

COMP 9322

Software Service Design and Engineering

Lecture 5 ChatBots

Disclaimer

- the slides are taken from Cognitive Services Engineering presentation by Sci Prof Boualem Benatallah and Shayan Zamani

ChatBots Future

- [Gartner](#) estimates that by 2020, chatbots will be handling 85 percent of customer-service interactions; they are already handling about 30 percent of transactions now”

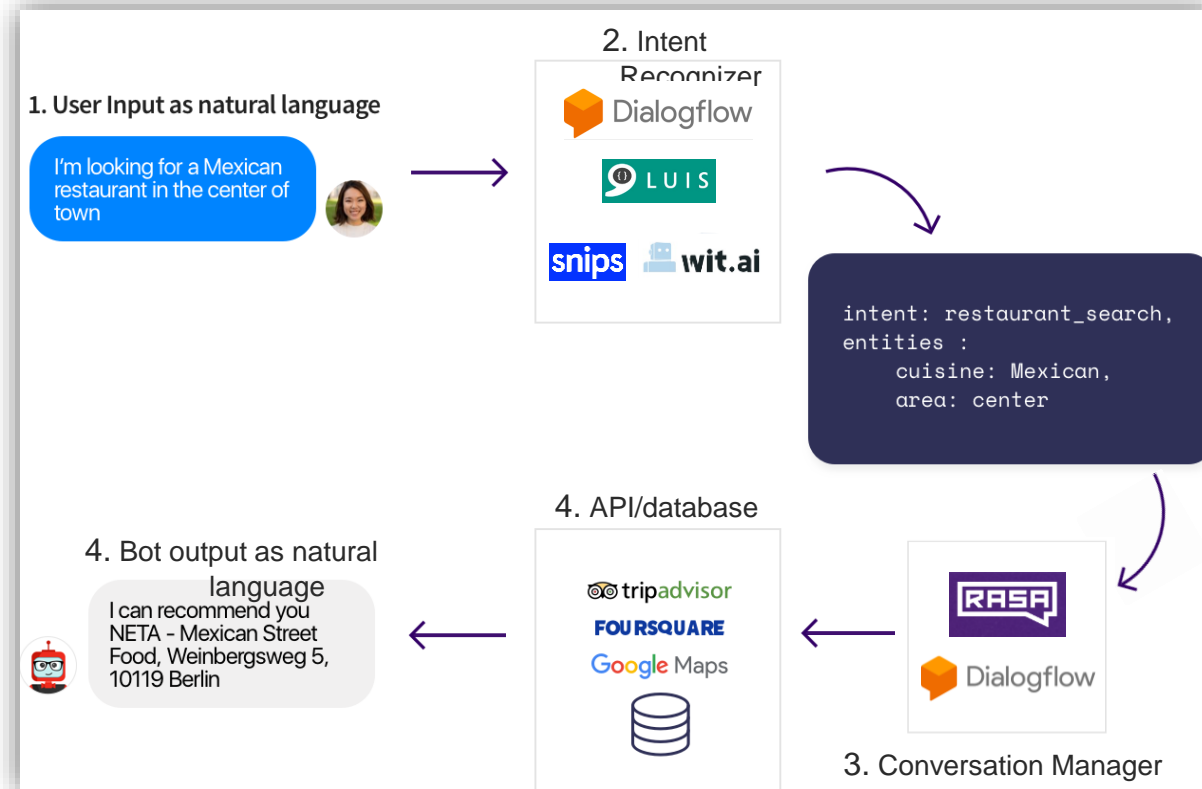
Types of Bots

- Template based (Rule Based)
- Machine Learning Based

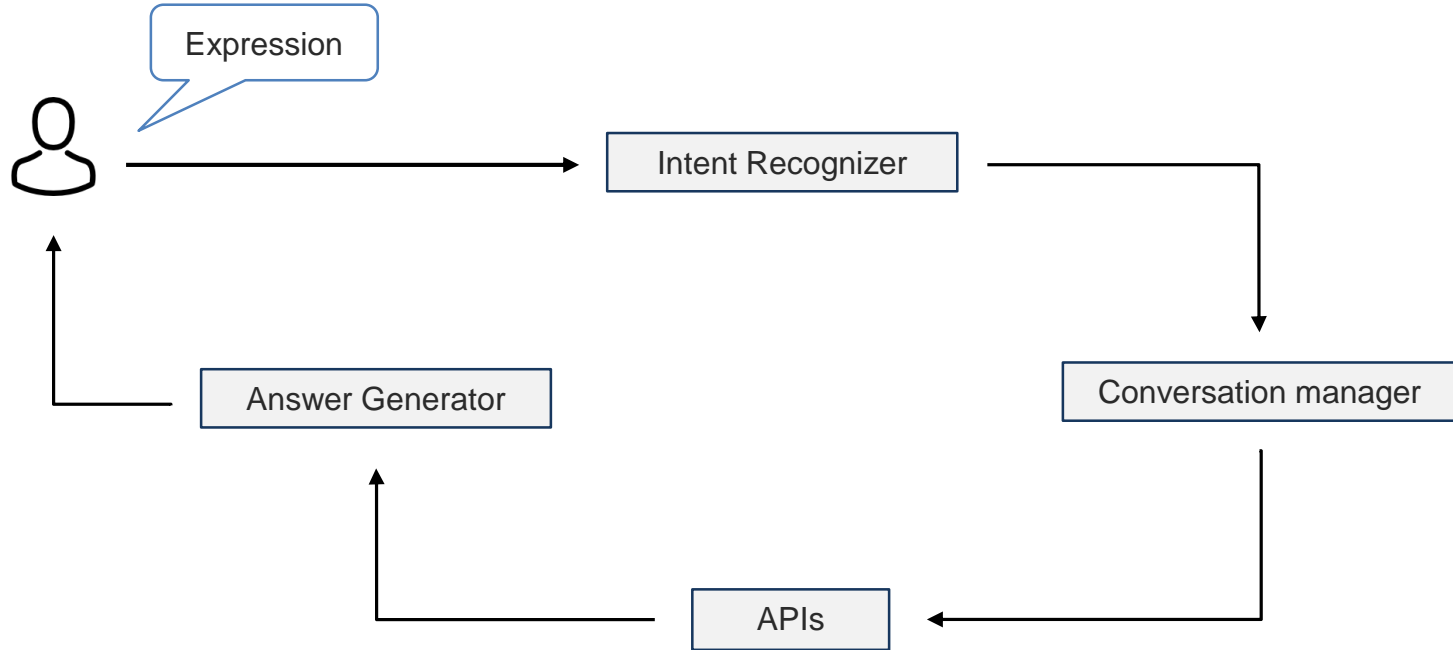
Machine Learning-Based Bots

- Need training dataset (user expressions) to train the NLU model
- Need backend-coding (define functions to handle different user intentions)
- Not 100% accurate
- Could be smart enough (amount of training expressions is important)
- Communicates in natural manner (can be controlled/non-controlled natural language)
- For simple/advanced use cases (e.g. order pizza, flight booking, schedule tasks)

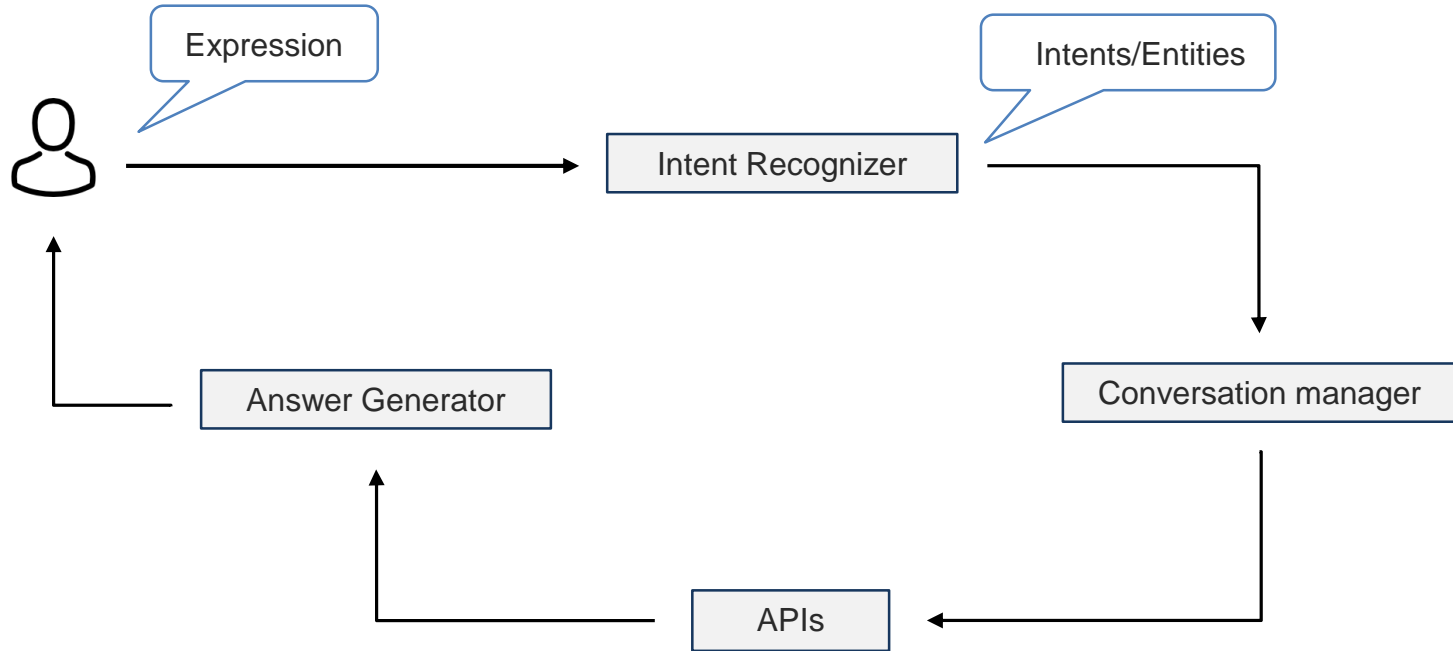
Architecture



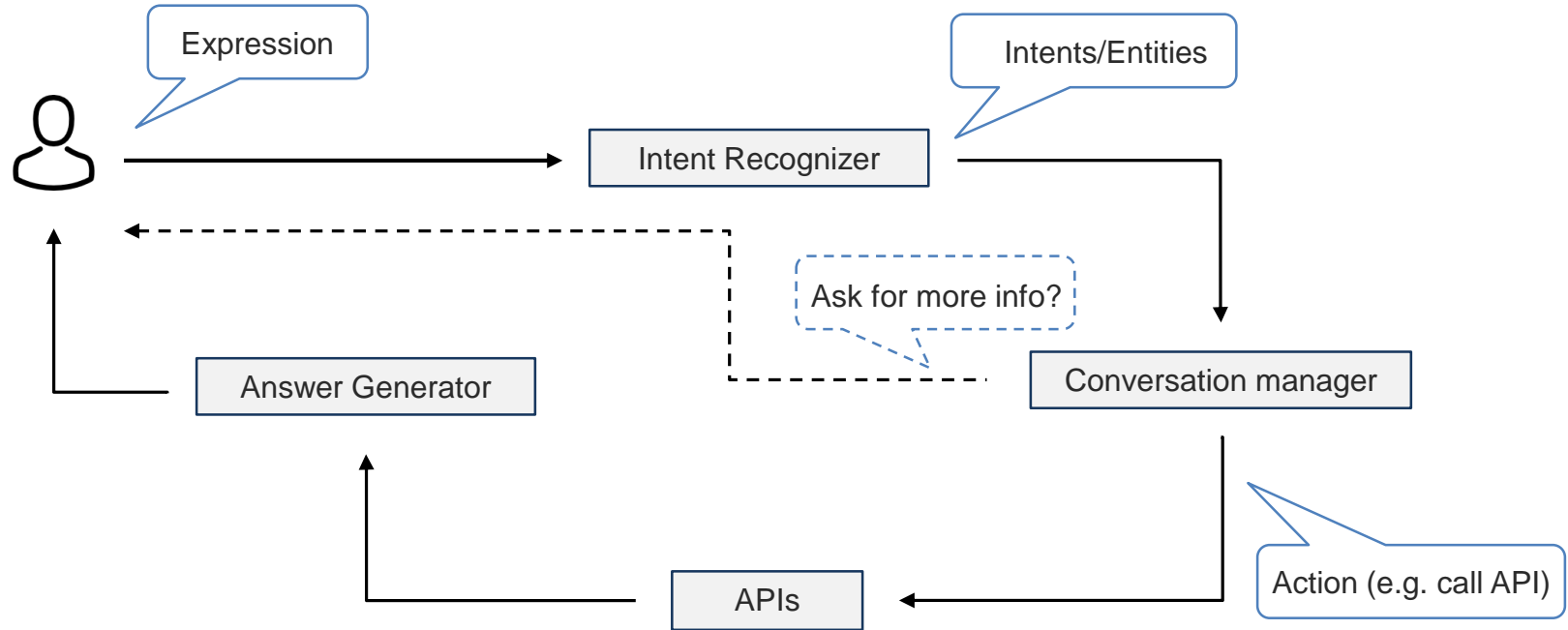
Machine Learning-Based Bots Example Flow



