A Reference Architecture for Applications with Conversational Components

Saurabh Srivastava, T.V. Prabhakar

Department of Computer Science & Engineering, IIT Kanpur, India

10th IEEE International Conference on Software Engineering and Service
Beijing, China

October 18-20, 2019

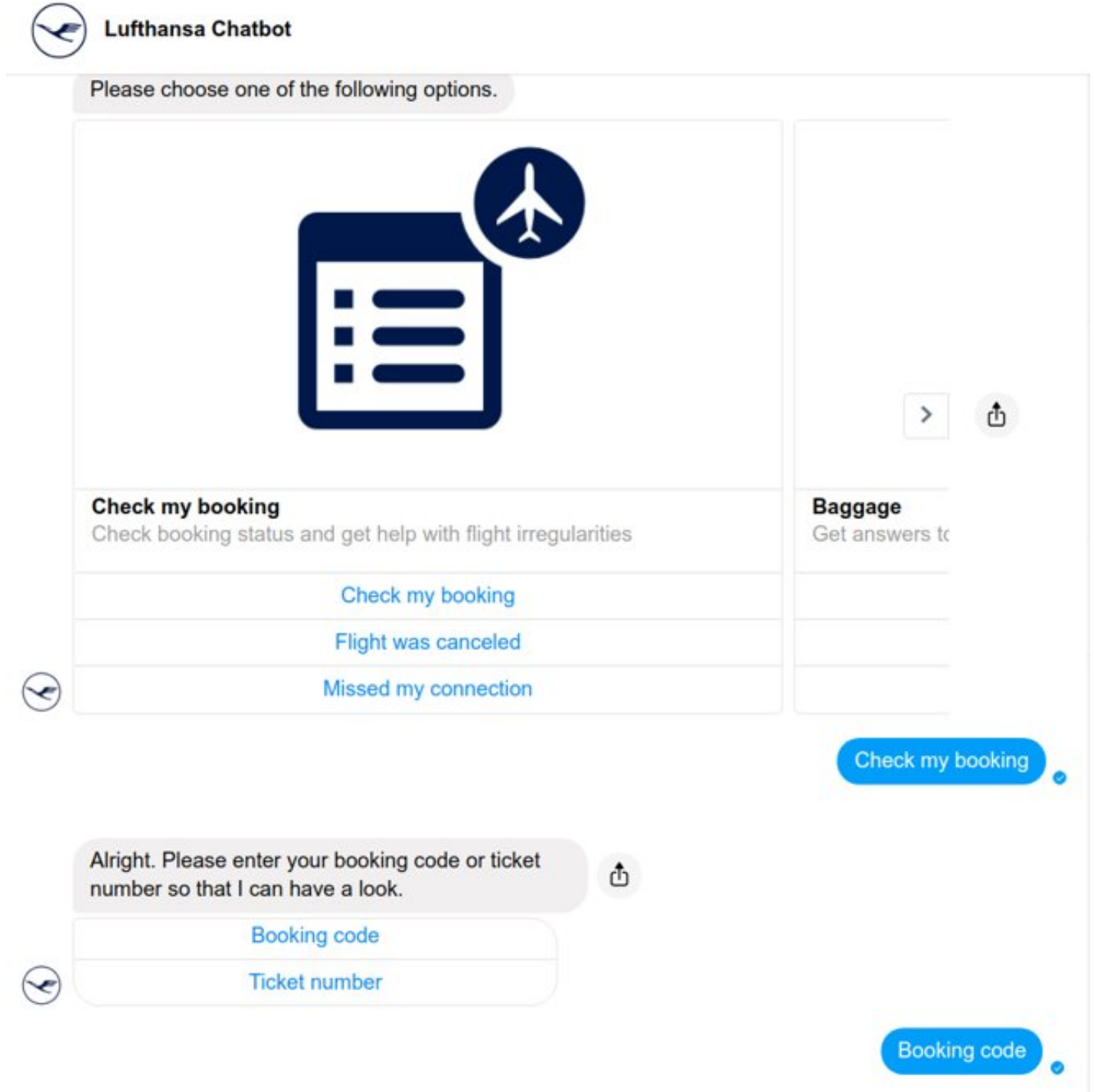
Agenda

- Introduction
  - Applications with Conversational Interfaces
  - Issues with Conversational Interfaces
- Reference Architecture for such applications
  - Major components
  - Common Variation Points
- Examples of Concrete Architectures
  - Using Google's Dialogflow
  - Using IBM's Watson Assistant
- Summary

Introduction

Conversational Interfaces

- User-interfaces involving communication through *Natural Language phrases*, say in English or Mandarin are becoming fairly common
- The conversation could be *textual*, i.e. "typed in a textbox", or it could be through *speech*, i.e. "spoken by a synthesised voice"
- The idea is to allow user to express her intentions in a language she is already familiar with
- These mechanisms to interact with a system are called *Conversational Interfaces*
- The colloqiual term for these interfaces is *Chatbots*



Example of a *Conversational interface* - **Lufthansa Airlines Chatbot** Over Facebook Messenger

Issues with Conversational Interfaces (1/3)

- Core NLP Issues
  - Processing Natural Languages is hard, we still haven't reached the 100% success rate !!
  - This means models, no matter how accurate, can still make mistakes
  - Often, these mistakes cannot be predicted beforehand
  - This adds a lot of "uncertainity" when adding these interfaces to systems
  - Common issues: converting speech to text and vice versa, understanding sarcasm, word sense disambiguation etc.

Issues with Conversational Interfaces (2/3)

- Expectations from a *human-like* machine
  - Attempting to be "human-like" may have unexpected consequences
  - The user assumes that the system has answers to questions which may be common knowledge for humans, but not for a machine
  - Example: "How is the weather outside?"
  - Finding out if a question is "answerable" or "not" with the given amount of data is a problem in itself

