Enhance customer experience with Conversational Interfaces

Tara E. Walker

Sr. Technical Evangelist

Amazon Web Services
@taraw

Agenda

✓ The What & Why of Conversational Interfaces

✓ Ins and Outs of Amazon Lex

✓ Putting Conversational Interfaces to Work

aws SUMMIT

What & Why of Conversational Interfaces

"What we need now is to be able to simply talk with our devices. That's why I believe it's finally time for the conversational user interface, or "CUI."

This is the interface of the future, made even more necessary as computing propagates beyond laptops, tablets and smartphones to cars, thermostats, home appliances,and now even watches … and glasses."

~ Ron Kaplan (via WIRED magazine), Lead-Nuance Communications' NLU R&D Lab, Professor of Linguistics at Stanford University, former CTO of Powerset

aws SUMMIT

Why Conversational Interface Access

Natural

On-demand

Accessible

Efficient

Developer challenges

Security

Speech recognition

Authentication

Language understanding

Messaging platforms

Disparate systems

Availability

Business logic

Scale

Testing

Mobile

Conversational interfaces need to combine a large number of sophisticated algorithms and technologies

aws SUMMIT

Considerations when building Conversation Interfaces

- Understand the conversation flow
- What information do you need from the conversation
- What is the intent/goal of the conversation
- Validate your input
- Handle errors gracefully
- Add clarification prompts if required
- Test, test and more testing
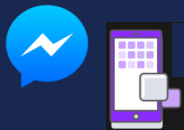- Understand Metrics after your bot or conversational interface is deployed in production

aws SUMMIT

"Alexa, What Is Amazon Lex?"

Amazon Lex - Overview

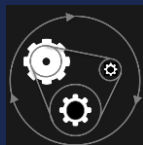Text and speech language understanding: powered by the same technology as Alexa

Build once and deploy to multiple platforms

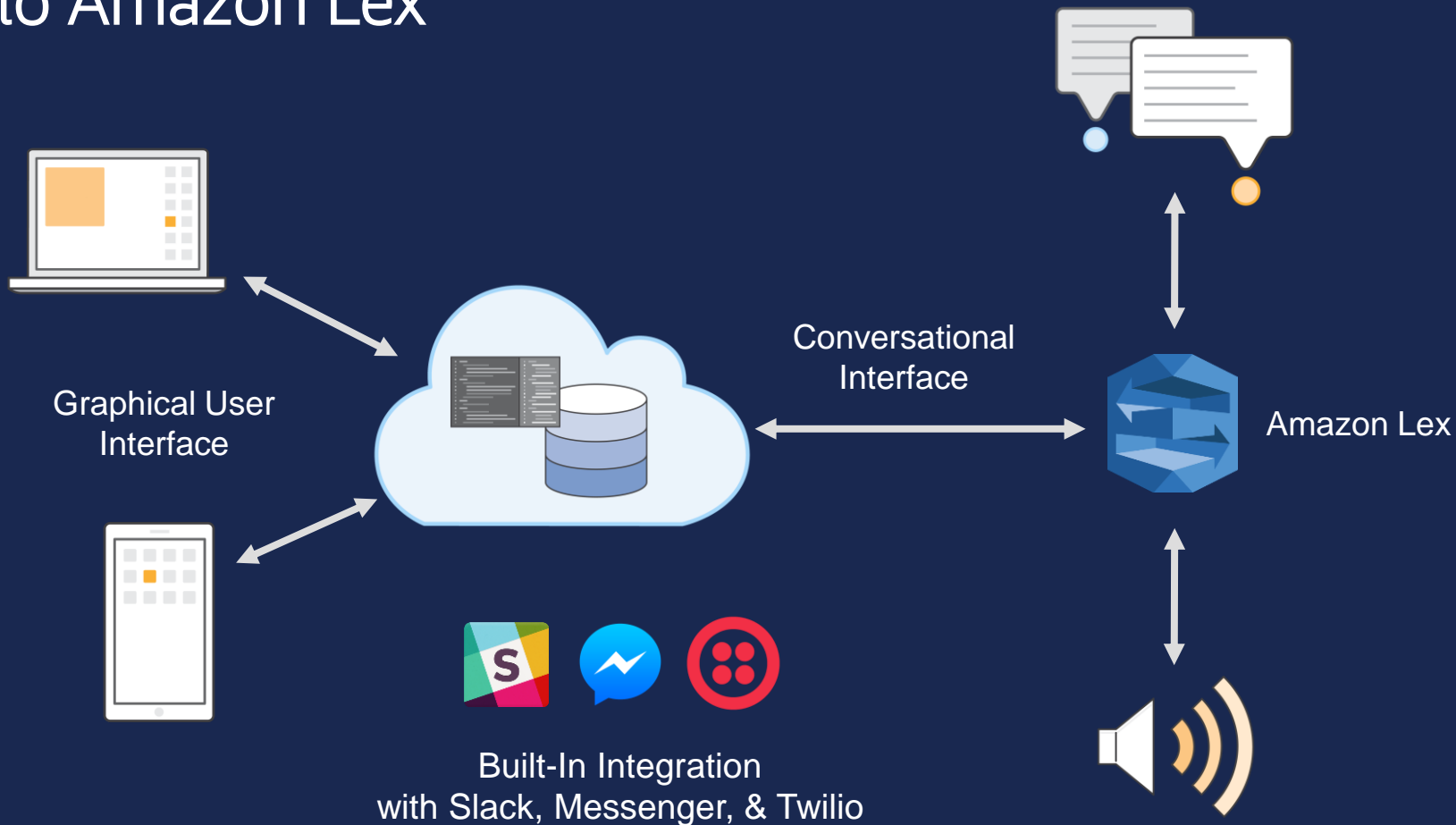Designed for builders: efficient and intuitive tools to build conversations; Scales automatically

Enterprise Ready: connect to enterprise systems via SaaS connectors; Versioning and alias support