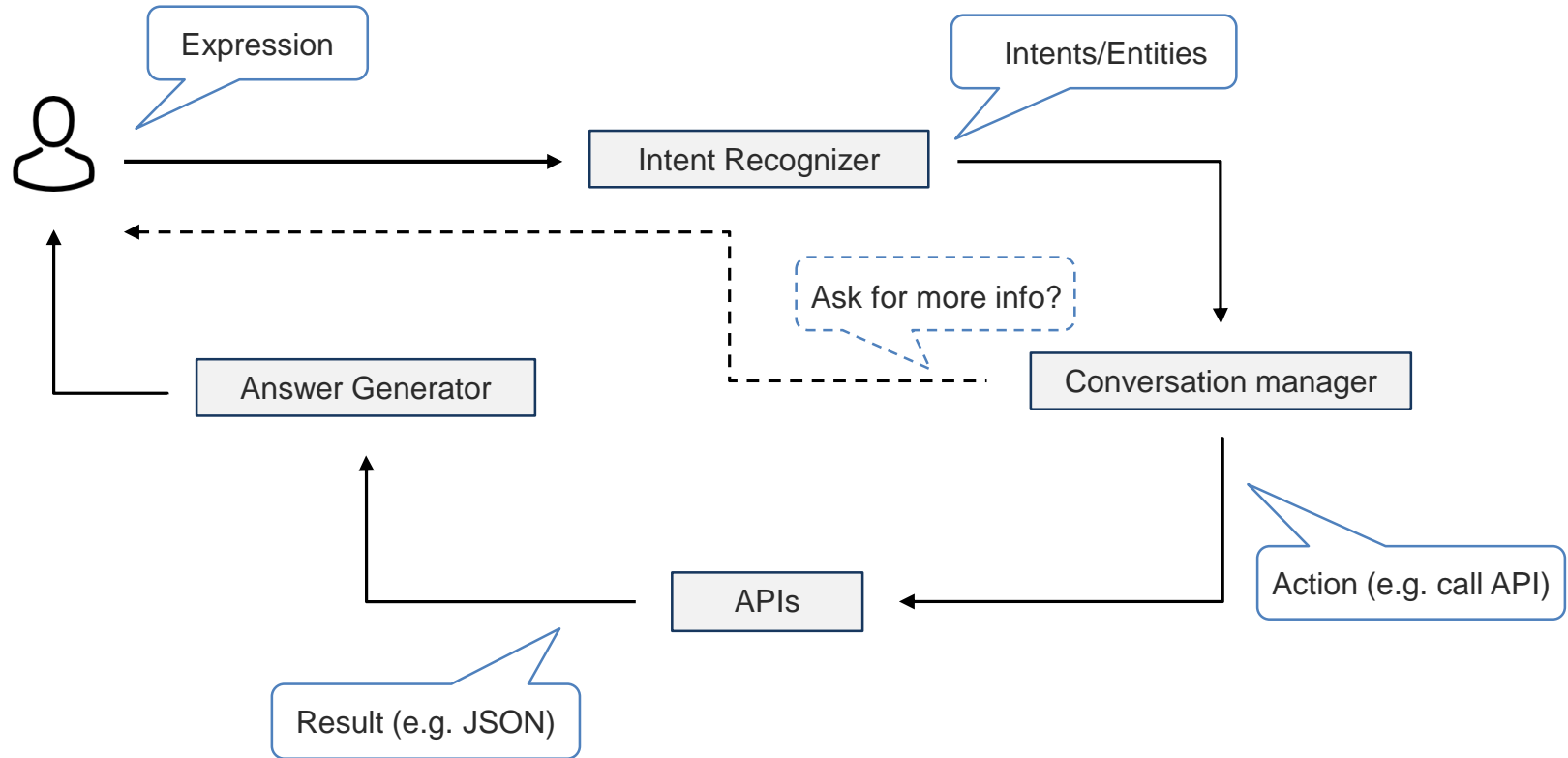
Machine Learning-Based Bots Example Flow



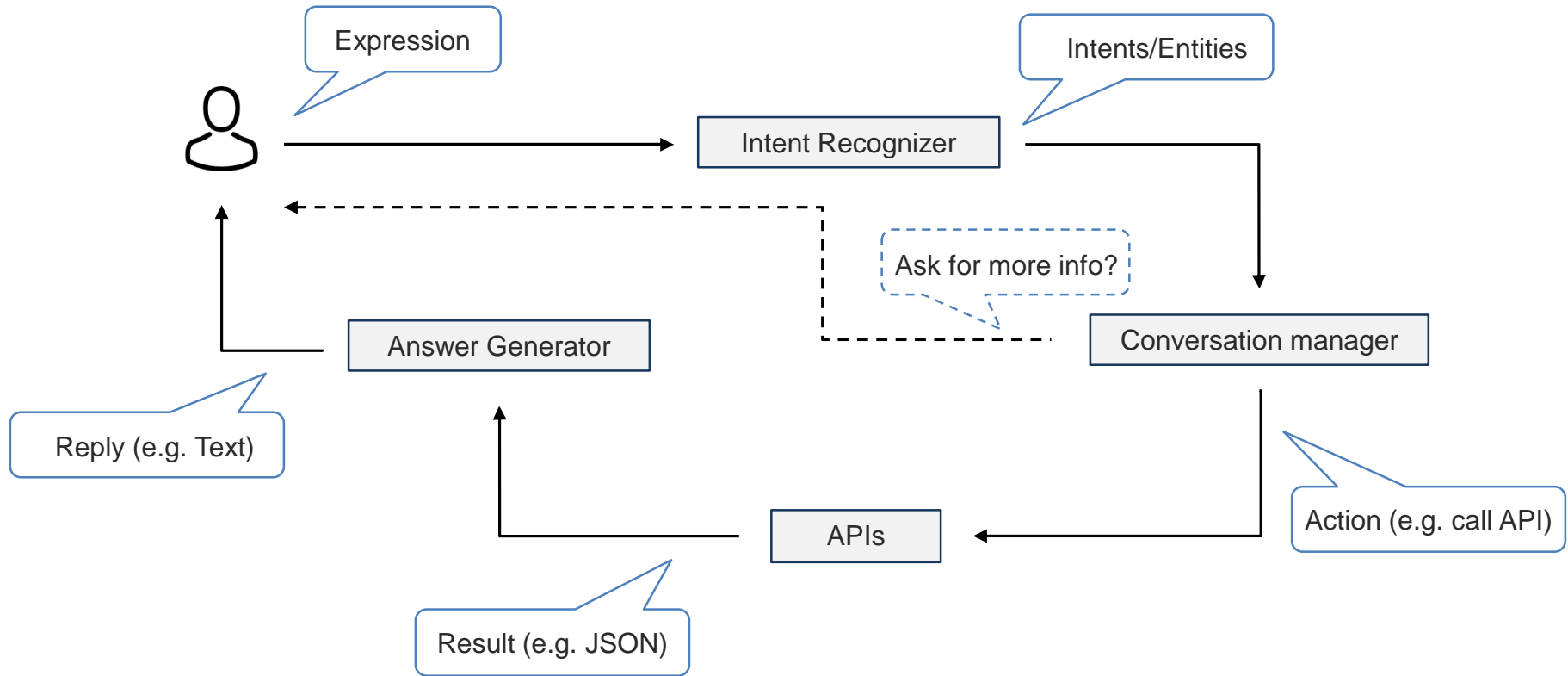
Machine Learning-Based Bots Example Flow



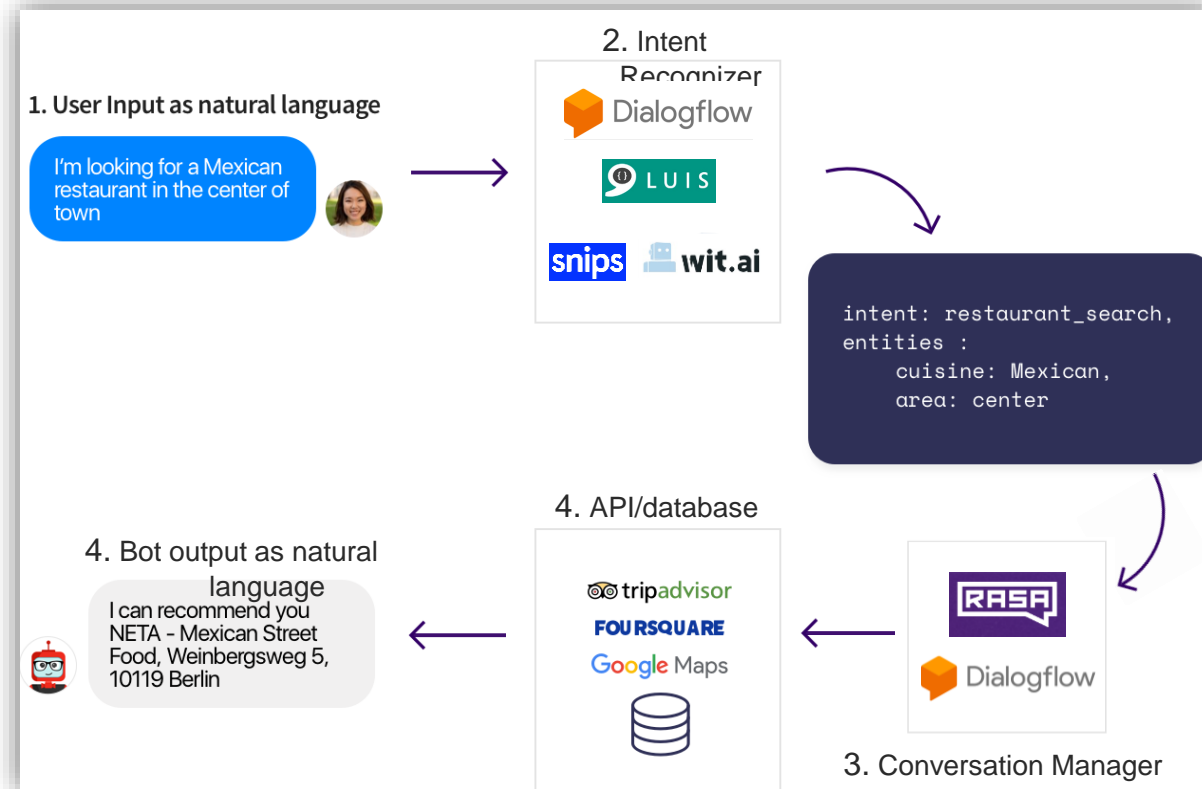
Machine Learning-Based Bots Example Flow



