

Building User Interfaces

Dialogflow 1

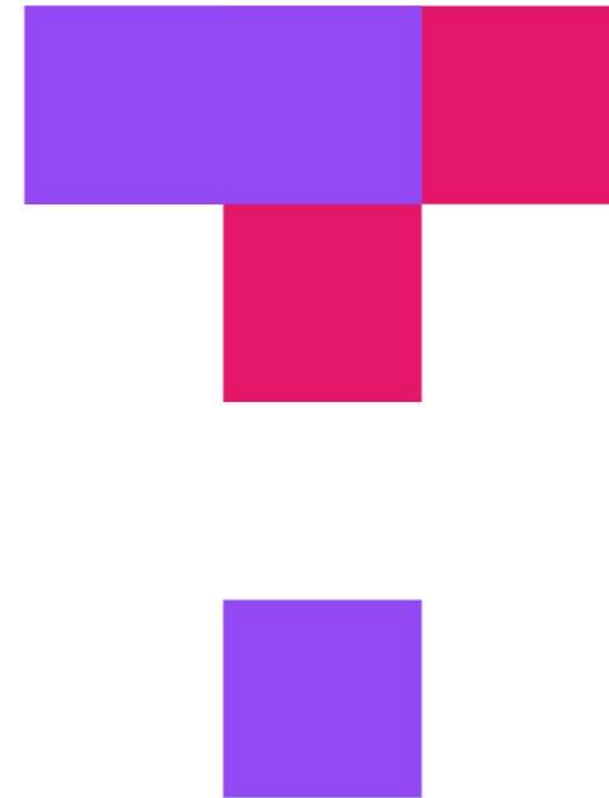
An Introduction

Professor Bilge Mutlu

What we will learn today?

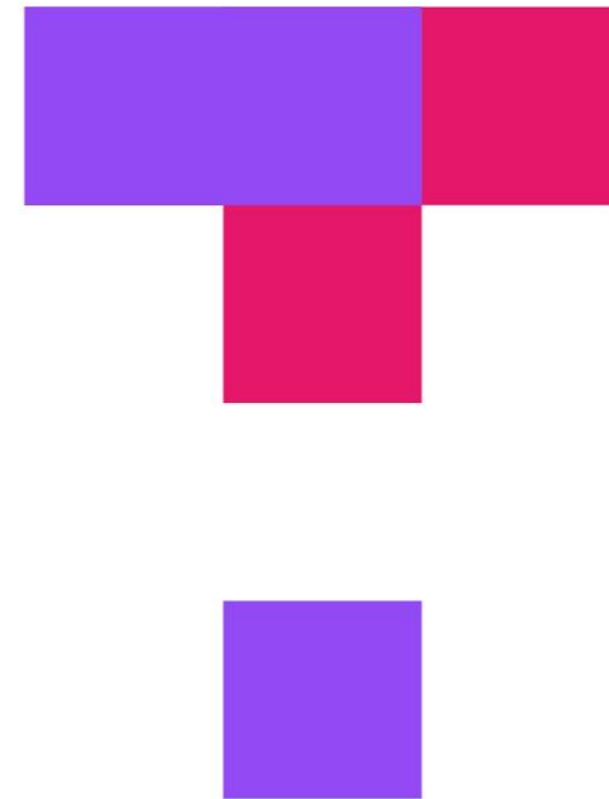
- >> Introduction to Conversational Interfaces
- >> Introduction to Dialogflow
- >> Dialogflow Building Blocks, Part 1
- >> Let's Make an Agent
- >> Assignment Preview

TopHat Attendance



TOP HAT

TopHat Questions



TOP HAT

Introduction to Conversational Interfaces

What is a conversational interface?

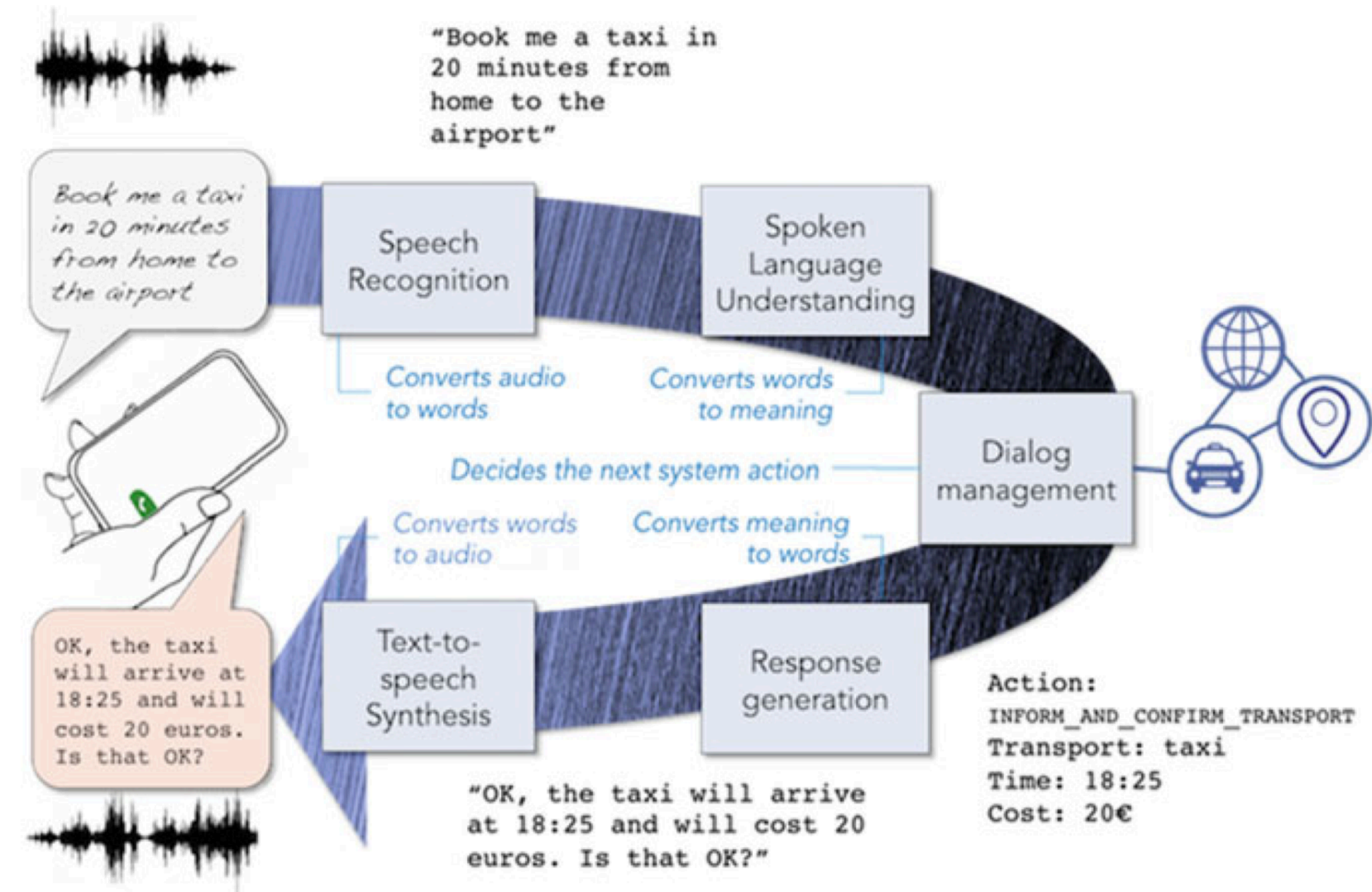
Definition: "Technology that supports conversational interaction with these VPAs [virtual personal assistants] by means of speech and other modalities."¹

