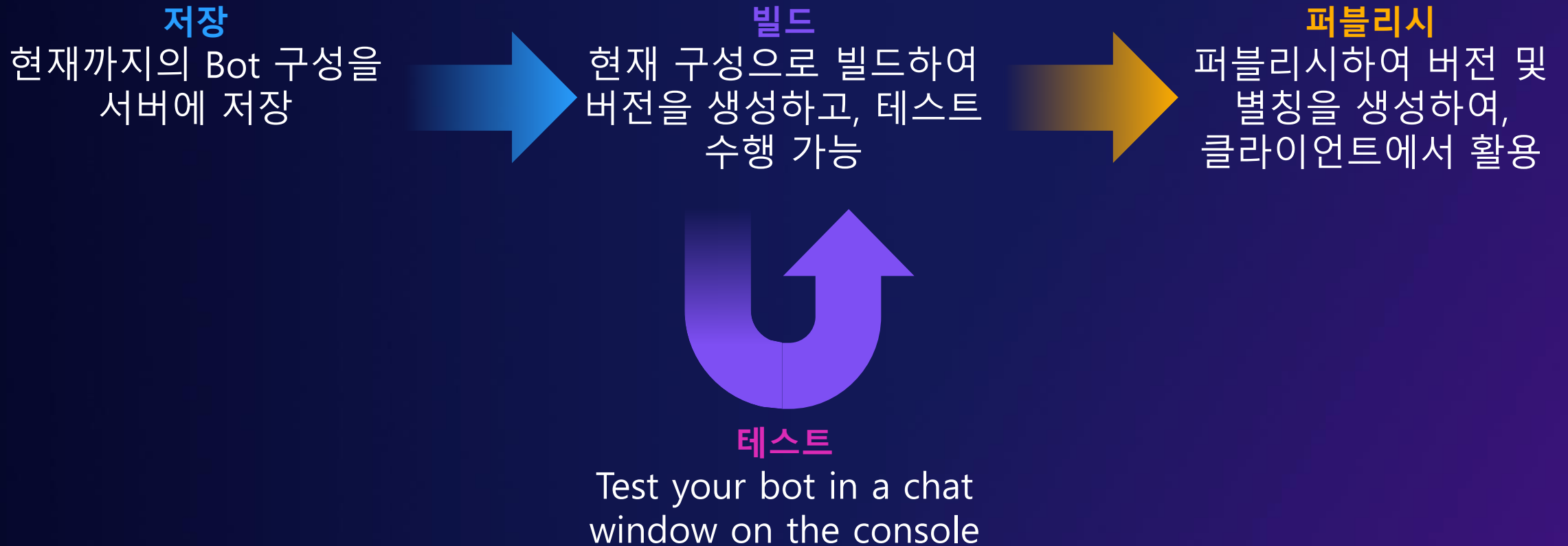
파악한 의도와 슬롯을 AWS Lambda 함수로 전달하여 비즈니스 로직을 수행합니다.



## 클라이언트로 전달

사용자의 입력에서 의도의 슬롯을 도출합니다.  
결과를 클라이언트로 전달하여 활용합니다.

