

**ORACLE®**

# Oracle Digital Assistant

## The Complete Training

**Conversation Designer**



Image courtesy of pixabay.com

Copyright © 2018, Oracle and/or its affiliates. All rights reserved. |

# Safe Harbor Statement

The following is intended to outline our general product direction. It is intended for information purposes only, and may not be incorporated into any contract. It is not a commitment to deliver any material, code, or functionality, and should not be relied upon in making purchasing decisions. The development, release, and timing of any features or functionality described for Oracle's products remains at the sole discretion of Oracle.

# Topic agenda

- 1 ➤ Challenge of conversation design
- 2 ➤ Getting familiar with the conversation designer
- 3 ➤ Things the bot can say
- 4 ➤ Things the user can say
- 5 ➤ What happens under the covers
- 6 ➤ Recommendations and best practices

# Topic agenda

- 1 ➤ Challenge of conversation design
- 2 ➤ Getting familiar with the conversation designer
- 3 ➤ Things the bot can say
- 4 ➤ Things the user can say
- 5 ➤ What happens under the covers
- 6 ➤ Recommendations and best practices

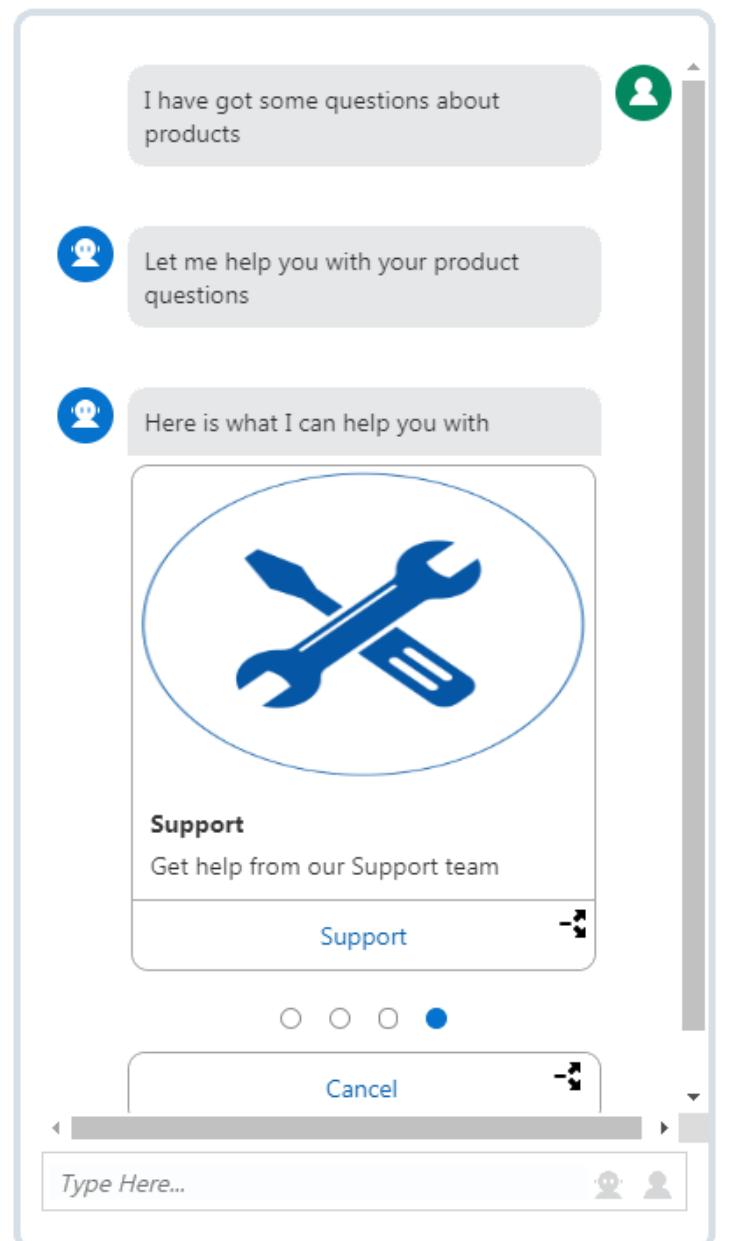
# Challenge of conversation design

- Conversational interfaces are, for most, a new concept
  - Business needs to understand what it is buying in to
  - Visualise what a conversation might look like
  - Understand tone of voice
  - Understand visuals and branding
- We encourage visualizing conversation flow
  - Stakeholder sign off
  - Starting point for developers
- Various options used today
  - Whiteboard, Botmock, Visio, PPT etc

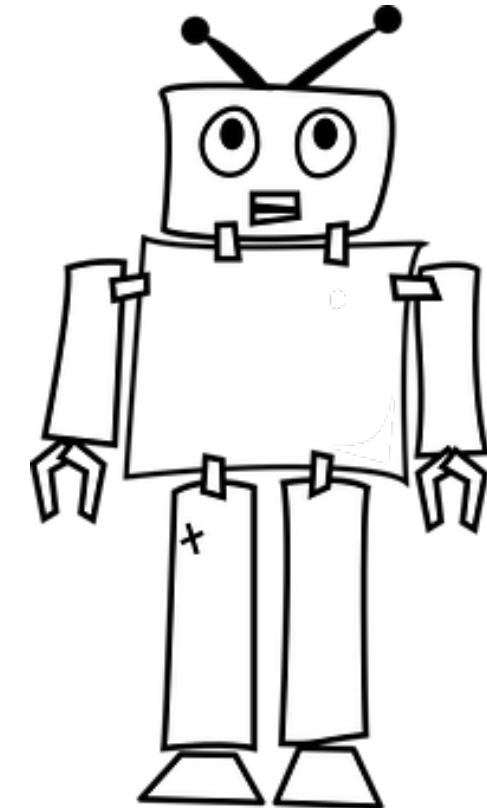


# Implementing conversation design

- Design a skill based on a sample conversation
- Allows user to mock up a conversation
  -  Define what the person says
  -  Define what the bot says
- Automatically detects intent and entities
- Conversation can be branched
- Generates a working skill



The **conversation designer** a great way for the business and stakeholders to get their hands on a something **real** and **representative**.

