

# Build Your Own Chatbot

Of course, using Java!

Arun Gupta @arungupta

Amazon Web Services

Docker Captain  
Java Champion  
JavaOne Rock Star (4 years)  
NetBeans Dream Team  
Silicon Valley JUG Leader  
Author  
Runner  
Lifelong learner



# Tell me a Yoda quote

# Developer challenges

Automated Speech  
Recognition

Scalability

Business Logic

Dialog Manager

Authentication

Availability

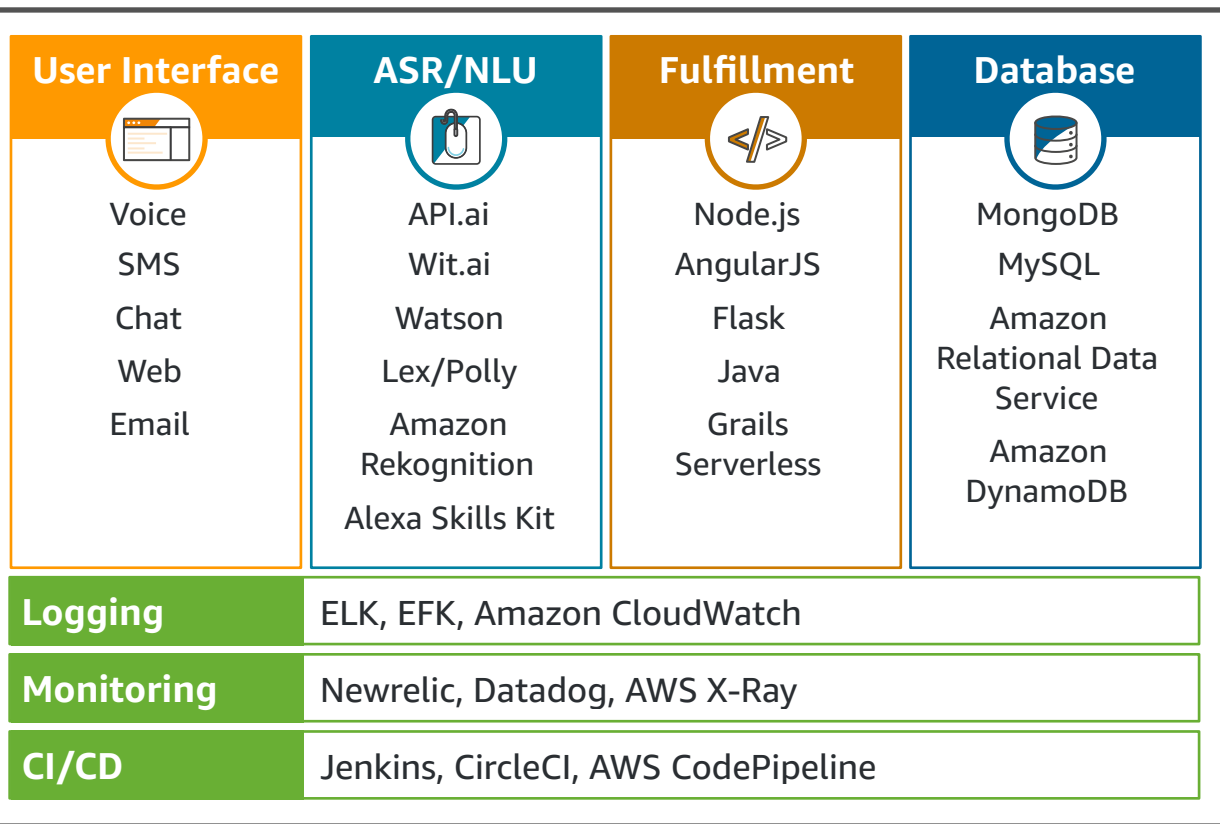
Testing

Natural Language  
Understanding

Mobile

Disparate Systems

# CHATBOT Components



## BookHotel

### Intents

A particular goal that the user wants to achieve

### Utterances

Spoken or typed phrases that invoke your intent

### Slots

Data the user must provide to fulfill the intent

### Prompts

Questions that ask the user to input data

### Fulfillment

The business logic required to fulfill the user's intent

I'd like to book a hotel.

Sure, which city?

New York City

What date do you check in?

...



November 30th.