Continuous Learning: monitor and improve your bot

aws SUMMIT

Hello Amazon Lex

Graphical User Interface

Conversational Interface

Amazon Lex

Built-In Integration
with Slack, Messenger, & Twilio

aws SUMMIT

Text and speech language understanding

Speech
recognition

Natural language
understanding

Powered by the same deep learning technology as Alexa

aws SUMMIT

Amazon Lex – Multi-platform

Mobile

Messaging Platforms

Web

IoT

SDKs: iOS & Android Mobile Hub

Facebook, Twilio SMS and Slack

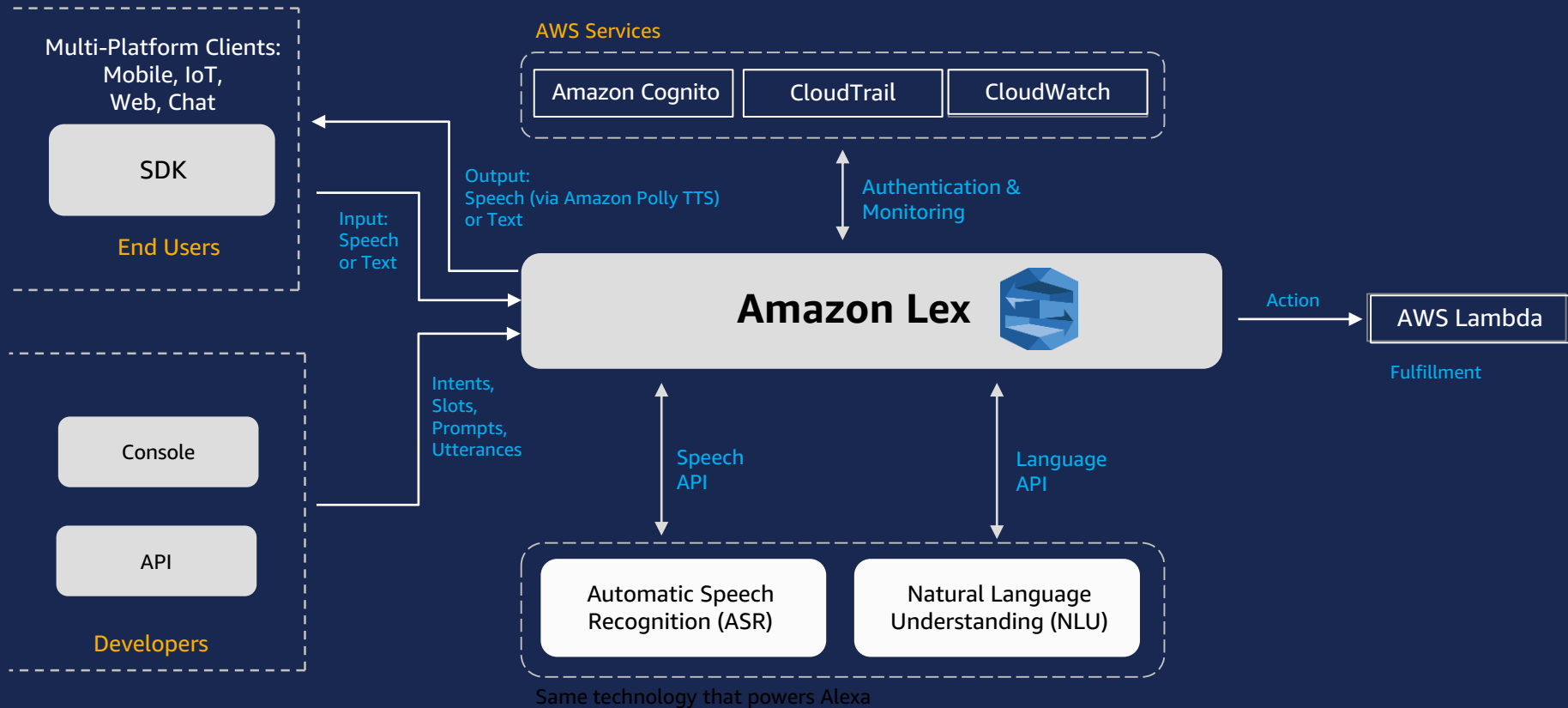**SDKs: Java, JavaScript, Python, CLI, .NET, Ruby on Rails, PHP, Go**
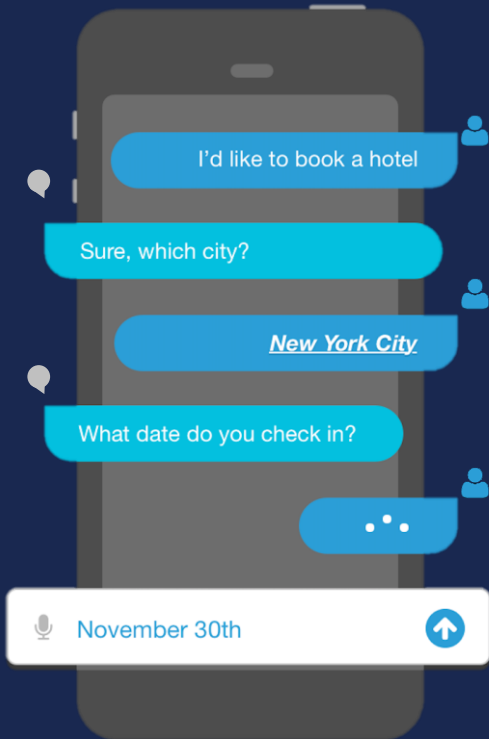
Integrated with AWS IoT via AWS Lambda

Build once and deploy to multiple platforms

Amazon Lex – Technology

Multi-Platform Clients:
Mobile, IoT,
Web, Chat

SDK

End Users

AWS Services

| Amazon Cognito | CloudTrail | CloudWatch |

Output:
Speech (via Amazon Polly TTS)
or Text

Input:
Speech
or Text

Authentication &
Monitoring

Amazon Lex

Action → AWS Lambda

Fulfillment

Console

API

Developers

Intents,
Slots,
Prompts,
Utterances

Speech
API

Language
API

Automatic Speech
Recognition (ASR)

Natural Language
Understanding (NLU)

Same technology that powers Alexa

Amazon Lex – Key Concepts

BookHotel

I'd like to book a hotel

Sure, which city?

