

CHATBOT



Cours de Programmation de ChatBots

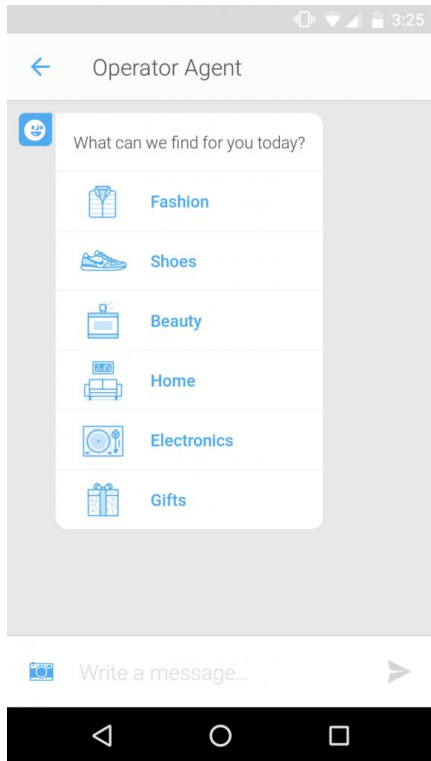
AVEC NODE.JS, DIALOGFLOW ET
SLACK

Sommaire

- Introduction
 - Qu'est-ce qu'un Chatbot ? – exemples
 - Historique
 - Usages
 - Caractéristiques d'une conversation (types de conversation) (cours ?)
 - GUI versus CUI et VUI (cours 2 ?)
- Architecture fonctionnelle
- Panorama des technologies (cours 2)
- Architecture et technologies du TP
- DialogFlow (TD1)
- Webhook, openAPI, Node.js et ngrok (TD2)
- Slack (TD3)
- Contexte et SlotFilling (TD4)

Introduction

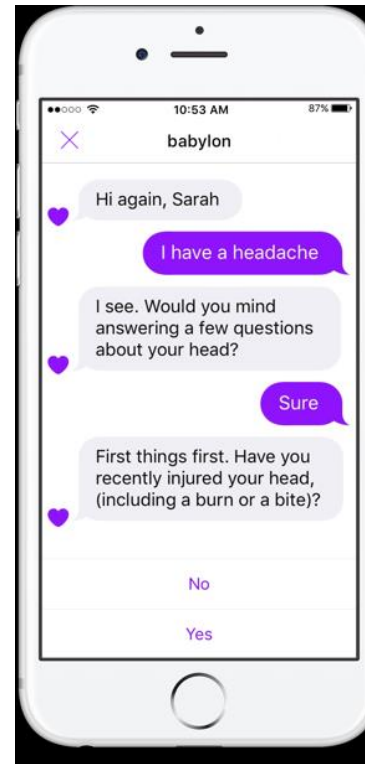
Chatbots !!