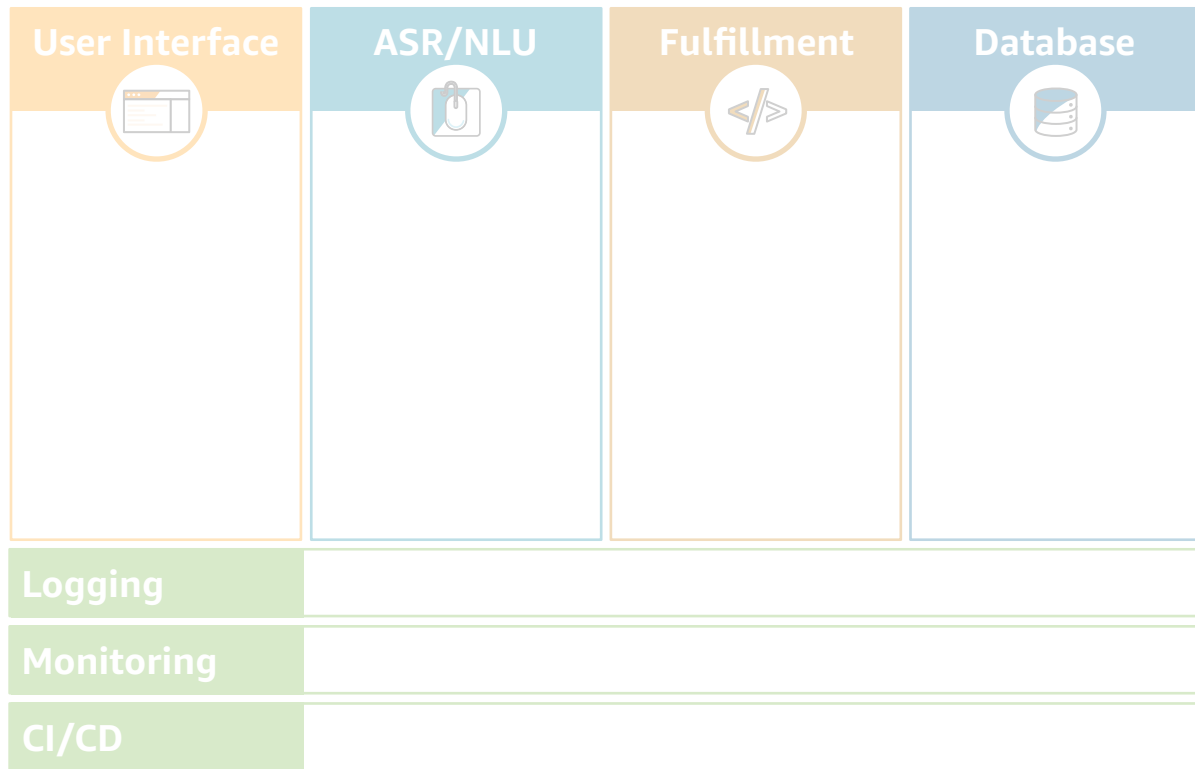


Are you sure you want to book the hotel in New York City?

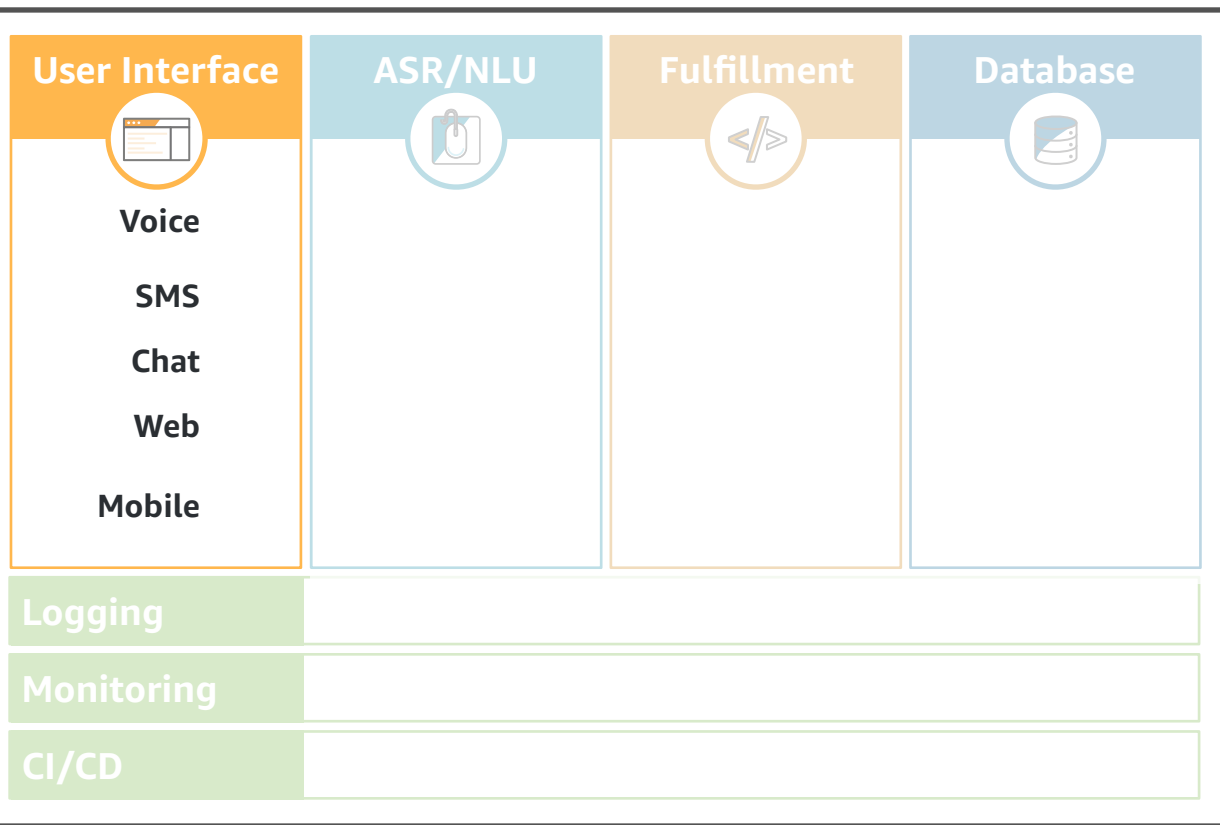
Yes.

Thank you. The reservation went through successfully.