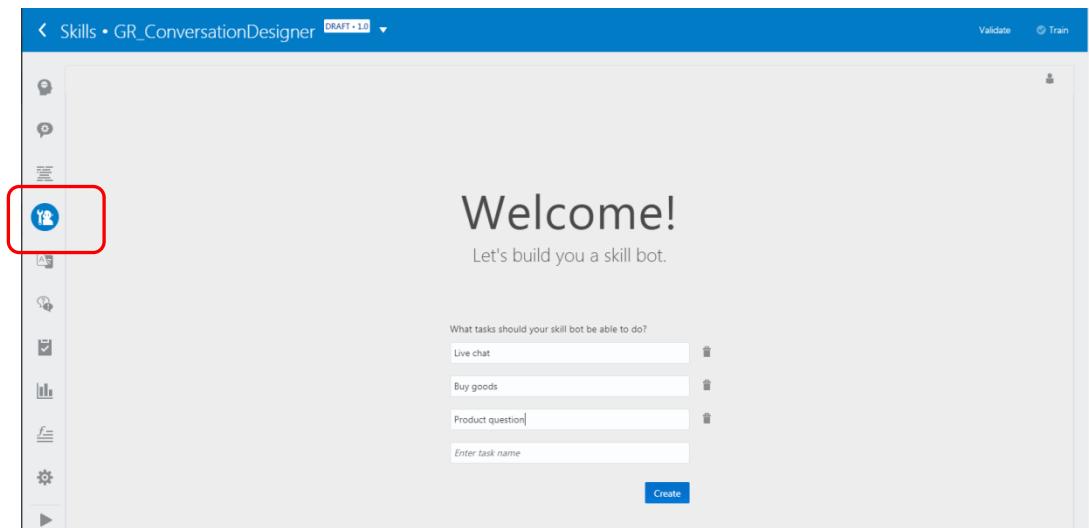
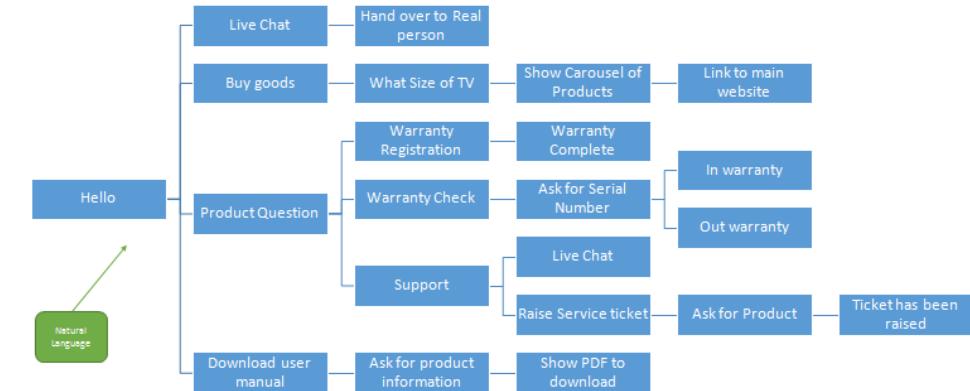


# Topic agenda

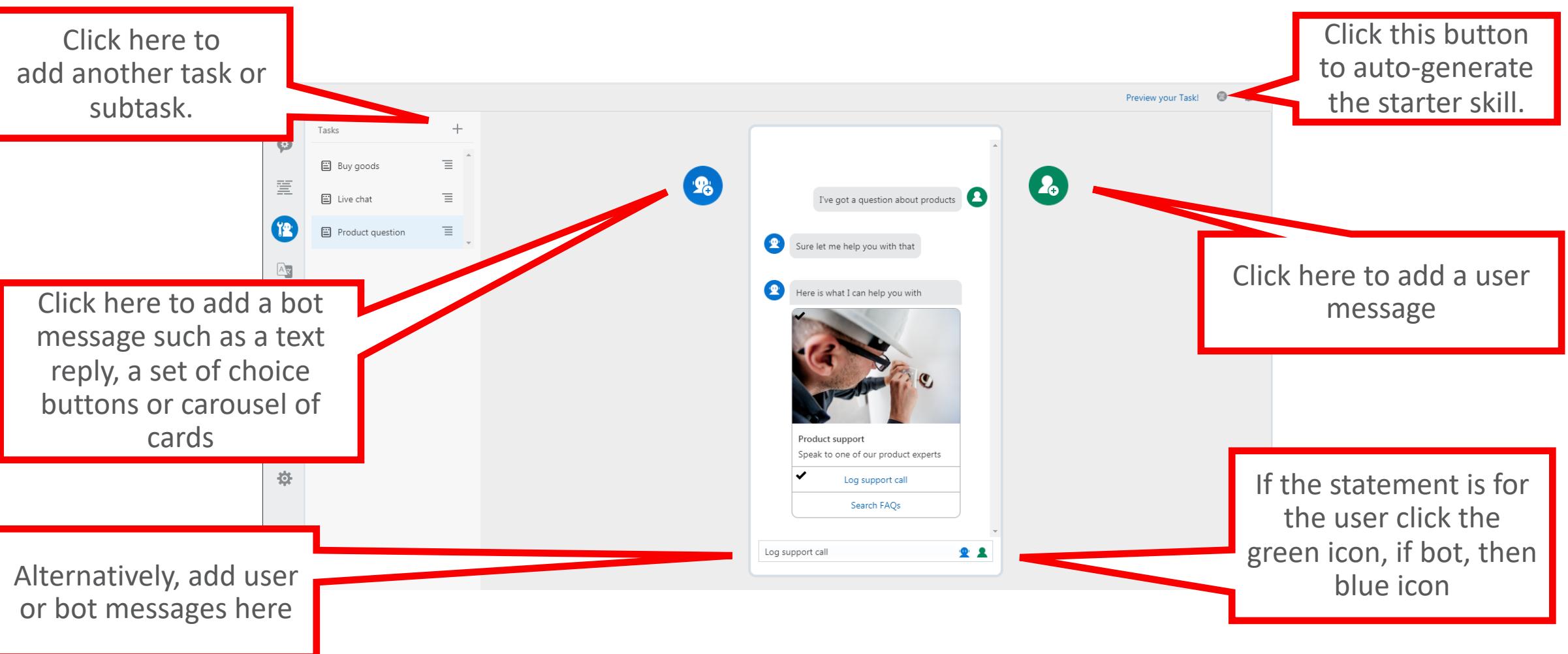
- 1 ➤ Challenge of conversation design
- 2 ➤ Getting familiar with the conversation designer
- 3 ➤ Things the bot can say
- 4 ➤ Things the user can say
- 5 ➤ What happens under the covers
- 6 ➤ Recommendations and best practices

# Getting familiar with the conversation designer

- Accessed within a skill
- Based on tasks which match to main conversation threads/branches
  - Live chat
  - Buy goods
  - Product question
  - Download user manual
- Additional tasks can be added later

