FLY ME

FLIGHTS BOOKING CHATBOT

Clément Fleury

Challenge

Fly Me is a travel agency offering *flight booking services* for private or professional customers.

In order to make the booking process easier, we want to automate it with a **chatbot**.
To be efficient, the bot must be able to :

- **interact** with the user in a natural way
- recognize the user's **intent** (book a flight, ...)
- recognize the desired **caracteristics** of the trip (origin and destination cities, ...)
- **ask** for more information if necessary
- allow the user to **cancel** the booking process
- raise an **alert** if the booking process fails too often

Goals

In this project, we are going to :

- deploy a **chatbot** in the *cloud*
- train a **language understanding** model
  - in order to *infer* the user's intent and trip caracteristics from the user's input
- implement **unit and integration tests**
  - to ensure that the chatbot is *working as expected*
- monitor the chatbot's **performance**
  - raise an *alert* in case of recurring issues

Exploratory Data Analysis

For this MVP, we used the Azure Frames Dataset :

- **1369 dialogues** between a human and a bot, composed of **~15 turns** on average
- the bot can :
  - **ask** the user for information
  - **suggest** a trip to the user
  - ask the user to **confirm**
- the user can :
  - ask the bot for **suggestions**
  - **inform** the bot about relevant information
  - **cancel** or **confirm** the trip

Dialog Example

```
0 - user says :
"Can you get me to Kyoto"
Known facts :
{'intent': 'book', 'dst_city': 'Kyoto'}

1 - wizard says :
"Ok! From where?"
Known facts :
{'intent': 'book', 'dst_city': 'Kyoto'}

2 - user says :
"I need to be there for at least four days"
Known facts :
{'intent': 'book', 'dst_city': 'Kyoto', 'min_duration': '4'}

3 - wizard says :
"Have you a budget?"
Known facts :
{'intent': 'book', 'dst_city': 'Kyoto', 'min_duration': '4'}

4 - user says :
"I'm on the road so I can head there from any origin point. Budget is 3500"
Known facts :
{'intent': 'book', 'dst_city': 'Kyoto', 'min_duration': '4', 'or_city': '-1', 'budget': '3500.0'}

5 - wizard says :
"Travelling alone?"
Known facts :
{'intent': 'book', 'dst_city': 'Kyoto', 'min_duration': '4', 'or_city': '-1', 'budget': '3500.0'}

6 - user says :
"two adults. oh and please find me a place near a park"
Known facts :
{'intent': 'book', 'dst_city': 'Kyoto', 'min_duration': '4', 'or_city': '-1', 'budget': '3500.0', 'park': True, 'n_adults': '2'}

7 - wizard says :
"I can get you 4 days in Kyoto for 1857.63USD if you leave from Sapporo."
Known facts :
{'intent': 'book', 'dst_city': 'Kyoto', 'min_duration': '4', 'or_city': 'Sapporo', 'budget': '3500.0', 'park': True, 'n_adults': '2', 'duration': '4', 'price': '1857.63'}
...
```

Step 1 : LUIS Model Training And Deployment

The first step in this project was to train and deploy a **language understanding** model using the LUIS *natural language service* :

- create an **Azure LUIS** resource
  - add the **intents** and **entities** to the *LUIS model*
- **format** the *raw data* to be compatible with Azure LUIS
  - add the **examples** to the LUIS model
- run the LUIS model **training**
- **deploy** the model to the *cloud*

Step 2 : Chatbot Development And Deployment

The second step in this project was to develop and deploy a **Chatbot** :