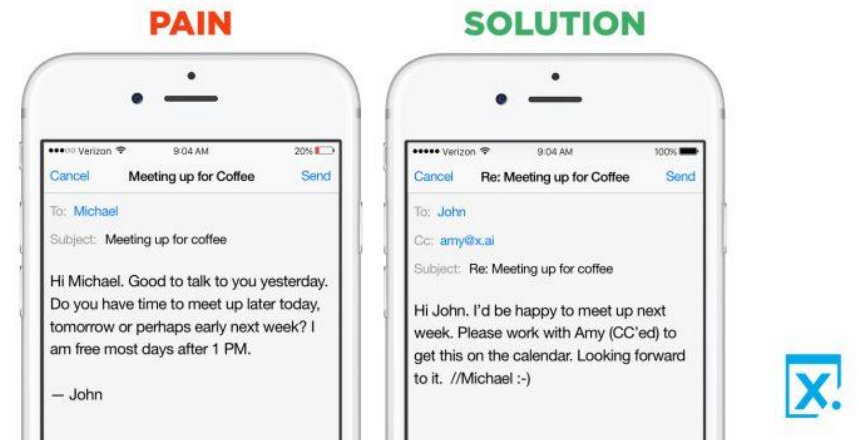
Ecommerce:Operator



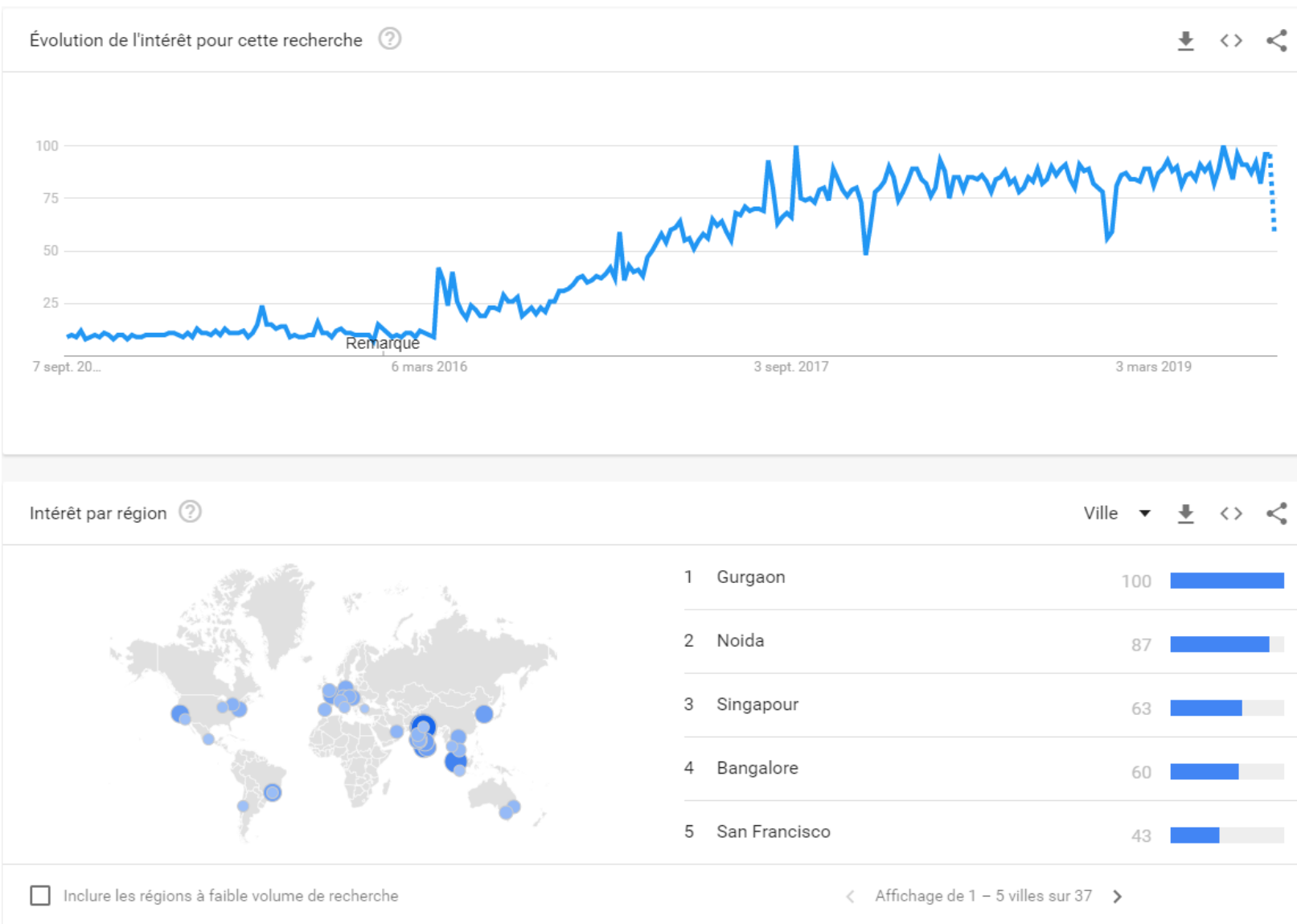
Duolingo



Babylon Health



x.ai



Evolution de l'intérêt (Google trends)

Qu'est-ce qu'un Chatbot ?

- Une « nouvelle » forme d'interface homme-machine qui échange avec l'utilisateur en langage naturel ou pseudo-naturel :
 - via un support texte (Messagerie instantanée, email, etc.)
 - via un support vocal (Reconnaissance et synthèse vocale)
- Un chatbot expose des services métiers au travers d'une interface conversationnelle.
- Suivant sa sophistication, il peut embarquer des modules d'IA (reconnaissance d'émotion, de visages, etc.)
- Permet de communiquer avec une entreprise via une application de messagerie
- Facebook, Kik, Slack, Skype, etc. investissent dans cette technologie (**entre 2015 et 2016 le temps passé sur messagerie a augmenté de 394% - un adulte américain moyen passe 5 heures / jour sur smartphone dont 50% sur messagerie**).

Historique

(MIT 1960)

ELIZA

*Pattern
matching*

(Carnegie Mellon 1995)

A.L.I.C.E

*AIML
heuristiques*

(AOL, Messenger 2000)

SmarterChild



(IBM 2006)



(Siri Apple 2010)



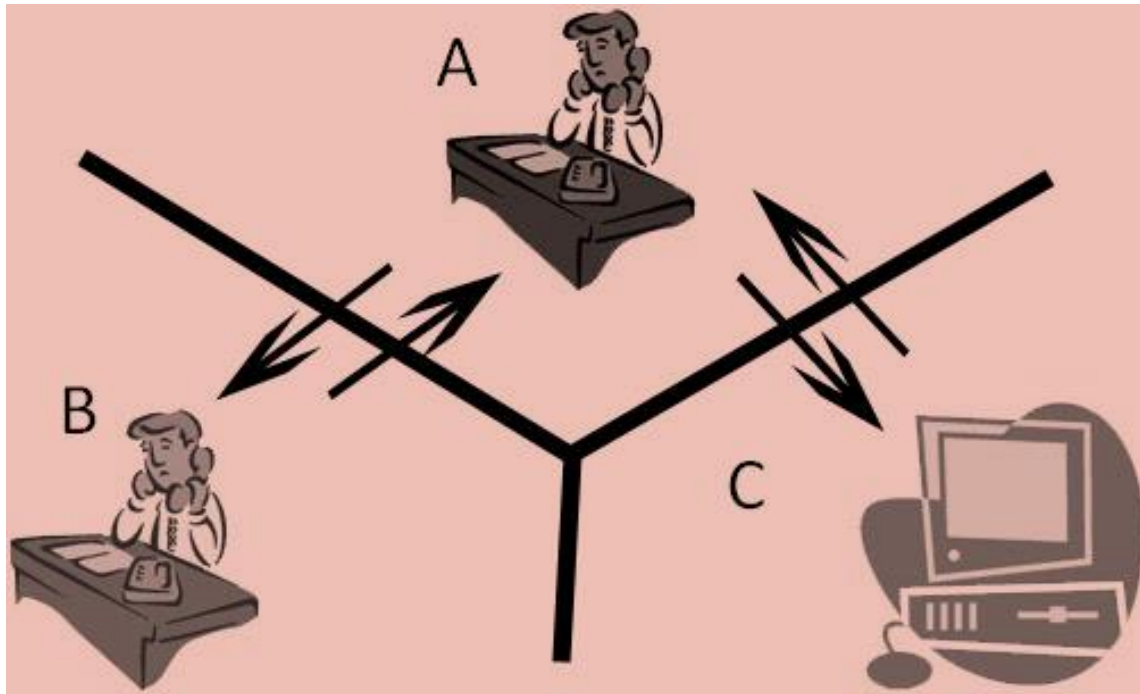
(Amazon Alexa 2015)