# CHATBOT Architecture

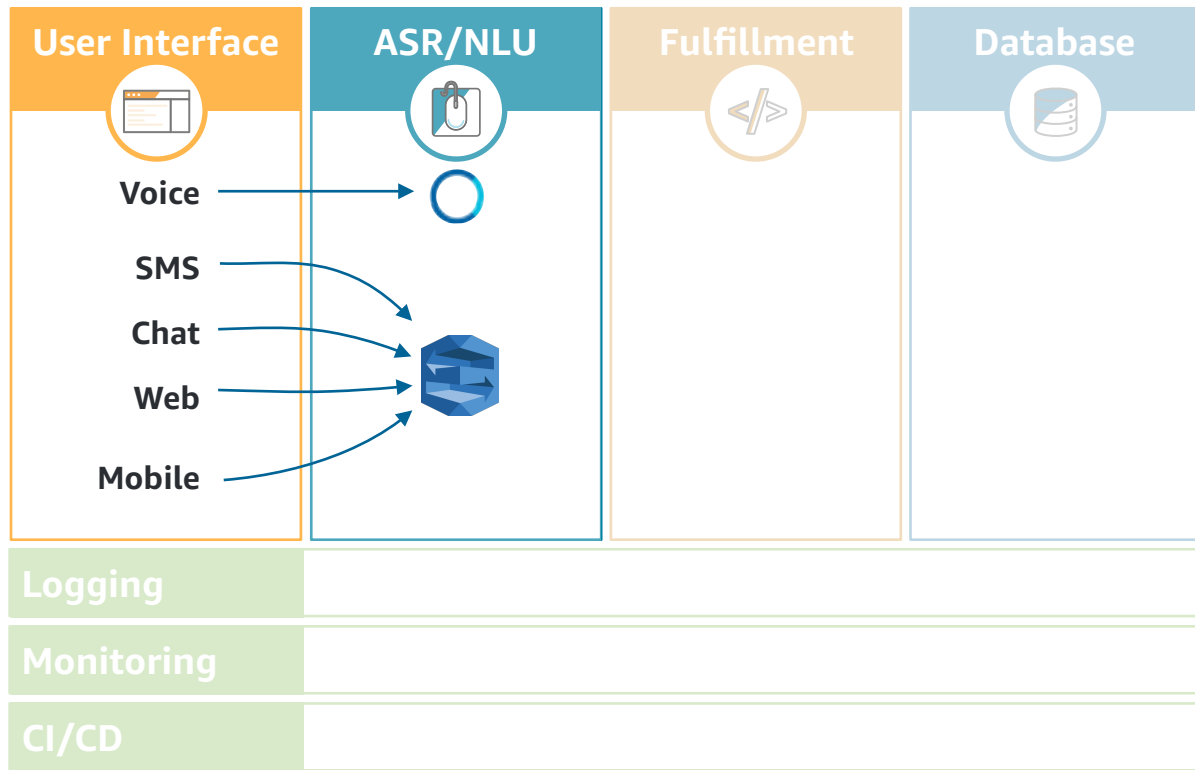


# CHATBOT Architecture

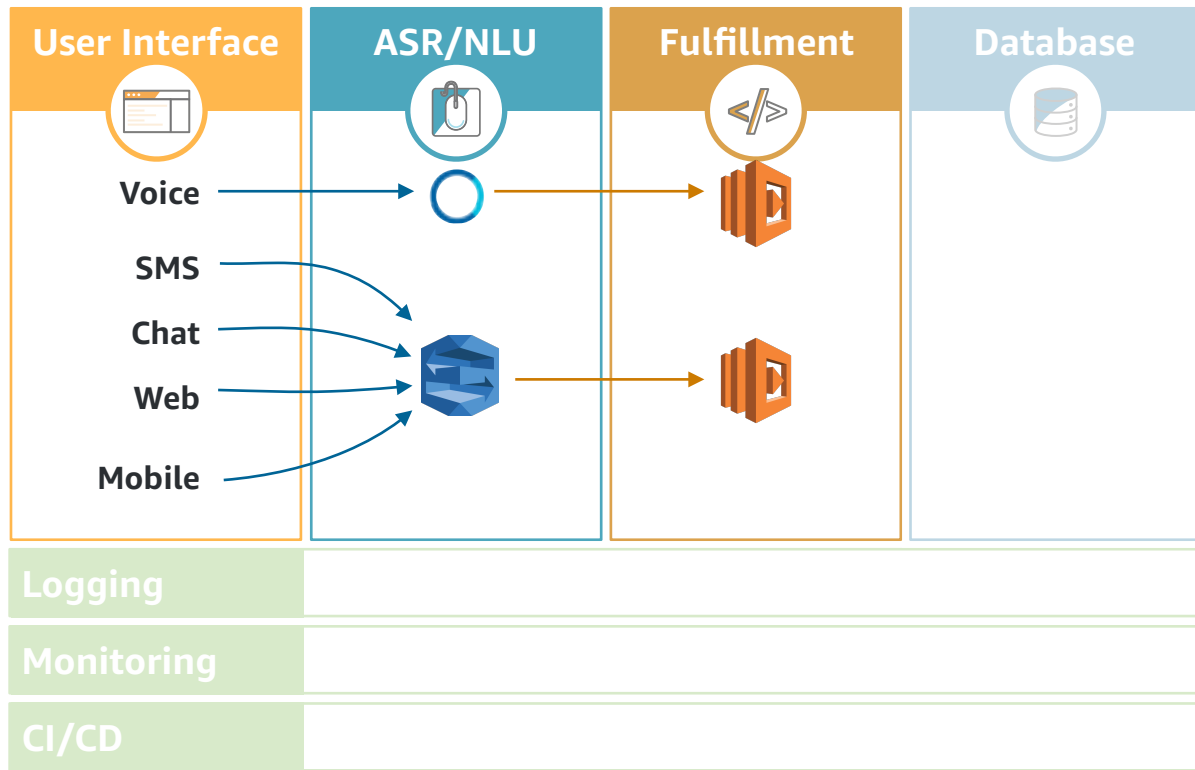




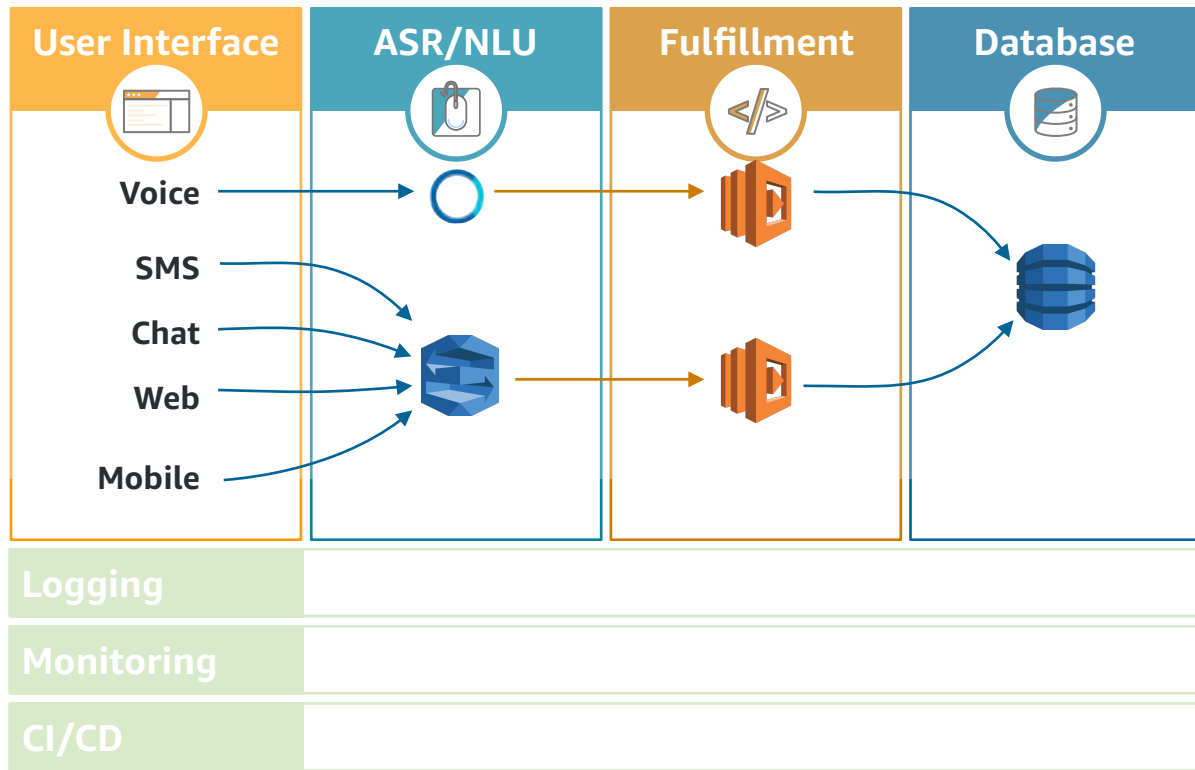
# CHATBOT Architecture



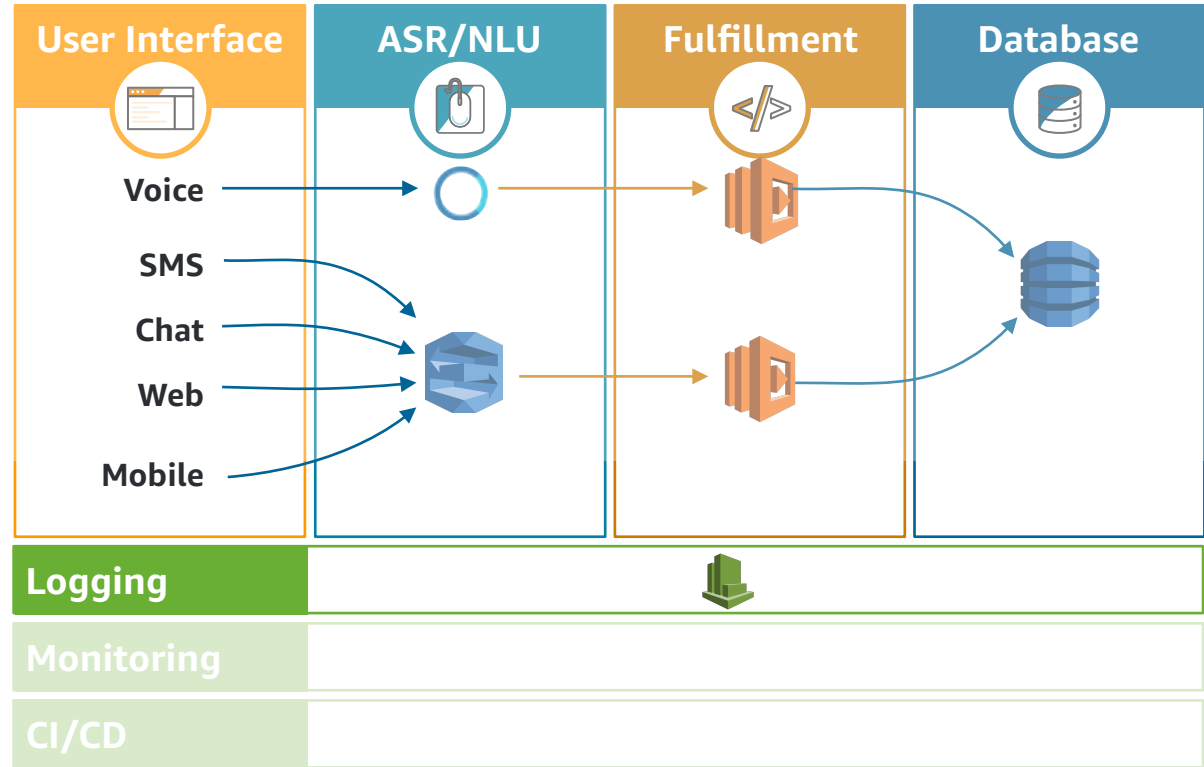
# CHATBOT Architecture



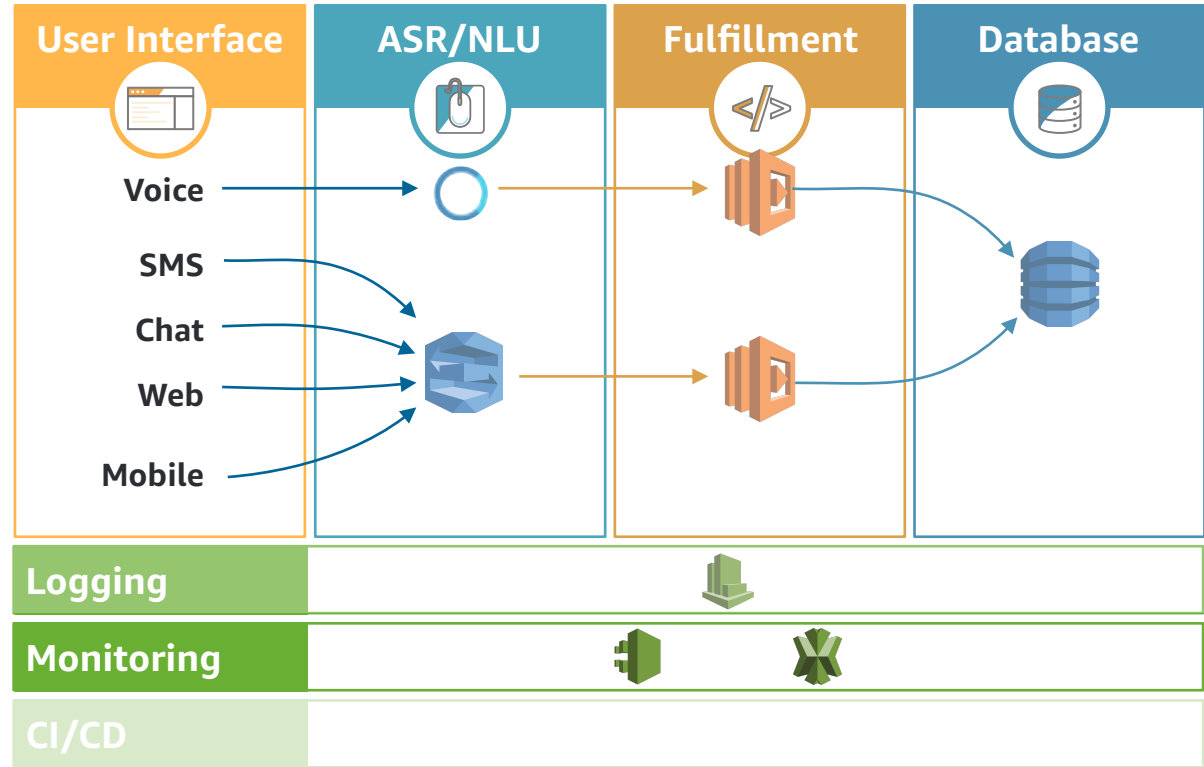
# CHATBOT Architecture



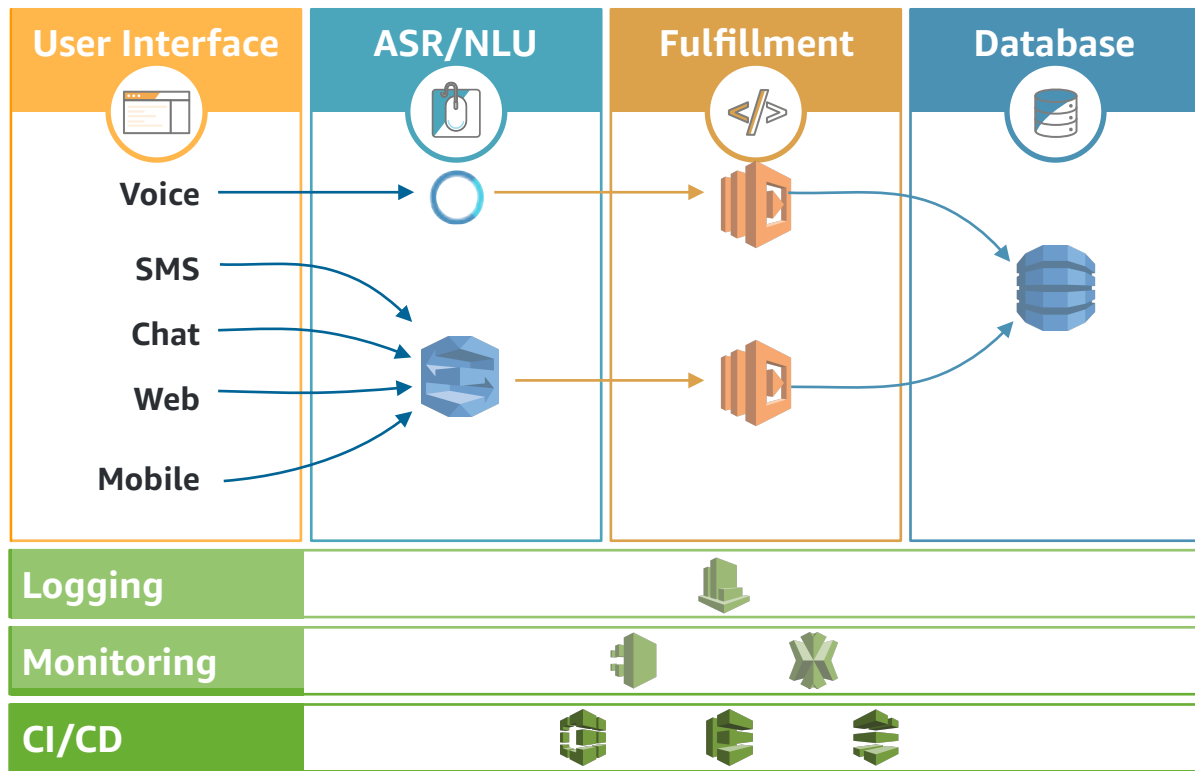
# CHATBOT Architecture



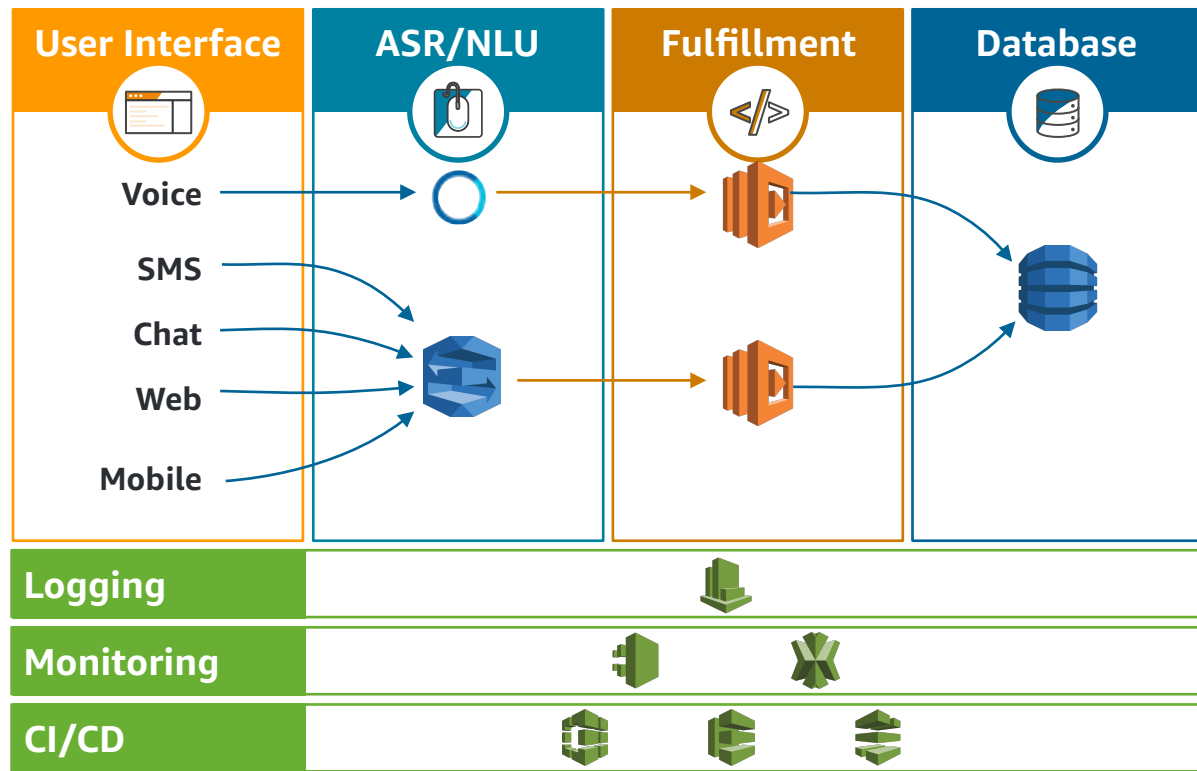
# CHATBOT Architecture



# CHATBOT Architecture



# CHATBOT Architecture



# BYOC

## Build Your Own Chatbot



### Build a Amazon Lex bot

#### Voice User Interface (VUI)

Define intent, utterances,  