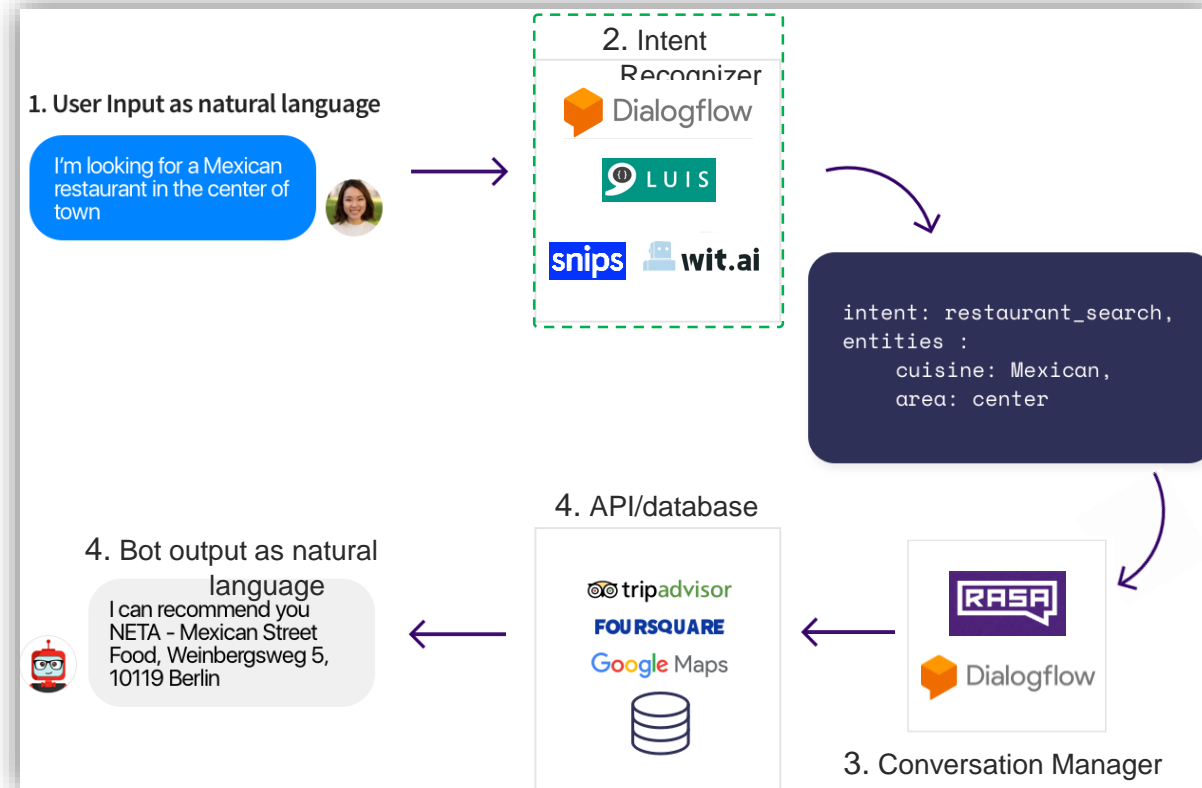
Machine Learning-Based Bots Example Flow



Architecture



Architecture



Intent Recognizer - Training

Training data for
FindRestaurant intent



```
{
  "text": "I am looking a restaurant in 29432",
  "intent": "restaurant_search",
  "entities": [
    {
      "start": 29,
      "end": 34,
      "value": "29432",
      "entity": "location"
    }
  ]
},
{
  "text": "I am looking for mexican indian fusion",
  "intent": "restaurant_search",
  "entities": [
    {
      "start": 17,
      "end": 38,
      "value": "mexican indian fusion",
      "entity": "cuisine"
    }
  ]
},
```

Intent Recognizer - Training

Training data for
FindRestaurant intent



expression

```
{
  "text": "I am looking a restaurant in 29432",
  "intent": "restaurant_search",
  "entities": [
    {
      "start": 29,
      "end": 34,
      "value": "29432",
      "entity": "location"
    }
  ]
},
{
  "text": "I am looking for mexican indian fusion",
  "intent": "restaurant_search",
  "entities": [
    {
      "start": 17,
      "end": 38,
      "value": "mexican indian fusion",
      "entity": "cuisine"
    }
  ]
},
```

Intent Recognizer - Training

Training data for
FindRestaurant intent



expression

intent name

```
{
  "text": "I am looking a restaurant in 29432",
  "intent": "restaurant_search",
  "entities": [
    {
      "start": 29,
      "end": 34,
      "value": "29432",
      "entity": "location"
    }
  ]
},
{
  "text": "I am looking for mexican indian fusion",
  "intent": "restaurant_search",
  "entities": [
    {
      "start": 17,
      "end": 38,
      "value": "mexican indian fusion",
      "entity": "cuisine"
    }
  ]
},
```


Intent Recognizer - Training

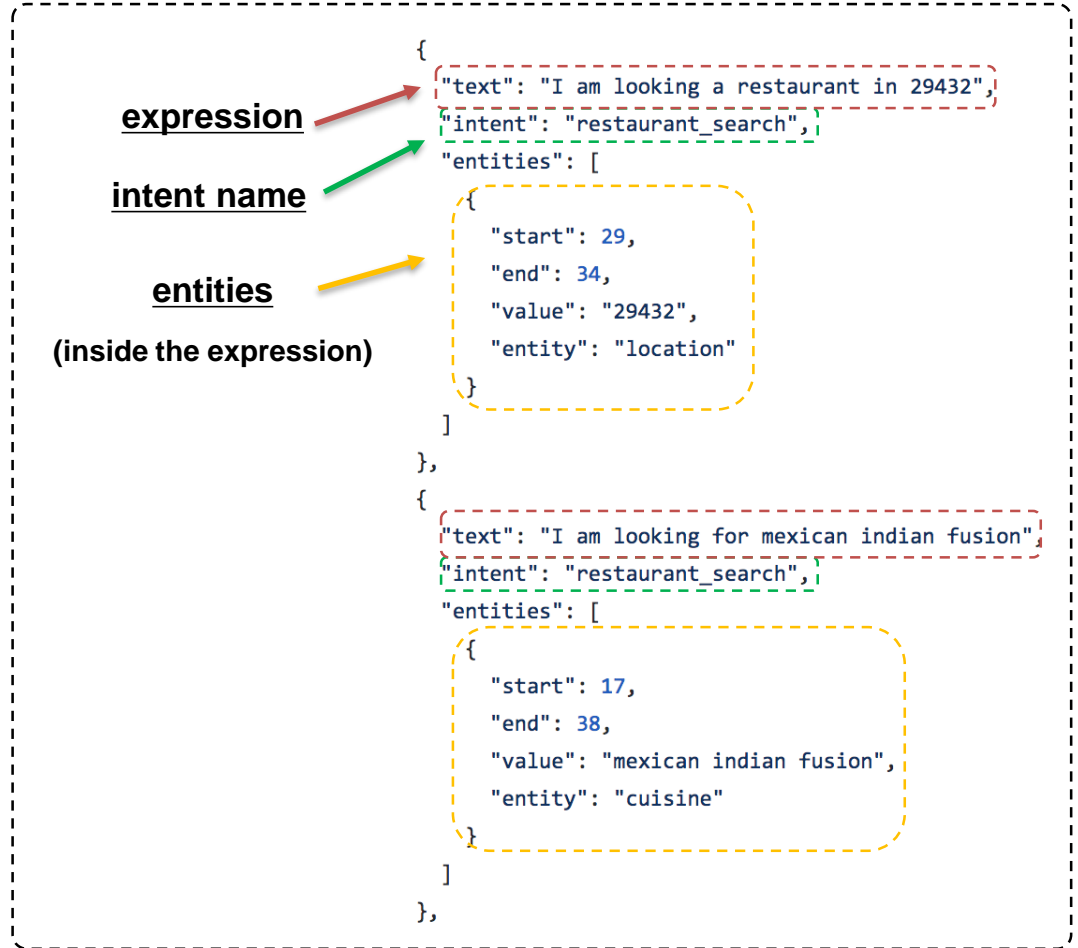
Training data for
FindRestaurant intent



expression → {
intent name → "text": "I am looking a restaurant in 29432",
entities → "intent": "restaurant_search",
(inside the expression) → "entities": [
 {
 "start": 29,
 "end": 34,
 "value": "29432",
 "entity": "location"
 }
]
},
{
 "text": "I am looking for mexican indian fusion",
 "intent": "restaurant_search",
 "entities": [
 {
 "start": 17,
 "end": 38,
 "value": "mexican indian fusion",
 "entity": "cuisine"
 }
]
},
}

Intent Recognizer - Training

Training data for
FindRestaurant intent



Intent Recognizer - Training

Training data for
Greeting intent

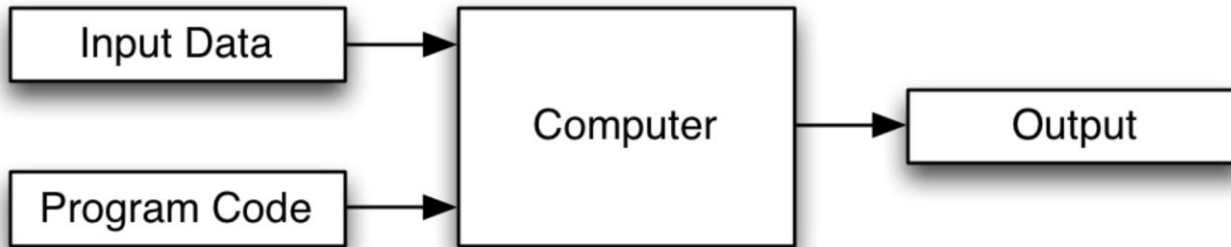


expression
intent name
entities
(inside the expression)

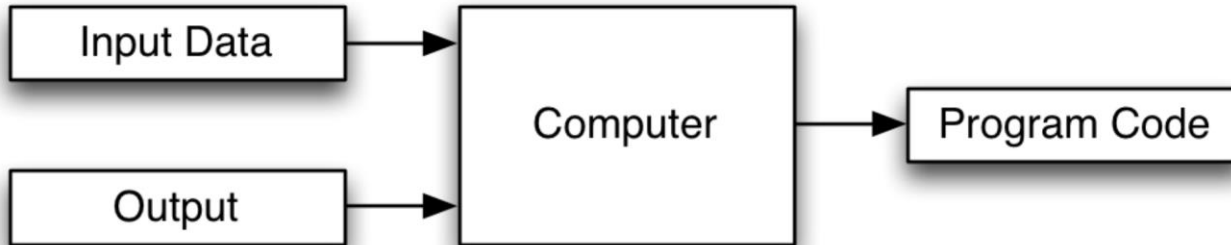
```
{  
  "text": "hey",  
  "intent": "greet",  
  "entities": []  
},  
{  
  "text": "howdy",  
  "intent": "greet",  
  "entities": []  
},  
{  
  "text": "hey there",  
  "intent": "greet",  
  "entities": []  
},  
{  
  "text": "hello",  
  "intent": "greet",  
  "entities": []  
},  
{  
  "text": "hi",  
  "intent": "greet",  
  "entities": []  
},  
...
```

Intent Recognizer - A Machine Learning Model

Traditional Software Development



Machine Learning Programming



Machine Learning Methods

- Linear regression based methods
- Neural Network
- Bayesian Network
- Support Vector Machine (SVM)
- Decision Tree based methods
- Nearest Neighbor
- Logistic Regression
- Naive Bayes
- K-Means
- Random Forest
- Dimensionality Reduction methods
- ...

Machine Learning Models



- **Predictive Modeling:** Predict the output using input values
- **Computer Vision:** Detect patterns in visual data
- **Time Series:** Detect patterns in in time e.g. financial applications, speech recognition


Intent Recognizer - A Machine Learning Model

- Intent Classification
 - MITIE intent classifier - multi class linear SVM, Nearest Neighbors, Neural Embedding intent classifier, Sklearn intent classifier - Linear SVM, Naive Bayes...
- Entity Recognition
 - CRF (conditional random field), Duckling, MITIE, Spacy - BILUO transition model,

Intent Recognizer - Acquisition

- **Experts**
 - **Expensive/Quality is high**
- **Crowds**
 - **Cheaper/Quality is less than expert training**
- **Algorithms**
 - **Automatic Generating/Quality depends on algorithm**
 - **Examples: Chatito, Tracy**

Intent Recognizer (DialogFlow)

 Dialogflow

Test

en

Intents

Entities

Fulfillment

Integrations

Training

History

Analytics

Prebuilt Agents

Small Talk

Docs

Forum

Support

Account

Logout

restaurant_search

SAVE

Contexts

Events

Training phrases

Search training phrases

” Add user expression

” where can I find pepperoni pizza, I am in Kensington

” It's really cold outside, any place to drink cappuccino around UNSW

” Give me list of Italian cafe, I am in Randwick

” I wanna eat hokken noodle for lunch, where can I find it, I am in Harbin university

” Is there any Chinese resto around UNSW, I want to eat noodle soup

Action and parameters

find_restaurant_function

REQUIRED	PARAMETER NAME	ENTITY	VALUE	IS LIST	PROMPTS
<input type="checkbox"/>	cuisine	@cuisine	\$cuisine	<input type="checkbox"/>	—
<input checked="" type="checkbox"/>	type	@type	\$type	<input type="checkbox"/>	What type of fo...
<input checked="" type="checkbox"/>	address	@address	\$address	<input type="checkbox"/>	Where are you? ...
<input type="checkbox"/>	Enter name	Enter entity	Enter value	<input type="checkbox"/>	—

Try it now

See how it works in Google Assistant.