¹McTear et al., 2016, *The Conversational Interface*

Components of Conversational Interfaces²

Conversational interfaces integrate a number of technologies:

- >> Speech recognition
- >> Spoken language understanding
- >> Dialog management
- >> Response generation
- >> Text-to-speech synthesis



²Image source: [McTear et al., 2016, The Conversational Interface](#)

The First Conversational Interface³

ELIZA was the first conversational interface developed by Joseph Weizenbaum at the MIT Artificial Intelligence Laboratory in 1964-1966.

```
Welcome to
                EEEEEEE LL      IIII  ZZZZZZZZ  AAAAA
                EE      LL      II     ZZ      AA  AA
                EEEEE  LL      II     ZZZ     AAAAAA
                EE      LL      II     ZZ      AA  AA
                EEEEE  LLLLLL IIII  ZZZZZZZZ  AA  AA

Eliza is a mock Rogerian psychotherapist.
The original program was described by Joseph Weizenbaum in 1966.
This implementation by Norbert Landsteiner 2005.

ELIZA: Is something troubling you ?
YOU:   Men are all alike.
ELIZA: What is the connection, do you suppose ?
YOU:   They're always bugging us about something or other.
ELIZA: Can you think of a specific example ?
YOU:   Well, my boyfriend made me come here.
ELIZA: Is it important to you that your boyfriend made you come here ?
YOU:   He says I'm depressed much of the time.
ELIZA: I am sorry to hear that you are depressed.
YOU:   It's true. I am unhappy.
ELIZA: Can you explain what made you unhappy ?
YOU:   █
```

³Image source: [Wikipedia: ELIZA](#)

Types of Conversational Interfaces

- >> Spoken dialog systems
- >> Voice user interfaces
- >> Embodied conversational agents
- >> Social robots
- >> Chatbots

Platforms for Building Conversational Interfaces

In this course, we will be using Dialogflow by Google.

However, there are alternatives:

- >> Amazon Lex (Commercial)
- >> Mycroft (Open Source)
- >> CoreNLP (Academic)

Introduction to Dialogflow

What is Dialogflow?

Dialogflow is an end-to-end, build-once deploy-everywhere development suite for conversational interfaces for websites, mobile applications, and IoT devices (e.g., smart speakers).