New York City

What date do you check in?

• • •

November 30th

Intents
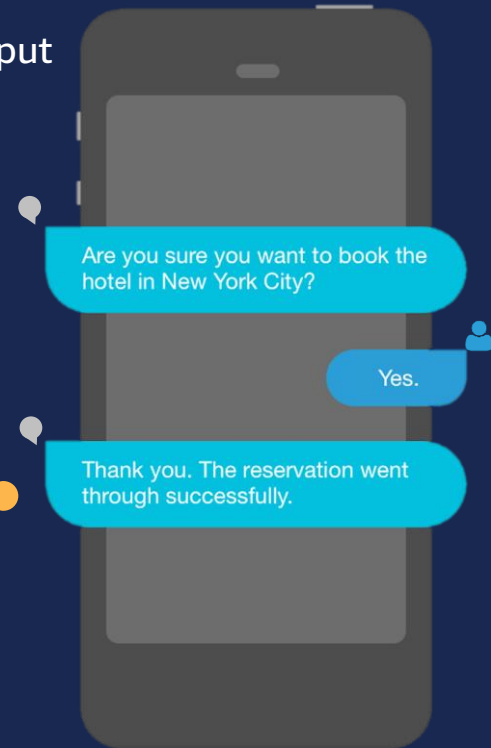An intent performs an action in response to natural language user input

Utterances
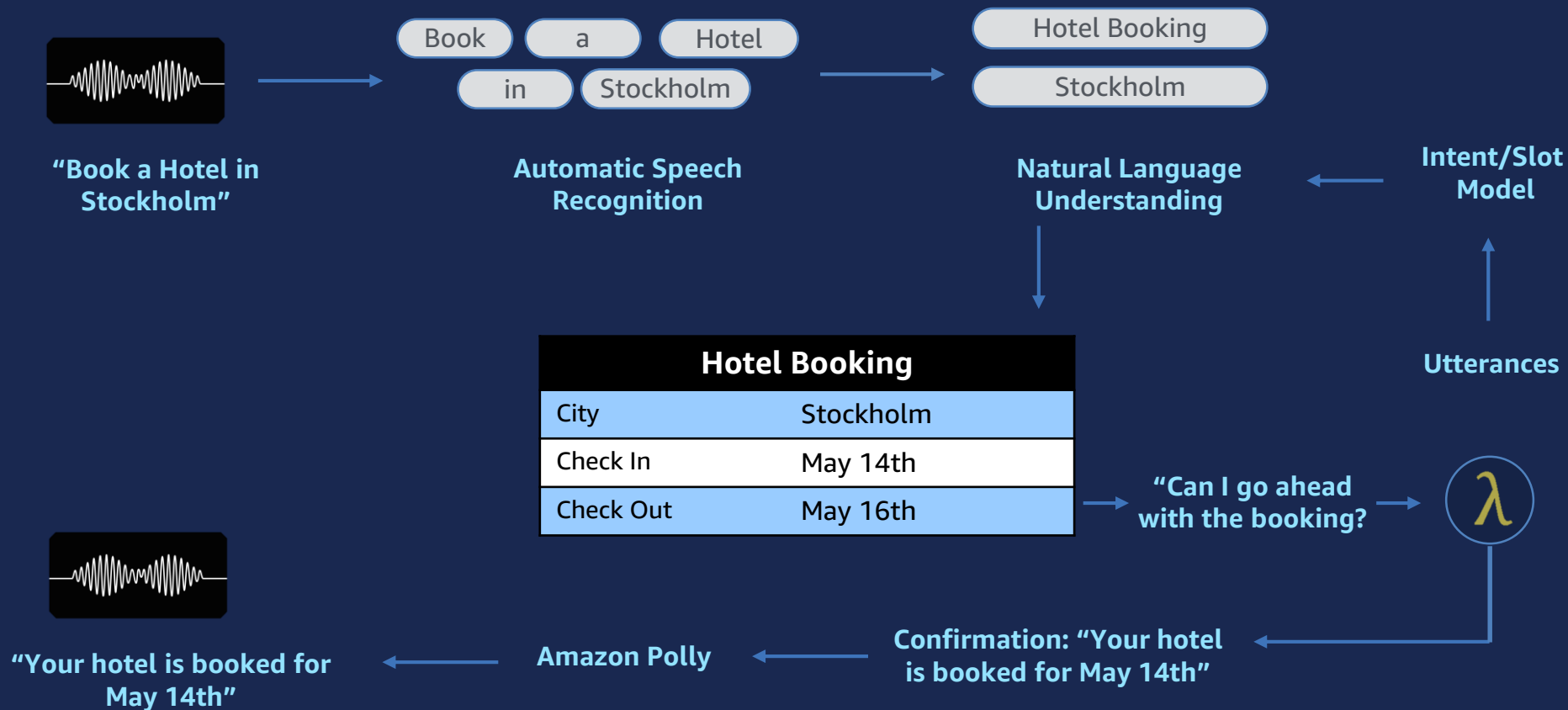Spoken or typed phrases that invoke your intent