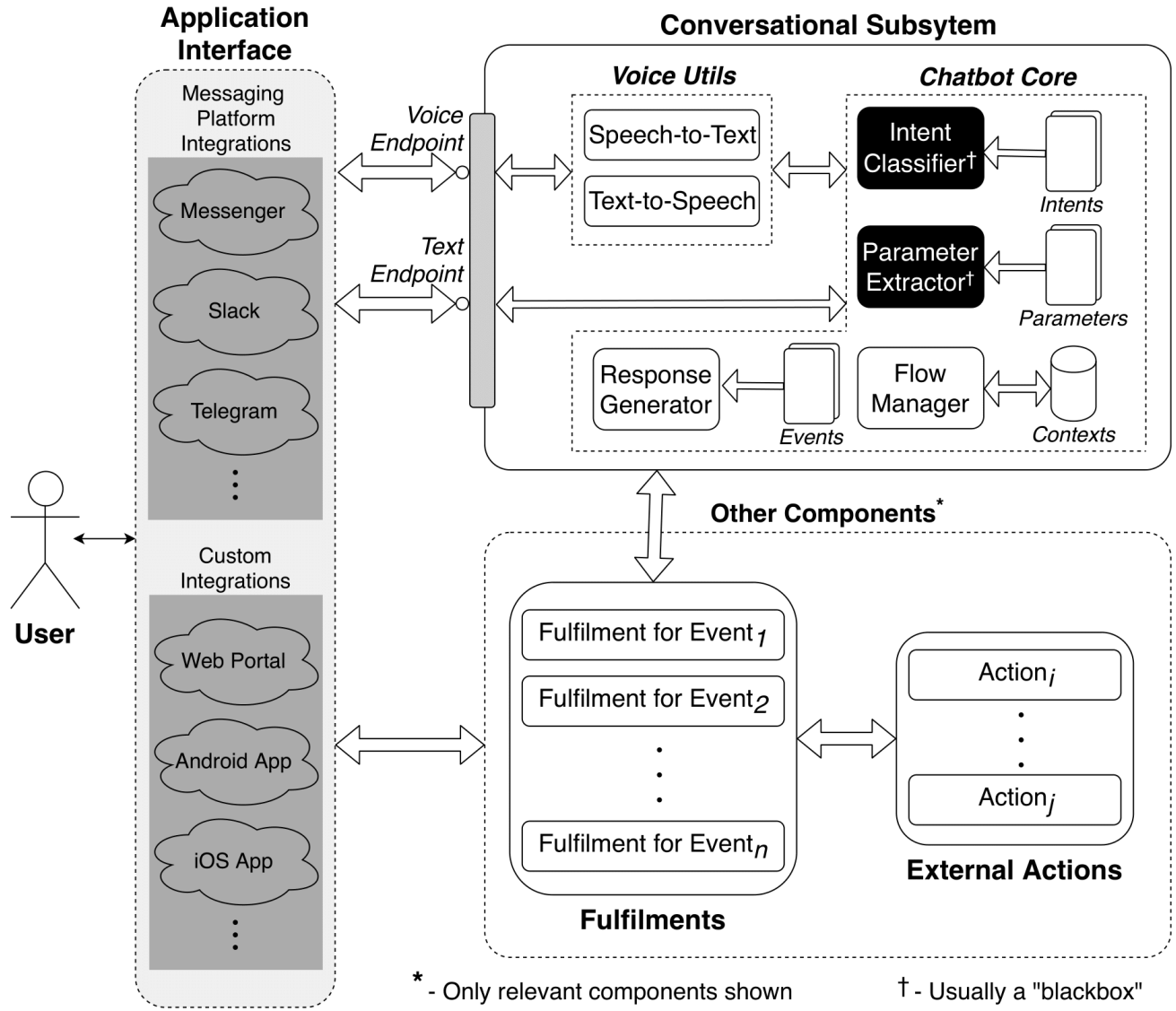
Issues with Conversational Interfaces (3/3)

- Operational problems
  - Building these interfaces require building Natural Language models
  - These models require a lot of *example user queries*
  - Even if such large amount of data is actually available, tuning the models to cover all possible types of user queries is not possible
  - Commercial platforms which are often used to build these models (like Google's Dialogflow or IBM's Watson Assistant), usually keep the models "blackboxed"

Reference Architecture

Workflow

- A typical workflow for an application with a conversational interface:
  - User enters a query, e.g. "What is the current price for apples"?
  - The chatbot core, contains business logic to find out the "intent" of the user from the supplied query
  - For example, here the intent could be *fruit_price_query*
  - The chatbot core also has a mechanism to find out any "parameters" that the query may have, e.g. here the value for *fruit_name* parameter is *apple*
  - The chatbot then looks for a way to "fulfil" this query – it may involve performing an external "action", such as calling an internal API
  - It may need some context, like user's location, which may have been provided during the conversational "flow" through previous queries
  - Finally, the chatbot prepares and either shows or "speaks" the response back



A Reference Architecture for applications with a conversational interface

Major Components in the Architecture

- *Intent Classifier*
  - This component is responsible for categorising a user query into one of the defined "categories"
  - For example, for a store that sells fruits, a user query could be categorised as *fruit_price_enquiry* (a query asking for the price of a particular fruit), *store_address_enquiry* (a query about the location of the store within the city), *fruit_availability_enquiry* (a query about availability of a particular fruit) etc.
  - These categories are pre-defined by the chatbot developer, with some examples of each query type

Major Components in the Architecture

- *Parameter Extractor*
  - This component is responsible for finding instances of real-world values in the user query
  - For example, for a store that sells fruits, to answer a query about the price of a fruit, it needs the value of the *Fruit* (name of the fruit) parameter, e.g. "apple", "guava" etc.
  - Similar to intents, parameters are also pre-defined by the chatbot developer

Major Components in the Architecture

- *Fulfilments*
  - Fulfilments are the code fragments executed for "fulfilling" a user query
  - This essentially involves performing any processing tasks in the background, like running SQL queries or calling external APIs
- *Actions*
  - Actions are any tasks that can produce an effect in the "outside" world
  - For instance, "addition of a reminder to user's calendar" or "sending a mail on behalf of the user"
  - Actions are triggered as part of some fulfilments

Major Components in the Architecture

- *Voice-to-Text*
  - All the models are built to work over "textual" inputs, and provide "textual" outputs
  - If the user query is in speech form, a component is required to transcribe it in text form before supplying it to the model
- *Text-to-Voice*
  - If the user is expecting the response in audio format, a component must produce the same, from the output generated by the model
  - A system voice is required to synthesis this audio clip

Major Components in the Architecture

- *Response Generator*
  - Prepares a response for a given user query
  - The response could be an answer to the user's question, or a follow-up question (e.g. to get the value for a parameter, if it was not supplied in the original query)
  - The response can either be generated "internally" by the chatbot, by using a pre-defined template, or, it can be directly sent from the respective fulfilment, which in turn, may have been generated "externally"

Major Components in the Architecture

- *Flow Manager*
  - How does these two conversations differ from each other?

Bot: *What can I do for you?*	Bot: *What can I do for you?*
User: *I want to order fruits*	User: *I want to order fruits*
Bot: *Which fruit?*	Bot: *Which fruit?*
User: *What do you have now?*	User: *What do you have now?*
Bot: *Bananas and Apples.*	Bot: *Sorry, we don't have that!*
Bot: *Which fruit?*	Bot: *What can I do for you?*
User: *Apples*	User: *What fruits do you have?*
Bot: *Ok. Ordering Apples...*	Bot: *Bananas and Apples.*
	Bot: *What can I do for you?*
	User: *I want to order apples*
	Bot: *Ok. Ordering Apples...*

  - Answer – Flow Management (keeping track of previous context, and changing the response based on that)