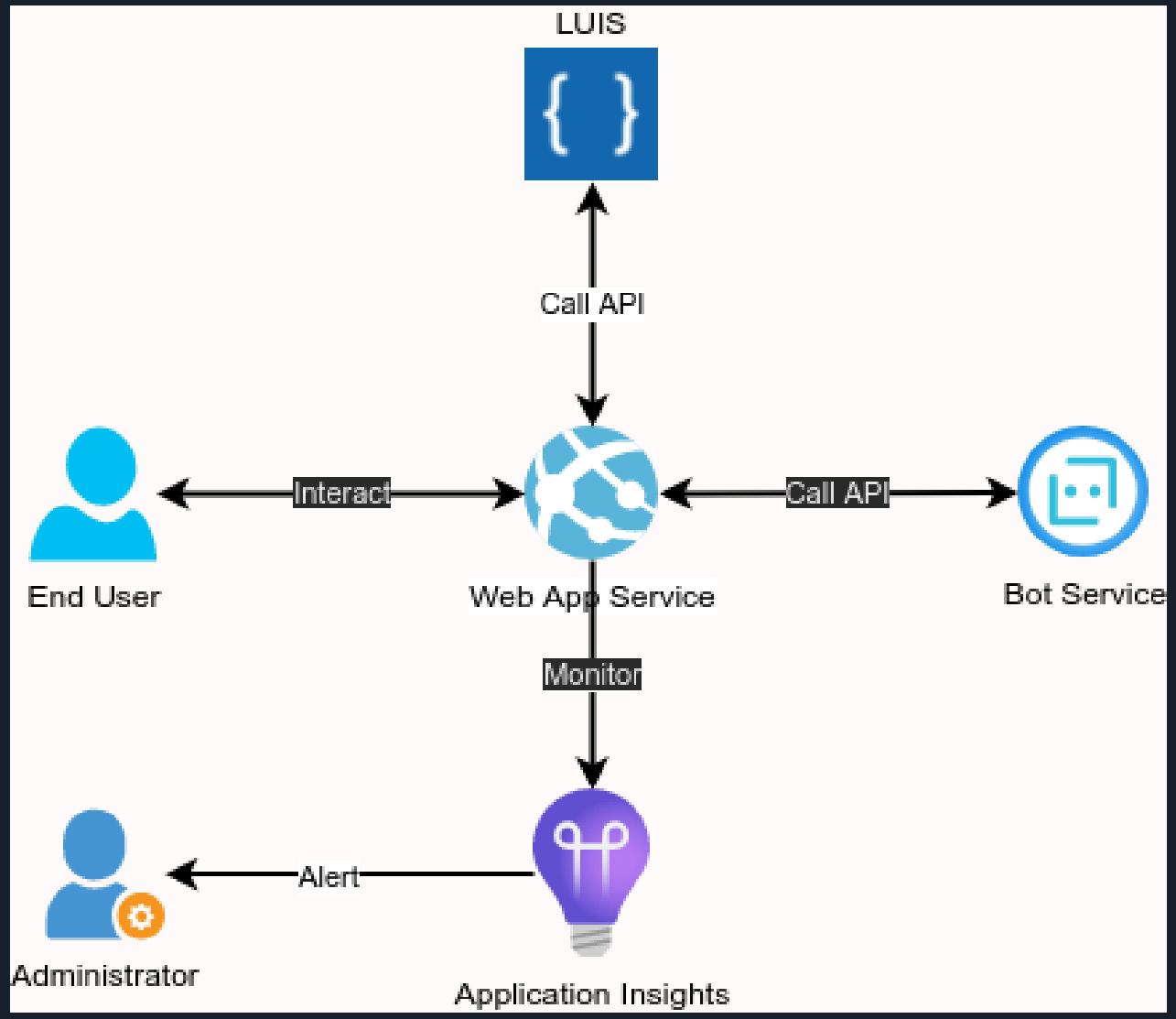
- create a **Bot Service** resource
- implement the bot with the **Bot Framework SDK**
  - implement *dialogs* : welcome, gather intention, ask informations, confirm, cancel, ...
  - integrate the *LUIS service*
  - add *unit and integration tests*
    - test the *LUIS service*
    - test the LUIS service *integration*
    - test a *standard dialog*
- deploy the bot to the **Azure Web App**