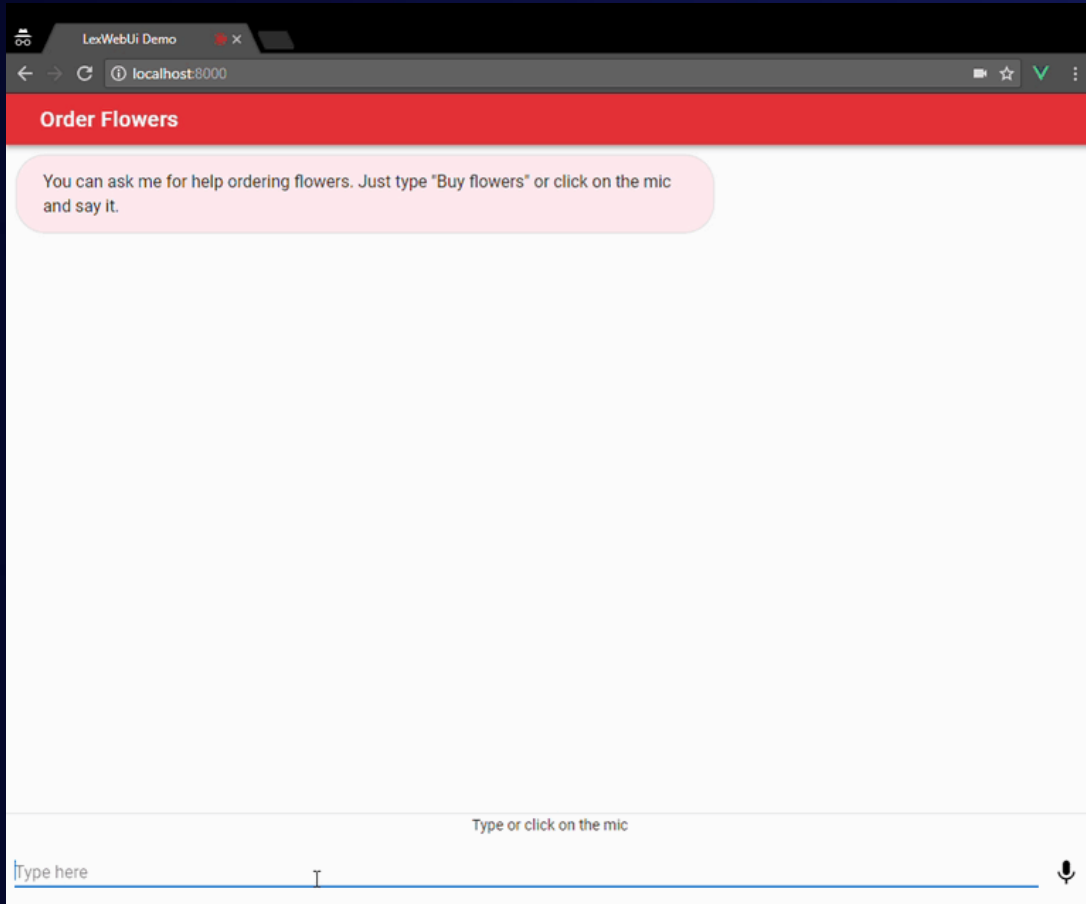
# 저장, 빌드, 퍼블리시



# 웹사이트에 Amazon Lex 통합하기

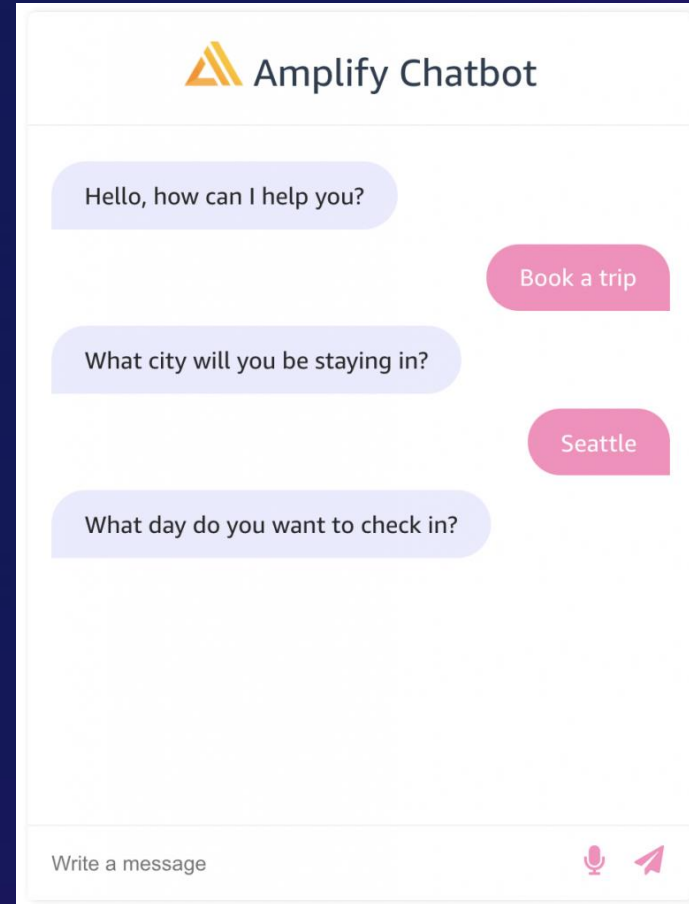
## Lex-Web-UI 활용

<https://github.com/aws-samples/aws-lex-web-ui>



## Amplify 활용

<https://ui.docs.amplify.aws/components/chatbot>