Agent

USER SAYS

I want to drink latte, is there any Italian cafe here in UNSW

DEFAULT RESPONSE

What type of food/drink?

CONTEXTS

restaurant_search_dialog_params_type

b03659af-09e2-4c6c-bb96-7f933e1e3418_i_d_dialog_context

restaurant_search_dialog_context

INTENT

restaurant_search

ACTION

find_restaurant_function

PARAMETER

cuisine

type

address

VALUE

Italian

latte

UNSW

DIAGNOSTIC INFO

Intent Recognizer (LUIS)

LUIS Go to Preview My Applications About Help Support Forum Pavlo Bashmakov Sign Out

App Development New utterances Search Suggest Review labels

Show all labeled utterances

Select text in an utterance to label an entity, or click to clear.

Model prediction
how to build an **ios** app in 6 weeks ? AppDevelopment (0.99)
how to build an **ios** app in 6 weeks ? AppDevelopment(0.99)

Model prediction
need to develop an **android** app for my service . AppDevelopment (1)
need to develop an **android** app for my service . AppDevelopment(1)

Model prediction
can you help me with **mac** application AppDevelopment (0.87)
can you help me with **mac** application AppDevelopment(0.87)

Performance analysis

Intents

AppDevelopment
5 utterances: 5 correctly predicted

None
0 utterances: 0 correctly predicted

AppDesign
3 utterances: 3 correctly predicted

Correctly predicted
Error (predicted as other intent)

Intent Recognizer (LEX)

The screenshot displays the Amazon Lex console interface for configuring an intent named 'startInstanceIntent' within a bot named 'awsChatBot'. The interface is divided into several sections:

- Navigation:** Includes 'Services', 'Resource Groups', and a search bar. The 'awsChatBot Latest' dropdown is visible.
- Actions:** 'Build' and 'Publish' buttons are located in the top right corner.
- Tabs:** 'Editor', 'Settings', 'Channels', and 'Monitoring' are present, with 'Editor' being the active tab.
- Left Sidebar:** Contains links for 'Intents' (with a plus icon), 'startInstanceIntent', 'Slot types' (with a plus icon), 'INSTANCE_NAME', and 'Error Handling'.
- Main Content Area:**
 - Intent:** 'startInstanceIntent' with a dropdown arrow.
 - Sample utterances:** Two examples are provided: 'i would like to start a sever (instance)' and 'i want to start a server (instance)'. The word 'instance' is highlighted in a blue box.
 - Lambda initialization and validation:** A section with an information icon.
 - Slots:** A list of slots. Slot 1 is checked and labeled 'Instance', with a corresponding dropdown menu showing 'INSTANCE_NAME' and a value of '1'.
 - Confirmation prompt:** A section with an information icon.
 - Fulfillment:**
 - Radio buttons for 'AWS Lambda function' (selected) and 'Return parameters to client'.
 - A dropdown menu showing 'awsLexAssistant'.
 - Radio buttons for 'Goodbye message', 'Follow-up message', and 'None'.
 - A text input field containing 'Thank you. starting (instance)'.
- Test Bot Panel:** A floating window on the right titled 'Test Bot' with a status of 'Build: Latest | Status: READY'. It shows a simulated conversation:
 - User: 'i want to start a srver' (note the typo)
 - Bot: 'which server you want to start?'
 - User: 'prod'
 - Bot: 'Are you sure you want to start prod'
 - User: 'yes'
 - Bot: 'Starting Server Successfully'
 - A 'Clear' button is at the bottom.

Intent Recognizer Examples



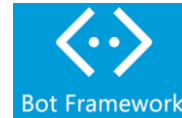
SnatchBot



Amazon Lex



amazon alexa



Dialogflow

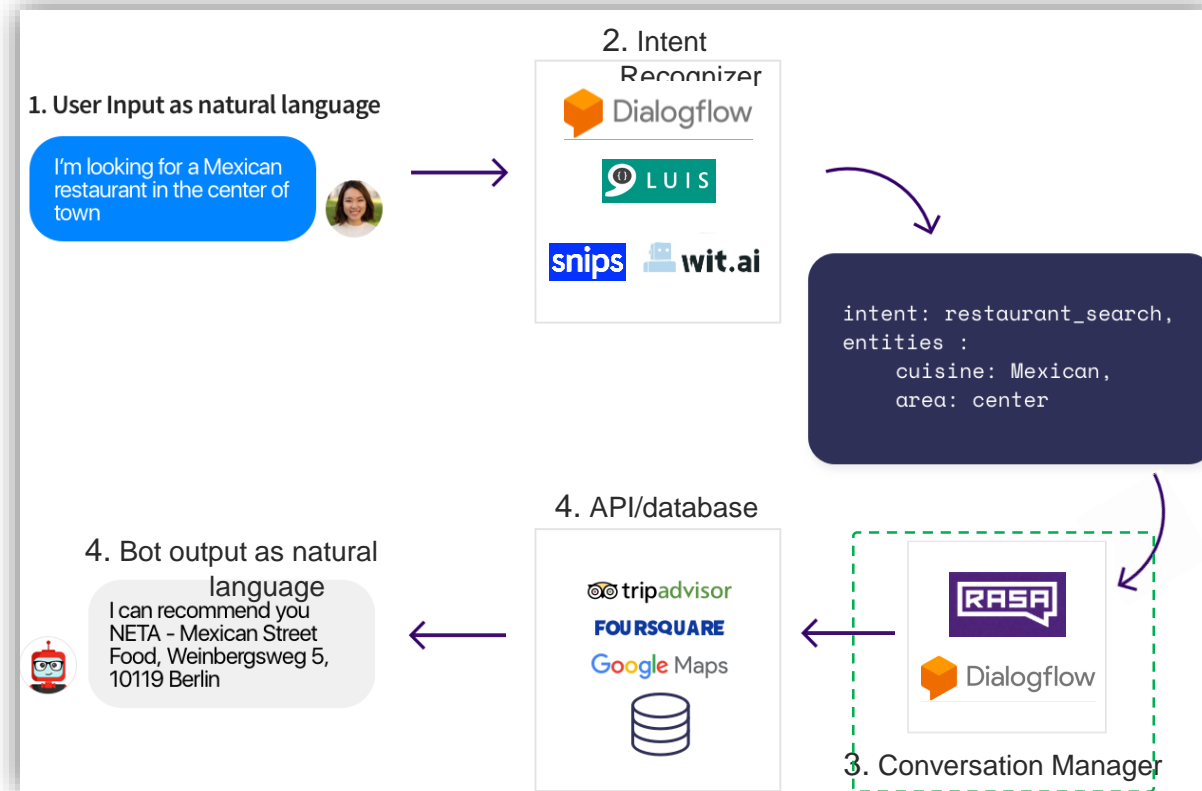


wit.ai



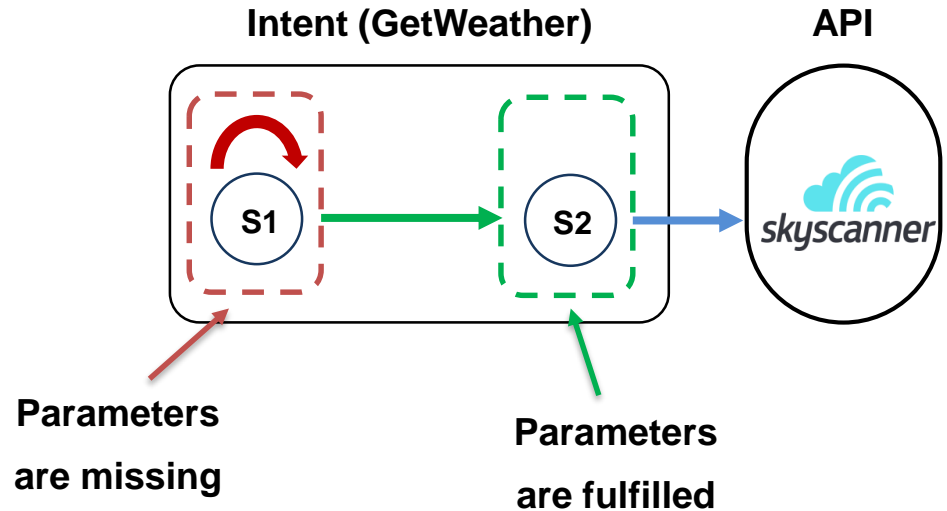
KIT.AI

Architecture

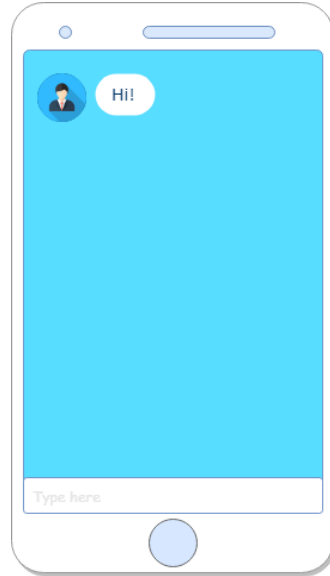


Conversation Manager

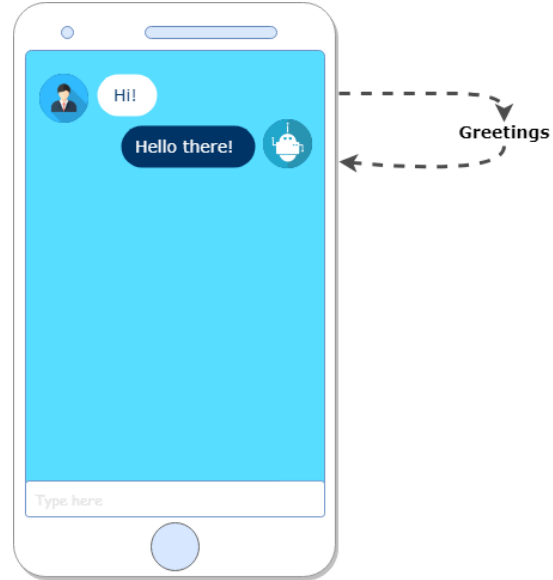
- Each intent (e.g. Get Weather) is a state machine
- Conversation Manager moves between states (to fill missing information)