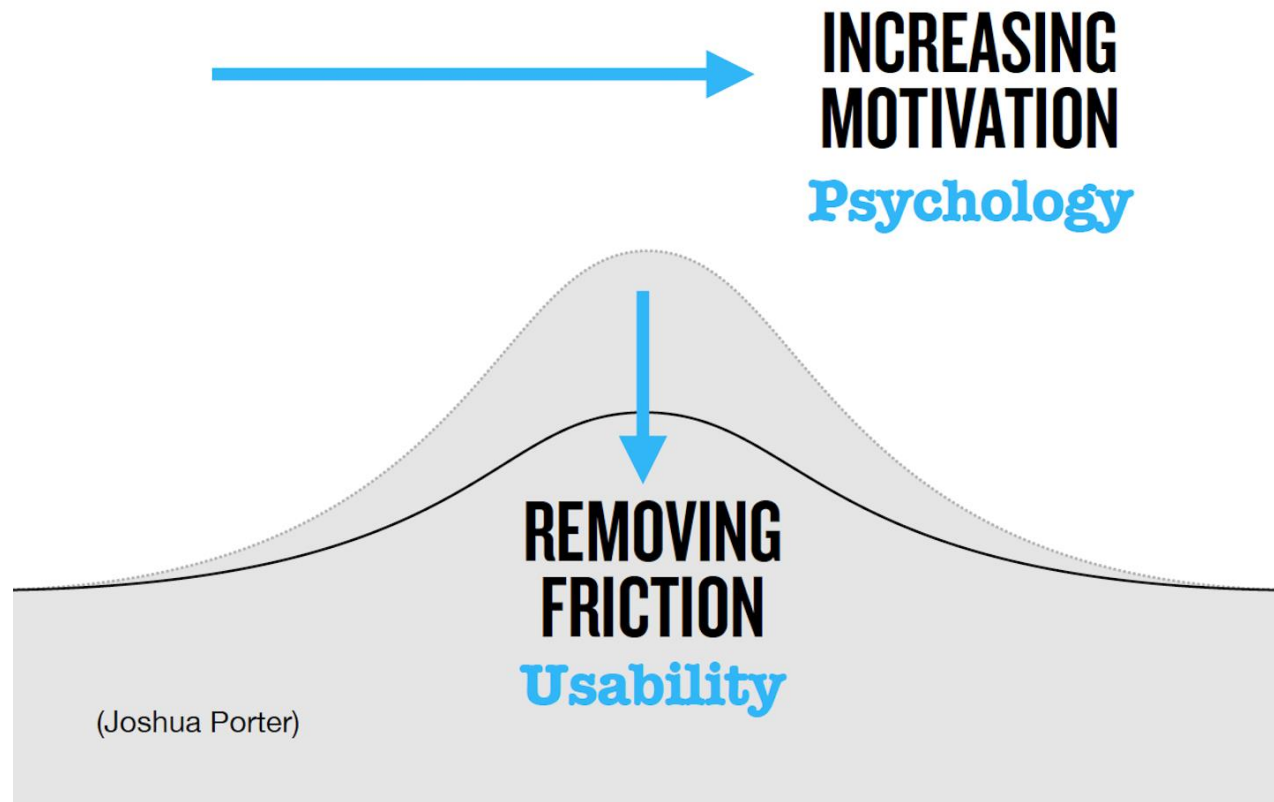
(Facebook messenger
Plateforme 2016)



Test de Turing



- Démontre que la conversation est considéré comme une des caractéristiques fortes de l'être humain – pour être intelligent il faut converser comme un être humain.
- Prix Loebner
 - ELIZA
 - ALICE
 - Mitsuku
- Royal Society 2014
 - Eugène Goostman (33% de juges dupés)



Quelle qualité
essentielle pour
une interface
homme machine
?

« Le vrai problème avec une interface, c'est que c'est une interface » Don Norman

« La meilleure interface, c'est pas d'interface » Golden Krishna

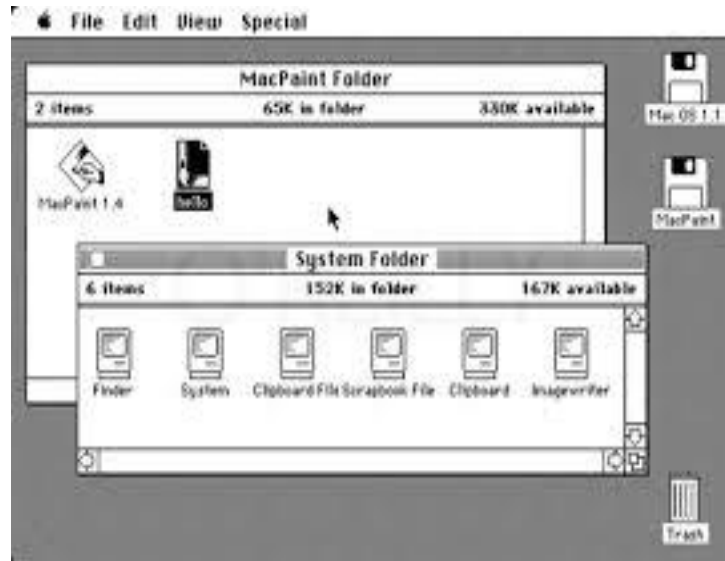
De la ligne de commande à l'interface conversationnelle

```
hightowe@utopia: ~/bin/kpcli.d$ ./kpcli-0.9.pl
KeePass CLI (kpcli-0.9) v0.9 is ready for operation.
Type 'help' for a description of available commands.
Type 'help <command>' for details on individual commands.

kpcli-0.9: /> open /tmp/db.kdb
Please provide the master password:
kpcli-0.9: /> ls /
=== Groups ===
eMail/
Internet/
test/
Backup/
kpcli-0.9: /> ls test
=== Entries ===
0. test
kpcli-0.9: /> show test/test

Title: test
Uname: test
Pass: test
URL: test
Notes: test
kpcli-0.9: />
```