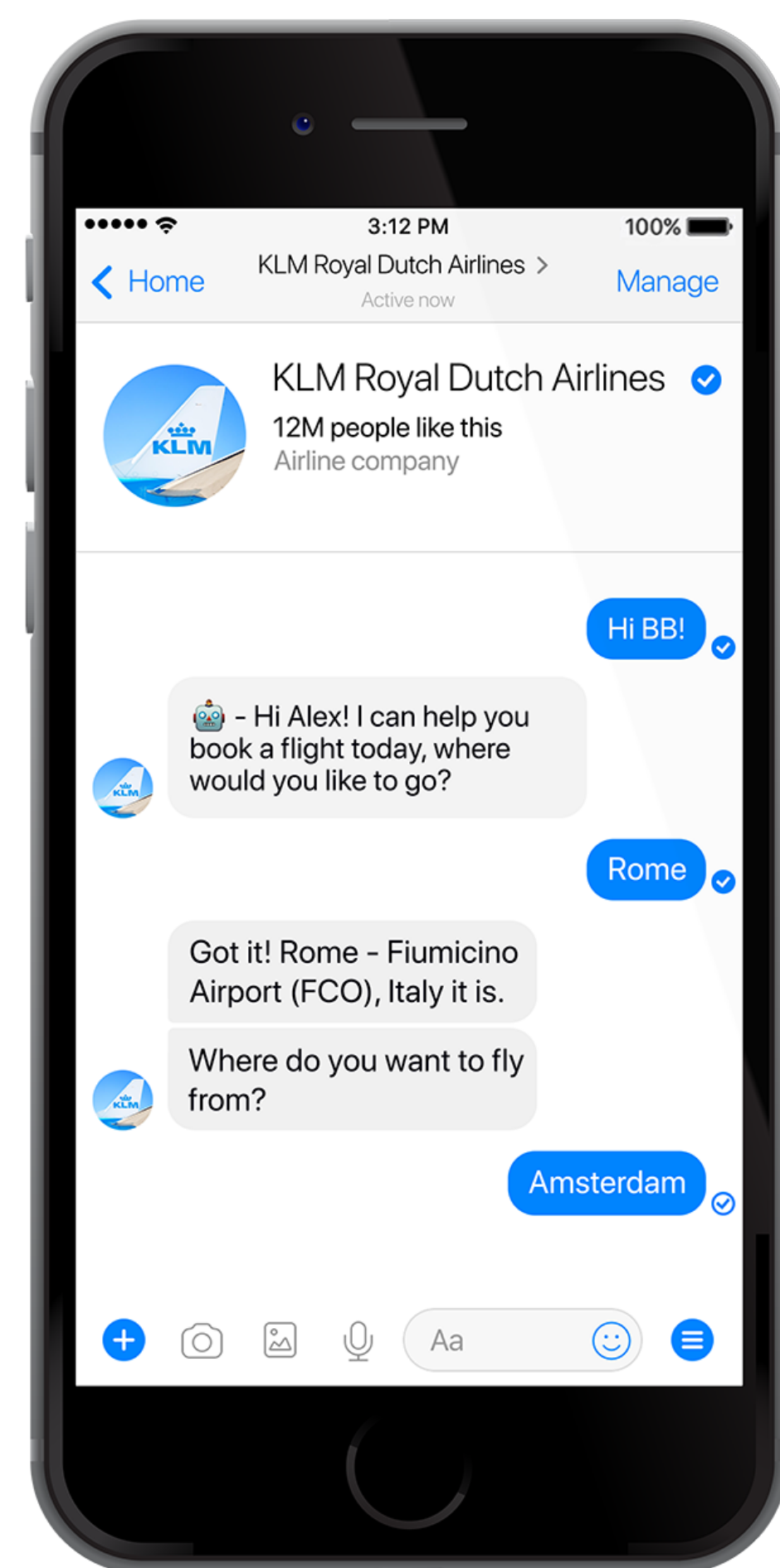
Intro to Dialogflow



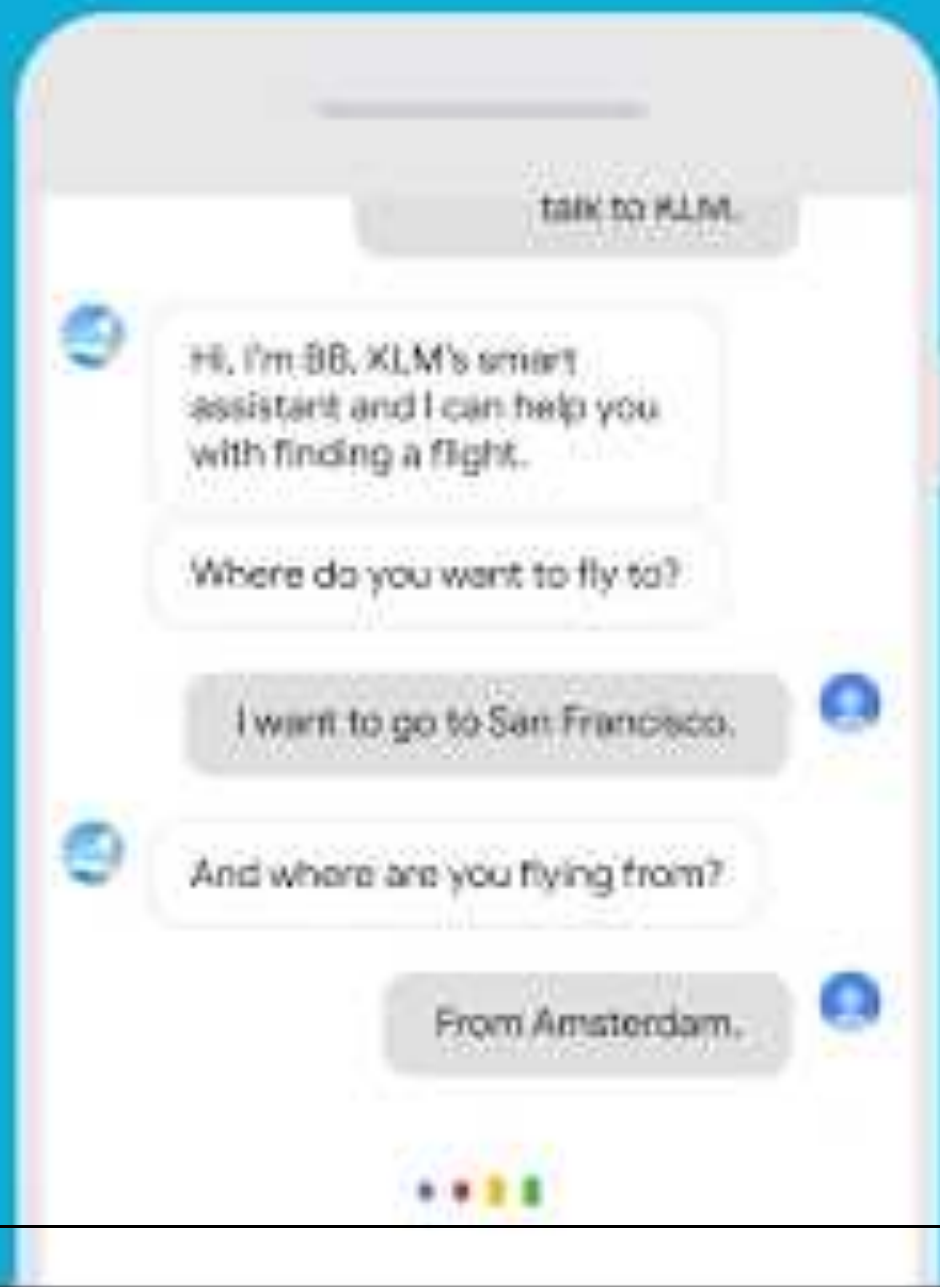
Case Study: KLM BB⁵

KLM used Dialogflow to create an agent to purchase travel as well as travel preparation.

Let's see how the KLM BB works...



⁵Image source [Dialogflow](#)