Step 3 : Chatbot Monitoring

The final step in this project was to monitor the **chatbot's performance** :

- create a **Application Insights** resource
- log a **Telemetry trace** every time the bot is used (`booking_accepted` or `booking_refused`)
- add an **Alert** if the bot fails too often :
  - if *More than 5 Booking Refused last 5 minutes*
  - then *Send Email and SMS to admins*

Curent MVP System Architecture

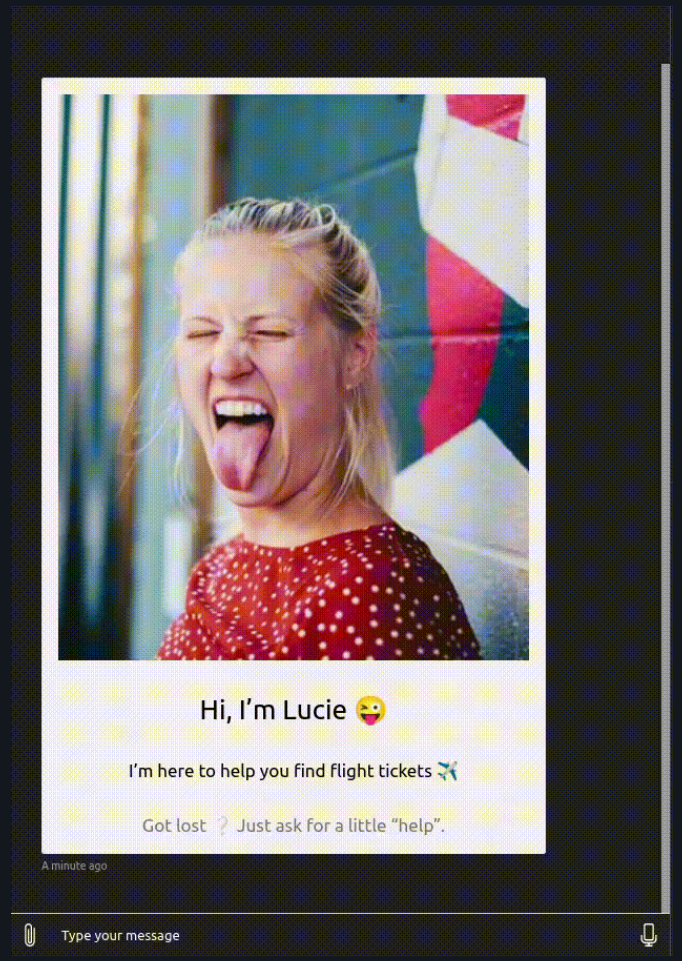


Resources Inventory

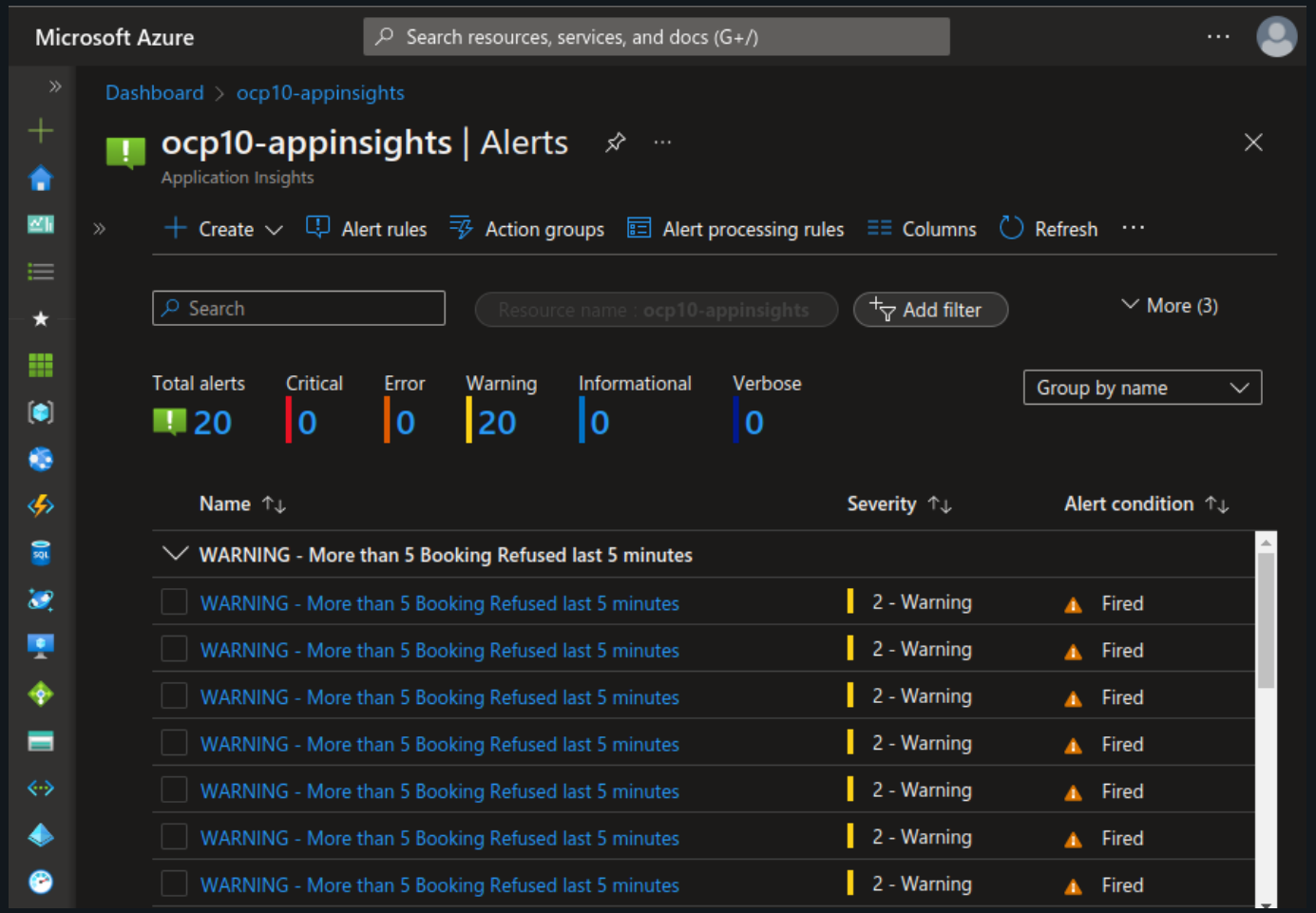
LUIS	Chatbot	Monitoring
Language understanding model	Bot service	Application Insights
Authoring resource	App Service Plan	Alert Rule
Prediction resource	App Service	Monitoring Dashboard