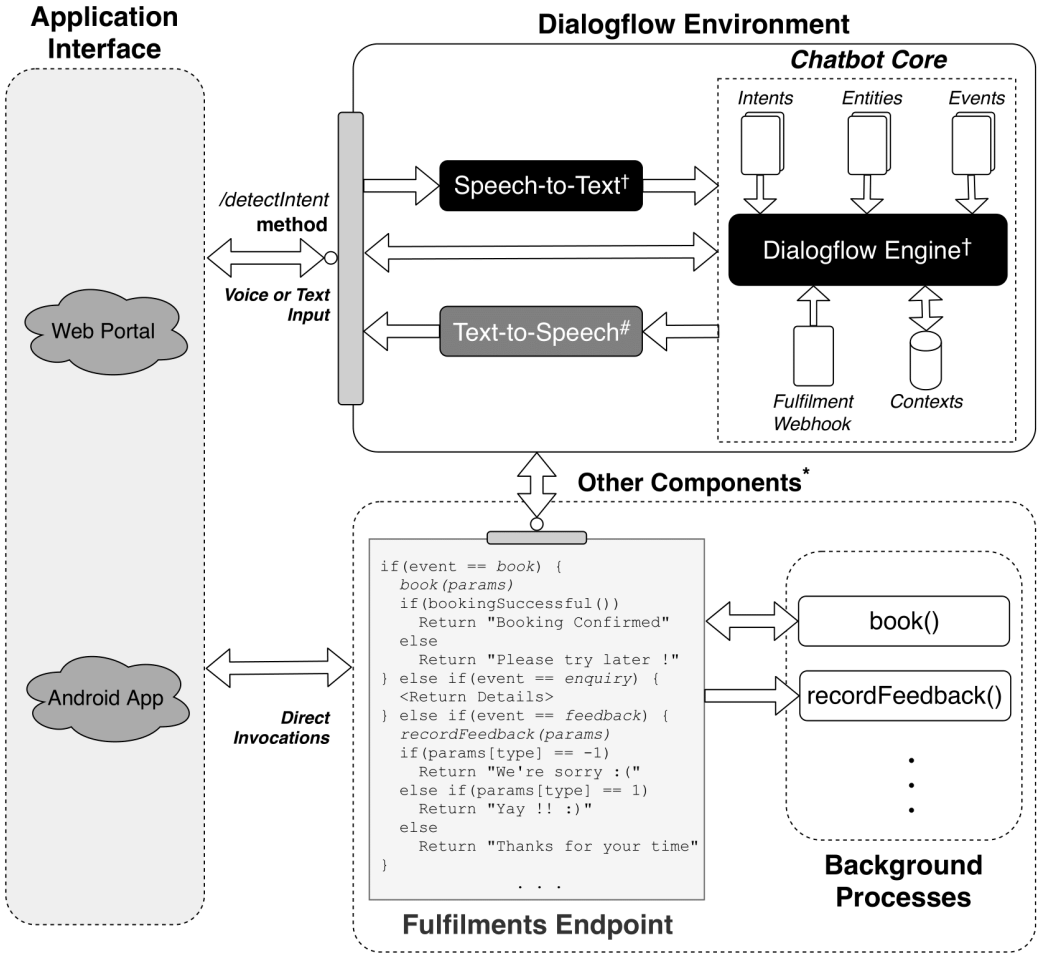
Common points of Variations

- Voice Utils
  - The Voice-to-Text and Voice-to-Text utilities are not required to be a part of the system, if the built chatbot supports only text
  - These tasks can be delegated to an external service or component, and invoked through API calls
- Fulfilments
  - The fulfilment code can be written in multiple technologies
  - The location of the code can also vary – it can be a part of the system itself, or, it could be invoked through API calls
- Flow Management
  - Flow Management is a "nice-to-have" feature
  - For some use cases, it may not even be required