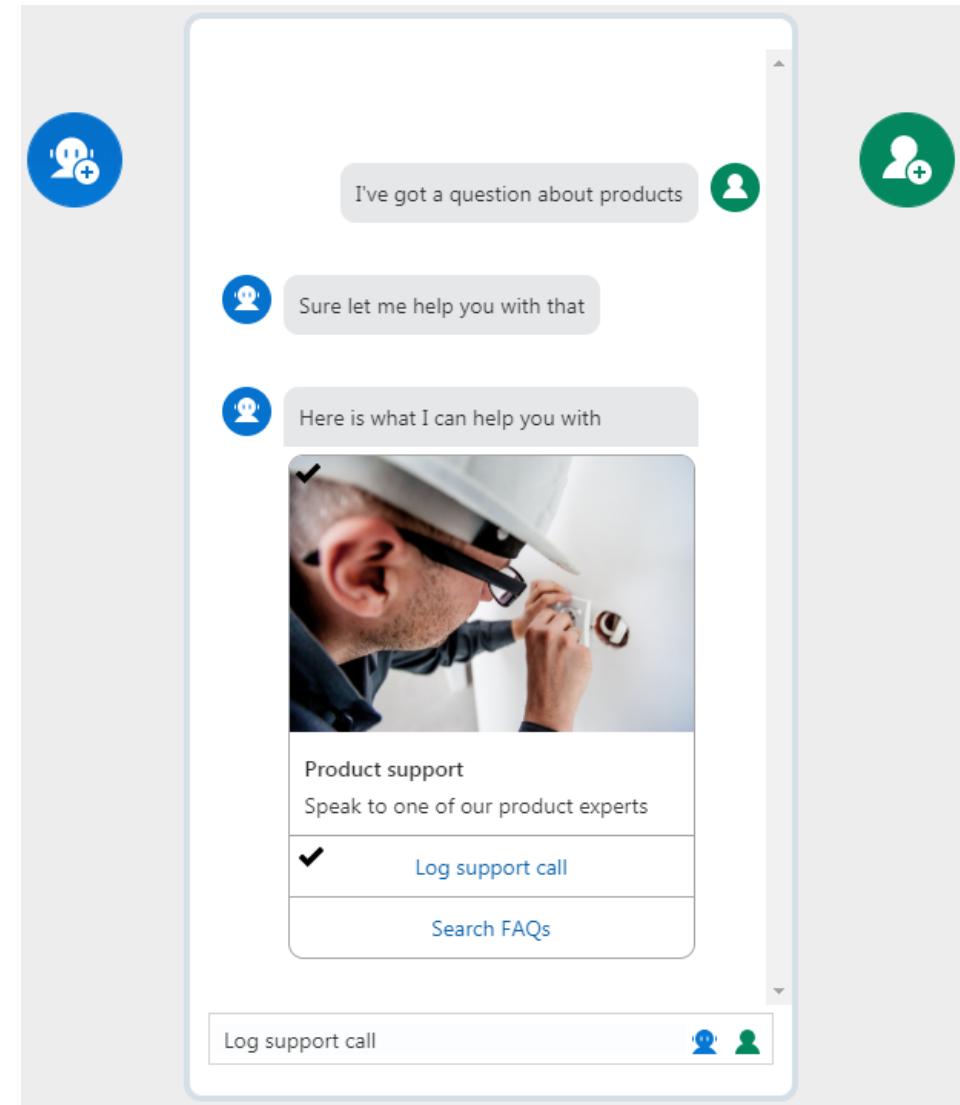


# Getting familiar with the conversation designer



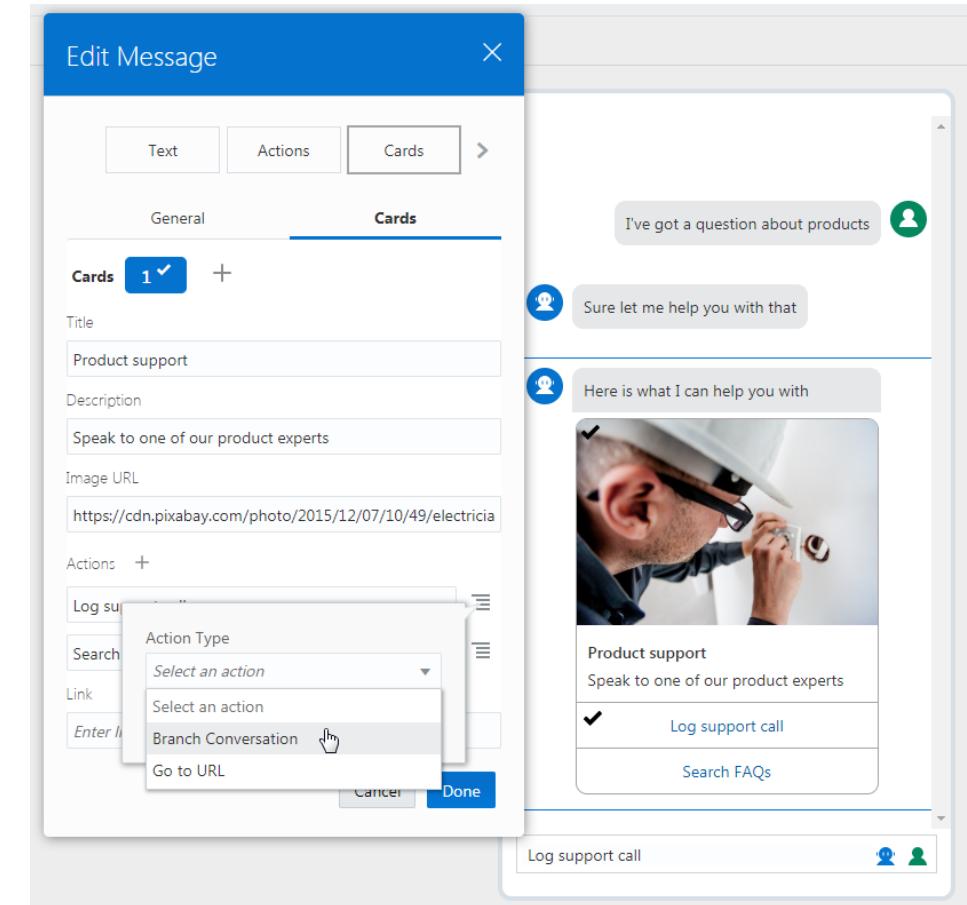
# Conversations have a pattern

- “Design by Example”
- The user makes a request
  - This is typically interpreted as an intent
  - The resolution of which will define which task to execute
- Bot then responds
  - A statement with information or data
  - A card or carousel of cards with info or data
  - A selection of buttons to further direct the dialog



# Conversations have a pattern

- Bot responses can define conversation branches
  - Would you like to speak to an agent?
- Branch to a sub task or URL
- Branch based on an action or entity value
  - Based on a value
    - E.g. If pizza size large offer stuffed crust
  - Based on action
    - E.g. Choose to speak to a support person