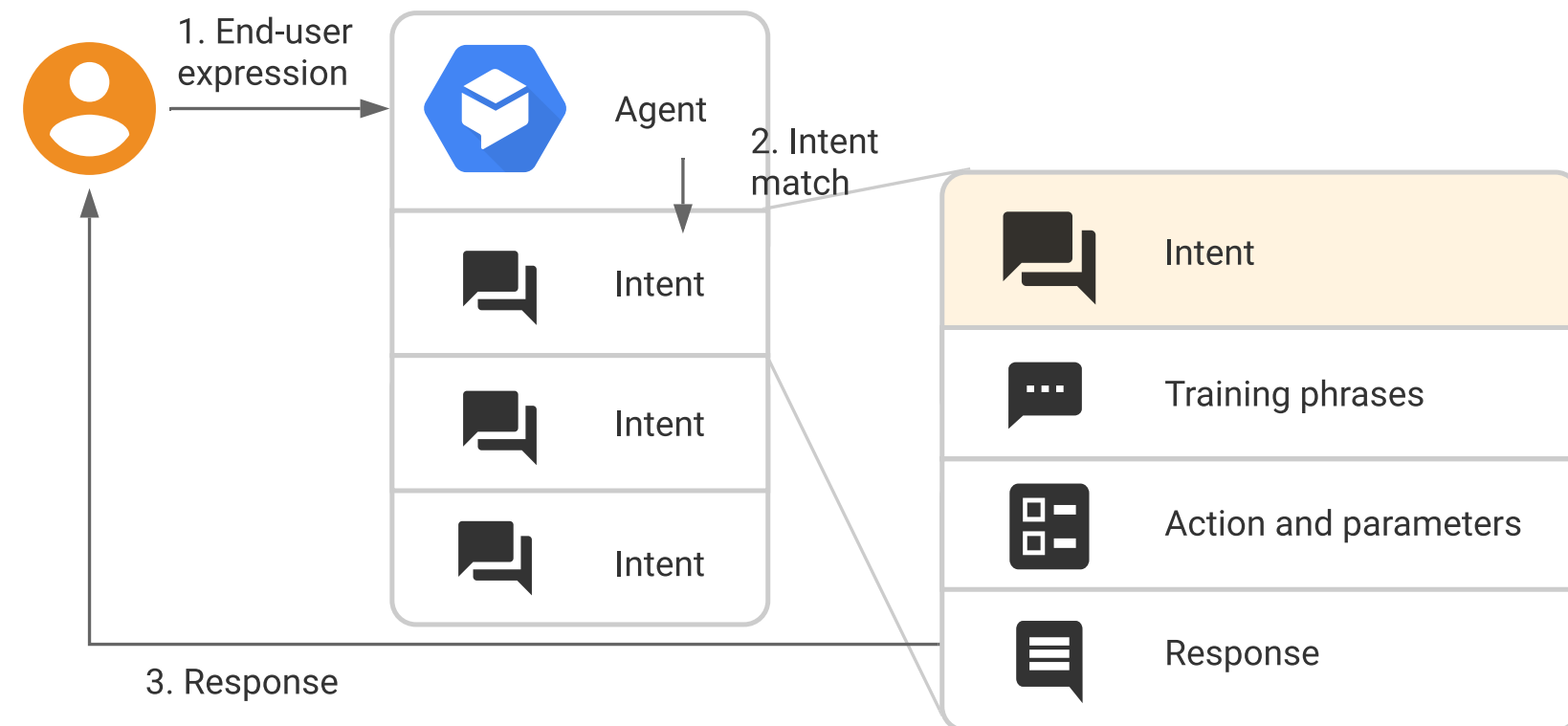
urco

Kit
KEM

How does Dialogflow work?⁸

The process within Dialogflow involves:

1. User expression
2. Intent matching
3. System response



⁸[Image source](#)

What is an *agent*?

Definition: A Dialogflow agent is a virtual agent that handles conversations with users (similar to a human call agent).

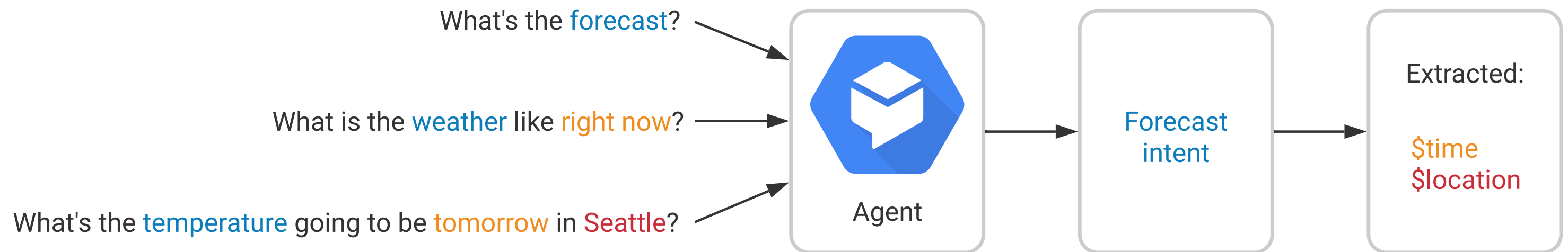


Image source

Agents are high-level containers for a number of building blocks:

>> Agent settings

>> Intents

>> Entities

>> Knowledge

>> Integrations

>> Fulfillment

The End-to-end Dialogflow Workflow

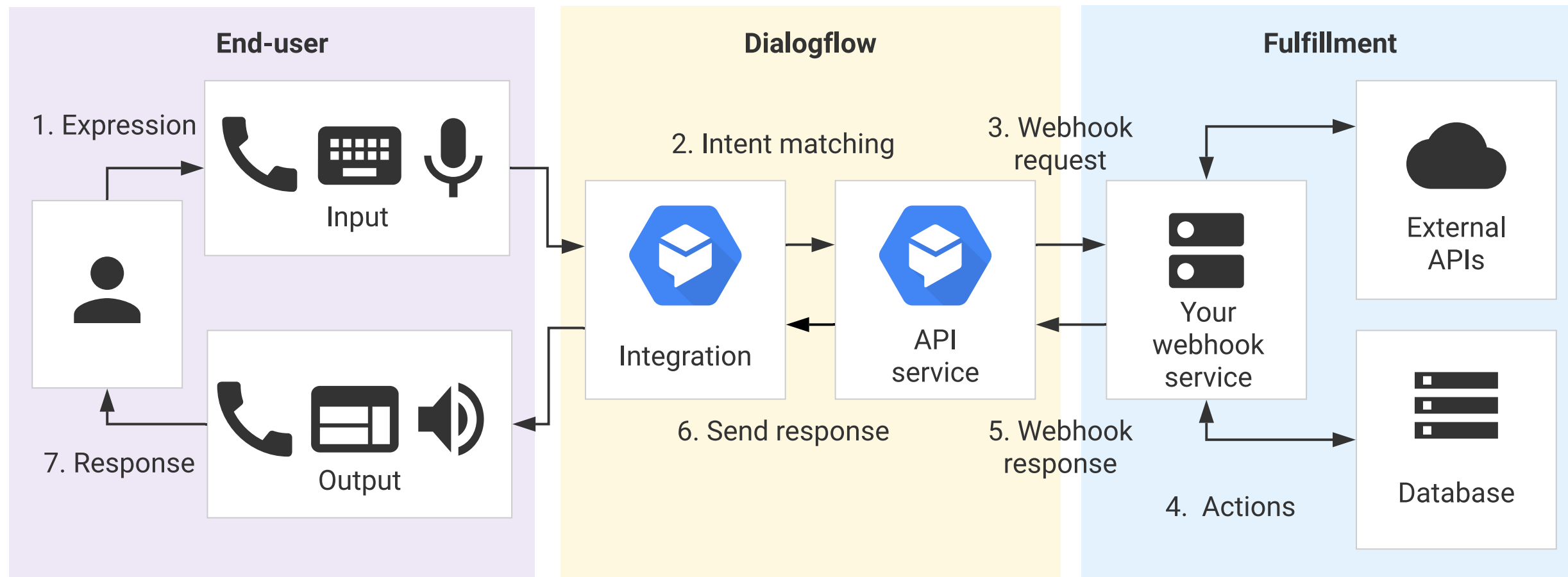
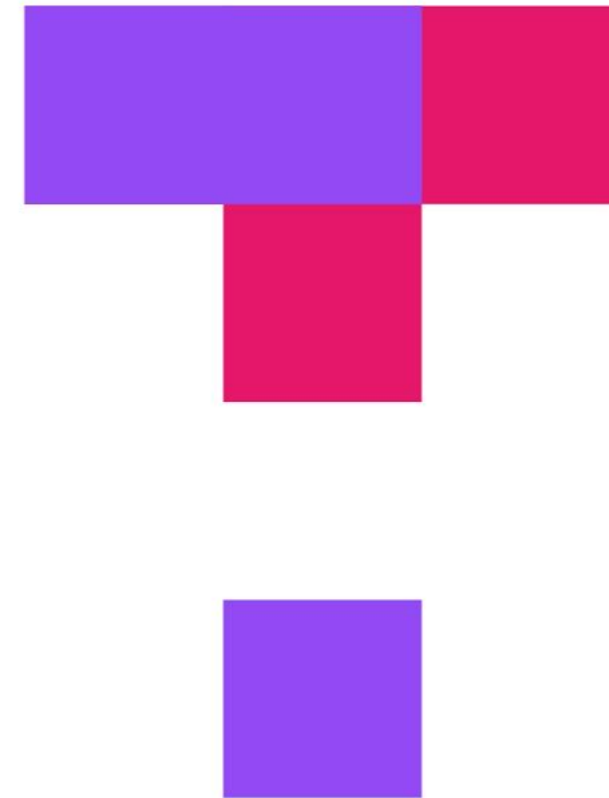


Image source

TopHat Quiz



TOP HAT

Dialogflow Building Blocks, Part 1

We will cover Part 2 in the next class.