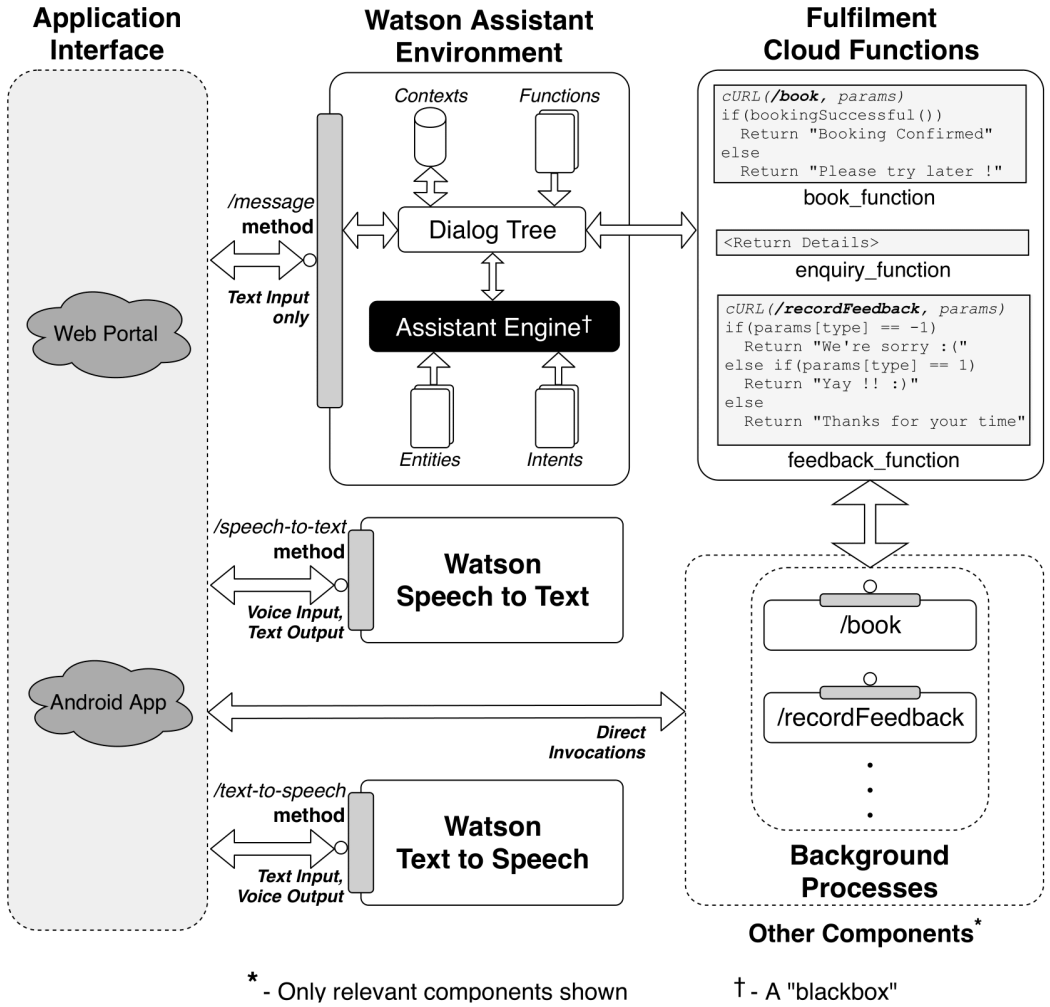
Concrete Architectures

Built using Commercial Chatbot Development platforms

Chatbots with Dialogflow and Watson Assistant

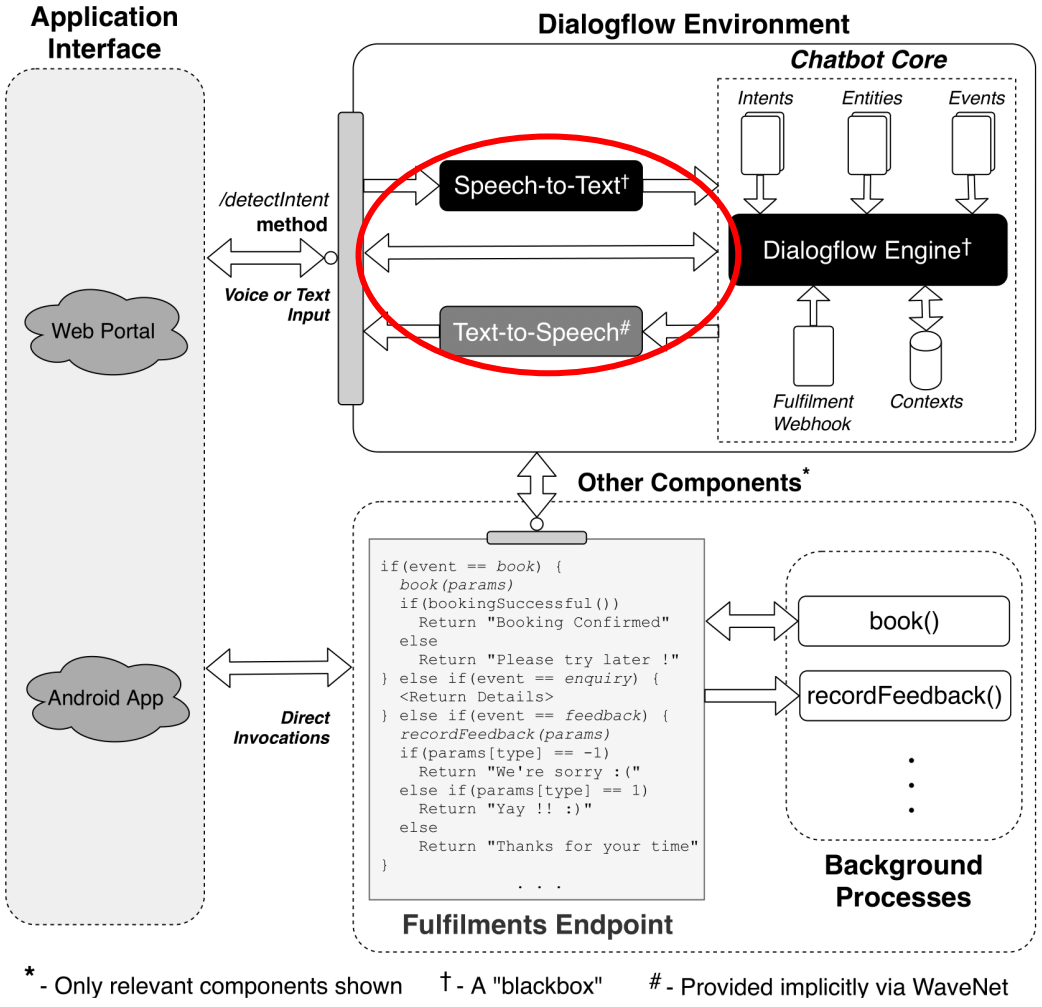


Application built with Dialogflow

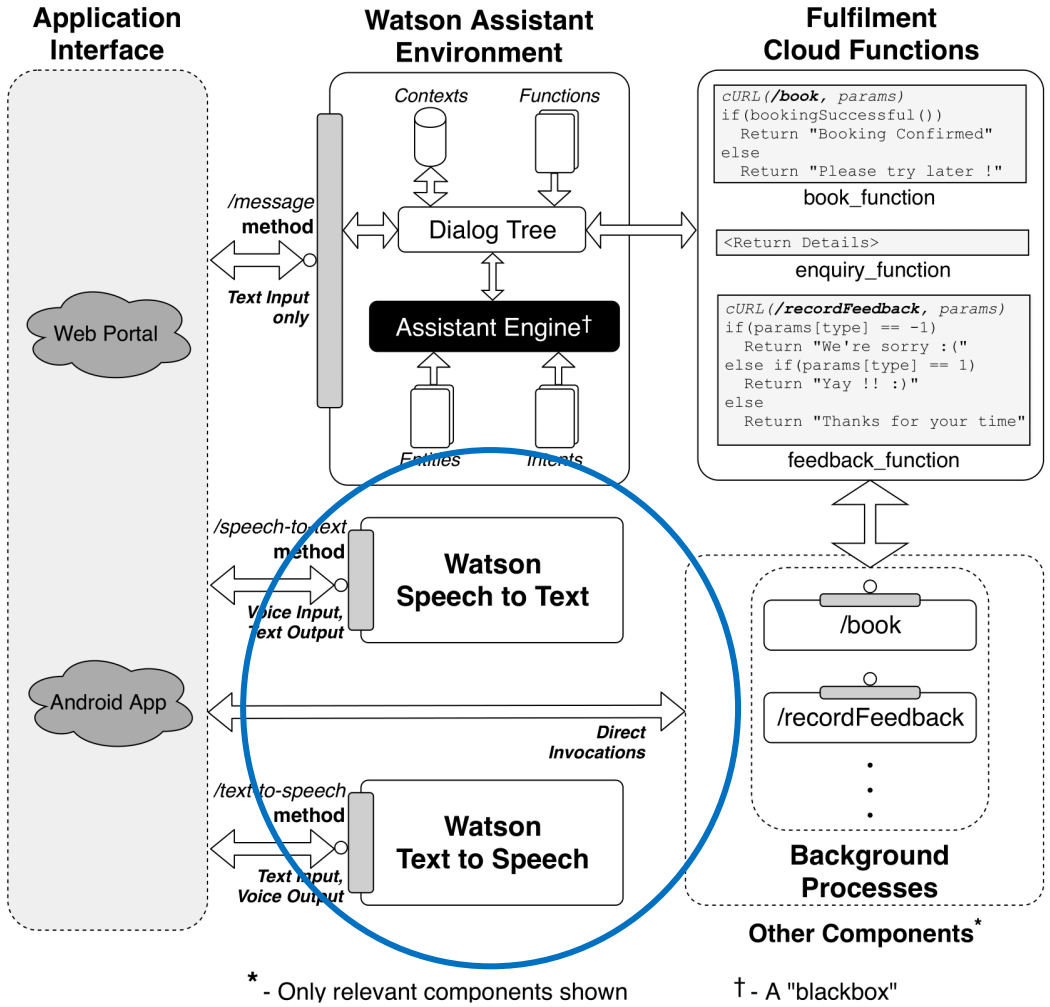


Application built with Watson Assistant

Chatbots with Dialogflow and Watson Assistant