# 실전 Chatbot 구성하기

# 예제 소개 : Banking Bot 만들기

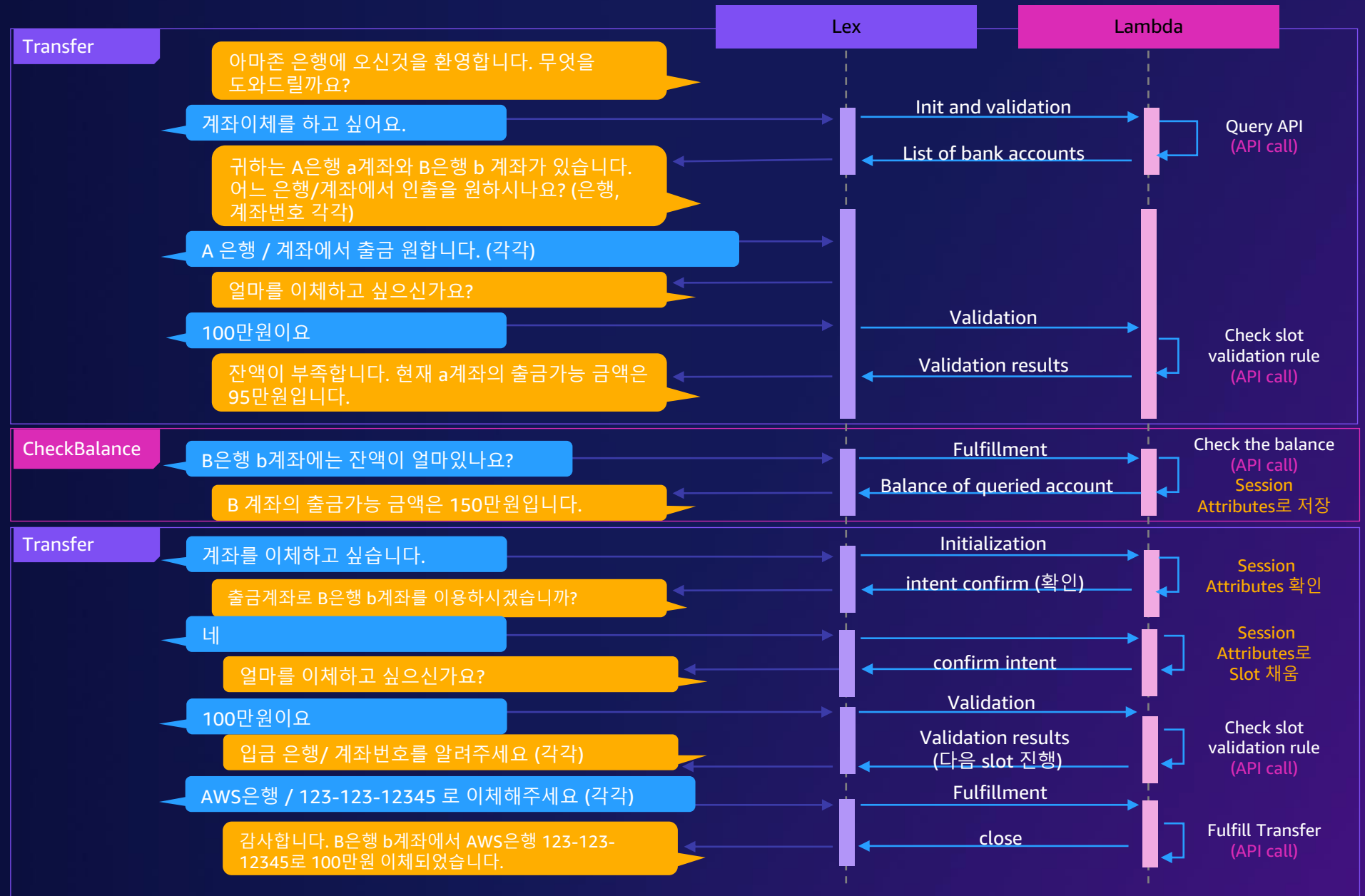
[HTTPS://GITHUB.COM/AWS-SAMPLES/AWS-AI-ML-WORKSHOP-KR/BLOB/MASTER/AISERVICES/LEX-KOREAN-WORKSHOP/BANKINGBOT.MD](https://github.com/aws-samples/aws-ai-ml-workshop-kr/blob/master/aishervices/lex-korean-workshop/bankingbot.md)