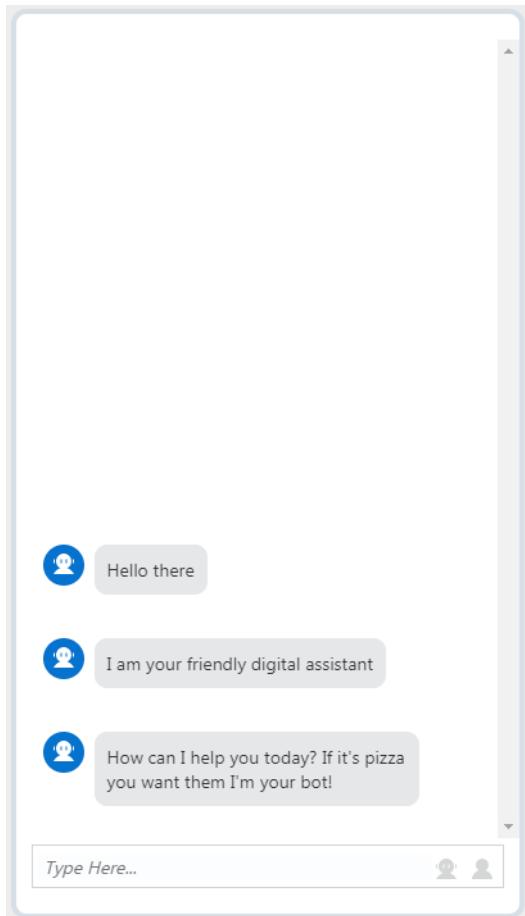


# Topic agenda

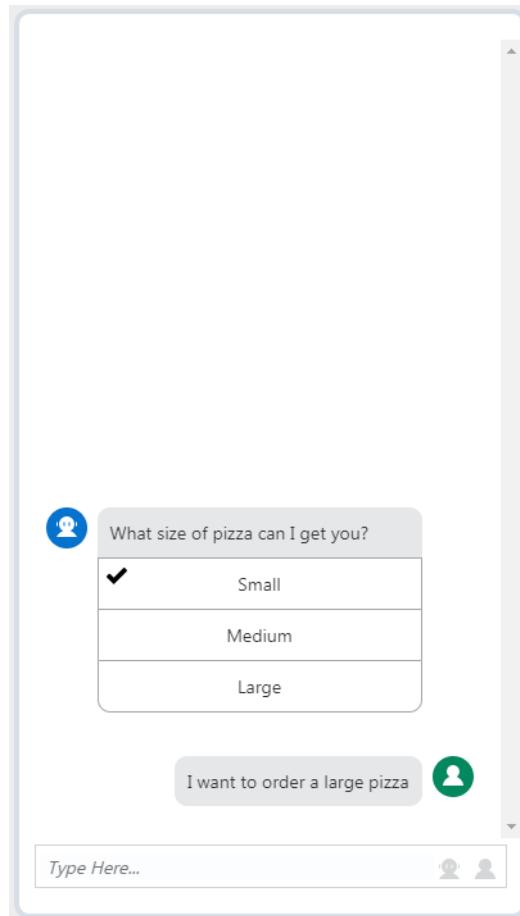
- 1 ➤ Challenge of conversation design
- 2 ➤ Getting familiar with the conversation designer
- 3 ➤ Things the bot can say**
- 4 ➤ Things the user can say
- 5 ➤ What happens under the covers
- 6 ➤ Recommendations and best practices