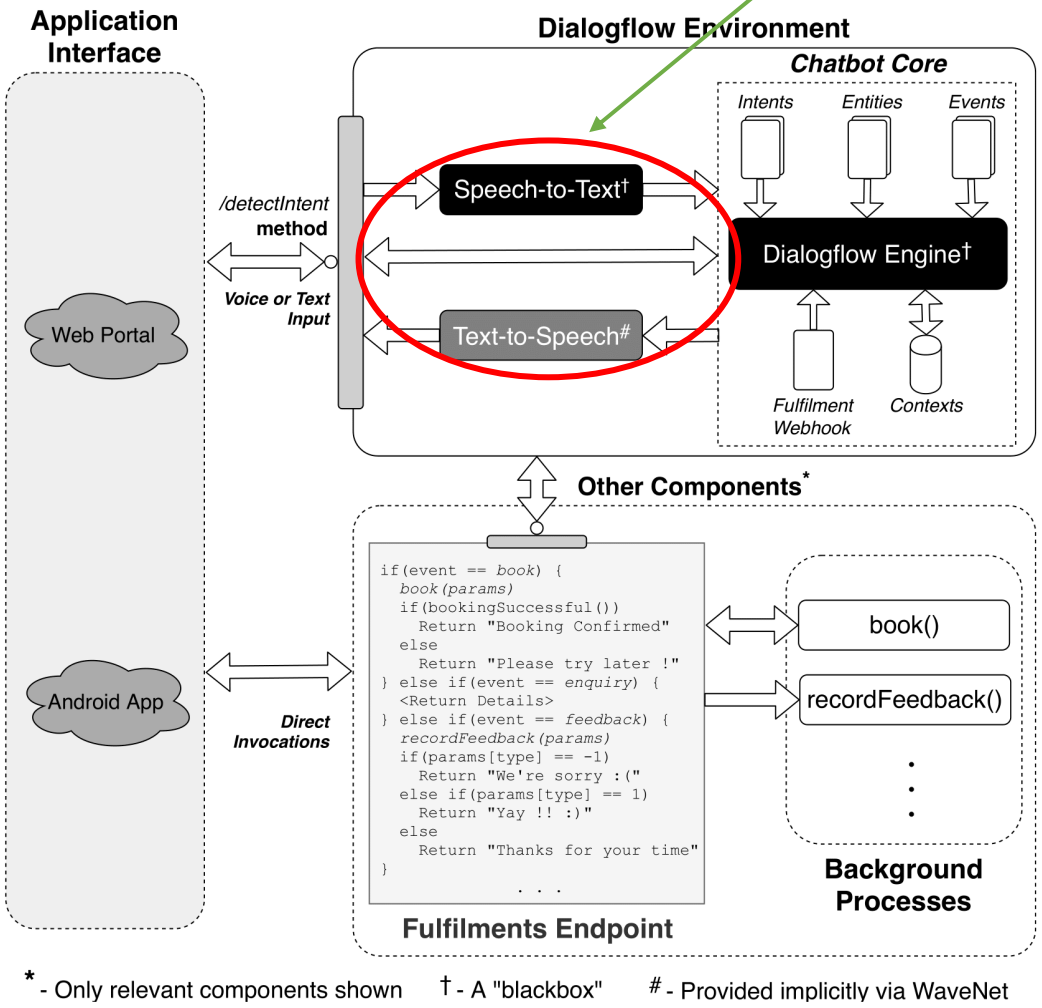
Application built with Dialogflow



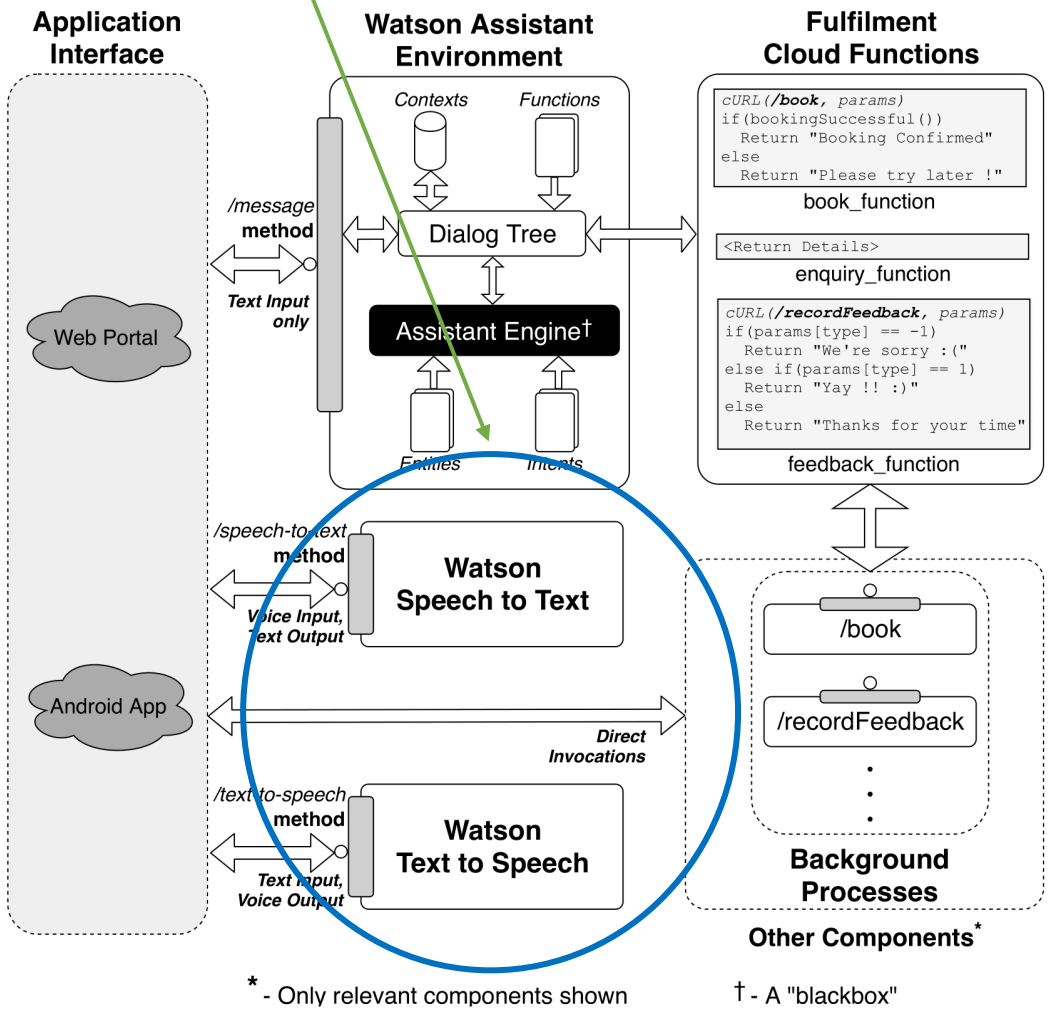
Application built with Watson Assistant

Internal vs External Voice Utils

Chatbots with Dialogflow and Watson Assistant

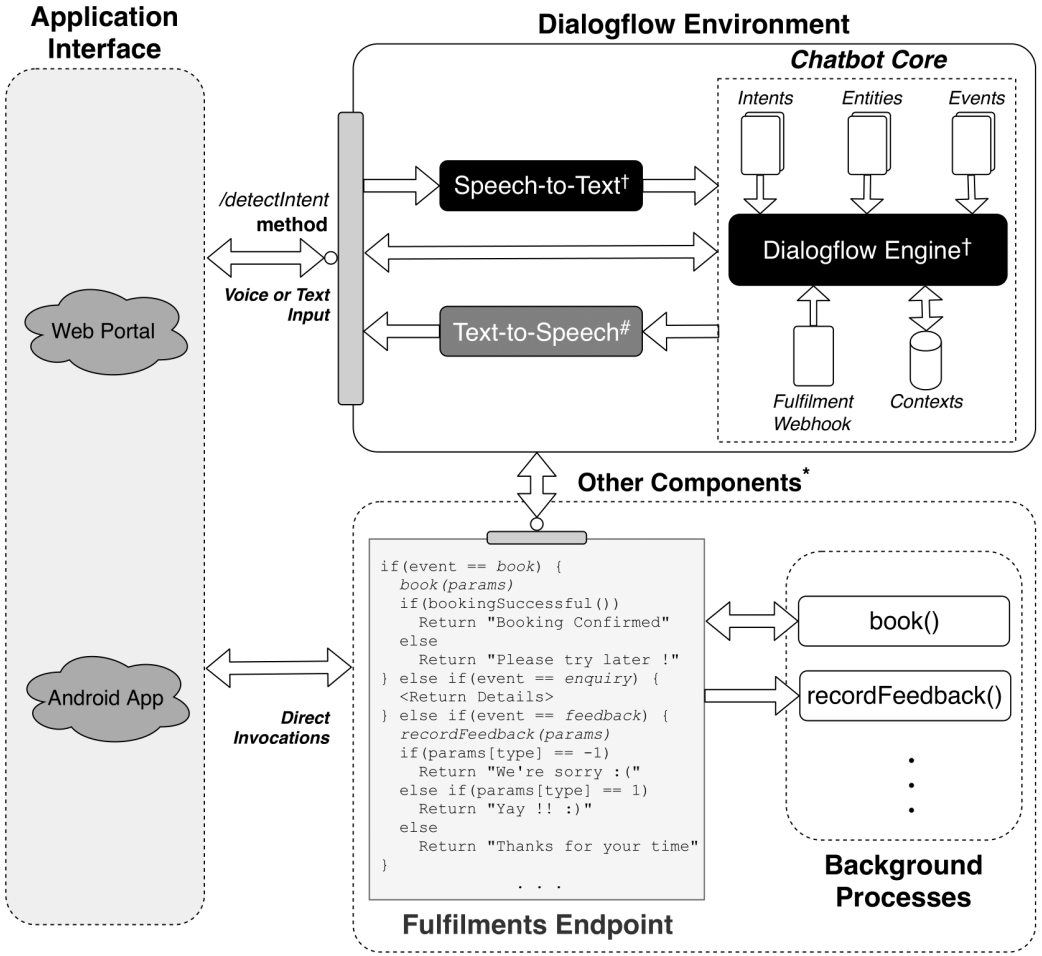


Application built with Dialogflow

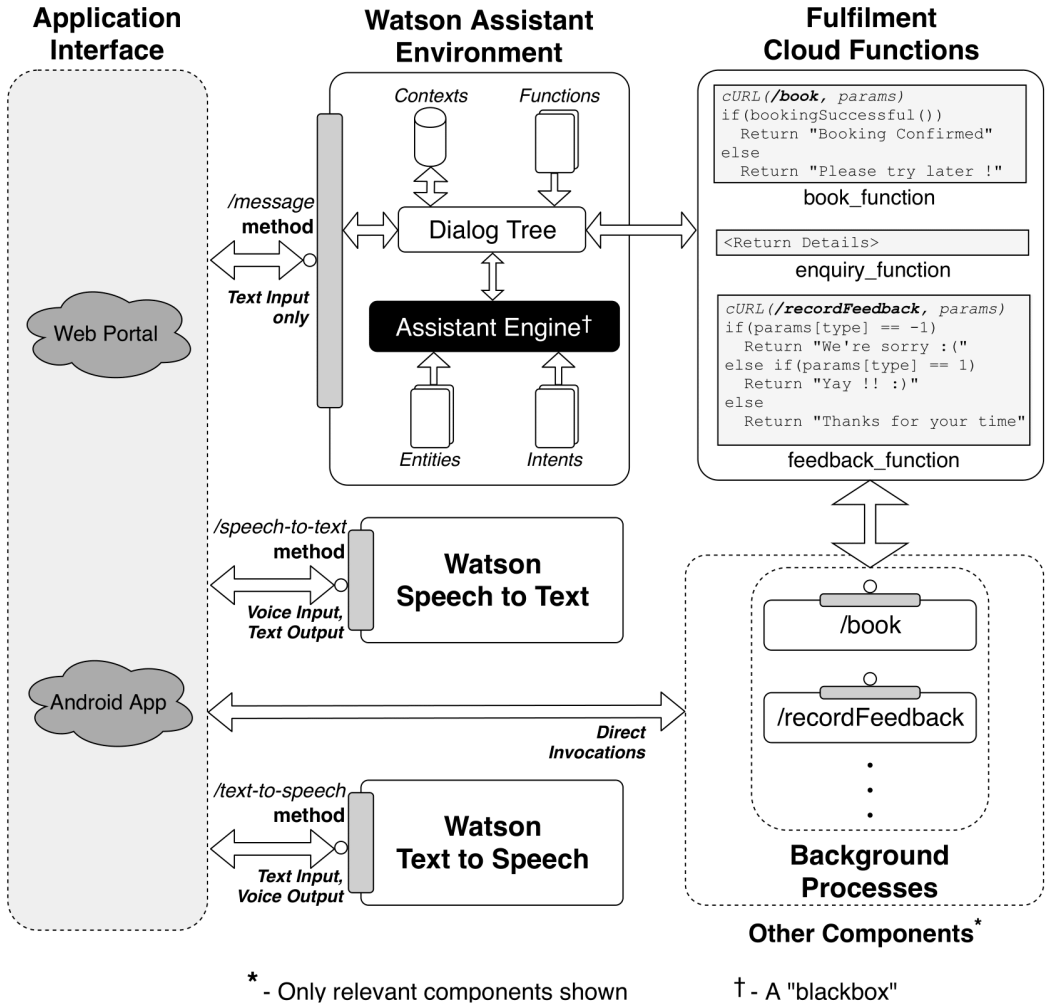