- 인터넷 뱅킹의 기본 기능을 구현한 한국어 Chatbot Workshop
- Chatbot 구성
  - **CheckBalance** 의도 : 계좌 조회
  - **Transfer** 의도 : 계좌 이체
- **세션 정보 전달 및 의도 전환** : 계좌 조회 후 이체를 진행하면 직전에 조회한 계좌번호를 이체에 사용할지 여부를 확인하고 활용하도록 하여 의도 전환을 구현합니다.
- **비즈니스 로직 통합** : Lambda 함수와 DynamoDB로 이체/ 계좌 조회, 잔고 조회 등이 구현되어 있습니다.

<input type="checkbox"/>	PK ⓘ ▲	SK ▼	Balance ▼
<input type="checkbox"/>	user-1	아마존은행#1234567890	1500000
<input type="checkbox"/>	user-1	한국은행#1111111111	1000
<input type="checkbox"/>	user-2	AWS은행#1234567890	10000

Table : BankingBot

# 대화 흐름



# Amazon Lex V2 구조

- 하나의 봇 내에서, 여러 언어를 지원하며, 각 언어는, 하나 이상의 의도를 포함합니다.

**Amazon Lex** X

Bots

BankingBot

Bot versions

Draft version

All languages

▼ Korean (KR)

Intents

Slot types

▼ Deployment

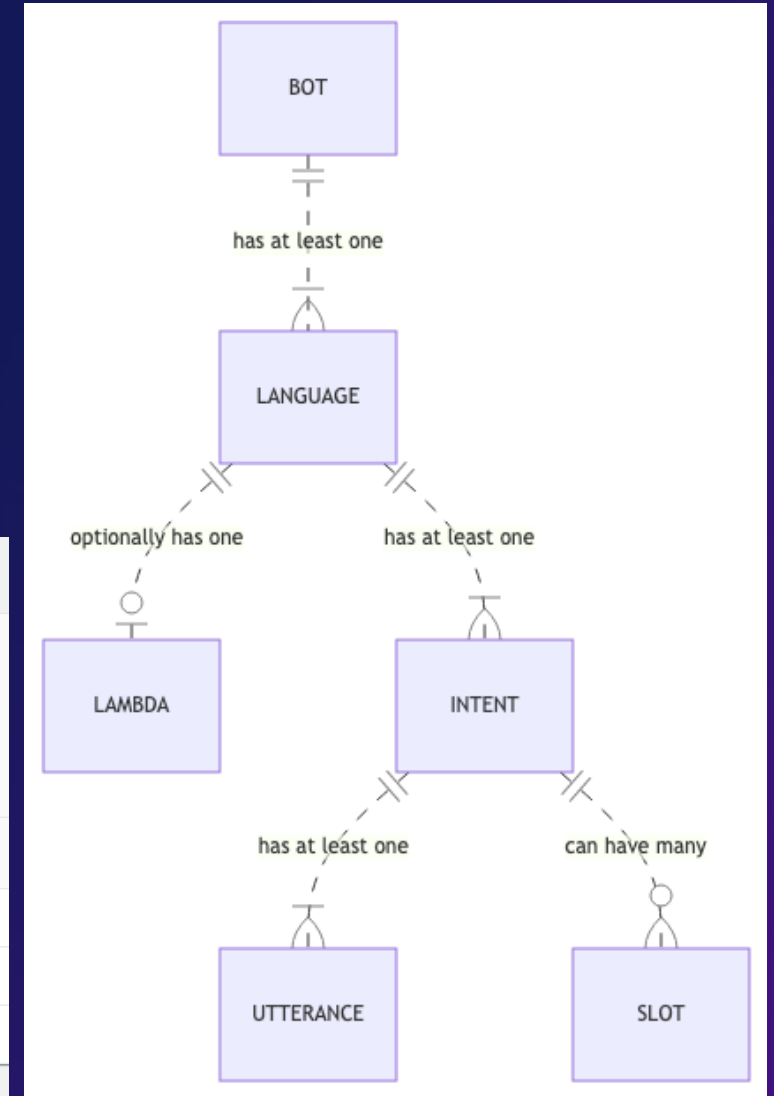
Aliases