# Things the bot can say

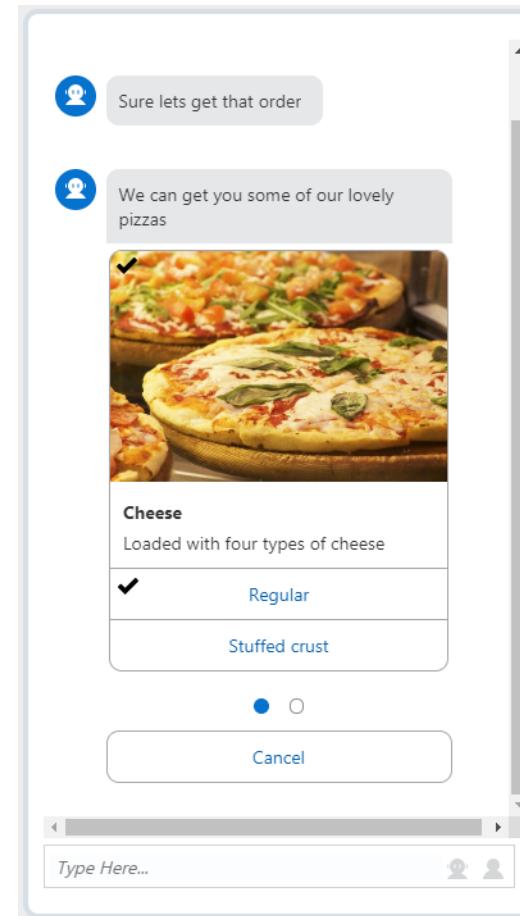
Text



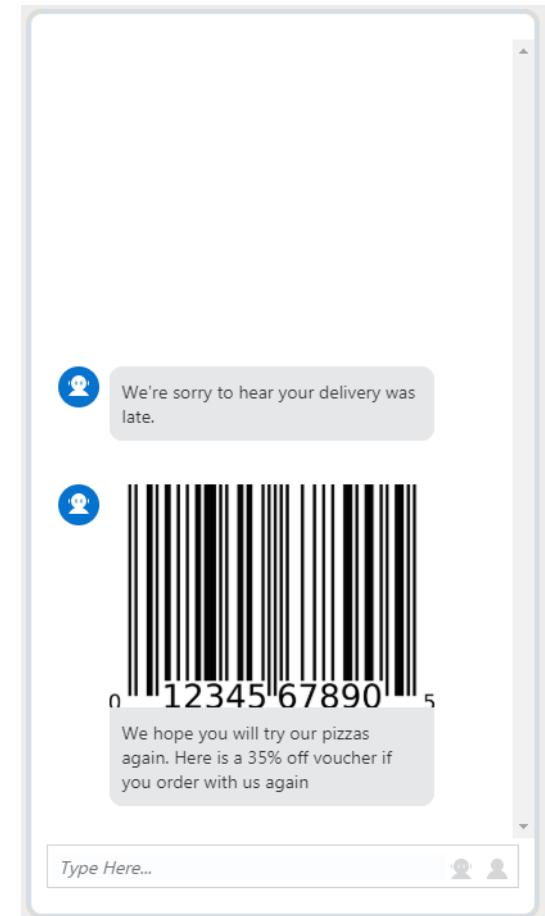
Action



Card

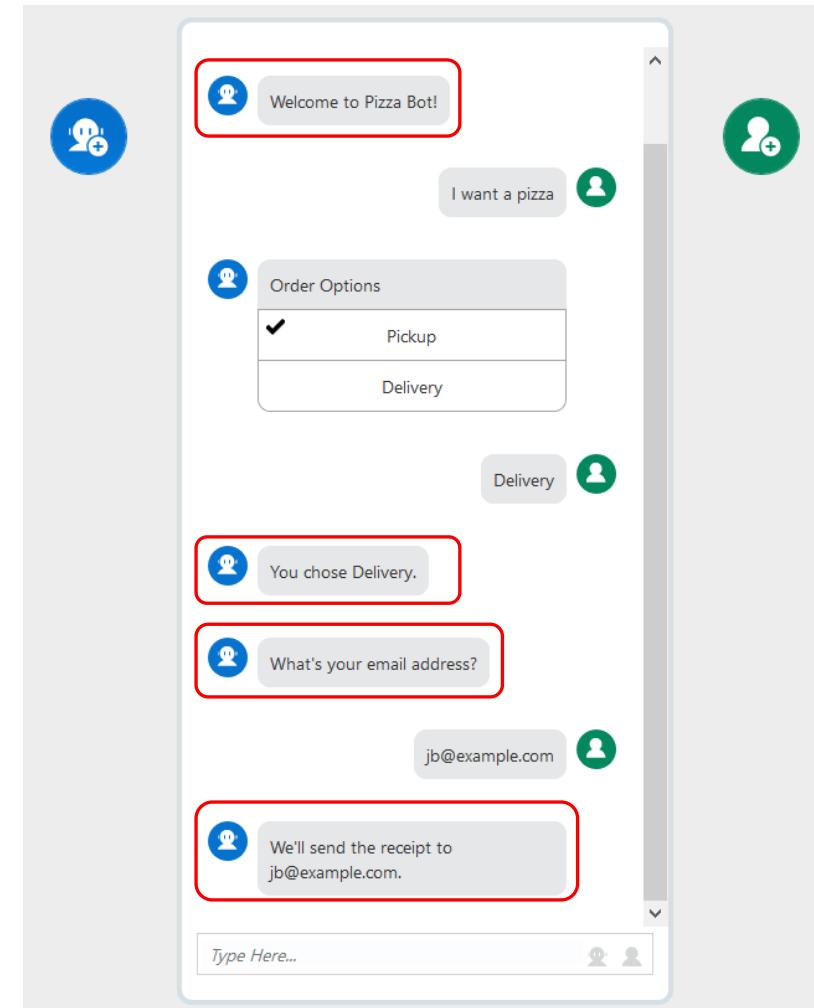


Attachment



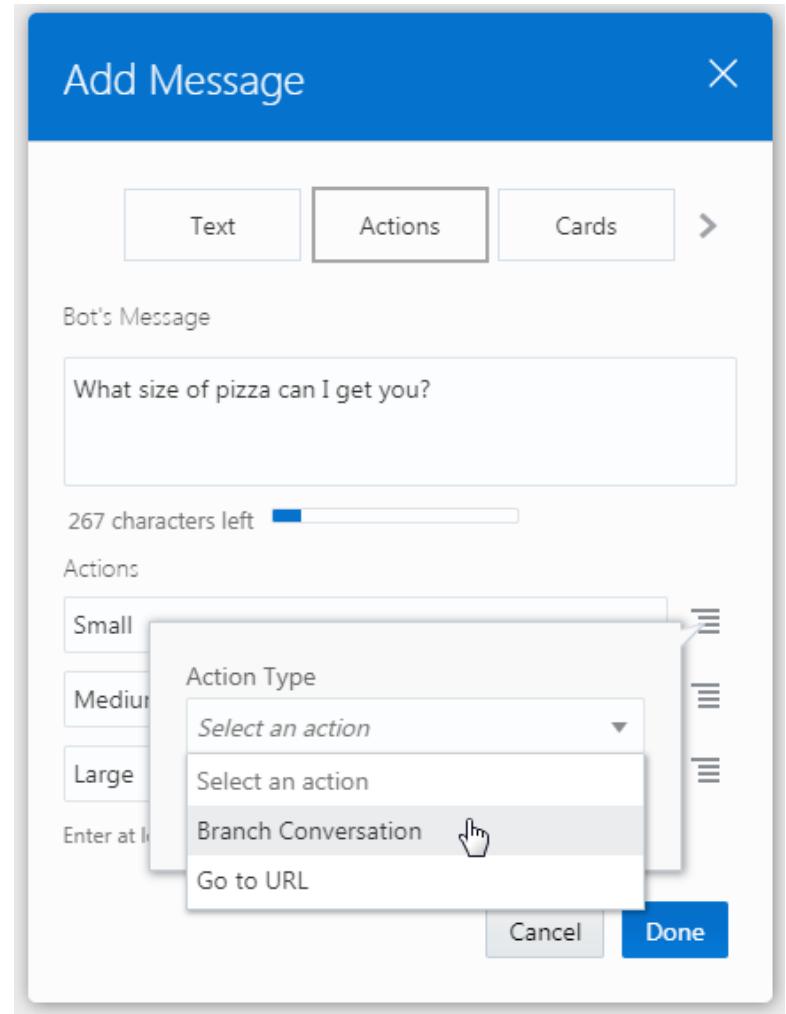
# Things the bot can say – Text

- Simple output to the user
  - Greetings
  - Acknowledgements
  - Prompts
  - Output Messages