« La mémoire de reconnaissance est plus efficace que la mémoire de rappel » Alan Baddeley



« Mais une GUI reste une création artificielle à laquelle les utilisateurs doivent s'adapter » Nick Babich



« Le langage est la meilleure et la plus effective technologie pour communiquer » Golden Krishna

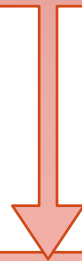
Pourquoi les chatbots maintenant ?

Pourquoi les chatbots ?

La langue naturelle est la plus ancienne forme de communication pour les humains

Usage main libre (pour les VUI, couplé à la réalité mixte, accès à l'IOT, etc.) se développent

Facilité de communication même pour des usagers n'utilisant pas régulièrement des ordinateurs



Pourquoi maintenant ?

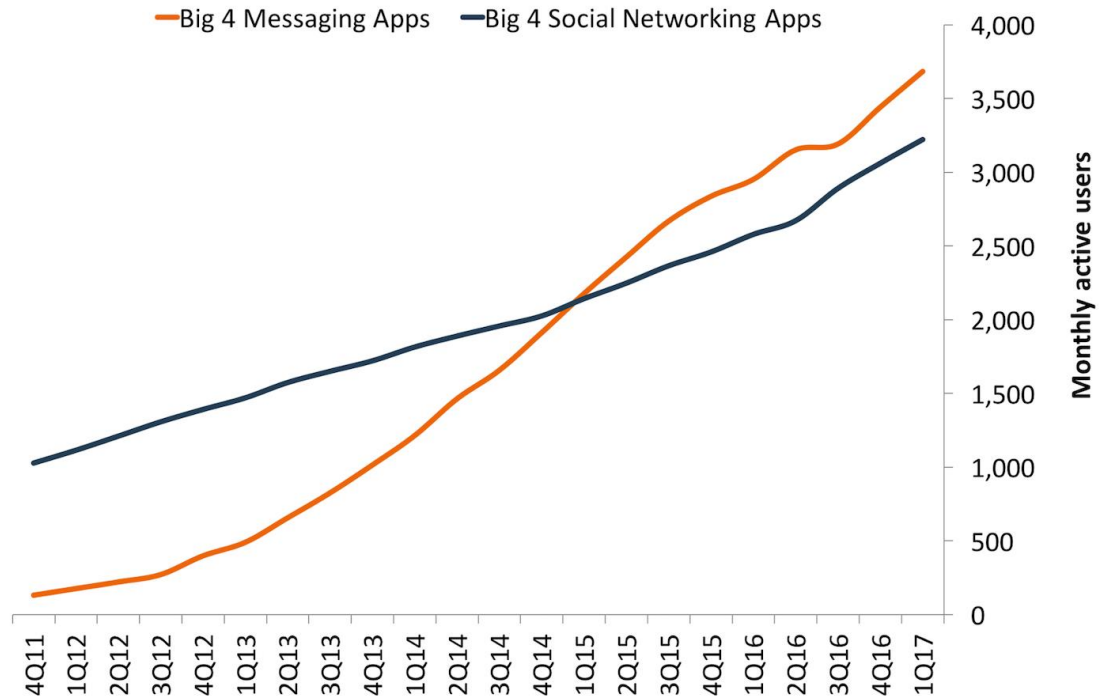
Mise en œuvre d'une approche probabiliste plutôt que grammaticale dans la compréhension du LN

- Progrès de la reconnaissance vocale
- Progrès de la compréhension du langage

Usage sans cesse grandissant des messageries au détriment des applications

Messaging Apps Have Surpassed Social Networks

Global monthly active users for the top 4 messaging apps and social networks,
In millions



Note: Big 4 messaging apps are WhatsApp, Messenger, WeChat, Viber.