Slots
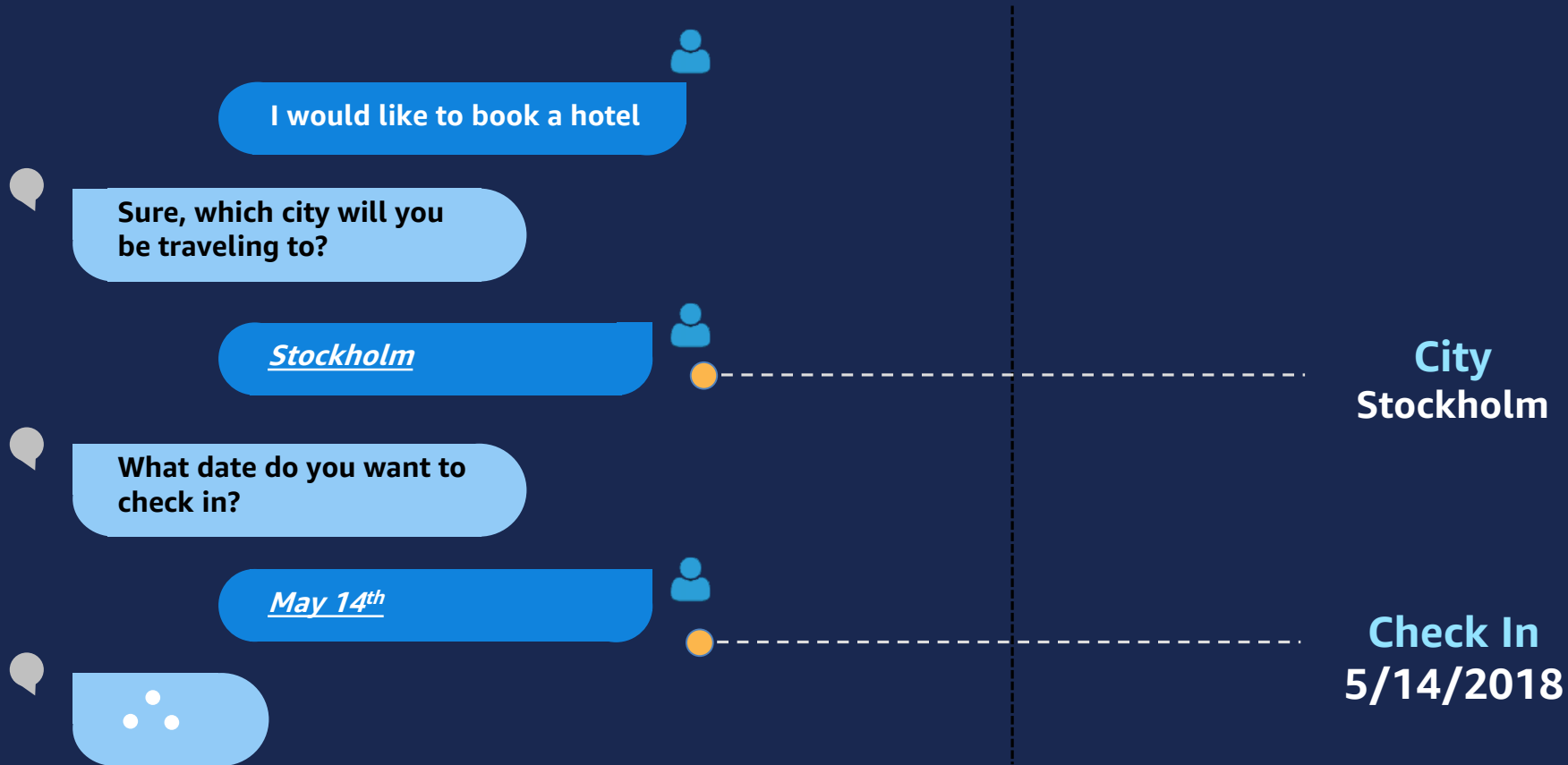Slots are input data required to fulfill the intent

Fulfillment
Fulfillment mechanism for your intent
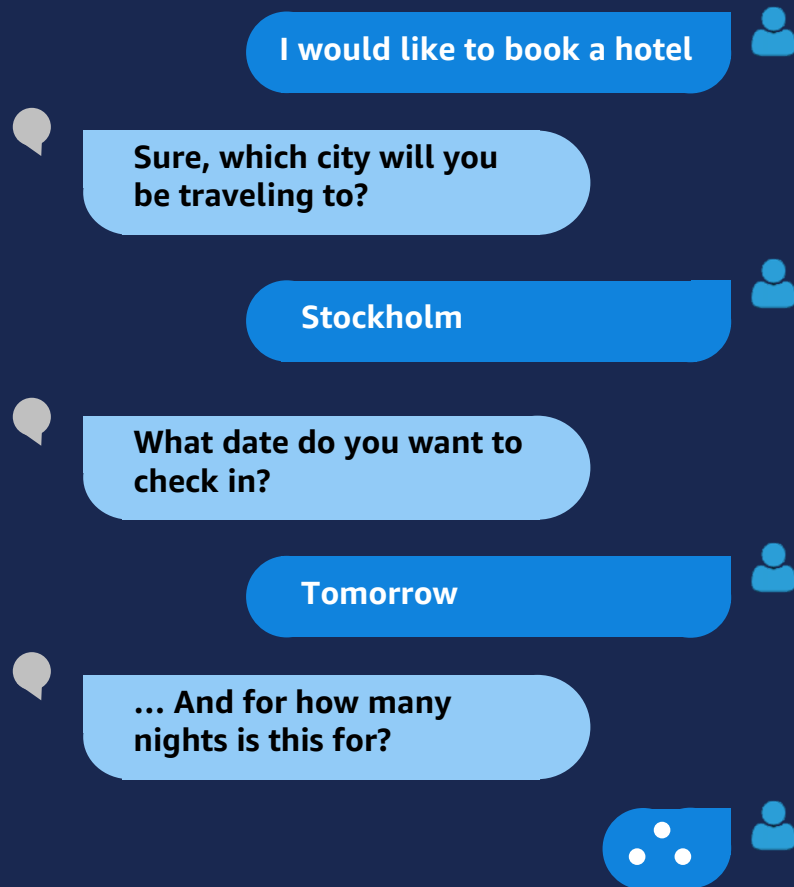
Are you sure you want to book the hotel in New York City?

Yes.

Thank you. The reservation went through successfully.