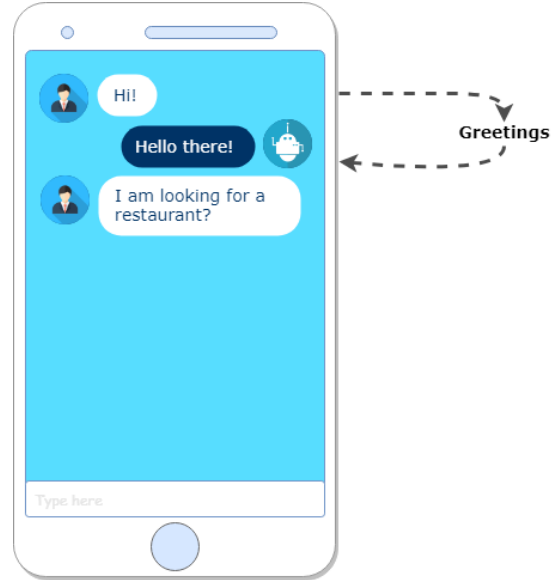
Conversation Manager



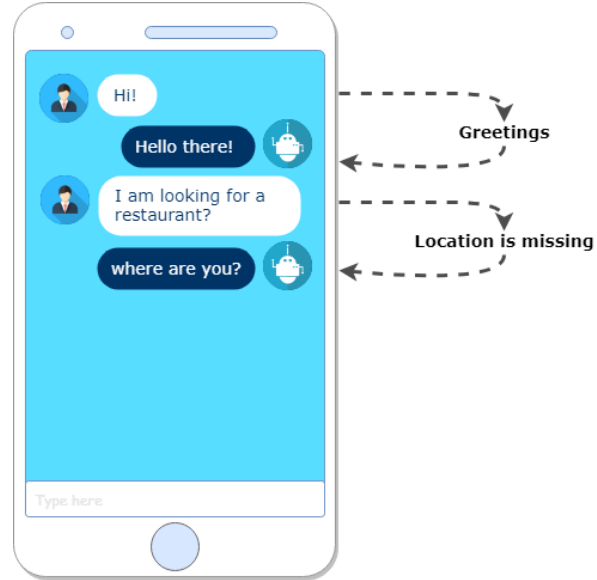
Conversation Manager



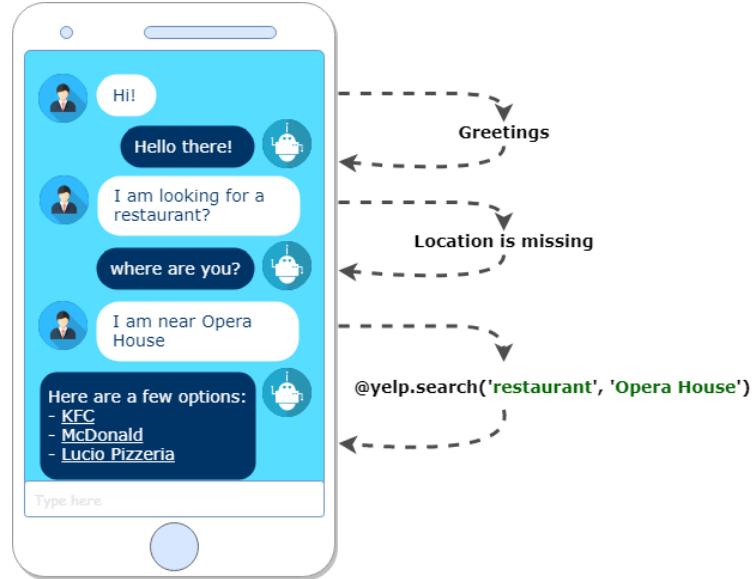
Conversation Manager



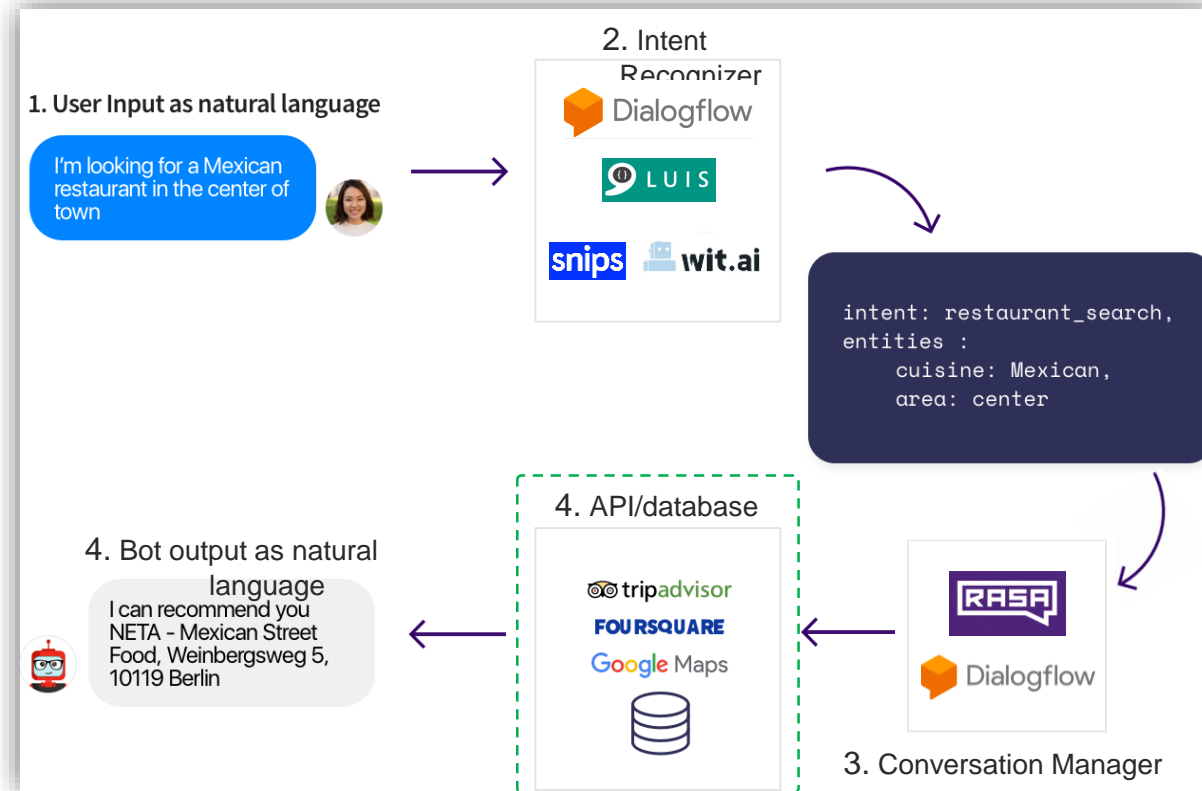
Conversation Manager



Conversation Manager



Architecture



User Utterances | Intents | APIs Integration

NL Expression:
e.g. "Show me list of flights from *Sydney* to *Paris* for *November 26*, and return flight is on *8th of Dec*"

NL Intent: **flightList**
NL Entities: [*Sydney*, *Paris*,
2017-11-26, *2017-12-08*]



NL Expression:
e.g. "Is there any *Italian cafe* near *UNSW*?"

NL Intent: **findResto**
NL Entities: [*Italian cafe*, *UNSW*]

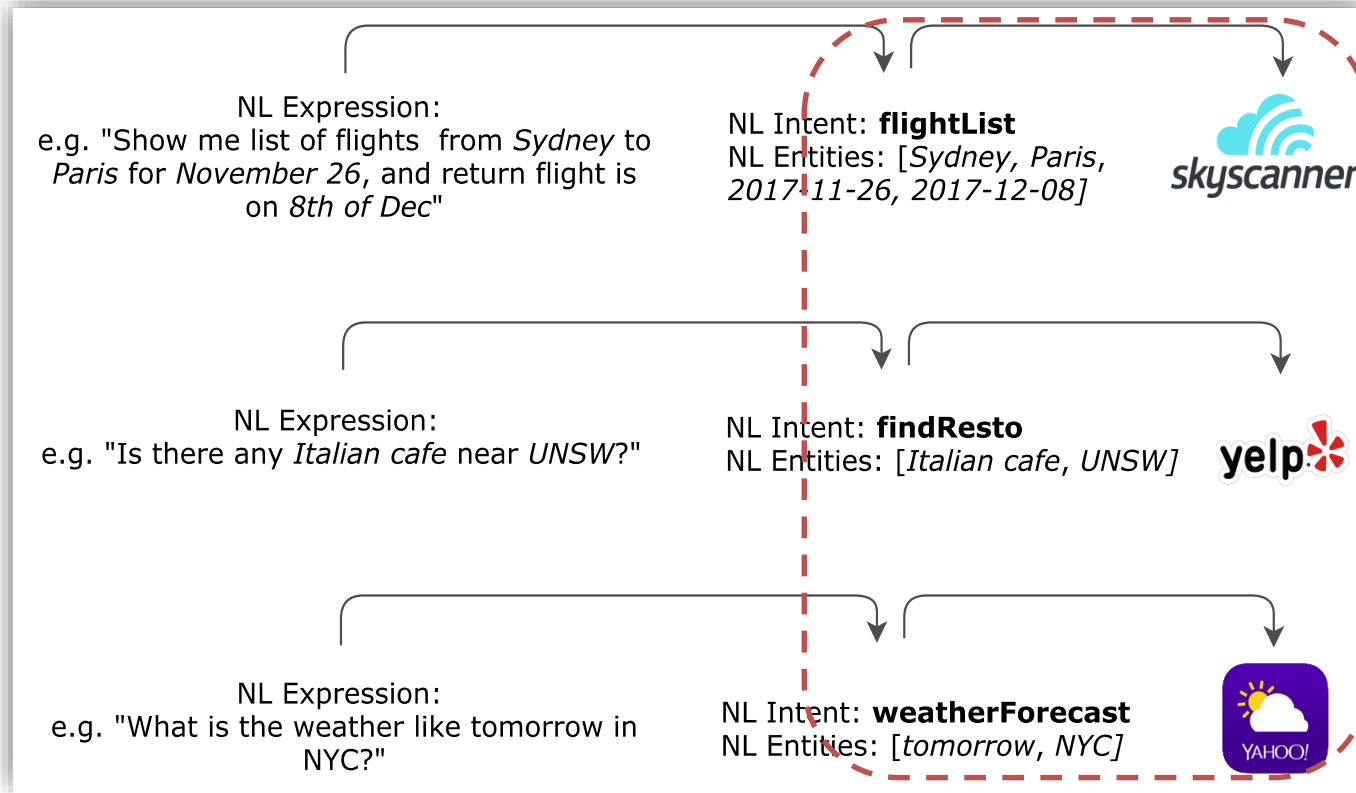


NL Expression:
e.g. "What is the weather like tomorrow in *NYC*?"

NL Intent: **weatherForecast**
NL Entities: [*tomorrow*, *NYC*]



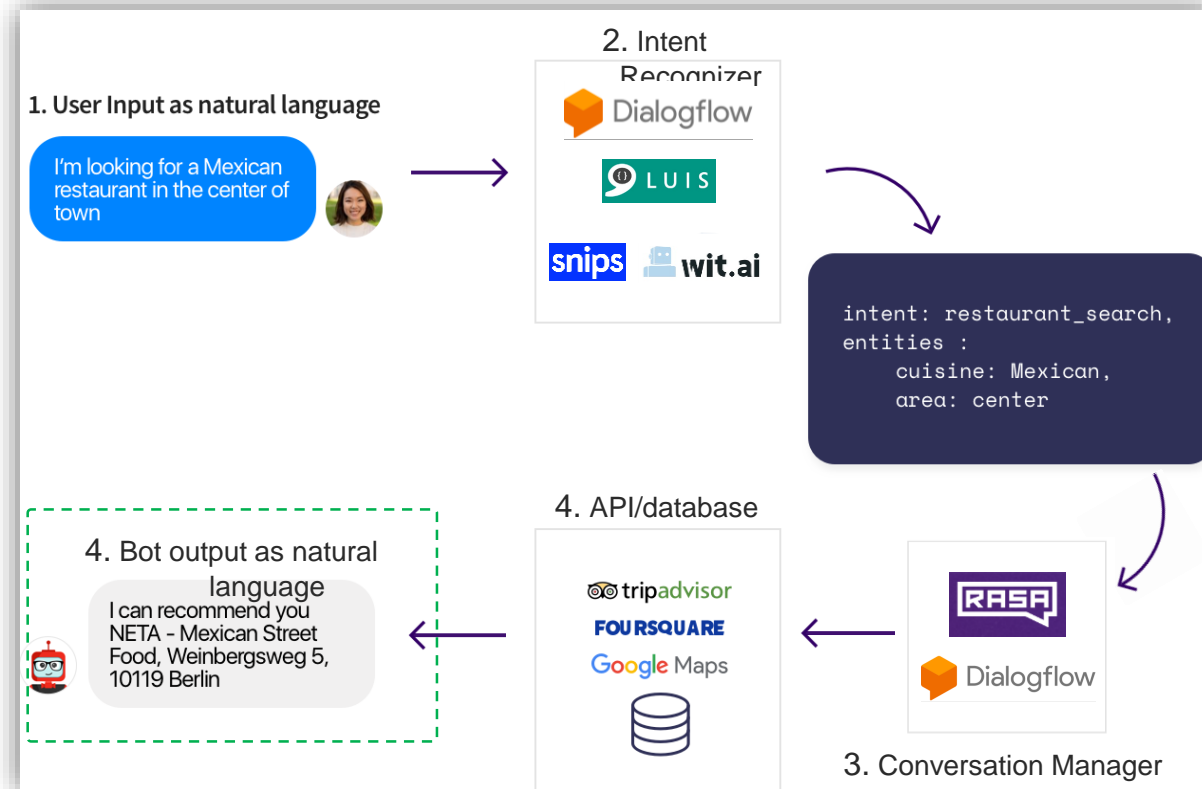
User Utterances | Intents | APIs Integration



Static Hardcoded
Mappings (e.g., Alexa
Skills)

Google DialogFlow,
Facebook Wit.ai,
Amazon Lex,
Microsoft LUIS

Architecture



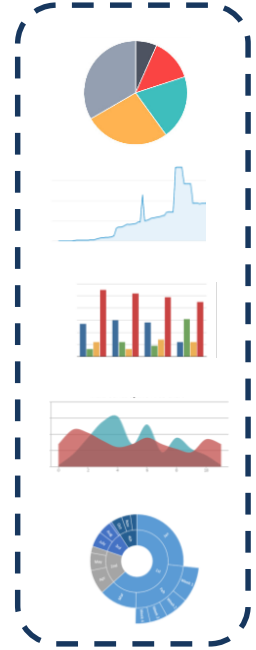
Answer Generator

```
templates
  weather_condition:
    - text: "The weather in {city} is {weather_description}."
```