Demo

Demo



Monitoring

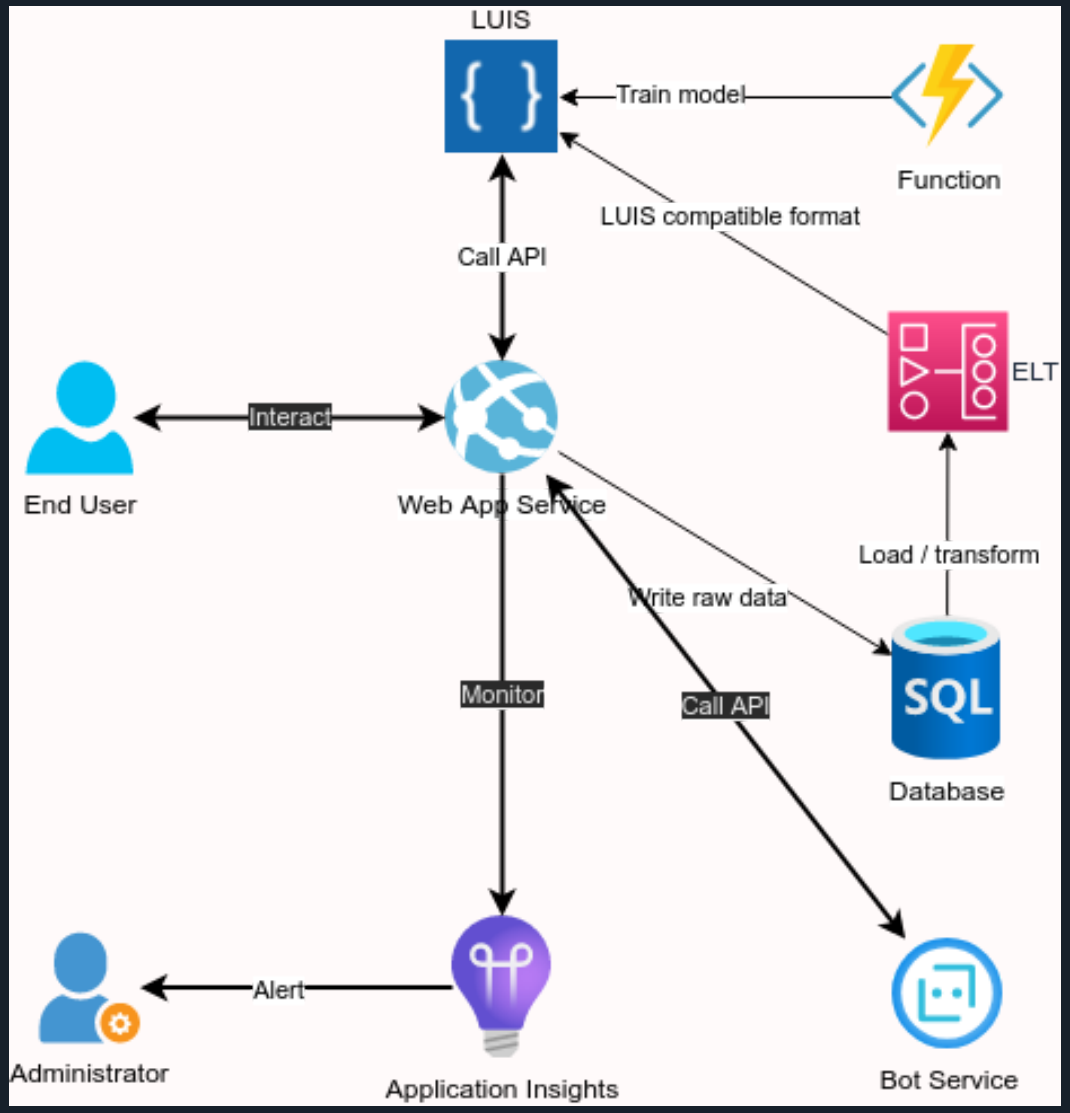


Target Performance Management Policy

In order to achieve the target performance, the following policy must be implemented :

- **store** each dialogs and inferred *intents* and *entities* in a *database*
- **transform and load** (ETL) the raw data in a *datawarehouse* in a *format compatible with the LUIS service*
- daily **re-train** the *language understanding model* with the new *intents* and *entities* of the **successful** bookings

Target Production Architecture



Next Steps

- **integrate** the bot with multiple *Channels* (Website, Discord, Teams, Slack, ...)
- **improve** the bot capacity to handle more *Intentions* and *Entities*
- **connect** the bot to an actual *Flight booking* system
- **monitor** more precisely the bot's performance : *errors*, *performance*, *availability*, ...
- **implement** the model continuous *training* and *deployment*

FLY ME

FLIGHTS BOOKING CHATBOT