Amazon Lex – Example: "Book a hotel"

Book a Hotel in Stockholm

Hotel Booking

Stockholm

"Book a Hotel in Stockholm"

Automatic Speech Recognition

Natural Language Understanding

Intent/Slot Model

Utterances

Hotel Booking	
City	Stockholm
Check In	May 14th
Check Out	May 16th

"Can I go ahead with the booking?"

λ

Confirmation: "Your hotel is booked for May 14th"

Amazon Polly

"Your hotel is booked for May 14th"

Amazon Lex – Slot elicitation

I would like to book a hotel

Sure, which city will you be traveling to?

Stockholm

City
Stockholm

What date do you want to check in?

May 14th

Check In
5/14/2018

Amazon Lex – Dialog management

I would like to book a hotel

Sure, which city will you be traveling to?

Stockholm

What date do you want to check in?

Tomorrow

… And for how many nights is this for?

Simple Declarative Model

Slots	Prompts
City	Which city will you be traveling to?
Check-In Date	What date do you want to check in?
Check-Out Date	How many nights is this for?

Build Multi-turn Conversations

Easy Setup in Console

Amazon Lex – Customize conversations

I would like to book a hotel

Would you prefer to stay in Downtown this time as well?

Yes

What date do you want to check in?

May 14th

Sorry no availability. Would a different location work?

λ

Personalize conversation based on user preferences

Validate user input and re-prompt as necessary
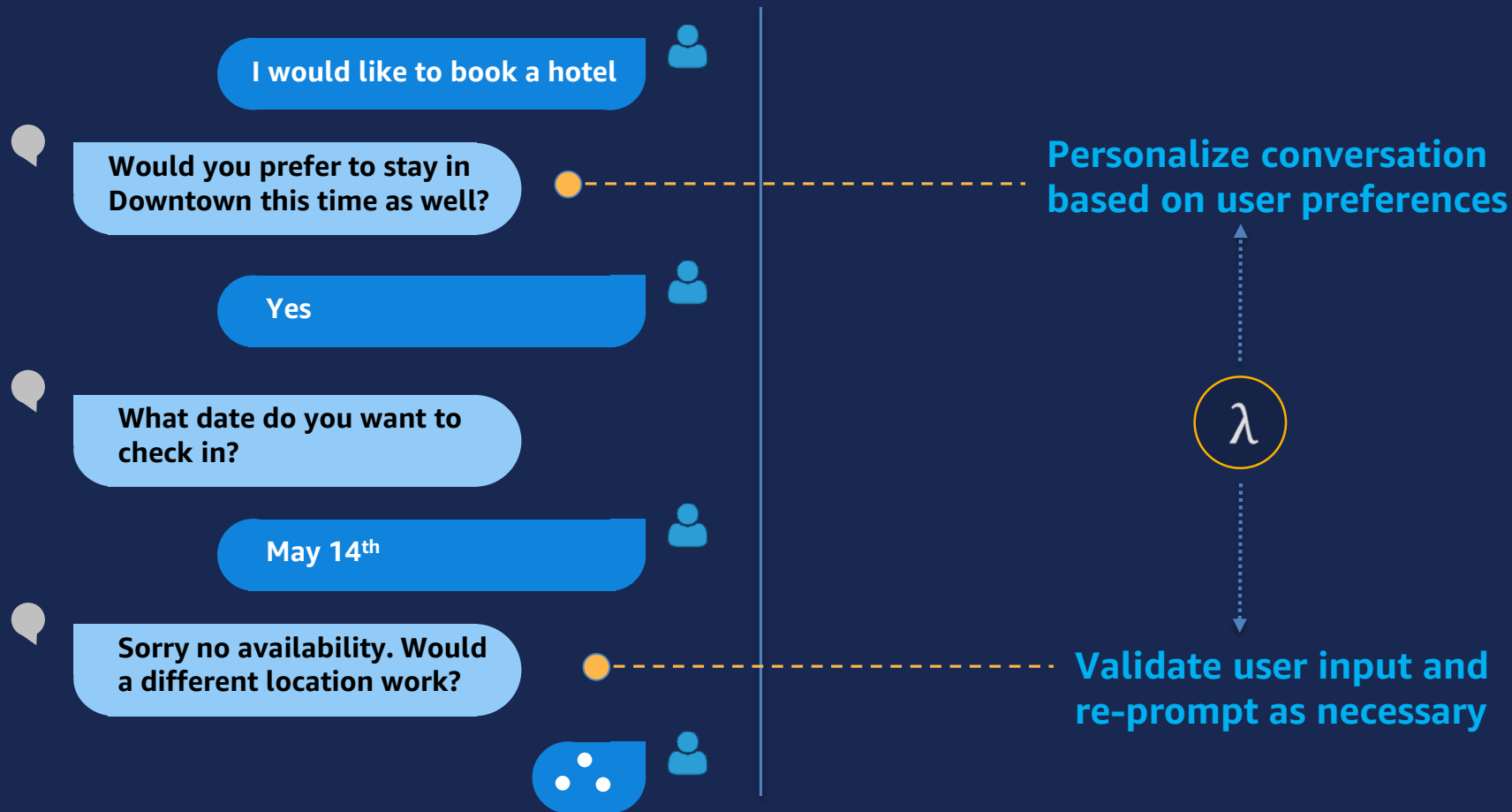
Amazon Lex – Error handling

I want to …. {garbled} …..

Sorry can you please repeat that?

Clarify by requesting user to repeat

I am having trouble understanding Can you please say that again?