Template

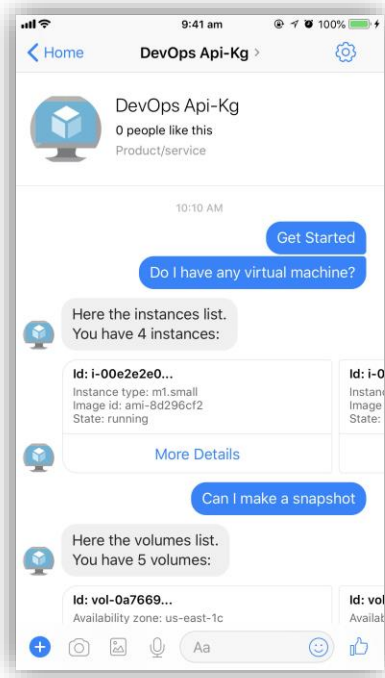
```
"coord": {
  "lon": 126.62,
  "lat": 45.77
},
"weather": [
  {
    "id": 800,
    "main": "Clear",
    "description": "clear sky",
    "icon": "01d"
  }
],
"base": "stations",
"main": {
  "temp": 302.15,
  "pressure": 1007,
  "humidity": 58,
  "temp_min": 302.15,
  "temp_max": 302.15
},
"visibility": 10000,
"wind": {
  "speed": 2,
  "deg": 130
},
"clouds": {
  "all": 0
},
"dt": 1531303200,
"sys": {
  "type": 1,
  "id": 7469,
```

Dump output
(e.g. JSON)



Widgets

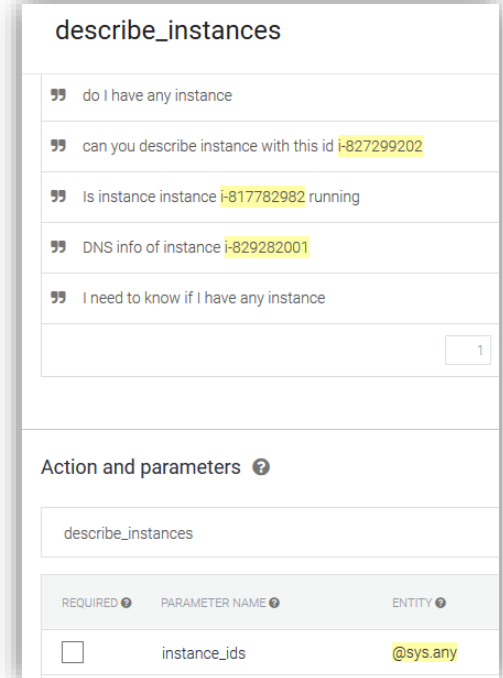
DevOps bot



 Messaging Interface
(Messenger)

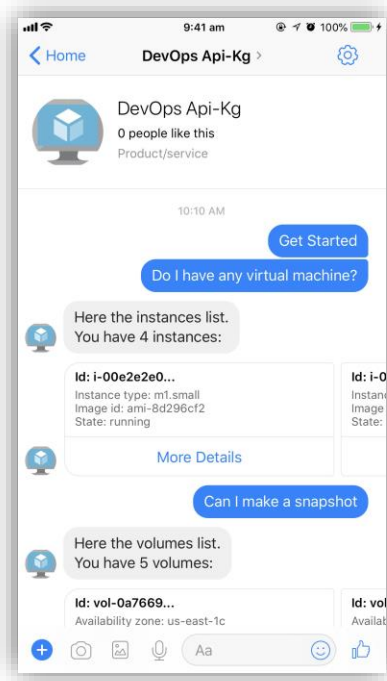


1



 Intent Recognizer
(Dialogflow)

DevOps bot



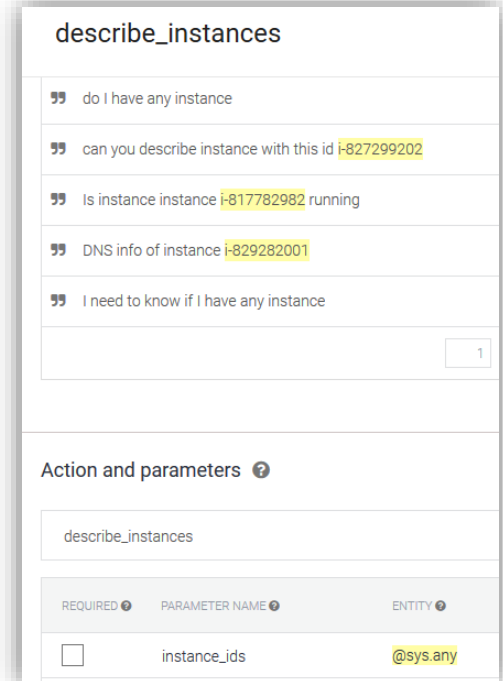
 Messaging Interface
(Messenger)



 API Backend

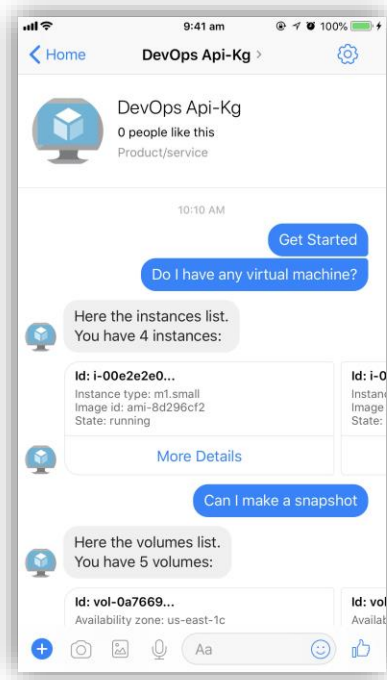


 Webhook
(NodeJS)

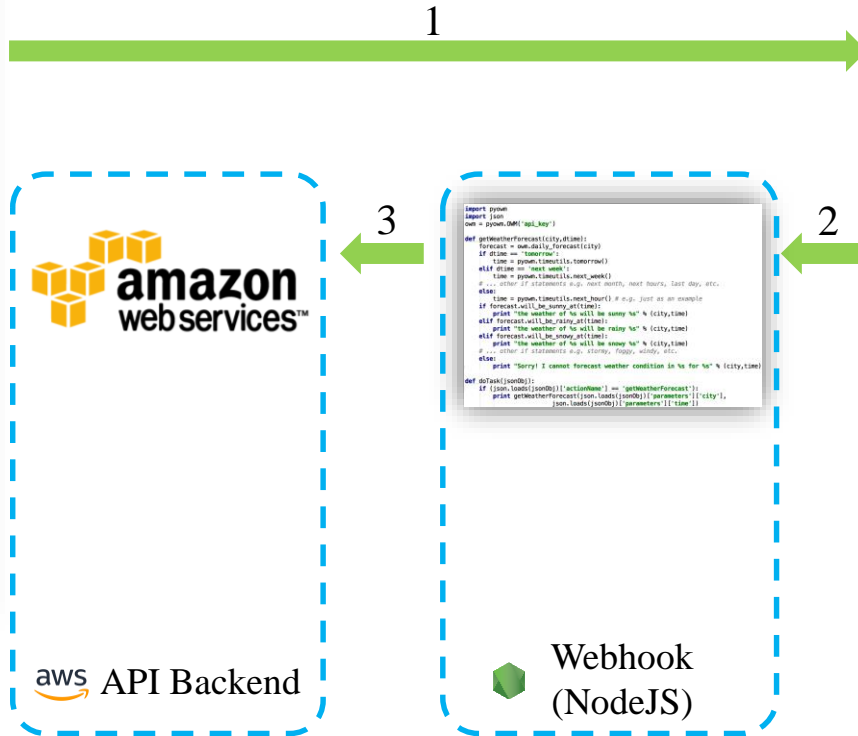


 Intent Recognizer
(Dialogflow)


DevOps bot



 Messaging Interface
(Messenger)



 API Backend

 Webhook
(NodeJS)

describe_instances

- do I have any instance
- can you describe instance with this id i-827299202
- Is instance instance i-817782982 running
- DNS info of instance i-829282001
- I need to know if I have any instance

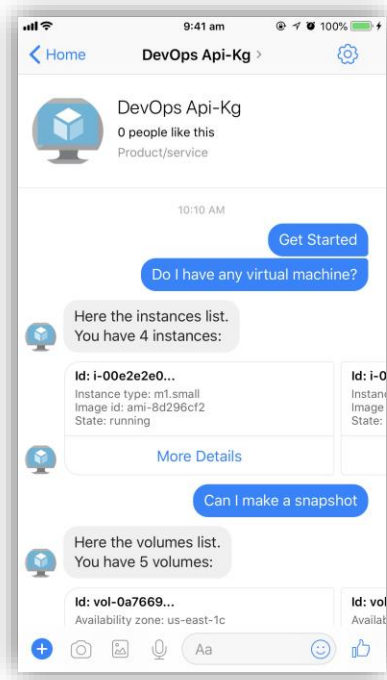
Action and parameters

describe_instances

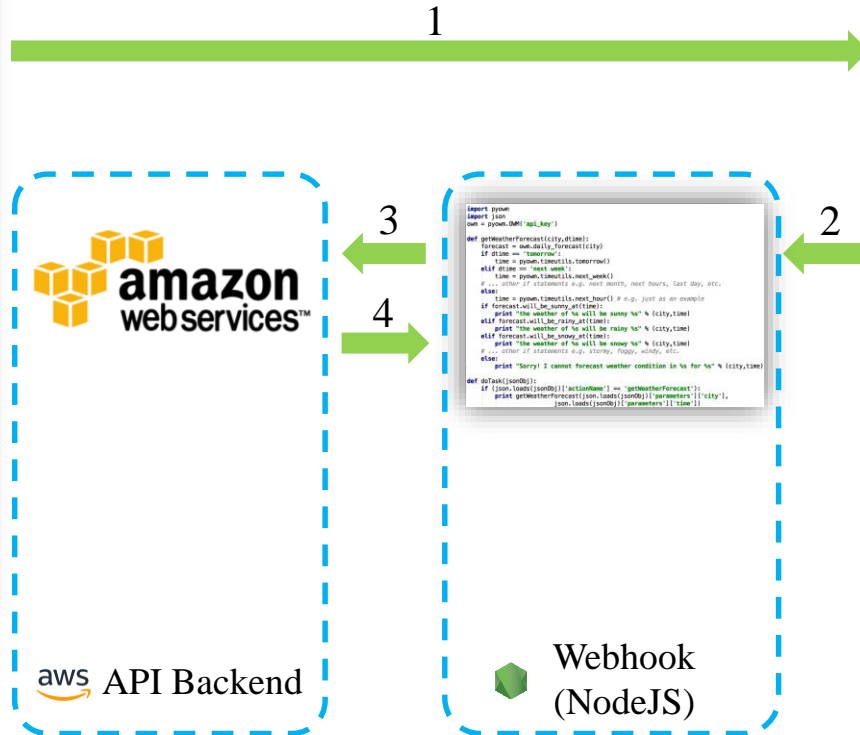
REQUIRED	PARAMETER NAME	ENTITY
<input type="checkbox"/>	instance_ids	@sys.any

 Intent Recognizer
(Dialogflow)


DevOps bot

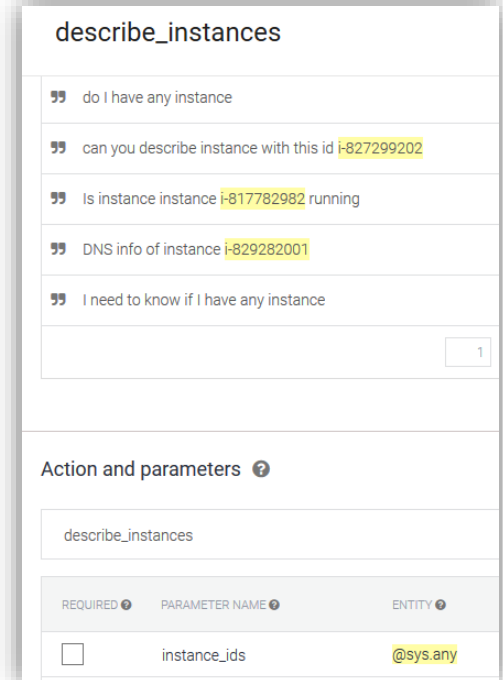


 Messaging Interface (Messenger)



 API Backend

 Webhook (NodeJS)



 Intent Recognizer (Dialogflow)