Getting Started with Dialogflow

1. Login to the Dialogflow
2. Go to the Dialogflow console
3. Create a new *agent*

Agent Settings

Get familiar with agent settings.

The screenshot shows the 'Agent Settings' page for 'CS639DemoAgent'. The page is divided into several sections:

- General:** Includes a description field ('Describe your agent'), a default time zone dropdown (set to '(GMT-6:00) America/Chicago'), and a note that date and time requests are resolved using this timezone.
- GOOGLE PROJECT:** A table showing the Project ID ('cs639demoagent-jmmvnmw') and Service Account ('dialogflow-cpgcaj@cs639demoagent-jmmvnmw.iam.gserviceaccount.com').
- API VERSION:** A radio button selection for 'V2 API', which is currently selected. A note states: 'Use Cloud API as default for the agent. Your webhook will receive and return V2 format messages.'
- BETA FEATURES:** A toggle switch for 'Enable beta features and APIs' is currently turned off. A note says: 'Be the first to get access to the newest features and latest APIs. (Full V2-beta API reference)'
- API KEYS (V1):** A table showing Client access token ('85c321bebf844e1bbe1732b6d1419b8') and Developer access token ('4aa35bdac21b402fab22caef2f675662').
- LOG SETTINGS:** Two toggle switches: 'Log interactions to Dialogflow' (checked) and 'Log interactions to Google Cloud' (unchecked). A note for the first toggle says: 'Collect and store user queries. Logging must be enabled in order to use Training, History and Analytics.'

At the bottom, there is a 'DANGER ZONE' section with a warning icon and the text 'Delete Agent'. Below this, it asks: 'Are you sure you want to delete agent CS639DemoAgent? This will destroy the agent with all corresponding data and cannot be undone!'. A red button labeled 'DELETE THIS AGENT' is visible.

On the right side of the page, there is a 'Try it now' button and a microphone icon. Below that, there is a message: 'Please use test console above to try a sentence.' and a link: 'See how it works in Google Assistant.'

Agent Exporting

The screenshot displays the Dialogflow console interface for a 'RobotPlanner' agent. The left sidebar contains navigation options: RobotPlanner (with a gear icon), en (language), Intents, Entities, Knowledge [beta], Fulfillment, Integrations, Training, Validation [beta], History, Analytics, and Prebuilt Agents. The main content area is titled 'RobotPlanner' and includes a 'SAVE' button and a 'Try it now' button with a microphone icon. Below the title, there are tabs for 'General', 'Languages', 'ML Settings', 'Export and Import' (which is active), 'Speech', 'Share', and 'Advanced'. The 'Export and Import' section features three blue buttons: 'EXPORT AS ZIP' (with the description 'Create a backup of the agent'), 'RESTORE FROM ZIP' (with the description 'Replace the current agent version with a new one. All the intents and entities in the older version will be deleted.'), and 'IMPORT FROM ZIP' (with the description 'Upload new intents and entities without deleting the current ones. Intents and entities with the same name will be replaced with the newer version.'). On the right side, there is a 'Try it now' section with an information icon and the text 'Please use test console above to try a sentence.', followed by a link to 'See how it works in Google Assistant' with an external link icon.

Agent Speech

The screenshot displays the Dialogflow RobotPlanner interface. On the left is a navigation sidebar with the Dialogflow logo and menu items: RobotPlanner (selected), Intents, Entities, Knowledge [beta], Fulfillment, Integrations, Training, Validation [beta], History, Analytics, Prebuilt Agents, and Small Talk. The main content area is titled 'RobotPlanner' and has a 'SAVE' button. Below the title are tabs for General, Languages, ML Settings, Export and Import, Speech (selected), Share, and Advanced. The 'Speech' tab contains two sections: 'IMPROVE SPEECH RECOGNITION QUALITY' and 'TEXT TO SPEECH'. The first section has two toggle switches: 'Enable Enhanced Speech Models and Data Logging (available for Enterprise Edition)' and 'Enable Auto Speech Adaptation [beta]'. The second section has a toggle for 'Enable Automatic Text to Speech'. Below this is a dropdown for 'Output Audio Encoding' set to '16 bit linear PCM (signed, little-end...)'. The 'VOICE CONFIGURATION' section includes a dropdown for 'Agent Language' set to 'en (English)' and a partially visible 'Voice' dropdown.

Dialogflow

RobotPlanner

en

Intents

Entities

Knowledge [beta]

Fulfillment

Integrations

Training

Validation [beta]

History

Analytics

Prebuilt Agents

Small Talk

RobotPlanner

SAVE

Try it now

General Languages ML Settings Export and Import **Speech** Share Advanced

IMPROVE SPEECH RECOGNITION QUALITY

Enable Enhanced Speech Models and Data Logging (available for Enterprise Edition)
By enabling data logging, you agree to the [terms and conditions](#) of the data logging agreement (which amends the agreement governing Customer's use of Google Cloud Platform services). I represent and warrant that I have authority to bind Customer to this data logging agreement. Please note that this setting will affect all data from this point onwards and does not impact prior logged data.

Enable Auto Speech Adaptation [beta]
Use Dialogflow agent information (e.g. intents, entities) to automatically improve speech recognition quality.

TEXT TO SPEECH

Enable Automatic Text to Speech
Automatically convert default text responses to speech in all conversations. The output audio will be included in [DetectIntentResponse](#) and [StreamingDetectIntentResponse](#).

Output Audio Encoding

16 bit linear PCM (signed, little-end... ▾)

VOICE CONFIGURATION

Configure your agent's synthesized voice in the V2 API and Telephony integration.

Agent Language

en (English) ▾

Voice

Please use test console above to try a sentence.

See how it works in [Google Assistant](#).

Intents

What are intents?

Consider the following user requests:

- >> What is the weather like today?
- >> Will it rain sometime today?

What is the intent of these requests?

What are intents?

Definition: Intents are the goals of the user that are expressed to the agent.

In the previous examples, despite their different framing, the user was expressing a desire to know what the weather will be.