Uses a different prompt every time

Sorry I am not able to assist you at this time

Hang up phrase to end the conversation
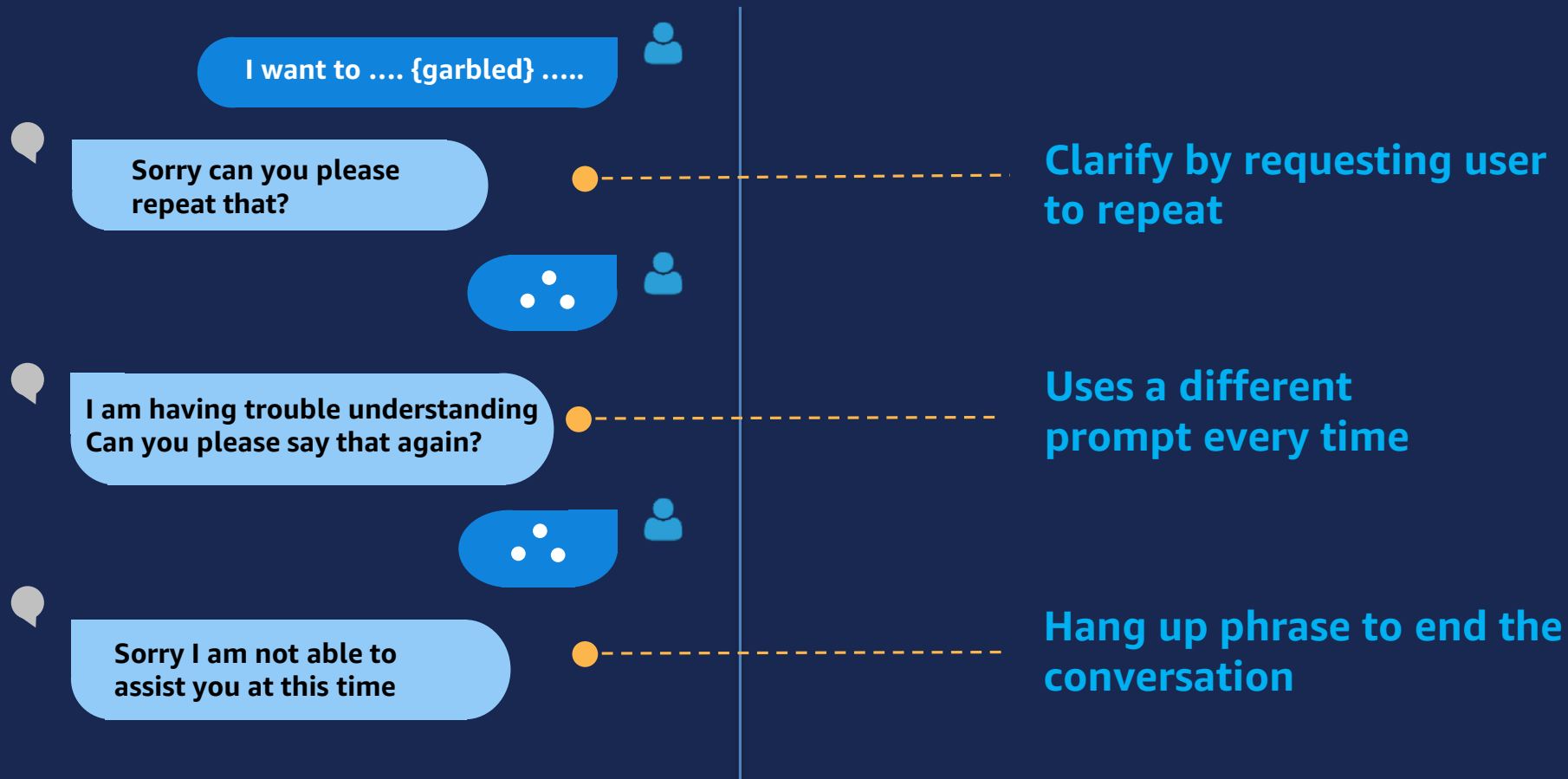
Amazon Lex – Conversation context