Application built with Watson Assistant

Chatbots with Dialogflow and Watson Assistant



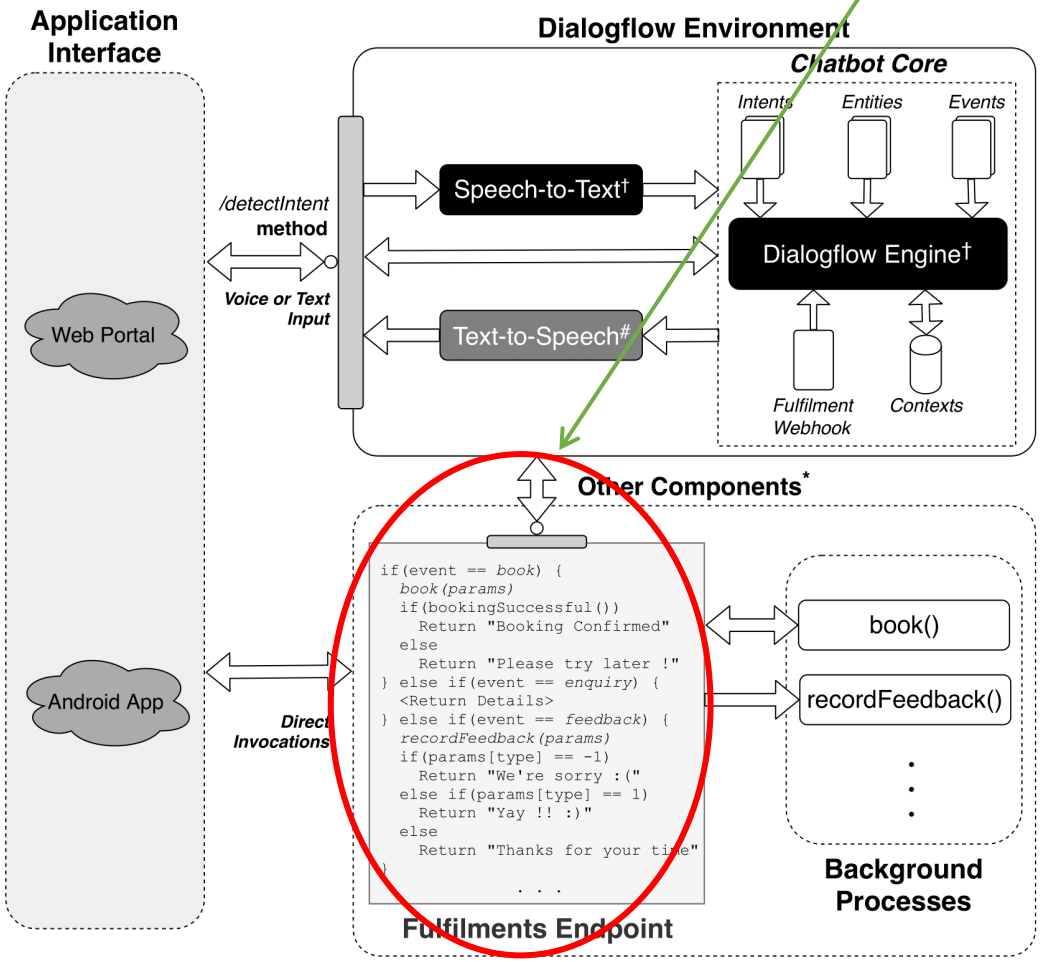
Application built with Dialogflow



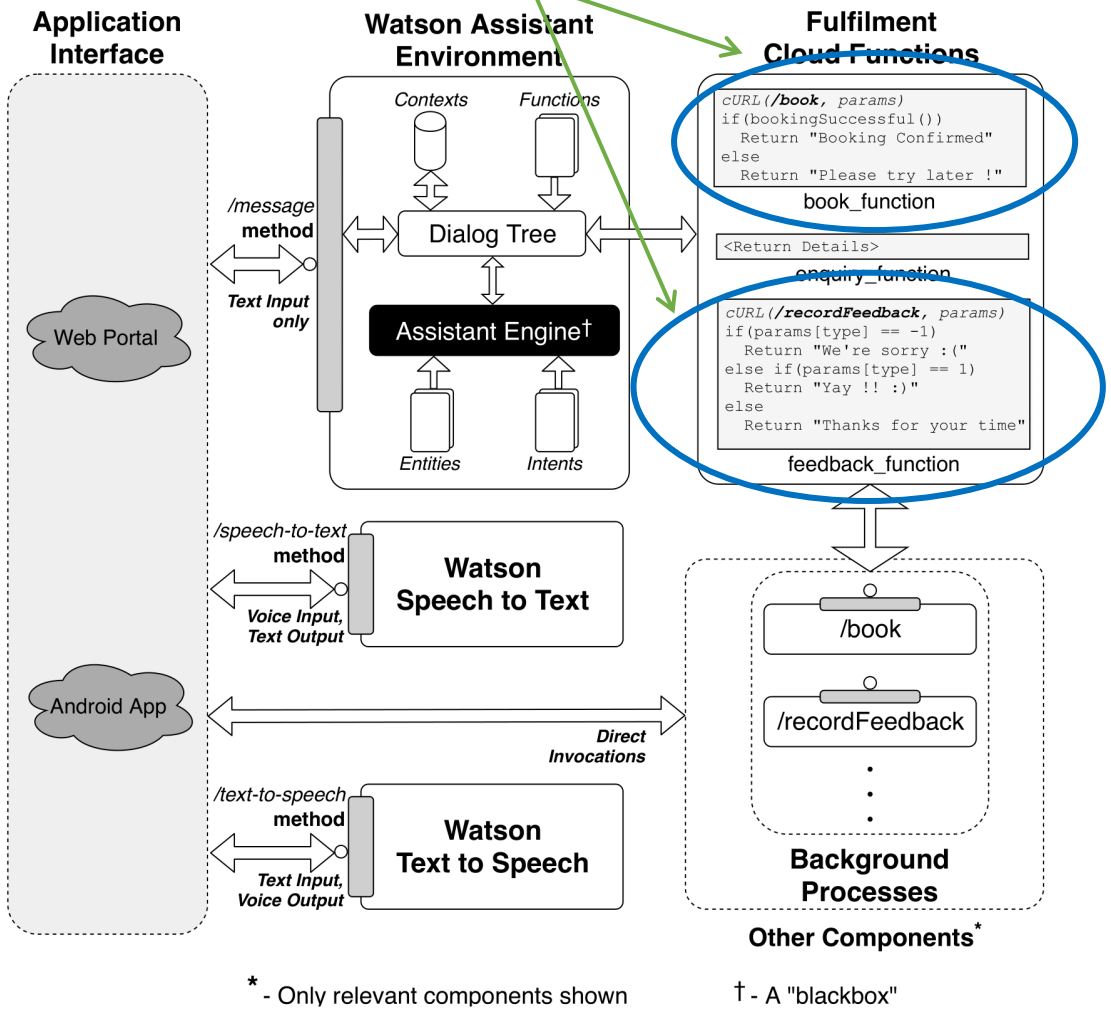
Application built with Watson Assistant

Single vs Multiple fulfilment hooks

Chatbots with Dialogflow and Watson Assistant



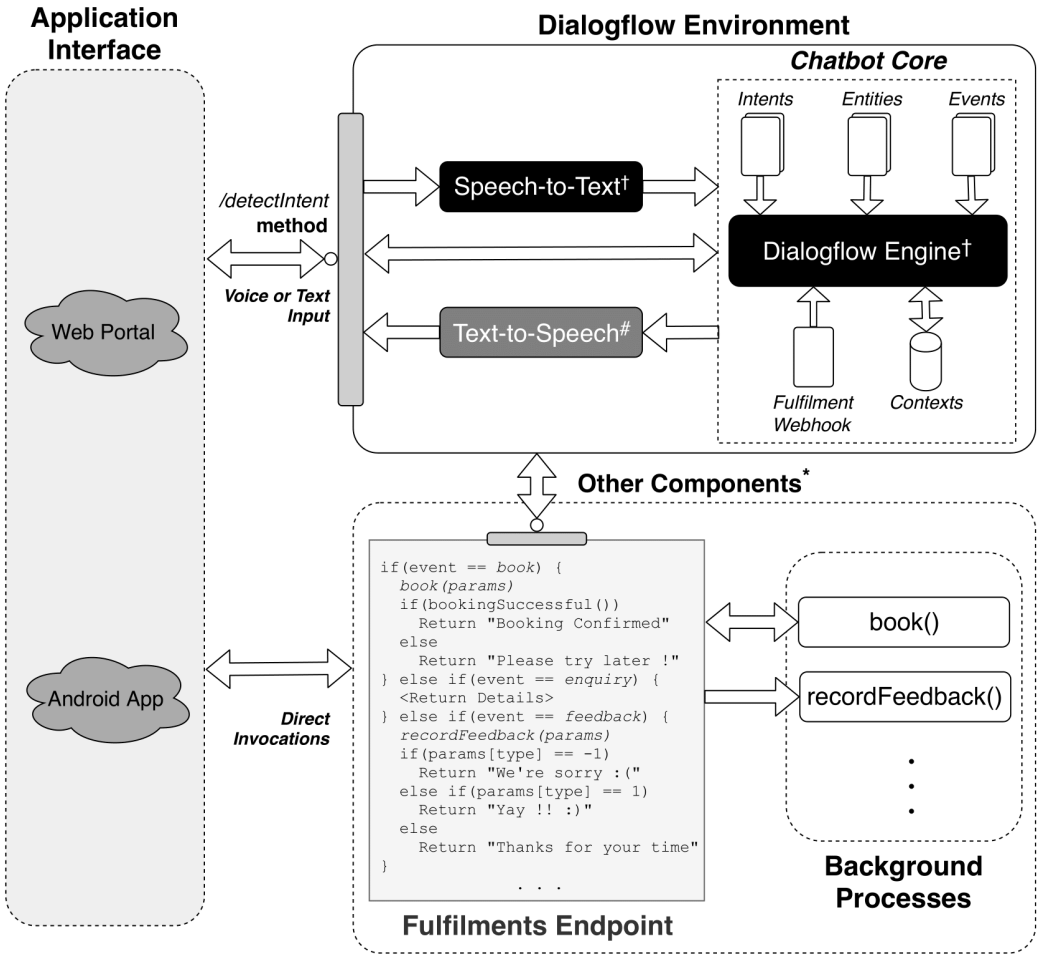