That is their *intent*.

More Intent Examples

Let's look at some more requests and identify their intents:

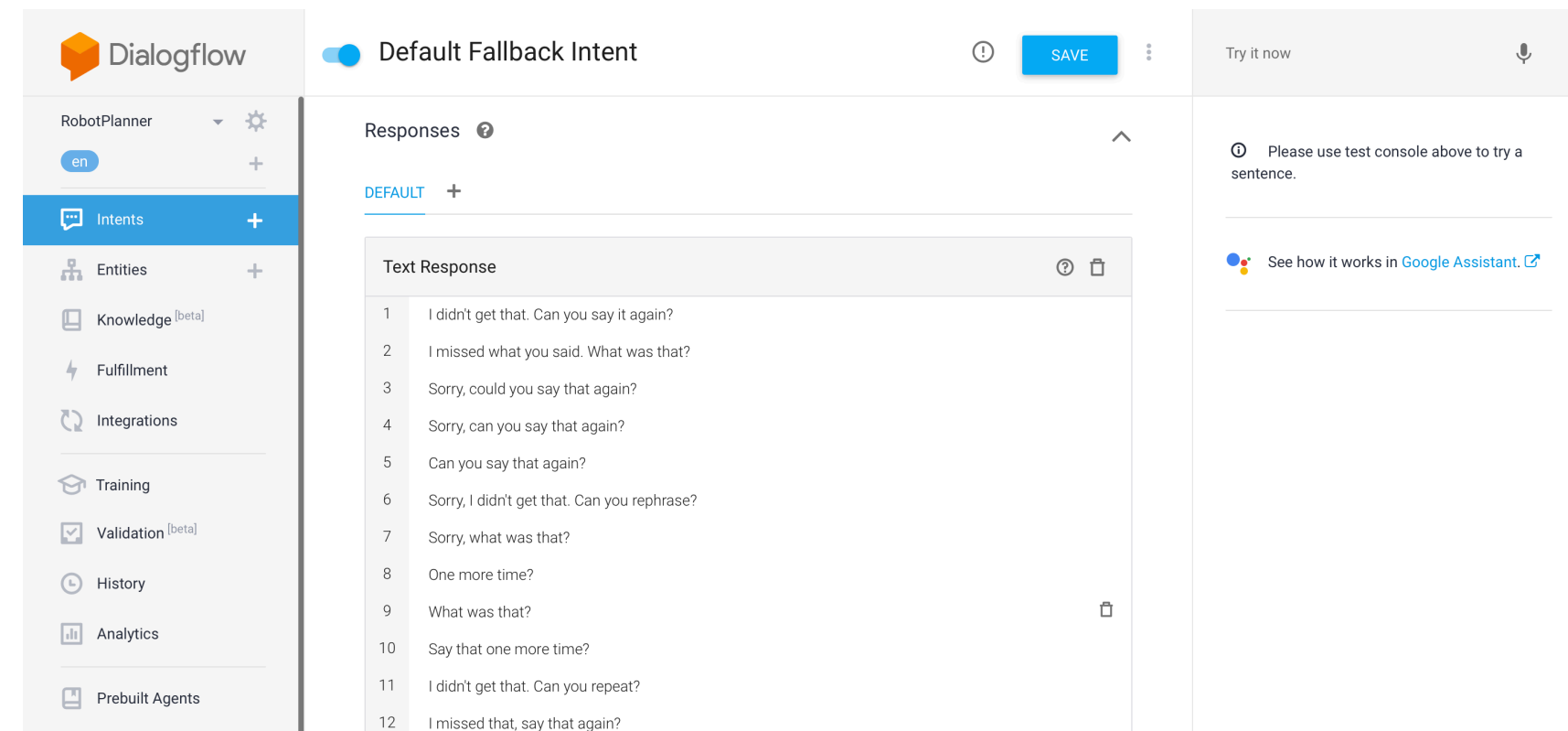
- >> How are you?
- >> How do I get to Middleton?
- >> What is the price of a basketball?
- >> Buy one box of tissues from Amazon.

Intents In Dialogflow

The screenshot shows the Dialogflow web interface. On the left is a sidebar with navigation options: RobotPlanner, Intents (selected), Entities, Knowledge [beta], Fulfillment, Integrations, Training, Validation [beta], History, and Analytics. The main area is titled 'Intents' and contains a 'CREATE INTENT' button, a search bar, and a list of two default intents: 'Default Fallback Intent' and 'Default Welcome Intent'. Below the list is an information icon and a message: 'No regular intents yet. [Create the first one.](#)' followed by an explanatory paragraph: 'Intents are mappings between a user's queries and actions fulfilled by your software. [Read more here.](#)' and another paragraph: 'Before you start, check out [Prebuilt Agents](#), a collection of agents developed by the Dialogflow team.' On the right side, there is a 'Try it now' button and a message: 'Please use test console above to try a sentence.' with a link to 'Set-up Google Assistant integration.'

Default Fallback Intents

- >> Engaged if no other intents are recognized.
- >> Could be a result of not being able to "hear" the user, or unable to parse what they said.
- >> Can provide training examples of things that will serve as negative examples for your desired intents.



Creating Intents

- >> Allow the user to say that they want a robot to pick something up.
- >> We will start with training phrases.
- >> Should try to create at least 10.

The screenshot displays the Dialogflow console interface for creating a new intent. On the left is a navigation sidebar with options like RobotPlanner, Intents, Entities, Knowledge, Fulfillment, Integrations, Training, Validation, History, Analytics, Prebuilt Agents, Small Talk, and Docs. The main content area is titled 'Intent name' and features a 'SAVE' button. Below the title are sections for 'Contexts', 'Events', 'Training phrases', 'Action and parameters', and 'Responses'. The 'Training phrases' section is currently expanded, providing instructions on how to train the intent with user phrases and a button to 'ADD TRAINING PHRASES'. The 'Action and parameters' section is also expanded, explaining how to extract actions and parameters from user requests and a button to 'ADD PARAMETERS AND ACTION'. On the right side of the console, there is a 'Try it now' button and a note about using the test console.

Creating Intents

- >> Can you get the screwdriver for me?
- >> Please get the green ball.
- >> Pick up that red cube.
- >> etc.

Notice how the color is highlighted? More on that next.

• Pickup Command

SAVE

Training phrases ?

Search training phrases 🔍 ^

” Add user expression

” Take that.

” Grab this from me.

” Take this wrench

” Pick up the hammer over there.

” How about getting that box of screws for me?

” Can you get the screwdriver for me?