Intent

Conversation

Prompt

Session
Attributes

Confirm

Slot Value

Slot Value

Yes/No
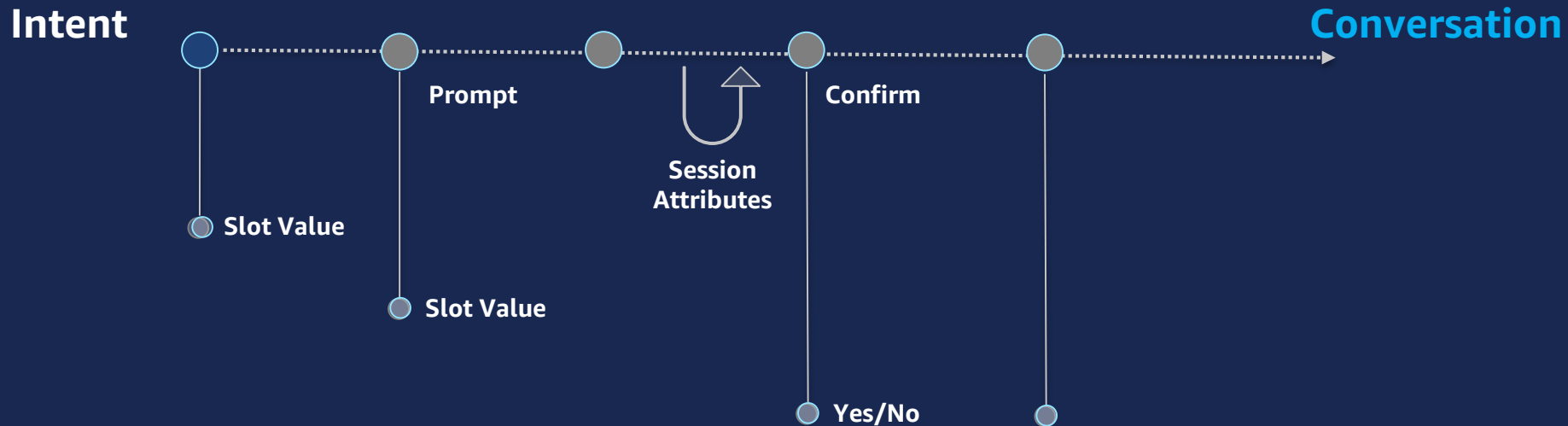
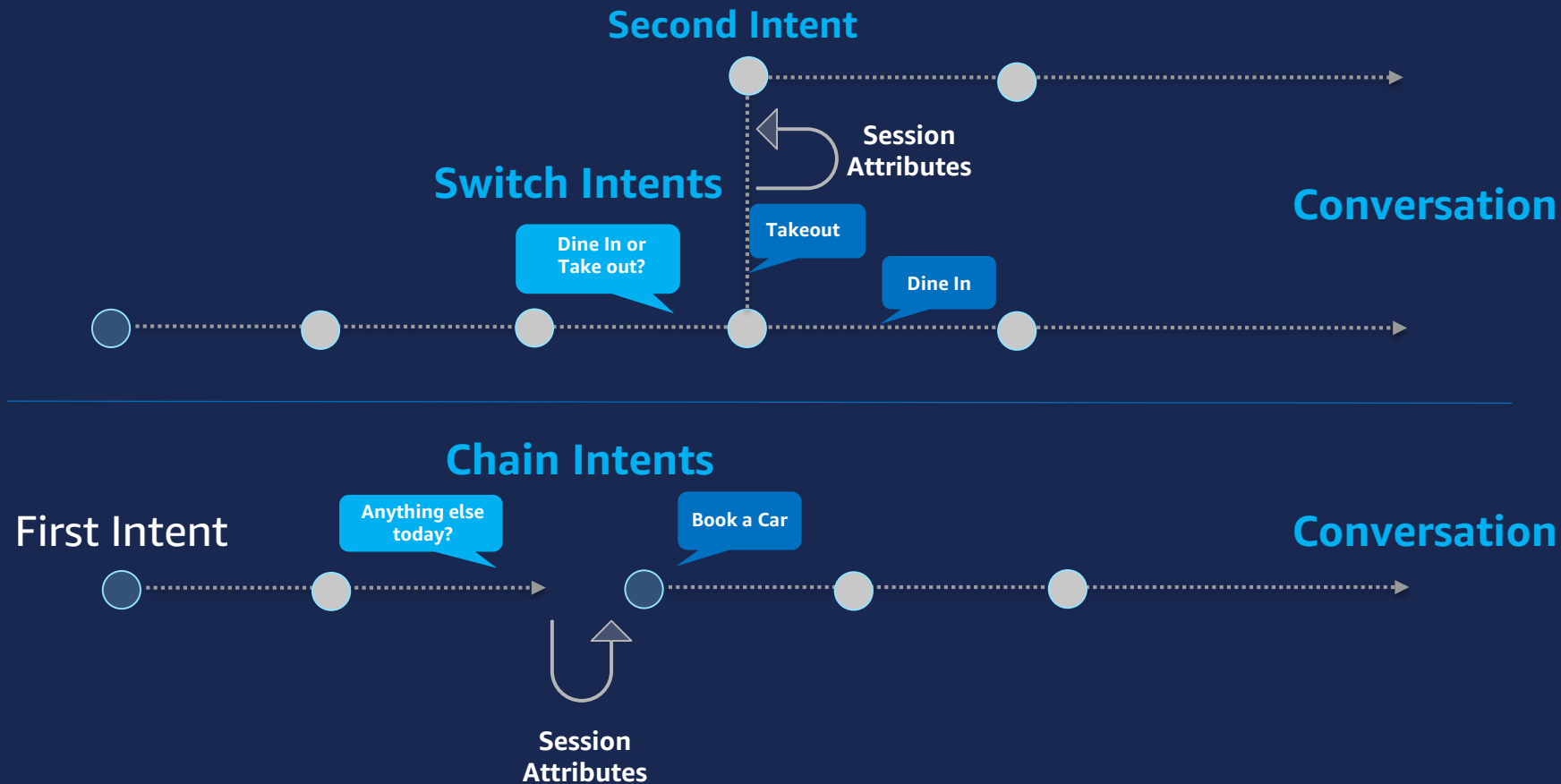**Lex maintains context by storing data throughout the conversation**

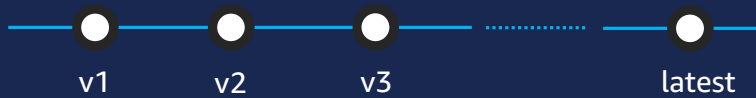Intents | Slot Values | Prompts | Session Attributes | Confirmations
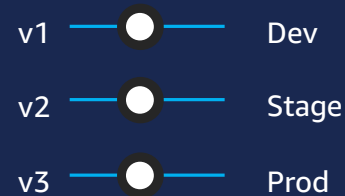
Amazon Lex – Dynamic conversation flow

Amazon Lex – Versioning and Alias support

v1   v2   v3   latest

v1 — Dev
v2 — Stage
v3 — Prod

- Supported for intents, slots, and bots
- Enables multideveloper environment
- Rollback to previous versions

- Deploy different aliases to different platforms
- Run different stacks for dev, stage and prod environments
- Target different user groups with different aliases

Versioning

Alias

Amazon Lex – Fulfillment & Response

Intents and slots passed to
AWS Lambda function for
business logic
implementation.