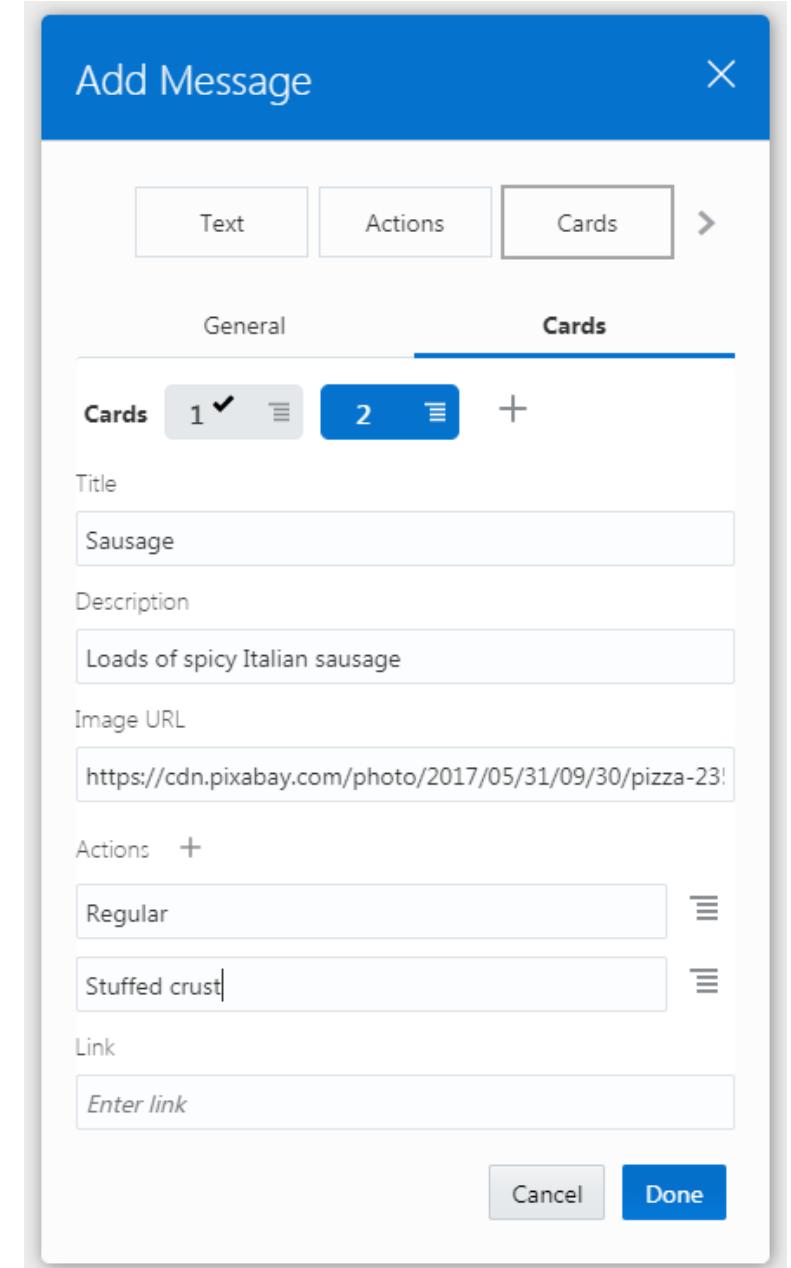
# Things the bot can say- Action messages

- Presents a list of options
  - Pizza size: small, medium, large
  - Confirm order: yes, no
- List options are enumerated:
  - Thus an entity will be created and set
  - User input validated against the entity value
- List options can trigger actions
  - Branch the conversation to another task or subtask
  - Go to a URL



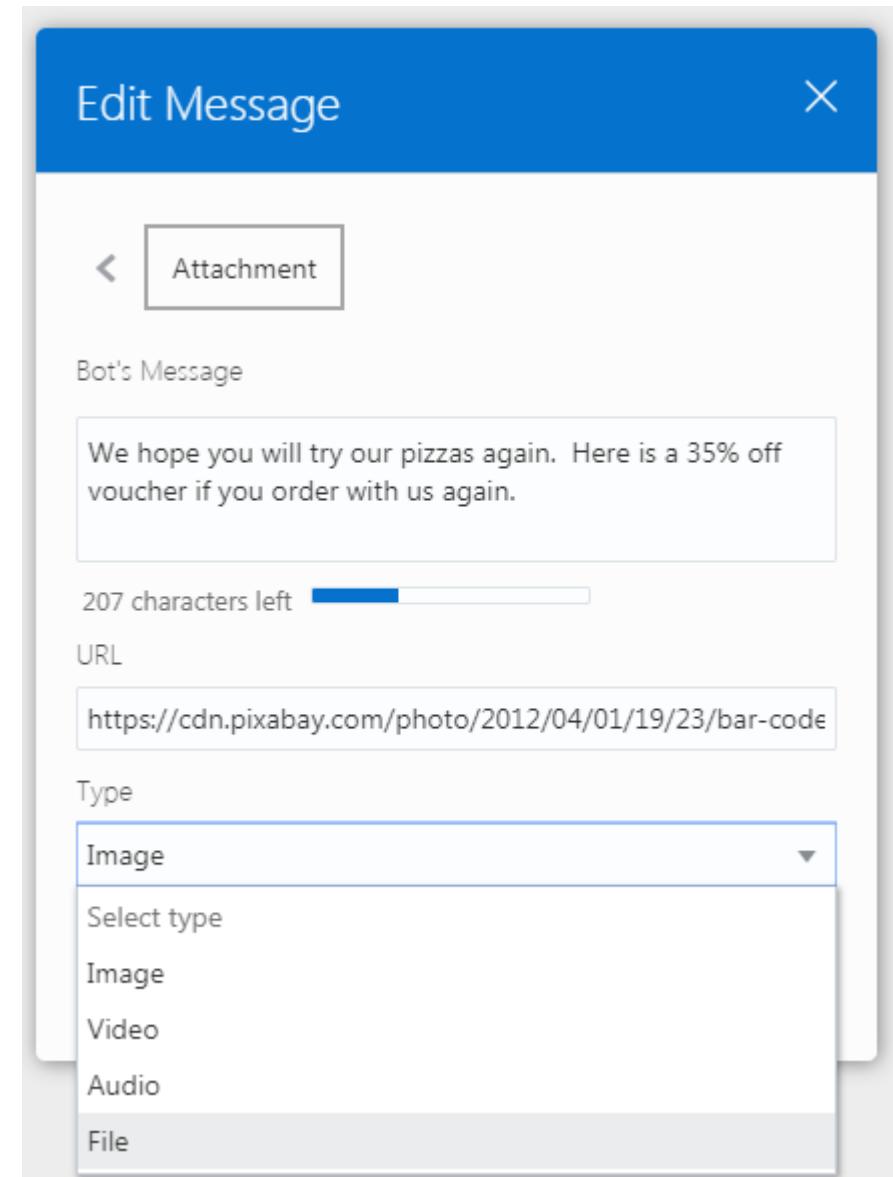
# Things the bot can say - Cards

- Presents a carousel that scrolls horizontally or vertically
- The card title defines the enumerated value
  - Thus an entity will be created and set
- Each card can have multiple actions
  - Branch to another task or subtask
  - Open a URL
- A carousel can also have a global action



# Things the bot can say - Attachment

- Allows an attachment to be displayed to the user
  - Image
  - Video
  - Audio
  - File