Big 4 social networks are Facebook, Instagram, Twitter, LinkedIn

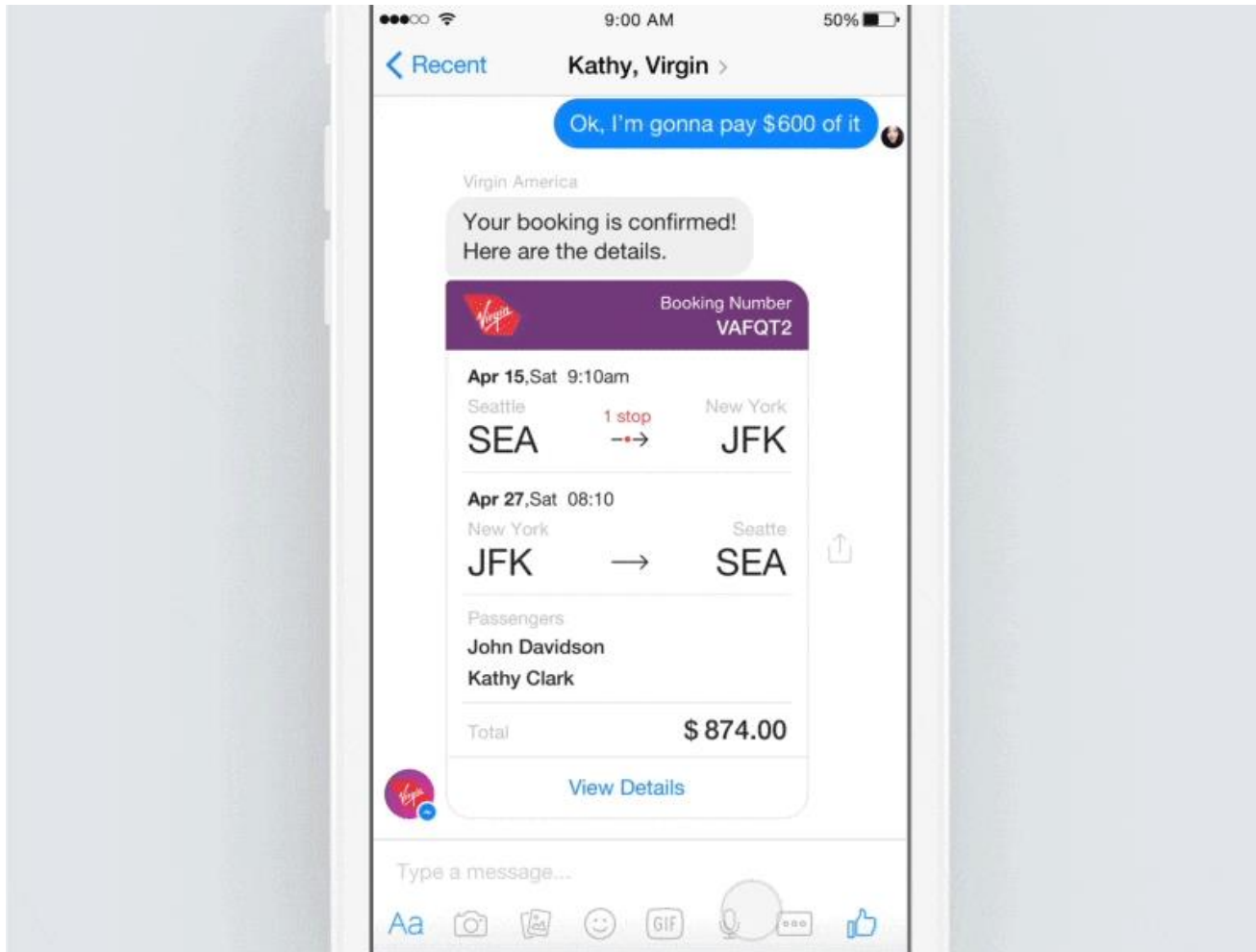
Source: Companies, Apptopia, TechCrunch, BI Intelligence estimates, 2017

BI INTELLIGENCE

Les messageries dépassent les réseaux sociaux

Avantages des chatbots

- Plus naturel que les applications – usage via le langage naturel
- Pas besoin d'apprendre une nouvelle interaction
- Permet une meilleure personnalisation – le CUI peut adapter son contenu à son interlocuteur
- Pas besoin de télécharger une application pour chaque marque
- Pas de jugement, les utilisateurs trouvent plus confortable de partager des informations avec un bot plutôt qu'avec une personne car les bots ne les jugent pas.
- Permet simplement de partager une interface / un service avec un ami ou un collègue.



Partager la
même interface

Usages d'aujourd'hui

○ **Commercial / Marketing B2C**

- Accéder aux clients via leurs messageries
- Notification aux client de nouveaux produits – garder le lien
- Service après-vente toujours disponible – libérer les agents des questions simples.
- Collecter des données sur le client lors d'une conversation « naturelle »

○ **Métier / RH B2B**

- Assister un collaborateur dans certaines de ses tâches quotidiennes
- Renseigner un agent commercial sur un produit, notifier l'arrivée de nouveau client, organiser une réunion, etc.
- FAQ
- Chatops

Usages d'aujourd'hui

○ **Coaching, Assistant personnel**

- Santé
- Sport
- etc.

○ **Jeux**

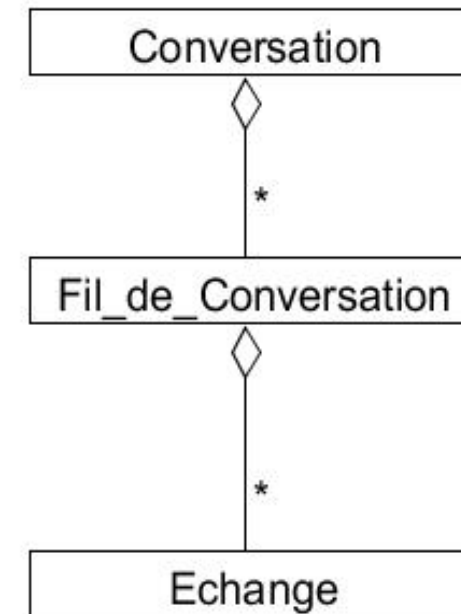
- IA des personnages de RPG
- etc.

Usages de demain

- Home Assistant
- Voiture autonome
- Robot d'Accueil d'entreprise
- Etc.