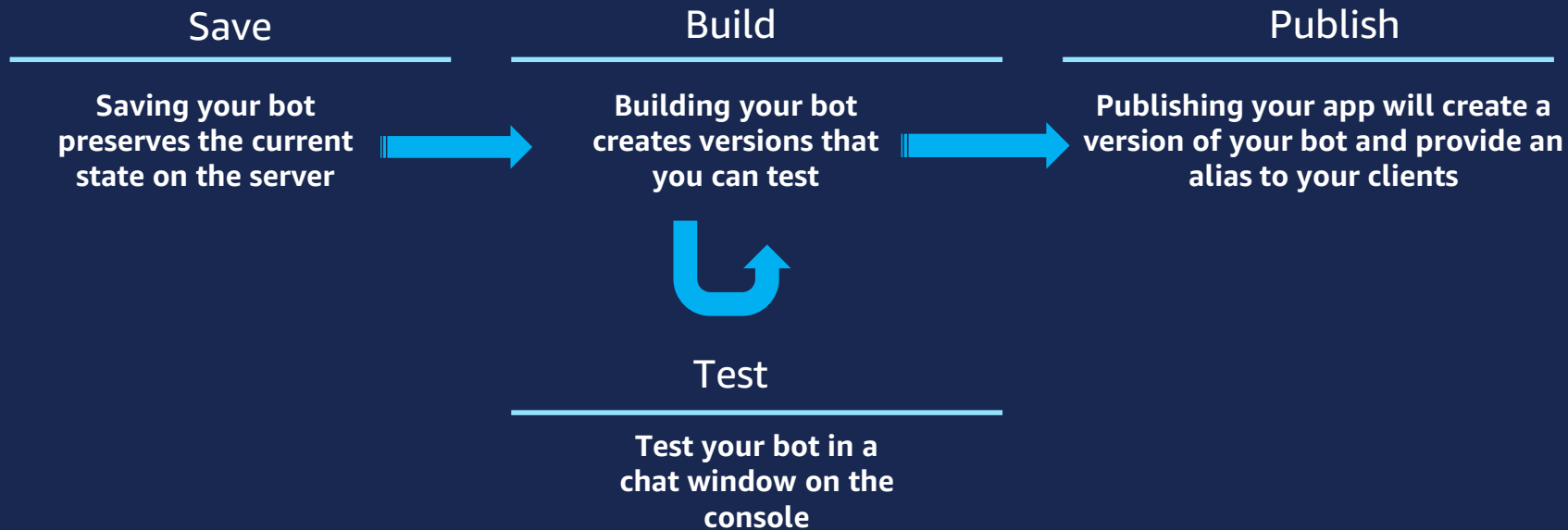User input parsed to derive
intents and slot values.
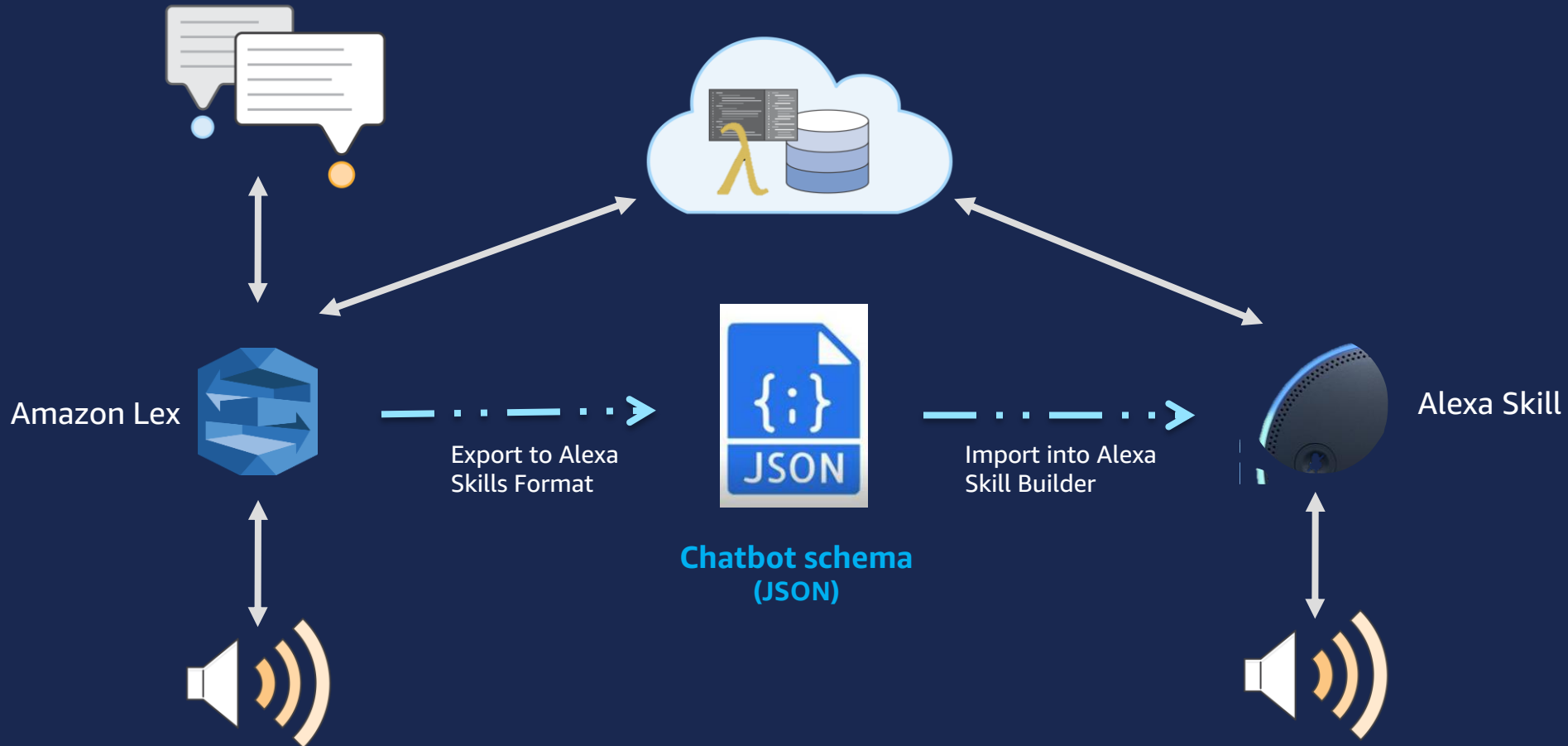Output returned to client for
further processing.

AWS Lambda integration

Return to Client

Amazon Lex – Deployment Cycle

Save

Saving your bot preserves the current state on the server

Build

Building your bot creates versions that you can test

Publish

Publishing your app will create a version of your bot and provide an alias to your clients

Test

Test your bot in a chat window on the console

Amazon Lex – Export to Alexa Skill

Amazon Lex

Export to Alexa
Skills Format

**Chatbot schema
(JSON)**

Import into Alexa
Skill Builder

Alexa Skill

Amazon Lex

Demo