1

```
import axios
import json

let s = process.env['url_key']

def getWeatherForecast(city, dtTime):
    forecast = new DailyForecast(city)
    if dtTime == "tomorrow"
        time = pytime.timeutils.tomorrow()
    elif dtTime == "next week"
        time = pytime.timeutils.next_week()
    # ... other dtTime values e.g., next month, next hours, last day, etc.
    else
        time = pytime.timeutils.next_hour() # e.g., just as an example
    if forecast.will_be_sunny_at(time):
        print "The weather of %s will be sunny %s" % (city, time)
    elif forecast.will_be_raining_at(time):
        print "The weather of %s will be rainy %s" % (city, time)
    elif forecast.will_be_snowing_at(time):
        print "The weather of %s will be snowy %s" % (city, time)
    # ... other dtTime values e.g., cloudy, foggy, windy, etc.
    else
        print "Sorry! I cannot forecast weather condition in %s for %s" % (city, time)

def doTask(jobId):
    if (job.load(jobId) != "actionName") {
        print "getWeatherForecast(%s, %s, %s)" % (job.load(jobId), job.parameters["city"],
                                                    job.load(jobId) || parameters["time"])
    }
```

amazon web services™

aws API Backend


Webhook (NodeJS)

2

3

4

5

 Webhook
(NodeJS)

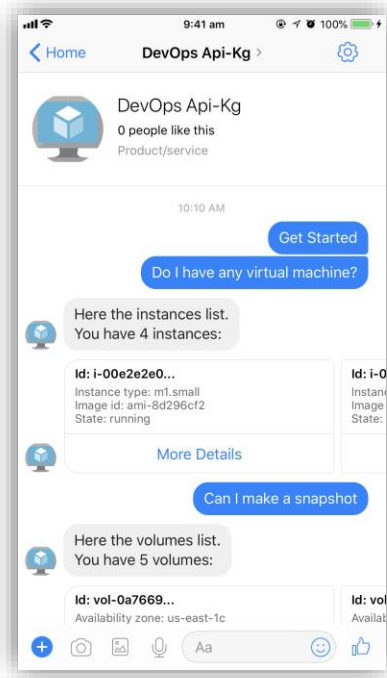
- do I have any instance
- can you describe instance with this id `i-827299202`
- Is instance `instance i-817782982` running
- DNS info of instance `i-829282001`
- I need to know if I have any instance

describe_instances

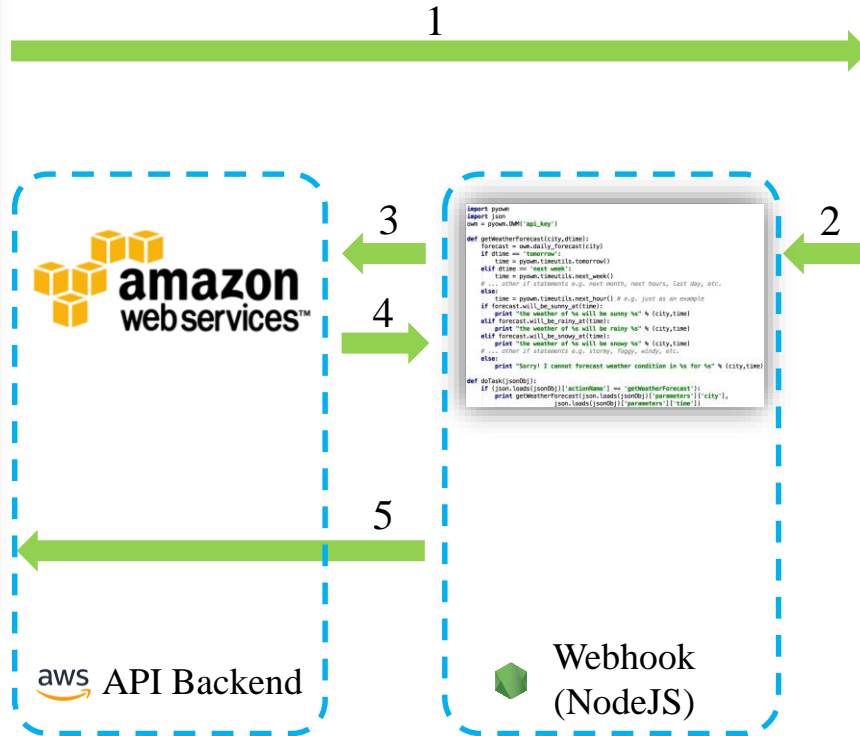
REQUIRED	PARAMETER NAME	ENTITY
	instance_ids	@sys.any

Intent Recognizer (Dialogflow)

DevOps bot



 Messaging Interface
(Messenger)



describe_instances

- do I have any instance
- can you describe instance with this id i-827299202
- Is instance instance i-817782982 running
- DNS info of instance i-829282001
- I need to know if I have any instance

Action and parameters ?

describe_instances

REQUIRED ?	PARAMETER NAME ?	ENTITY ?
<input type="checkbox"/>	instance_ids	@sys.any

 Intent Recognizer
(Dialogflow)

DevOps bot - Intents

1 OF 2	→
● authentication	
● create_image	
● create_key_pair	
● create_security_group	
● create_snapshot	
● create_volume	
🔖 Default Fallback Intent	
● Default Welcome Intent	
● delete_key_pair	
● delete_security_group	
● describe_availability_zones	
● describe_images	
● describe_instances	
● describe_key_pairs	
● describe_security_groups	
● describe_snapshots	
● describe_volumes	
● run_instance	
● start_instance	
● stop_instance	
1 OF 2	→

DevOps bot - Entities

@aws_zone

@image_name

@instance_id

@instance_type_entity

@key_pair_name

@security_group_name

DevOps bot - Training

Training phrases ?

Search training phrases 🔍

” Add user expression

” I need at least 3 instances for my work, create for me, use this image ami-127729200

” run 3 instances with this image ami-i1823720-29

” run instance with image id ami-1831248467, run 2 but if you have more resource make it 7

” I want you to build only 1 instance for me, image id is ami-1928467

” run 1 to 5 instances for this image ami-i1823720-29

” can you build a virtual machine for me, I need 1

” I need 4 instances or at least 2, use this image ami-i1812423219 and create all of them for me

” here is the image id ami-09211219, create 6 instance or at least 1

” this is the image ami-19242100, build 1 instance for me quickly

” I need a virtual machine, build it for me

1 OF 3

➔

DevOps bot - Conversation Manager

```
// This function run the specified instance with the specified parameters
var runInstance = function(senderId, agentName, generalContext){
    var ec2 = authParameterAWS_EC2();

    var imageId = generalContext.parameters.image_id;
    var maxCount = generalContext.parameters.max_count;
    var minCount = generalContext.parameters.min_count;
    var keyName = generalContext.parameters.key_pair_name;
    var instanceType = generalContext.parameters.instance_type;
    var zone = generalContext.parameters.zone;

    if (!minCount) {
        generalContext.parameters.request_parameter = 'min_count';
        generalContext.parameters.next_function = 'run_instance';
        var quickReplies = quickRepliesNumbers('payloadMinCount', 'number');
        var message = {
            'text': 'Give me the minimum number of instances, please.\nIf you need a different number, please specify only the number.',
            'quick_replies': quickReplies
        };
        sendTextToMessenger(senderId, message, agentName);
        saveContext(senderId, agentName, generalContext);
    } else if (!maxCount) {
        generalContext.parameters.request_parameter = 'max_count';
        generalContext.parameters.next_function = 'run_instance';
        var quickReplies = quickRepliesNumbers('payloadMaxCount', 'number');
```

DevOps bot - Conversation Manager

```
// This function terminate the specified instance with the specified parameters
var terminateInstance = function(senderId, agentName, generalContext){
    var ec2 = authParameterAWS_EC2();

    var instances = generalContext.parameters.instance_ids;

    if (typeof instances !== 'undefined' && instances.length > 0) {
        var params = {
            InstanceIds: instances,
        };

        ec2.terminateInstances(params, function(err, data) {
            if (err){
                var message = {
                    'text': err.message
                };
                console.log(err, err.stack); // an error occurred
                sendTextToMessenger(senderId,message,agentName);
            }
            else {
                generalContext.parameters.request_parameter = '';
                generalContext.parameters.next_function = '';
                generalContext.parameters.instance_ids = [];

                var terminateInstances = data.TerminatingInstances;
```

DevOps bot - Answer Generator

facebook for developers

Docs

Tools

Support

My Apps

Messenger Platform

Introduction

Getting Started

Messaging

Saving Assets

Quick Replies

Sender Actions

Message Tags

Handover Protocol

Templates

Generic Template

List Template

Button Template

Open Graph Template

Media Template

Receipt Template

Airline Templates

Buttons

Persistent Menu

Broadcast Messages

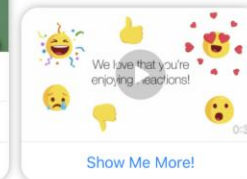
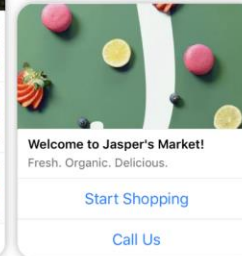
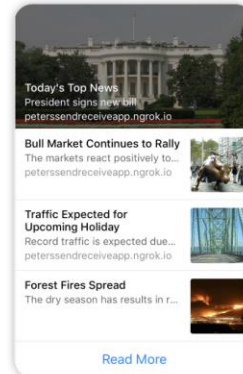
Webhooks

Webview

Payments (Beta)

Discovery & Re-engagement

Message Templates

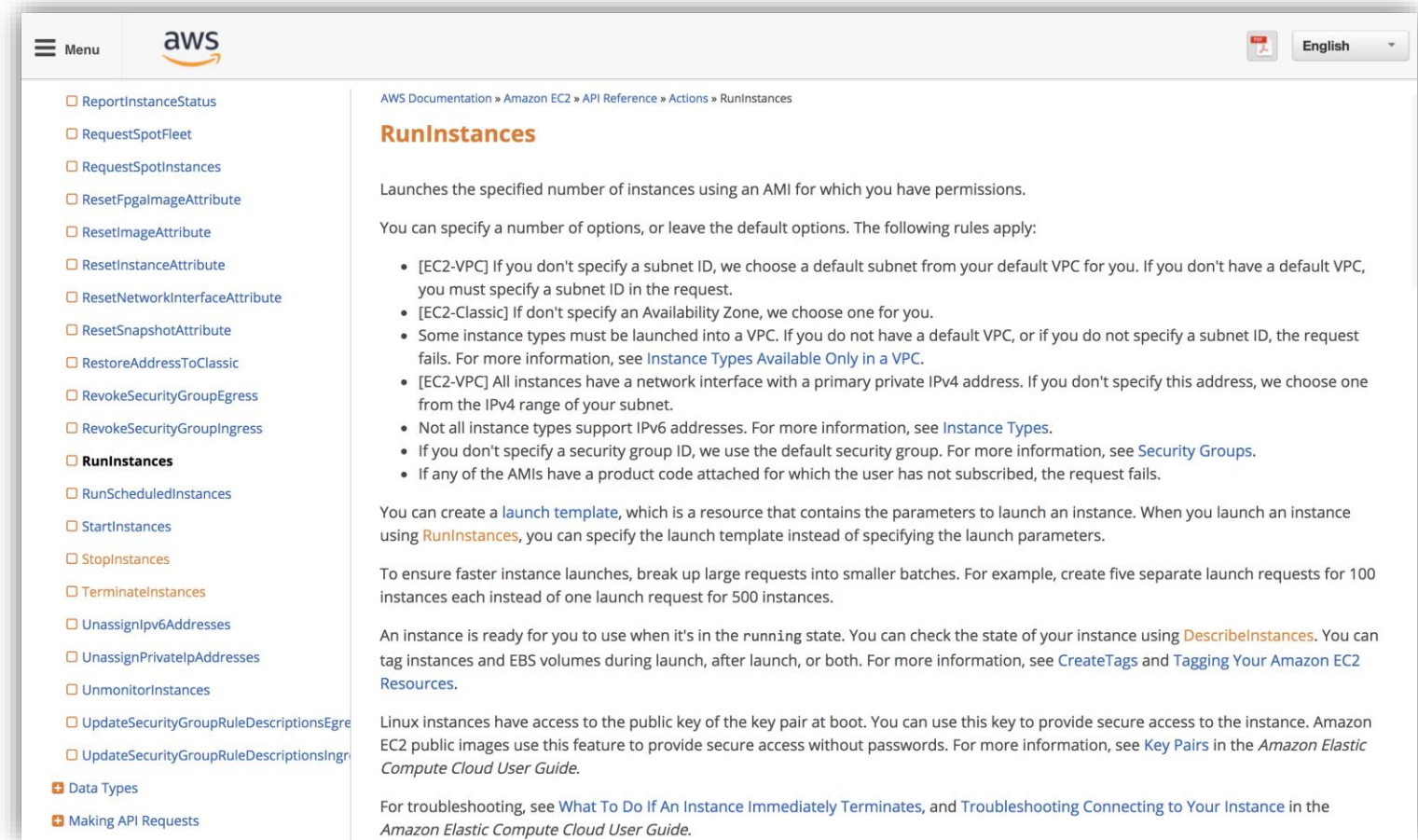


Message templates offer a way for you to offer a richer in-conversation experience than standard text messages by integrating buttons, images, lists, and more alongside text a single message. Templates can be use for many purposes, such as displaying product information, asking the message recipient to choose from a pre-determined set of options, and showing search results.