Caractéristiques d'une conversation

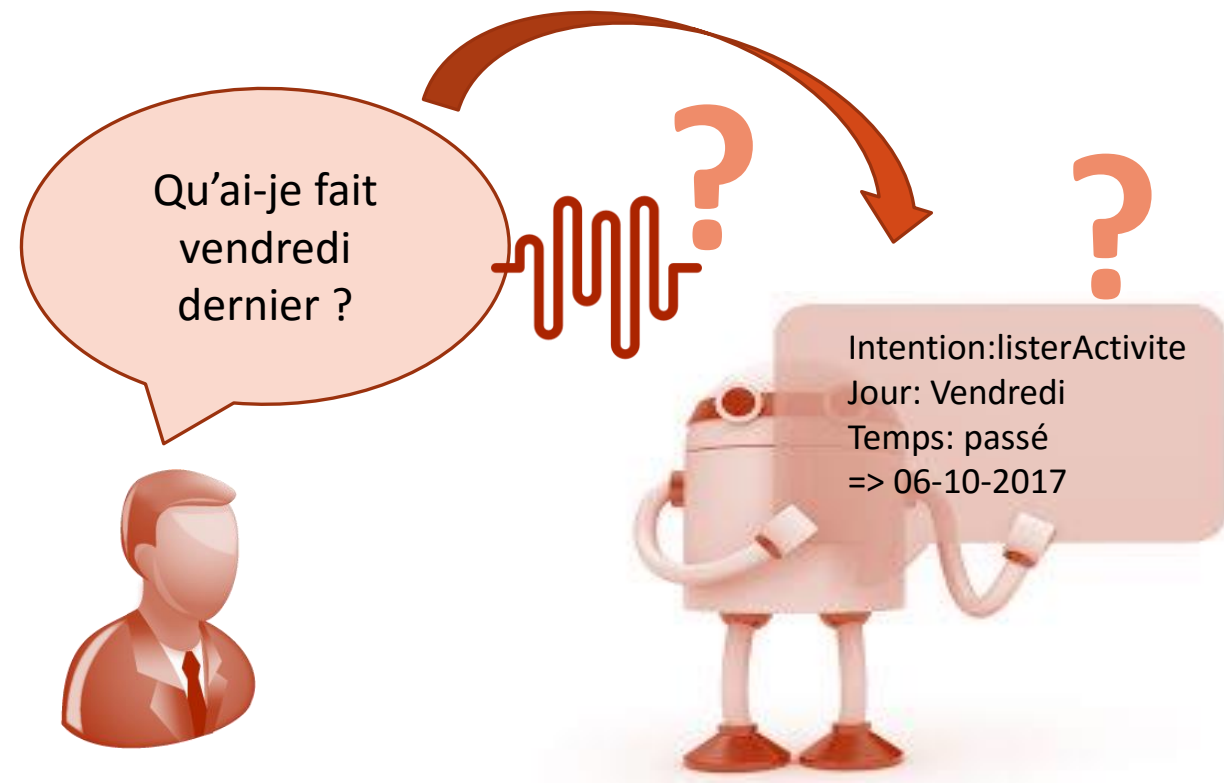
- Echange élémentaire
 - Question / réponse
 - Initiative de l'utilisateur ou du bot
- Fil de conversation
 - Ensemble d'échanges visant la récupération d'informations (commander un café, demander la météo, s'identifier auprès du bot, etc.)
- Conversation
 - Ensemble de fils de conversation
 - **Séquentiels** ou entrelacés (mémoire de conversation)



GUI vs CUI (VUI)



VS



GUI versus VUI ou CUI

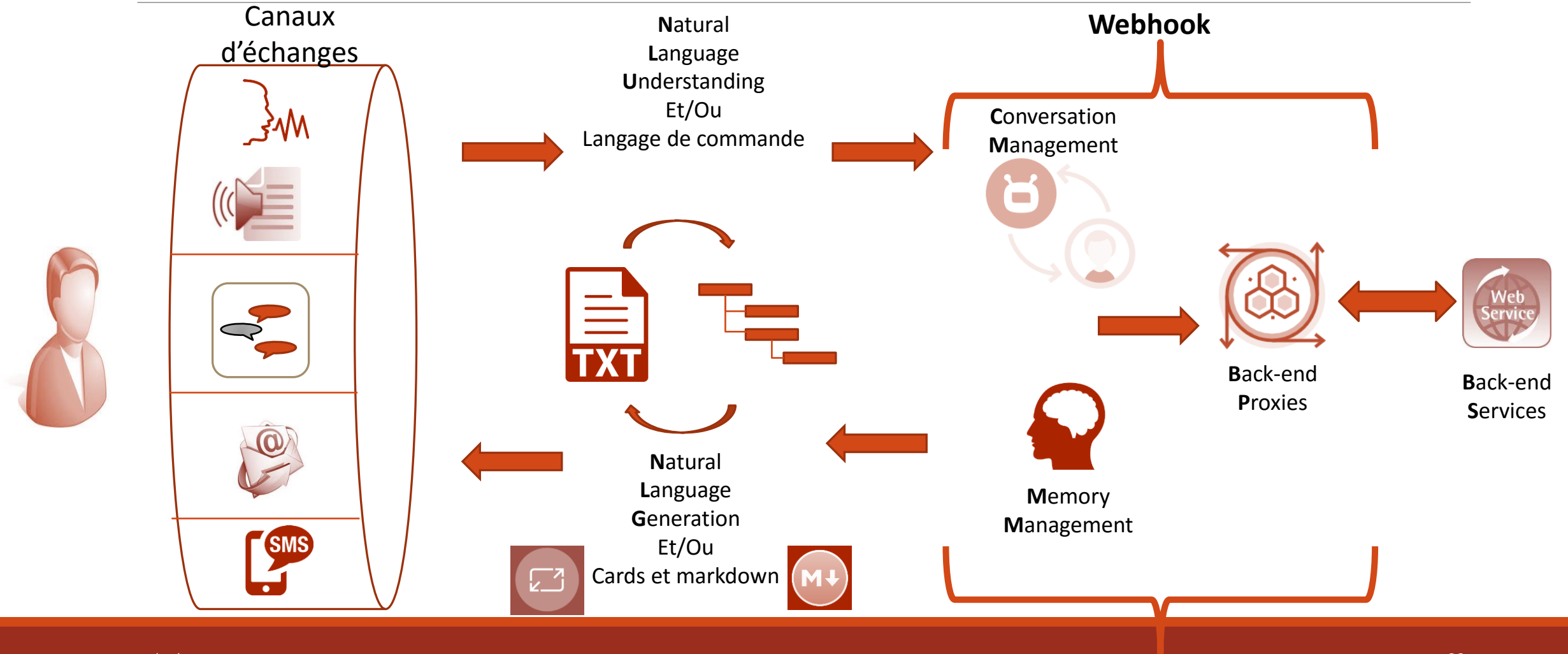
- **Graphical User Interface** : c'est l'utilisateur qui s'adapte au langage graphique proposé par l'informatique – langage basé sur des interactions clavier + souris + écran(s).
- Dynamique globale car le regard embrasse tout l'écran **MAIS nécessite de focaliser sa vision.**
- Permanence des informations affichées
- Multi-tâches
- Plus complexe de prime abord
- Très bien adapté à l'affichage de données complexes
- **Conversational ou Voice User Interface** : basé sur du texte ou de la voix, c'est ici l'ordinateur qui s'invite dans un mode d'échanges purement humain depuis des millénaires – c'est à lui de s'adapter sous peine de rejet.
- Dynamique séquentielle par essence même de la voix **MAIS libère le regard.**
- Volatilité des données parlées
- Mono-tâche
- Plus naturelle de prime abord
- Très adapté à un langage de commandes et quand l'utilisateur sait ce qu'il veut