slot types, slots, prompts

Write Lambda function  
for business logic  
(intent → database)

Prepare database, setup indexes

Integrate with Twilio/SMS,  
Slack, Facebook, Web, mobile



### Build an Alexa skill

Export Lex bot  
configuration

Write Lambda function  
for business logic  
(intent → database)





---

# Amazon DynamoDB

Fast, fully-managed NoSQL database service

Both key/value and document

Capable of handling any amount of data

Durable and highly available

All SSD storage


Simple and cost effective

FREE Tier: 200 million requests/month

25GB indexed data storage

# JSON Document

```
{
  "dead": true,
  "force-sensitive": true,
  "force-side": "light",
  "id": 1,
  "lightsaber": "green",
  "planet": "Dagobah",
  "quotes": [
    "When nine hundred years old you reach, look as good you will not.",
    "Truly wonderful, the mind of a child is",
    "A Jedi uses the Force for knowledge and defense, never for attack",
    "That is why you fail.",
    "Adventure. Excitement. A Jedi craves not these things.",
    "Judge me by my size, do you?",
    "Fear is the path to the dark side",
    "Wars not make one great",
    "Do, or do not. There is no try",
    "Size matters not",
    "The dark side clouds everything",
    "Impossible to see the future is",
    "Clear your mind must be",