Contents

- [Available Templates](#)
- [Sending a Message Template](#)
- [Using Buttons](#)

Building Bot | API Integration - DevOps bot – API Endpoints



The screenshot shows the AWS documentation page for the `RunInstances` API action. The left sidebar contains a menu with various API actions, with `RunInstances` highlighted. The main content area displays the `RunInstances` title, a brief description, and a list of rules that apply to the action. The page also includes a breadcrumb trail, a language selector, and a 'Menu' button.

Menu

aws

English

AWS Documentation » Amazon EC2 » API Reference » Actions » RunInstances

RunInstances

Launches the specified number of instances using an AMI for which you have permissions.

You can specify a number of options, or leave the default options. The following rules apply:

- [EC2-VPC] If you don't specify a subnet ID, we choose a default subnet from your default VPC for you. If you don't have a default VPC, you must specify a subnet ID in the request.
- [EC2-Classical] If don't specify an Availability Zone, we choose one for you.
- Some instance types must be launched into a VPC. If you do not have a default VPC, or if you do not specify a subnet ID, the request fails. For more information, see [Instance Types Available Only in a VPC](#).
- [EC2-VPC] All instances have a network interface with a primary private IPv4 address. If you don't specify this address, we choose one from the IPv4 range of your subnet.
- Not all instance types support IPv6 addresses. For more information, see [Instance Types](#).
- If you don't specify a security group ID, we use the default security group. For more information, see [Security Groups](#).
- If any of the AMIs have a product code attached for which the user has not subscribed, the request fails.

You can create a [launch template](#), which is a resource that contains the parameters to launch an instance. When you launch an instance using `RunInstances`, you can specify the launch template instead of specifying the launch parameters.

To ensure faster instance launches, break up large requests into smaller batches. For example, create five separate launch requests for 100 instances each instead of one launch request for 500 instances.

An instance is ready for you to use when it's in the running state. You can check the state of your instance using [DescribeInstances](#). You can tag instances and EBS volumes during launch, after launch, or both. For more information, see [CreateTags](#) and [Tagging Your Amazon EC2 Resources](#).

Linux instances have access to the public key of the key pair at boot. You can use this key to provide secure access to the instance. Amazon EC2 public images use this feature to provide secure access without passwords. For more information, see [Key Pairs](#) in the *Amazon Elastic Compute Cloud User Guide*.

For troubleshooting, see [What To Do If An Instance Immediately Terminates](#), and [Troubleshooting Connecting to Your Instance](#) in the *Amazon Elastic Compute Cloud User Guide*.

ReportInstanceStatus

RequestSpotFleet

RequestSpotInstances

ResetFpgaImageAttribute

ResetImageAttribute

ResetInstanceAttribute

ResetNetworkInterfaceAttribute

ResetSnapshotAttribute

RestoreAddressToClassic

RevokeSecurityGroupEgress

RevokeSecurityGroupIngress

RunInstances

RunScheduledInstances

StartInstances

StopInstances

TerminateInstances

UnassignIpv6Addresses

UnassignPrivateIpAddresses

UnmonitorInstances

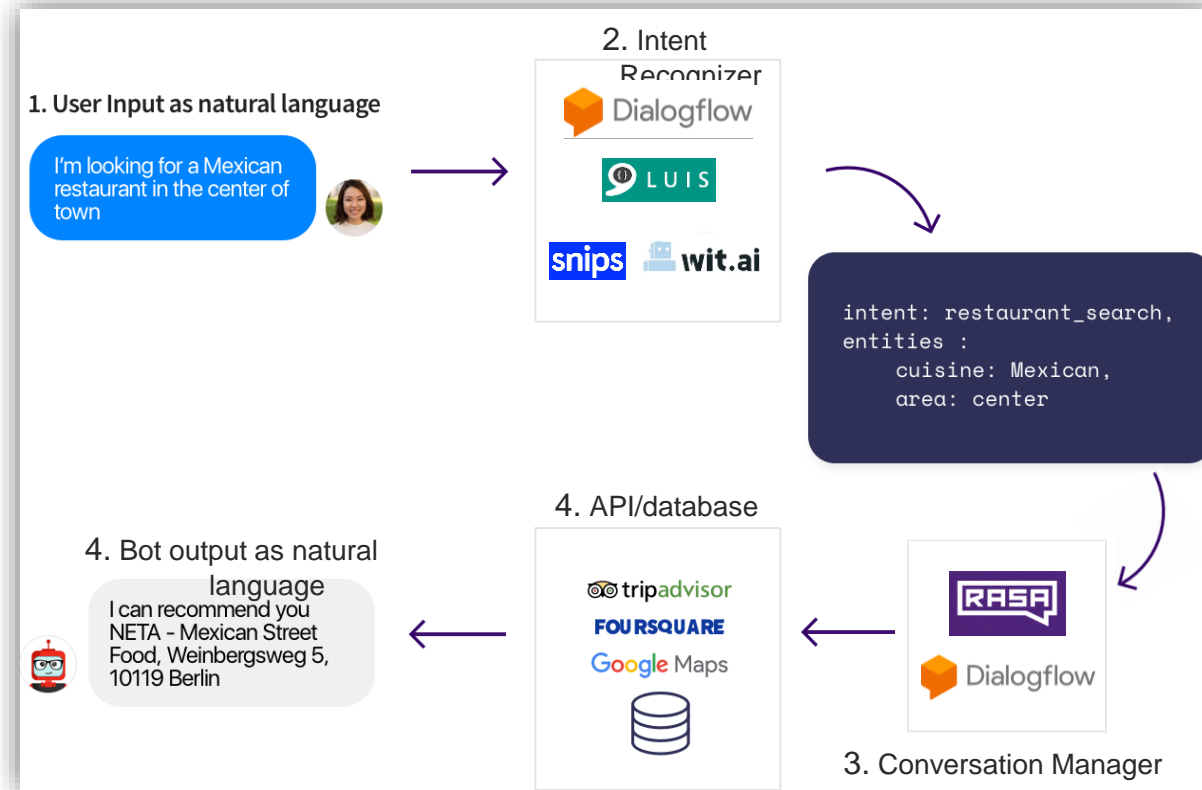
UpdateSecurityGroupRuleDescriptionsEgress

UpdateSecurityGroupRuleDescriptionsIngress

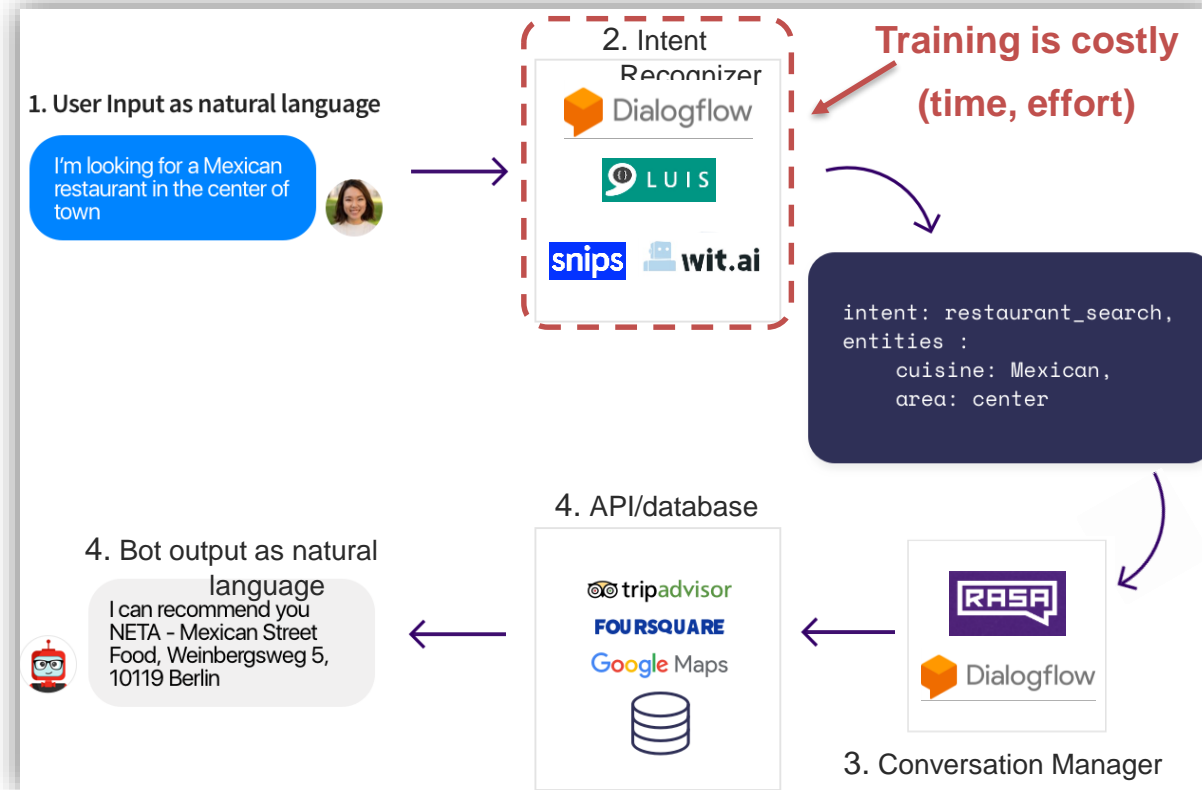
Data Types

Making API Requests

Machine Learning-Based Bots



Machine Learning-Based Bots



Intent|APIs Integration Challenges

APIs

- **APIs | Composition constructs are not meant to be intuitive** (variables, control flow, symbols)

Intent|APIs Integration Challenges

APIs

- **APIs | Composition constructs are not meant to be intuitive** (variables, control flow, symbols)
- **Low-level API description models** (mostly XML, code snippets)

Intent|APIs Integration Challenges

APIs

- **APIs | Composition constructs are not meant to be intuitive** (variables, control flow, symbols)
- **Low-level API description models** (mostly XML, code snippets)

Bots

- User Expressions are **fuzzy** (Natural Language - NL)

Intent|APIs Integration Challenges

APIs

- **APIs | Composition constructs are not meant to be intuitive** (variables, control flow, symbols)
- **Low-level API description models** (mostly XML, code snippets)

Bots

- User Expressions are **fuzzy** (Natural Language - NL)
- API description: **limited knowledge, short NL description, low-level interfaces**

Intent|APIs Integration Challenges

APIs

- **APIs | Composition constructs are not meant to be intuitive** (variables, control flow, symbols)
- **Low-level API description models** (mostly XML, code snippets)

Bots

- User Expressions are **fuzzy** (Natural Language - NL)
- API description: **limited knowledge, short NL description, low-level interfaces**

“Mapping Intents to APIs requires more knowledge than API descriptions”

Where we are now

- SOA
- Web Services and Web APIs
- Microservices
- Cloud Services
- Cognitive Services

How do they affect Organizations?

IT Architecture

- An IT architecture provides the necessary technical foundation for an effective IT strategy, which is the core of any successful modern business strategy
- Specifically, an IT architecture defines the **components or building blocks** that make up the overall information system. It provides a plan from which products can be procured, and systems developed that will work together to implement the overall system. It thus enables you to manage your IT investment in a way that meets the needs of your business.

So How is this Relevant to What We have Learned

- Understanding software development styles and technologies, their benefits, opportunities and limitations can help in:
 - Defining the principles that guide technology decisions for the enterprise
 - Analysis of the current IT ecosystem to detect critical deficiencies and recommend solutions for improvement
 - Development of technology infrastructure that can be highly leveraged across multiple enterprise solutions
 - And more...

Questions?