” Please get the **green** ball 

” Get the **green** ball

” Grab the toy

” Pick up that **red** cube

Test

Test your agent using the *Default Welcome Intent*

The screenshot displays the Dialogflow console interface. On the left, a sidebar lists various management tools: Intents (selected), Entities, Knowledge [beta], Fulfillment, Integrations, Training, Validation [beta], History, Analytics, Prebuilt Agents, Small Talk, Docs, Standard Free (with an Upgrade button), Support, Account, and Logout. The main content area is titled 'Intents' and features a 'CREATE INTENT' button. A search bar is present above a list of intents, which currently shows 'Default Fallback Intent' and 'Default Welcome Intent'. Below the list, an information icon is followed by the text: 'No regular intents yet. [Create the first one.](#) Intents are mappings between a user's queries and actions fulfilled by your software. [Read more here.](#) Before you start, check out [Prebuilt Agents](#), a collection of agents developed by the Dialogflow team.' The right sidebar, under 'Try it now', includes a link to 'See how it works in Google Assistant', an 'Agent' section, a 'USER SAYS' section with the input 'Hello!', a 'DEFAULT RESPONSE' section with the output 'Hi! How are you doing?', an 'INTENT' section showing 'Default Welcome Intent', an 'ACTION' section showing 'input.welcome', and a 'DIAGNOSTIC INFO' button.

Entities

What are entities?

Let's consider those requests again.

- >> How do I get to *Middleton*?
- >> What is the price of a *basketball*?
- >> Buy *one* box of *tissues* from *Amazon*.

What are entities?

Sometimes, users' intents are more specific, and have an intent based around a certain item or *entity*.

Definition: Entities allow for more specificity of requests, without exploding the intent space.

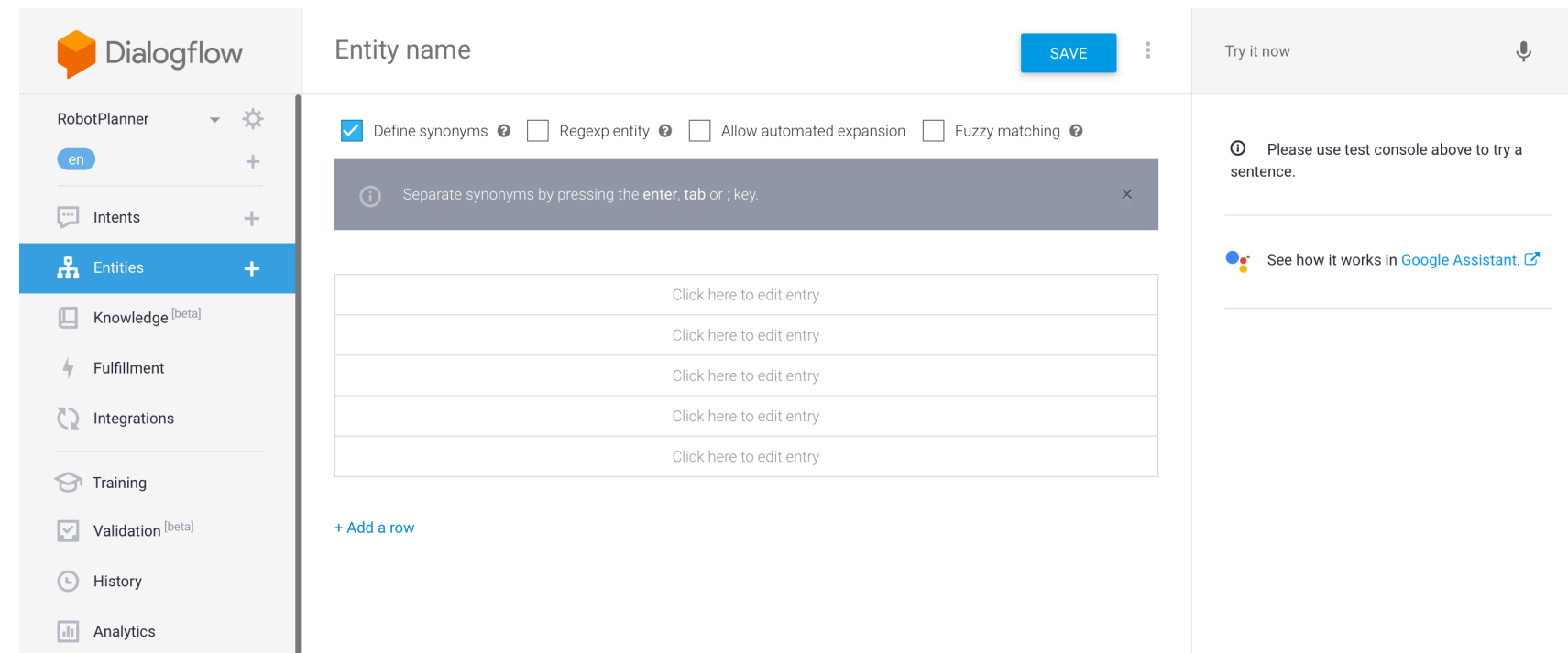
Thus, if the request was:

What is the weather like today in Seattle?

The *intent*: weather inquiry; *entity*: Seattle

Entities in Dialogflow

Let's define some things that the robot can pick up.



The screenshot displays the Dialogflow console interface for configuring an entity. On the left is a navigation sidebar with options: RobotPlanner, Intents, Entities (selected), Knowledge [beta], Fulfillment, Integrations, Training, Validation [beta], History, and Analytics. The main area is titled 'Entity name' and includes a 'SAVE' button. Below the title are configuration options: 'Define synonyms' (checked), 'Regex entity', 'Allow automated expansion', and 'Fuzzy matching'. A grey tooltip box contains the instruction: 'Separate synonyms by pressing the enter, tab or ; key.' Below this is a table with five rows, each containing a 'Click here to edit entry' link. At the bottom left of the main area is a '+ Add a row' link. On the right side, there is a 'Try it now' button with a microphone icon and a test console area with a message: 'Please use test console above to try a sentence.' and a link: 'See how it works in Google Assistant.'

Entities in Dialogflow

Remember I mentioned some objects when creating my intents.
Let's add those here.

>> Cube

>> Sphere

>> Screwdriver

>> etc.