    "Much to learn you still have ... my old padawan"
  ],
  "weapon": "lightsaber",
  "whoami": "yoda"
}
```



The screenshot shows the AWS IAM console interface. The 'Groups' tab is selected, and the user 'yoda' is highlighted in the list. The table below shows the details of the user.

Name	Status	Type	Partition key	Sort key	Attributes	Read capacity	Write capacity	Size
yoda	Active	DB	Amazon (DB)	-	N/A	0	0	1,024



# Amazon Lex

Service for building conversational interfaces using voice and text

## Provides

Automatic Speech Recognition (ASR): speech → text

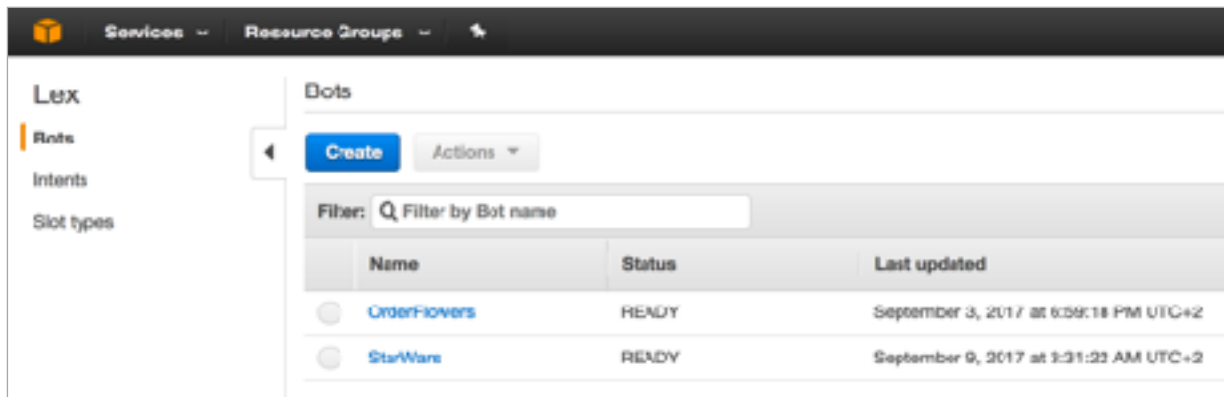
Natural Language Understanding (NLU): text → intent