Ils se complètent et on devrait s'orienter vers des interfaces multi-modales
D'ailleurs les bots textuels intègrent déjà des images, tableaux, etc.

Architecture Fonctionnelle

Architecture fonctionnelle et dynamique des flux



Canaux d'échanges

□ Vocal

- ASR : Automatic Speech Recognition – Reconnaissance vocale
 - Transformer des ondes sonores en texte
- TTS : Text to speech – Synthèse vocale
 - Transformer du texte en sons

□ Textuel

- Messagerie – FB messenger, Teams, Slack, etc.
 - Récupérer une saisie de texte
 - Afficher un texte – enrichi par d'autres interacteurs (bouton, image, carroussel, etc.)
- Email
- SMS

Modules du Chatbot

❑ NLP Natural language Processing

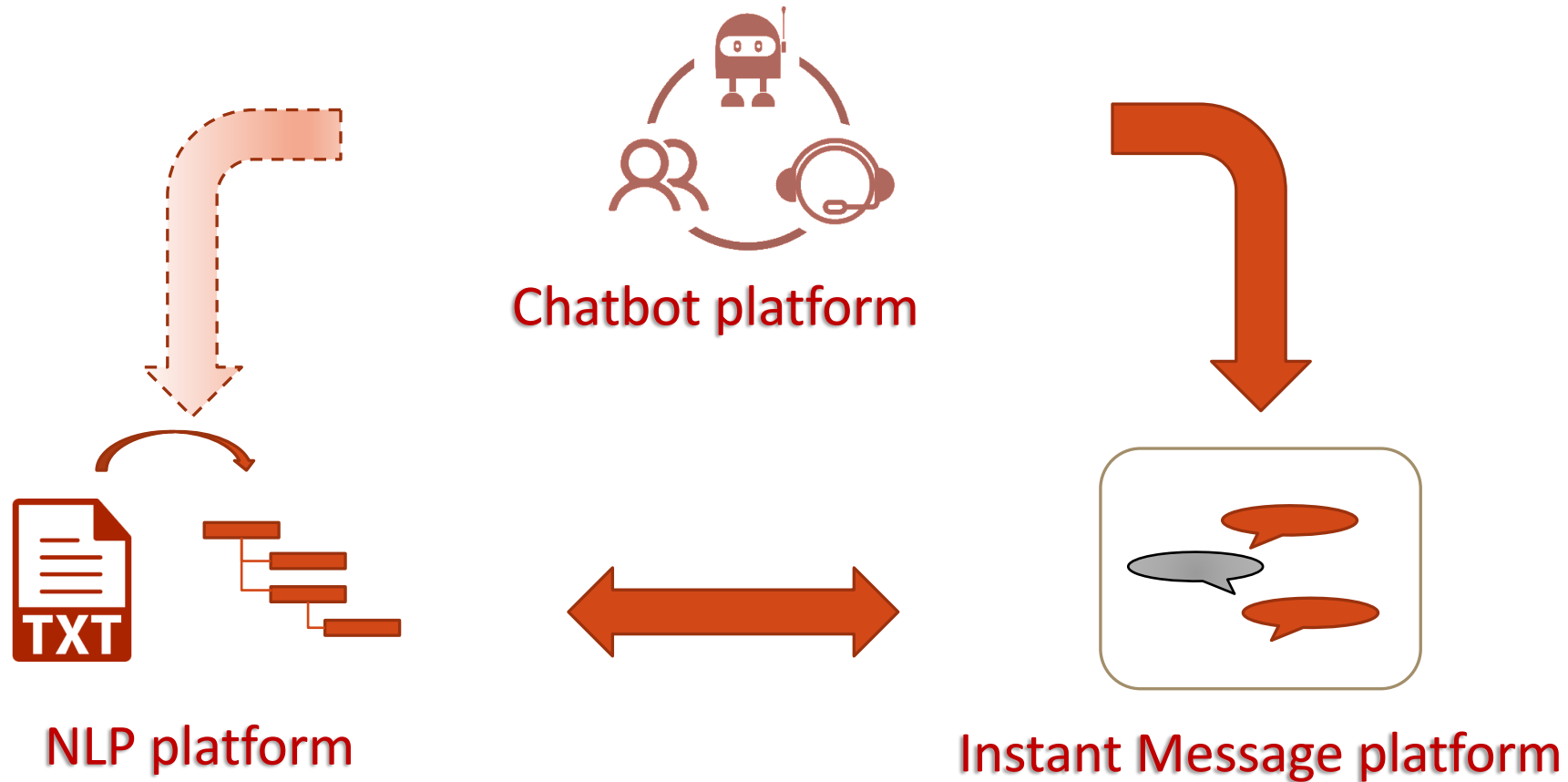
- NLU : transformer du texte en une structure d'intention et paramètres
- Langages de commandes
- NLG : transformer une structure de données en un texte en langage naturel
 - être capable de s'adapter aux variations et d'engendrer des variations
 - Mise en œuvre de techniques de Machine Learning (régression logistique, réseaux de neurones, etc.)
- Cards et markdown : tuiles reprenant des aspects des GUI (carroussels, boutons, liste, graphiques, etc.)