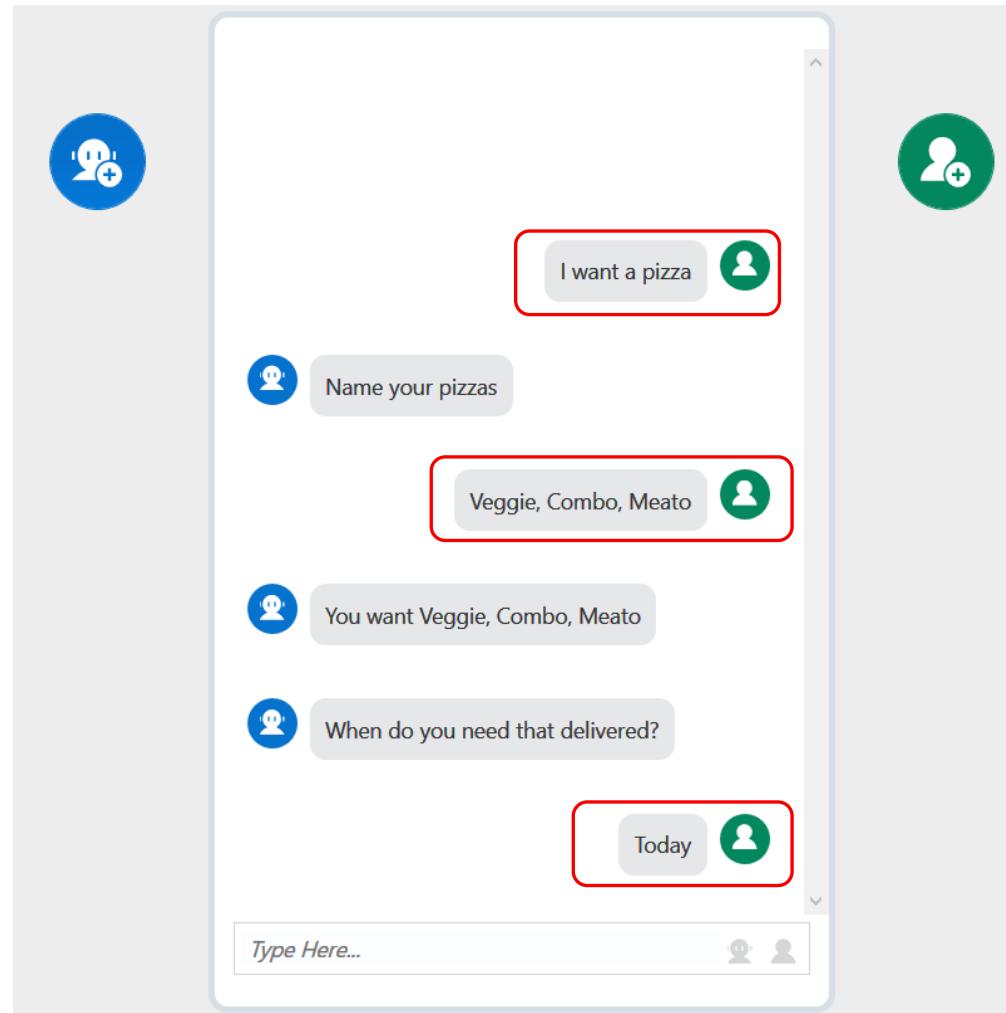


# Topic agenda

- 1 ➤ Challenge of conversation design
- 2 ➤ Getting familiar with the conversation designer
- 3 ➤ Things the bot can say
- 4 ➤ Things the user can say
- 5 ➤ What happens under the covers
- 6 ➤ Recommendations and best practices

# Things the user can say - Text

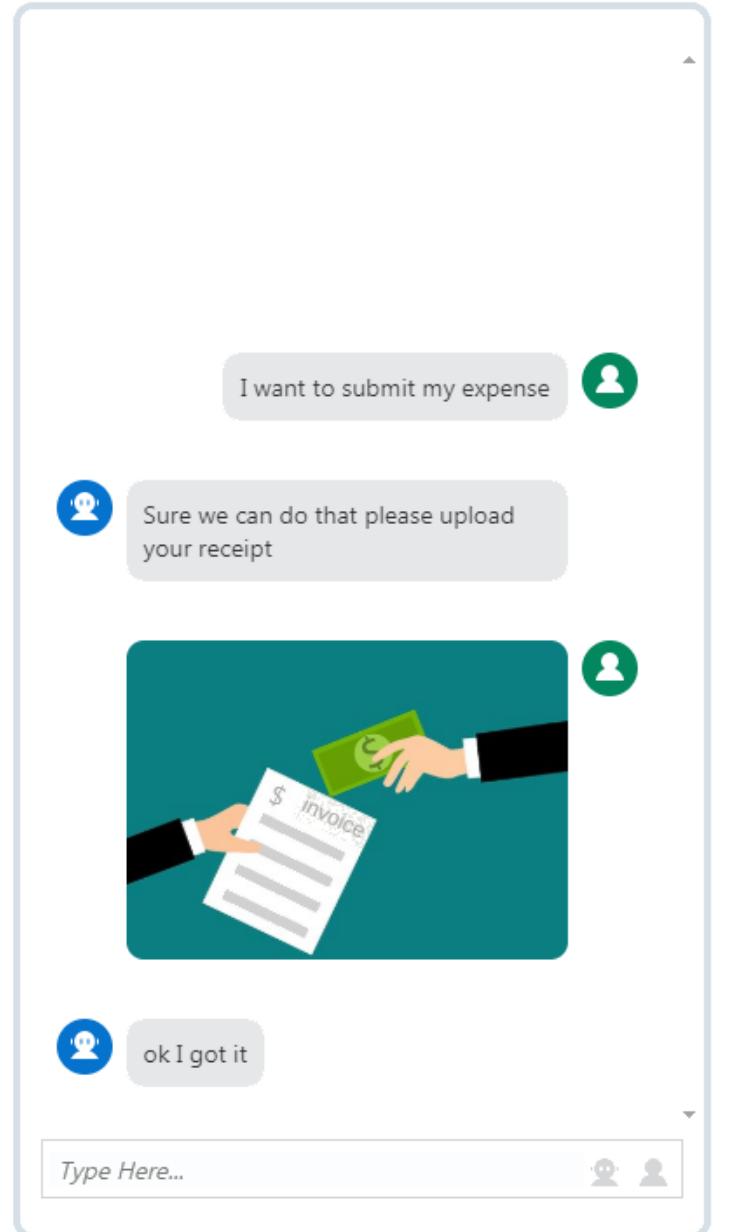
- Represents input from a user
- The Conversation Designer interprets as:
  - Intent Utterances
    - An imperative statement that ends with a noun
  - Custom entity values
  - Built-in entity values
    - Currency
    - Date
    - URL
    - Phone Number
    - Time
    - Email



# Things the user can say - Attachment

Allows a user to submit an attachment:

- Image
- Video
- Audio
- File



# Bot and user message classification

- Each message is classified depending on context
- This influences the way the bot behaves and the structures generated

Edit Message X

Text Actions Cards >

Bot's Message

What is your email address?