Powered by the same deep learning technologies as Alexa

<http://aws.amazon.com/lex>



# Build a Amazon Lex Bot



Name	Status	Last updated
<a href="#">OrderFlowers</a>	READY	September 3, 2017 at 6:59:18 PM UTC+2
<a href="#">StarWars</a>	READY	September 9, 2017 at 1:31:23 AM UTC+2



# Amazon Lex and Twilio

## SMS INTEGRATION

**Twilio:** Build software that communicates with everyone in the world

Create a Twilio SMS endpoint

Specify Twilio credentials in Amazon Lex bot

Create a Twilio SMS endpoint (phone #)

Set Callback URL in Messaging  
of Twilio SMS endpoint

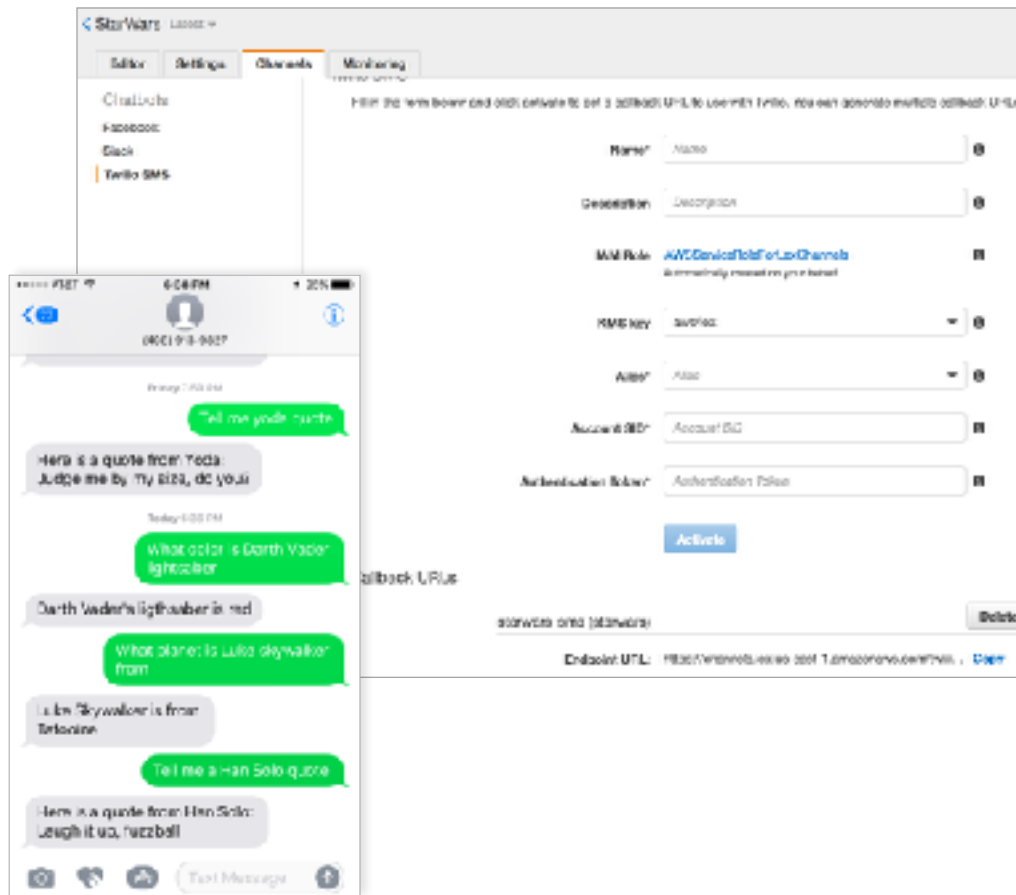
When message comes in

<https://docs.aws.amazon.com/lex/latest/dg/twilio-bot-association.html>



# Amazon Lex and Twilio

## SMS INTEGRATION





# Amazon Lex and Slack INTEGRATION

Create a Slack application: [api.slack.com](https://api.slack.com)

Add a Bot to the application

Always online

Enable interactive messages

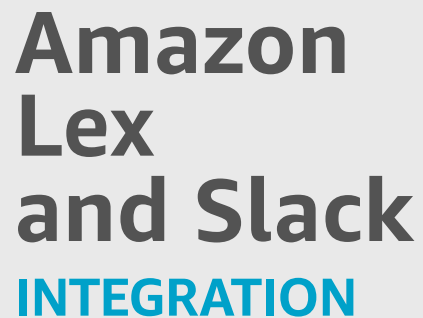
Specify Slack credentials in Amazon Lex bot

Specify URLs in Slack application

Postback: Amazon Lex bot's endpoint for Slack events

OAuth: Lex bot's handshake with Slack

<https://docs.aws.amazon.com/lex/latest/dg/slack-bot-association.html>







# Amazon Lex and Facebook

## INTEGRATION

Create a Facebook application

Specify application credentials in Amazon Lex bot

Gives Callback URL

Setup webhooks for Facebook application

Specify Callback URL and enable subscription

Optionally, publish the application  
(for public consumption)