Application built with Dialogflow



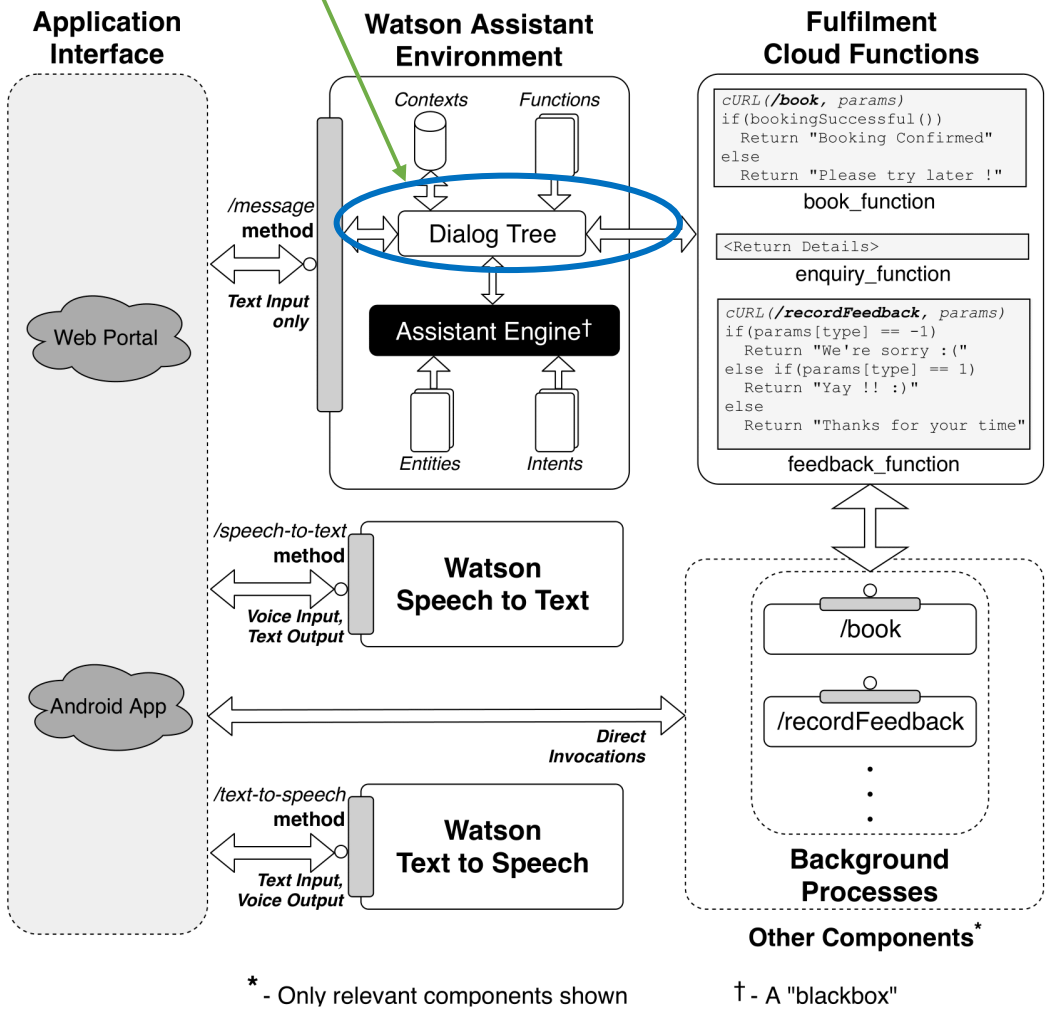
Application built with Watson Assistant

Implicit vs Explicit Flow Management

Chatbots with Dialogflow and Watson Assistant



Application built with Dialogflow



Application built with Watson Assistant

In a nutshell

- We presented a Reference Architecture for applications with conversational interfaces
- We showed the variation points in the architecture, which can yield different concrete architectures
- We showed concrete architectures of two applications, built using two commercial chatbot-building platforms – Google's Dialogflow and IBM's Watson Assistant
- We showed the variations in the application's architecture, in the above two cases

Thank You !!