Lex > Bots > Bot: BankingBot > Versions > Version: Draft > All languages > Language: Korean (KR) > Intents

Intents (3) Info

An intent represents an action that the user wants to perform.

	Name	Description
<input type="radio"/>	<a href="#">FallbackIntent</a>	-
<input type="radio"/>	<a href="#">Transfer</a>	-
<input type="radio"/>	<a href="#">CheckBalance</a>	-



# Amazon Lex V2 구조

- 하나의 언어의 모든 의도가 validation/initialization과 Fulfillment에 사용되는 **Lambda**를 하나로 공유하며,
- V1와 다르게 의도 별로 Lambda를 매핑하지 않습니다.
- Lambda Handler 내에서 의도 별로 분기하도록 구현해야 합니다.

Lex > Bots > Bot: BankingBot > Aliases > Alias: ProdBot > Alias language support: Korean (KR)

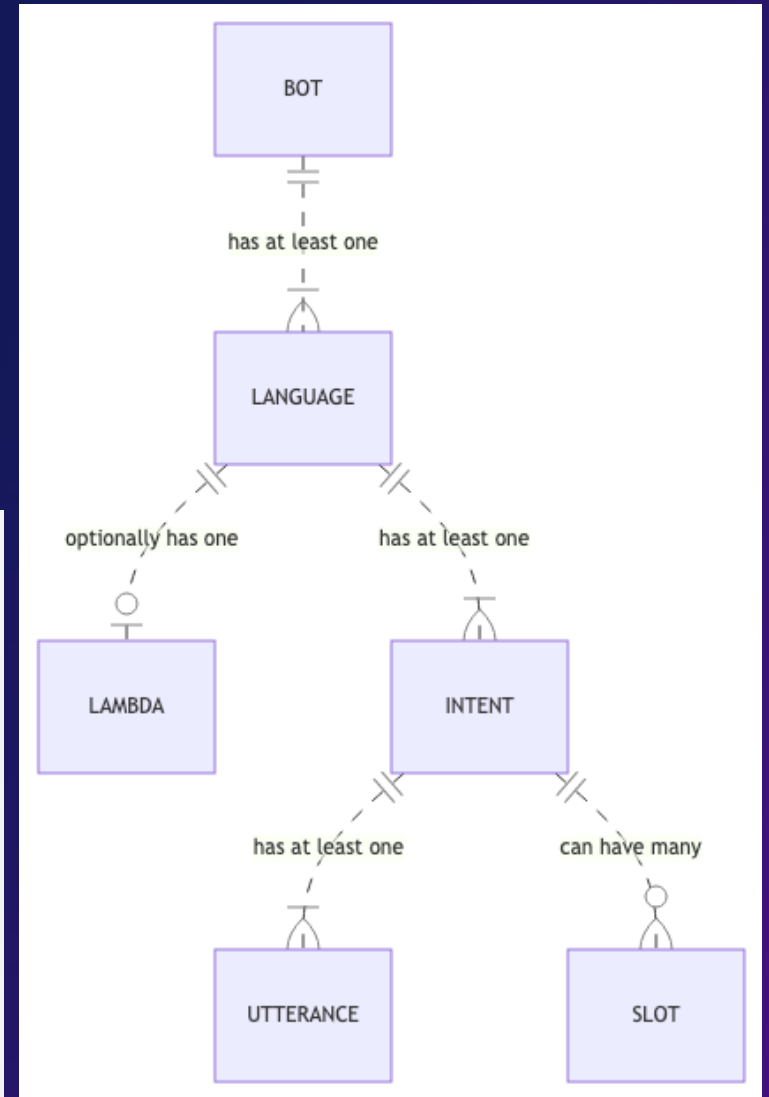
## Alias language support: Korean (KR)

▼ **Lambda function - optional**  
The Lambda function is invoked for initialization, validation, and fulfillment.