<http://docs.aws.amazon.com/lex/latest/dg/fb-bot-association.html>



# Amazon Lex and Facebook INTEGRATION

The screenshot displays the Amazon Lex console interface. In the foreground, a chatbot preview window titled "Star Wars Chatbot" shows a conversation history. The chatbot's name is "Star Wars Chatbot" and it has 1 person who likes this. The conversation history includes:

- 5:21 PM: User: "tell me a yoda quote" (blue bubble)
- 5:27 PM: Chatbot: "Here is a quote from Yoda: When nine hundred years old you reach, look as good as you will not." (orange bubble)
- User: "what is the color of luke skywalker lightsaber" (blue bubble)
- Chatbot: "Luke Skywalker's lightsaber is blue" (orange bubble)
- User: "what planet is darth vader from?" (blue bubble)
- Chatbot: "Darth Vader is from Tatooine" (orange bubble)

The background shows the "Channels" tab of the console, with a list of channels including Facebook, Slack, and Twilio SMS. The right sidebar shows configuration options for the chatbot, including Name, Description, IAM Role, and various input/output settings.



# What is Alexa



**Alexa** is a **cloud-based service** that can answer questions, play music, read the news and more



**Echo** is an **always-on always-connected hands-free device** that connects to Alexa



# Alexa Has Skills

[amazon.com/skills](https://amazon.com/skills)

Let the fireworks begin.

"Alexa, open eHarmony."

Get started

"Alexa, what are your top skills?"

- Sleepand Relaxation Sounds**  
"Alexa open Sleep Sounds"  
★★★★☆ 5,255
- iHeartM**  
"Alexa, play iHeartM Vinyon StrusM"  
★★★★☆ 1,219
- FOX NEWS**  
"Alexa, what's my Flash Briefing?"  
★★★★☆ 582
- Song Quiz**  
"Alexa, start Song Quiz"  
★★★★☆ 652
- longairly!**  
★★★★☆ 5

***Alexa, ask Star Wars tell me a Yoda quote***



## Customer

## Alexa

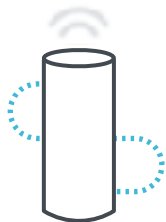
## Your Alexa Skill



**1** Customer asks a question or gives a command

**2** Alexa identifies your skill's name, analyzes and understands the customer's request, then sends your service a structured representation of the user's request

**3** Your service processes the request and returns a text and graphical response



**5a** Customer hears the response from Alexa's voice

**4a** Alexa converts the returned text to speech and streams it to the device



**5b** Customer sees a graphical representation in the companion app

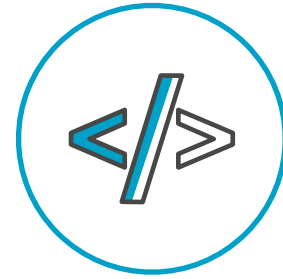
**4b** Your graphical representation (if any) is rendered in the companion app

# Two Sides of an Alexa Skill



Voice User  
Interface

+



Programming  
Logic

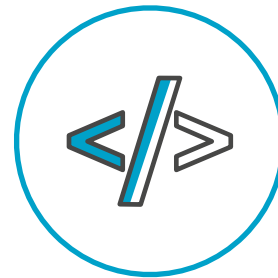
# Two Sides of an Alexa Skill



Voice User  
Interface

[developer.amazon.com](https://developer.amazon.com)

+



Programming  
Logic

[aws.amazon.com](https://aws.amazon.com)





# Build an Alexa Skill

Design a **Voice User Interface**

**Setup** the Skill in the Developer Portal

Can be done using ASK CLI or REST API

Use the Voice Design to **Build Your Interaction Model**

Intents, sample utterances, and the dialog model

**Write and Test the Code** for your Skill

AWS Lambda or web service on any cloud provider

**Beta Test** your Skill

Submit your Skill for **Certification**



# Alexa Skills Kit

```
<dependency>  
  <groupId>com.amazon.alex</groupId>  
  <artifactId>alex</artifactId>  
  <version>1.4.0</version>  
  <scope>compile</scope>  
</dependency>
```

echosim.io provided by Quinix Media. Special thanks to Sam Mathis for his Alexa in the browser [project](#).

## Alexa Skill Testing Tool

Click and **hold** the microphone button or hold down the space bar on your keyboard to activate the microphone.

Tapnet Speech



Clear

Collapse / Expand

SpeechRecognizer.ExpectSpeech 2017-09-21 11:21:18

```
{
  "directive": {},
  "header": {
    "dialogRequestId": "dialogRequestId-10b1f102-d922-4e60-b854-4f9993f011f9",
    "namespace": "SpeechRecognizer",
    "name": "ExpectSpeech",
    "messageId": "64b44121-2924-4901-05a5-847f0f1766b"
  },
  "payload": {
    "timeoutInMilliseconds": 8000
  }
}
```

TemplateRuntime.RenderTemplate 2017-09-21 11:21:18

Welcome to Star Wars Trivia, you can ask quotes

Star Wars  
Star Wars Welcome

```
{
  "directive": {},
  "header": {
    "dialogRequestId": "dialogRequestId-10b1f102-d922-4e60-b854-4f9993f011f9",
    "namespace": "TemplateRuntime",
    "name": "RenderTemplate",
    "messageId": "0015d9d1-ec01-4e01-b12f-f0e020e0005"
  },
  "payload": {
```

[echosim.io](https://echosim.io)

# Alexa is now in the Amazon app





# LOGGING Amazon CloudWatch

# Maven Plugin

```
<groupId>com.github.seanroy</groupId>  
<artifactId>lambda-maven-plugin</artifactId>  
<version>2.2.3</version>
```



---

## References

### **Amazon Lex**

[aws.amazon.com/lex](https://aws.amazon.com/lex)

### **Developer Portal**

[developer.amazon.com](https://developer.amazon.com)

### **Slides & code**

[github.com/arun-gupta/chatbot](https://github.com/arun-gupta/chatbot)