273 characters left [progress bar]

Classification  
PROMPT

Cancel Done

Add Message X

Text Attachment

User's Message

gr@example.com

286 characters left [progress bar]

Classification  
EMAIL

Cancel Done

Add Message X

Text Attachment

User's Message

I want a pizza

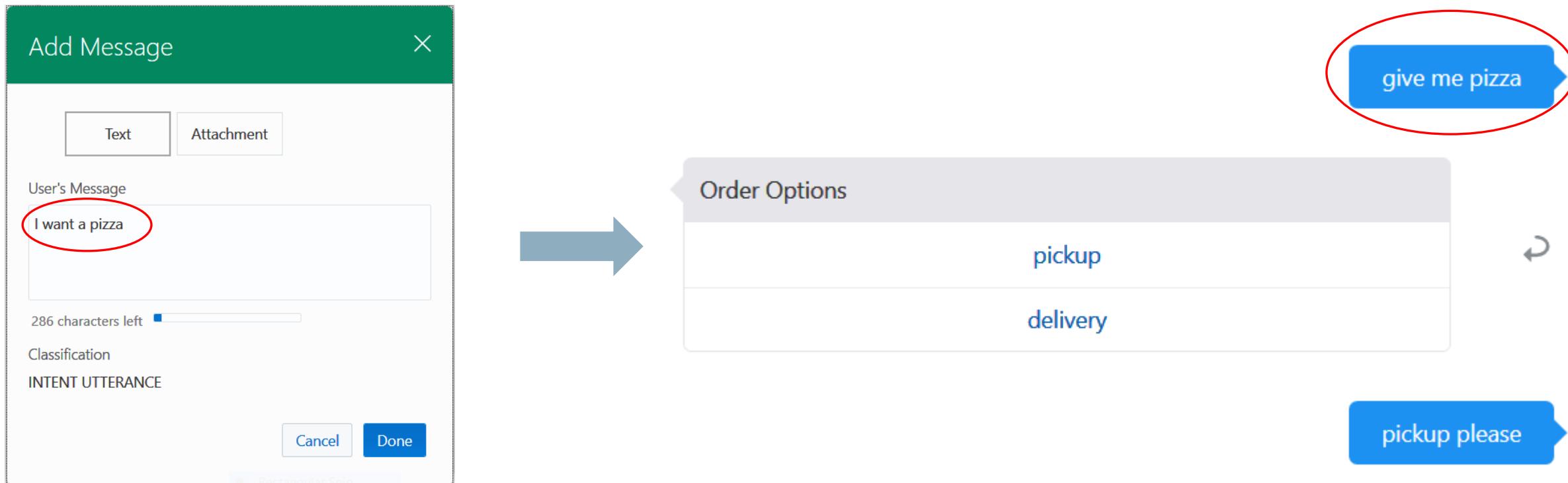
286 characters left [progress bar]

Classification  
INTENT UTTERANCE

Cancel Done

# Bot and user message classification

- User input classified as intent
  - At runtime, NLP is used to determine which task to execute



# Bot and user message classification

- By classifying we can impose entity validation
  - Entity values highlighted
- Depends on message position and context
  - User message interpreted as intent utterance
  - Bot message interpreted as prompt
  - User message interpreted as ent

Edit Message X

Text Actions Cards >

Bot's Message

What is your email address?

273 characters left

Classification

PROMPT

Cancel Done

Add Message X

Text Attachment

User's Message

I want a pizza

286 characters left

Classification

INTENT UTTERANCE

Cancel Done

Delivery User

You chose Delivery

What is your email address?

gr@example.com

We'll send the receipt to gr@example.com

Type Here...

Add Message X

Text Attachment

User's Message

gr@example.com

286 characters left

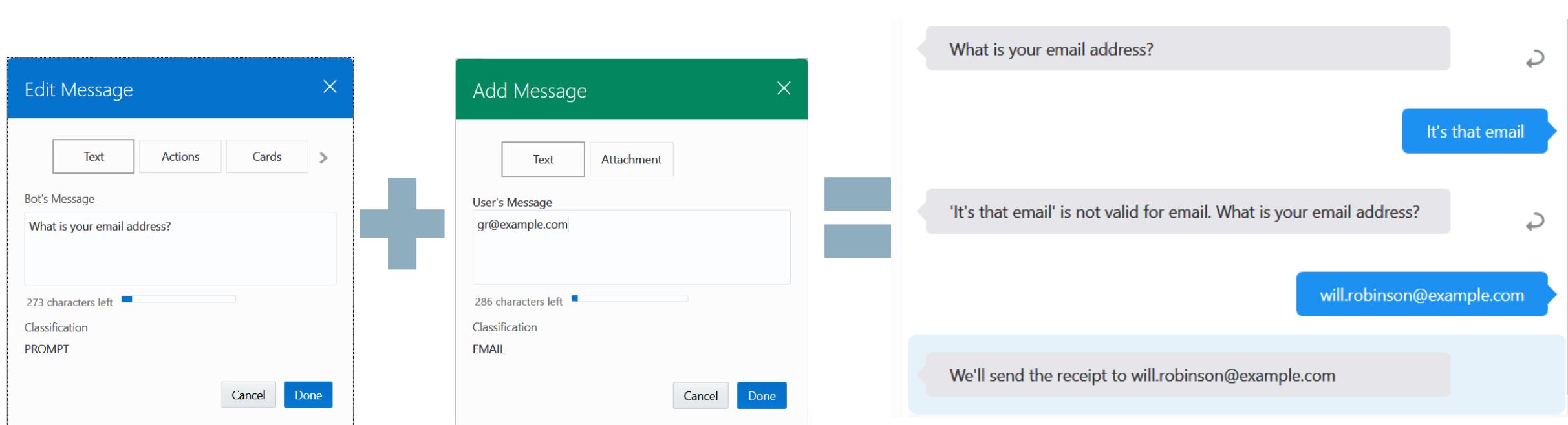
Classification

EMAIL

Cancel Done

# Bot and user message classification - example

- Creates an entity and incorporates the prompt message
- At runtime, the format of the user input is validated



# Topic agenda

- 1 ➤ Challenge of conversation design
- 2 ➤ Getting familiar with the conversation designer
- 3 ➤ Things the bot can say
- 4 ➤ Things the user can say
- 5 ➤ What happens under the covers
- 6 ➤ Recommendations and best practices

# What happens under-the-covers?

- Generate and run in tester, or preview
- Design definition JSON in dialog flow
- When you generate:
  - The dialog (YAML) is generated
  - For each task, an intent is created along with some training utterances
  - For each task, a set of entities (templates for the request details) are generated
    - E.g. toppings = pepperoni, size = large, medium, small
- Bot generation is one-way
  - Hand editing is NOT reflected in design

# Topic agenda

- 1 ➤ Challenge of conversation design
- 2 ➤ Getting familiar with the conversation designer
- 3 ➤ Things the bot can say
- 4 ➤ Things the user can say
- 5 ➤ What happens under the covers
- 6 ➤ Recommendations and best practices

# Recommendations and best practices

- Give task names such as
  - Product Question, Live Human
  - Don't use “task” “subtask” as suffix.
- Start each task with a user input which indicates the “intent” of the flow
  - Use imperative voice
  - Ideally ends with noun
    - “I want to order pizza”, “Show me currently open orders”
- Cards need actions

# Integrated Cloud Applications & Platform Services

**ORACLE®**