Source

Lambda function version or alias

[Learn more about Lambda](#)





# 비즈니스 로직 연결 - Lambda 통합

- 의도 내에서 Lambda가 호출되는 시점
  - **Initialization** : 의도가 파악되고 첫번째 슬롯을 위한 질문하기 전
  - **Validation** : 각 슬롯에 대한 응답이 왔을 때
  - **Fulfillment** : 모든 슬롯이 채워졌을 때 이며,
- 언어별 Lambda 연결 후, 아래와 같이 **각 의도별로, Checkbox를 활성화**해야만 트리거 됩니다.

## Initialization와 Validation

### ▼ Code hooks - optional [Info](#)

#### ☒ Use a Lambda function for initialization and validation

Lambda function is executed at every turn of the conversation. You can use this function to initialize values or validate user input.

## Fulfillment (Advanced options 내)

### Fulfillment Lambda code hook [Info](#)

You can enable Lambda functions to initialize the conversation, validate user input, and execute fulfillment.

#### ☒ Use a Lambda function for fulfillment

You can use AWS Lambda to fulfill your intent. The Lambda function is invoked after slot elicitation and confirmation. Use this function to fulfill your intent.

# 의도 전환 구현

- **Session Attributes**를 활용하여, 이전 의도의 결과를 전달하고, 이를 다른 의도에서 활용할 수 있습니다.
- 예제에서, 계좌 조회 후 이체를 진행하면 직전에 조회한 계좌번호를 이체에 사용할지 여부를 확인하고 활용하도록 구현되어있습니다.
- **CheckBalance** 의도의 Fulfillment 시점에 Session Attribute로 저장



# Demo



master aws-ai-ml-workshop-kr / aiservices / lex-korean-workshop / BankingBot.md Go to file ...

serithemage update BankingBot.md Latest commit b6a5051 Feb 15, 2022 History

1 contributor

243 lines (136 sloc) 11 KB

<> Raw Blame

# 뱅크 봇 만들기

## 개요

앞서서는 Amazon Lex 콘솔만을 사용하여 간단한 HelloWorldBot을 만들어 보았습니다. 이번에는 AWS Lambda를 이용해 인터넷 뱅킹을 수행하는 좀 더 복잡한 작업을 하는 봇을 구현해 보겠습니다. 이번 섹션에서는 AWS CloudFormation을 사용하여 봇의 생성 및 구성을 자동화 하는 방법도 함께 살펴 보겠습니다.

## CloudFormation

AWS CloudFormation은 AWS 리소스 모음을 손쉽게 생성할 수 있는 클라우드 배포 자동화 서비스입니다. CloudFormation을 사용하면 다음과 같은 이점이 있습니다.

- 인프라 관리 간소화: 리소스를 개별적으로 관리하는 대신 템플릿을 사용하여 필요할때마다 전체 스택을 한번에 생성, 업데이트, 삭제 할 수 있습니다.
- 신속한 인프라 복제: 애플리케이션의 가용성을 확보하기 위해서나 다양한 테스트를 위해서 환경을 복제하거나 여러 리전에 배포해야 하는경우 템플릿을 이용해 일관되고 반복적으로 리소스를 생성할 수 있습니다.
- 손쉬운 인프라 변경 사항 제어 및 추적: CloudFormation 템플릿을 AWS CodeCommit과 같은 버전관리 시스템을 이용해 관리할 수 있습니다. 이를 통해 과거로부터의 이력관리나 코드리뷰와 같은 작업들을 인프라에 대해서도 수행할 수 있습니다. 만약 업데이트한 새로운 버전의 환경에 문제가 있을경우 이전버전으로 되돌리는 작업도 간단하게 수행할 수 있습니다.

여러분의 소중한 피드백을 기다립니다.  
강연 종료 후, 강연 평가에 참여해 주세요!

# 감사합니다