Entity Entries and Synonyms

- >> Cube (Box)
- >> Container (Box, Bin)
- >> Sphere (Ball)
- >> Screwdriver

object

SAVE

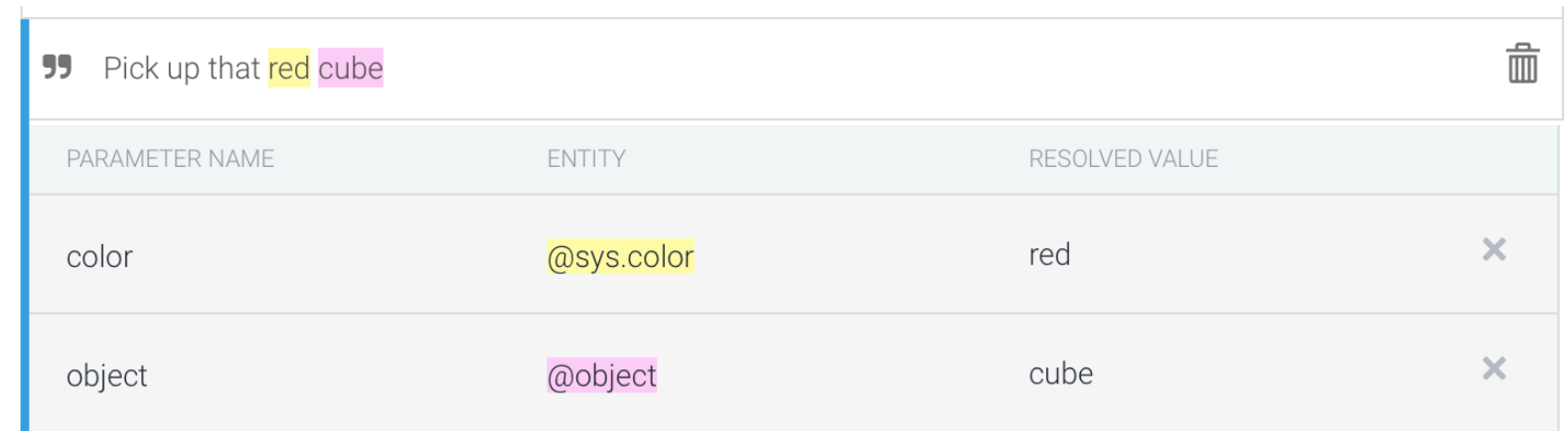
Define synonyms [?] Regexp entity [?] Allow automated expansion Fuzzy matching [?]

cube	cube
container	container, box, bin
sphere	sphere, ball
screwdriver	screwdriver
hammer	hammer, mallet
wrench	<input type="text" value="wrench"/> Enter synonym
Click here to edit entry	

[+ Add a row](#)

Tagging Entities in Intents

Entities can be explicitly tagged in intents, if they are not automatically detected.



The screenshot shows a dialog flow editor interface. At the top, there is a text input field containing the sentence "Pick up that red cube". The word "red" is highlighted in yellow, and "cube" is highlighted in purple. To the right of the input field is a trash icon. Below the input field is a table with three columns: "PARAMETER NAME", "ENTITY", and "RESOLVED VALUE". The table contains two rows of data. The first row has "color" in the first column, "@sys.color" in the second column, and "red" in the third column. The second row has "object" in the first column, "@object" in the second column, and "cube" in the third column. Each row has a small 'x' icon in the rightmost column, indicating that the entity can be removed.

PARAMETER NAME	ENTITY	RESOLVED VALUE
color	@sys.color	red
object	@object	cube

Automated expansion

Allows dialogflow to extrapolate to new objects

object

SAVE

Define synonyms [?] Regexp entity [?] Allow automated expansion Fuzzy matching [?]






cube	cube
container	container, box, bin
sphere	sphere, ball
screwdriver	screwdriver
hammer	hammer, mallet
wrench	wrench
Click here to edit entry	

[+ Add a row](#)

Required Entities

Suppose you want to require the user provide some entity. You can make it required, and specify how you want the agent to respond if it isn't provided.

Action and parameters

REQUIRED 	PARAMETER NAME 	ENTITY 	VALUE	IS LIST 	PROMPTS 
<input type="checkbox"/>	color	@sys.color	\$color	<input type="checkbox"/>	—
<input checked="" type="checkbox"/>	object	@object	\$object	<input type="checkbox"/>	Define prompt s...
<input type="checkbox"/>	Enter name	Enter entity	Enter value	<input type="checkbox"/>	—

[+ New parameter](#)

Specifying prompts

Specify in the modal what prompts to use to query the required entity.

The screenshot shows a modal window titled "Prompts for 'object'". It contains a table with the following data:

NAME	ENTITY	VALUE
object	@object	\$object

Below the table is a section titled "PROMPTS" containing a list of three items:

- 1 What object do you want me to pick up?
- 2 Which one?
- 3 Enter a prompt variant

A "CLOSE" button is located at the bottom right of the modal. A "Responses" tab is visible at the bottom left of the modal's container.

Tweaking your intents for required entities

Remove the entity from the prompt if it was too general. Then the agent will inquire if it gets that prompt.

Note: This could also be handled by context, which will be discussed next build class.

Training phrases ?

Search training phrases ^

” Add user expression

” Take that.

” Grab this from me.

” Take this wrench

” Pick up the hammer over there.

” How about getting that box of screws for me?

” Can you get the screwdriver for me?

” Please get the green ball

” Get the green ball

” Grab the toy

” Pick up that red cube

Dialogflow Documentation

Full Documentation

Assignment Preview

Assignment Overview

- >> Available 11/25
- >> Due 12/13 (Last possible day to submit 12/20)
- >> 10 points (No intermediate assignments)
- >> GUI code will be available on GitHub

Assignment Goals (1/3)

- >> Create a voice agent that allows a user to navigate a visual shopping interface.
- >> Allow the user to navigate from anywhere back to home, or to their cart.
- >> Allow the user to return to the previous page.

Assignment Goals (2/3)

- >> Allow the user to specify the category of items they want to look at.
- >> Allow the user to specify attributes/tags to search in.
- >> Allow the user to navigate to product pages.

Assignment Goals (3/3)

- >> Allow the user to add items (when on that product page) to their cart.
- >> Allow the user to go to checkout and complete the process.

Server API

Route	Auth Required	Token Required	Get	Post	Put	Delete
/login	✓		✓			
/users				✓		
/users/ <username>		✓	✓	✓	✓	✓
/tags			✓			
/categories			✓			
/products			✓			
/products/ <product_id>			✓			
/products/ <product_id> /tags			✓			
/products/ <product_id> /reviews			✓			
/products/ <product_id> /reviews/ <review_id>			✓			
/application		✓	✓		✓	
/application/tags		✓	✓			✓
/application/tags/ <tag_value>		✓		✓		✓
/application/messages		✓	✓	✓		✓
/application/messages/ <message_id>		✓	✓		✓	✓
/application/products		✓	✓			✓
/application/products/ <product_id>		✓		✓		✓

Pages

home

category:<category>

product:<product>

cart-current

cart-review

cart-confirmed

Tips

There is no deliverable, but we suggest you start looking at the API and figuring out what your entities and intents need to be. Next build class we will look into how you use these intents and entities to modify the application state.

Other Notes:

There will be no clinic hours on Wednesday, November 27. Please come Monday or Tuesday if you need assistance!

What have we learned today?

- >> Introduction to Conversational Interfaces
- >> Introduction to Dialogflow
- >> Dialogflow Building Blocks, Part 1
- >> Let's Make an Agent
- >> Assignment Preview