❑ Conversation Management

- Thread conversation Management : gérer les fils de conversations
- Memory Management : gérer une mémoire de conversation
- Back-end Proxies : s'adapter en entrées / sorties aux API back-end qui offrent les fonctionnalités interfacées par le Chatbot

Panorama des technologies

Panorama des technologies



NLP Platform

- Fonction première : compréhension d'une phrase pour la transformer en une structure intention et entités.
 - Détecter la bonne intention
 - Extraire les valeurs des entités qui paramètrent l'intention
 - Interpréter correctement tous les exemples d'entraînement mais aussi être capable de généraliser au-delà
- Fonction secondaire : génération de réponse souvent limitée à l'usage de template pouvant être tirés au hasard
- Caractéristiques
 - Cloud / On-premise
 - Open source / propriétaire
 - Pure NLU / NLU +NLG + gestion de conversation
- Solutions
 - Snips, rasa, DialogFlow, Luis, wit.ai, IBM Watson

Instant Message Platform

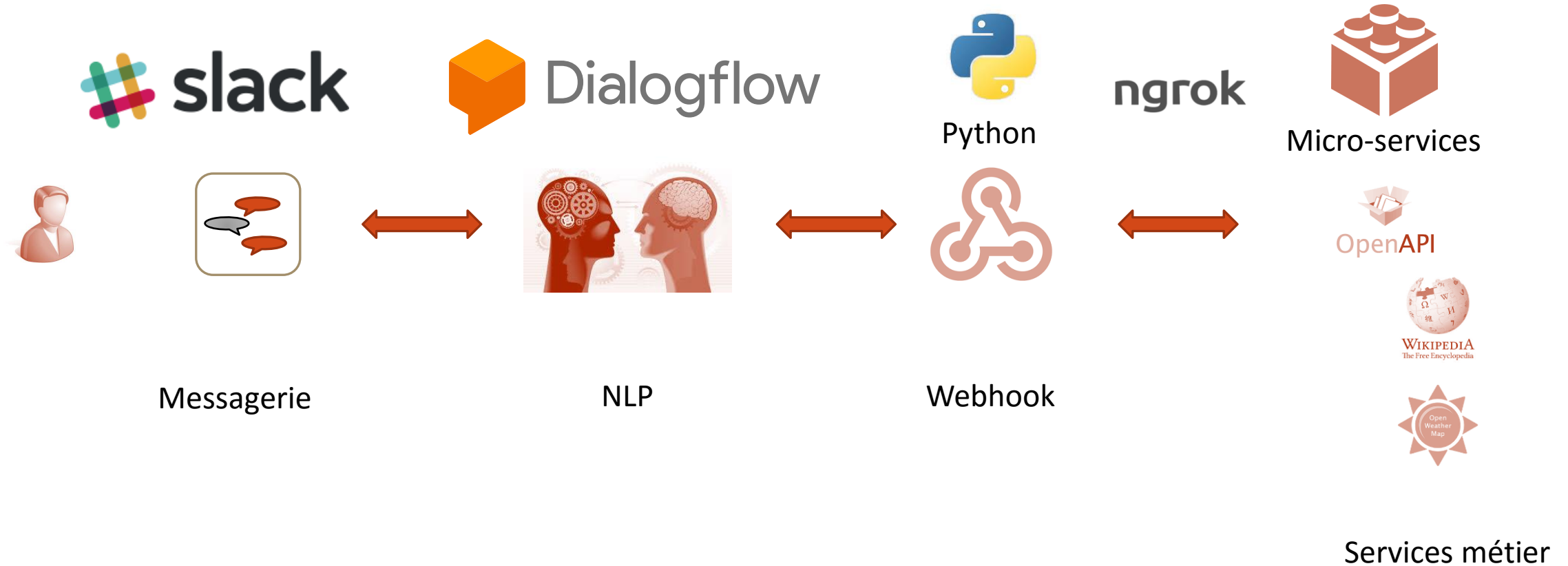
- Usage Grand public
 - Facebook Messenger
 - Telegram
 - Whatsapp
 - Kik
 - Skype
- Usage Professionnel
 - Slack
 - Skype
 - Teams
 - Hang-out

Chatbot platform

- Sans code – Outil de programmation visuel
 - Chatfuel FB principalement
 - Botsify FB mais aussi Web site
 - Flow XO FB, Slack, Telegram, la plupart des messageries
- Avec code
 - IBM Watson
 - Microsoft bot framework FB, Teams, Skype, Slack, etc.
 - Botkit (outil d'intégration) FB, Slack, Skype, DialogFlow, etc.

Architecture et technologies des TD

Vue d'ensemble

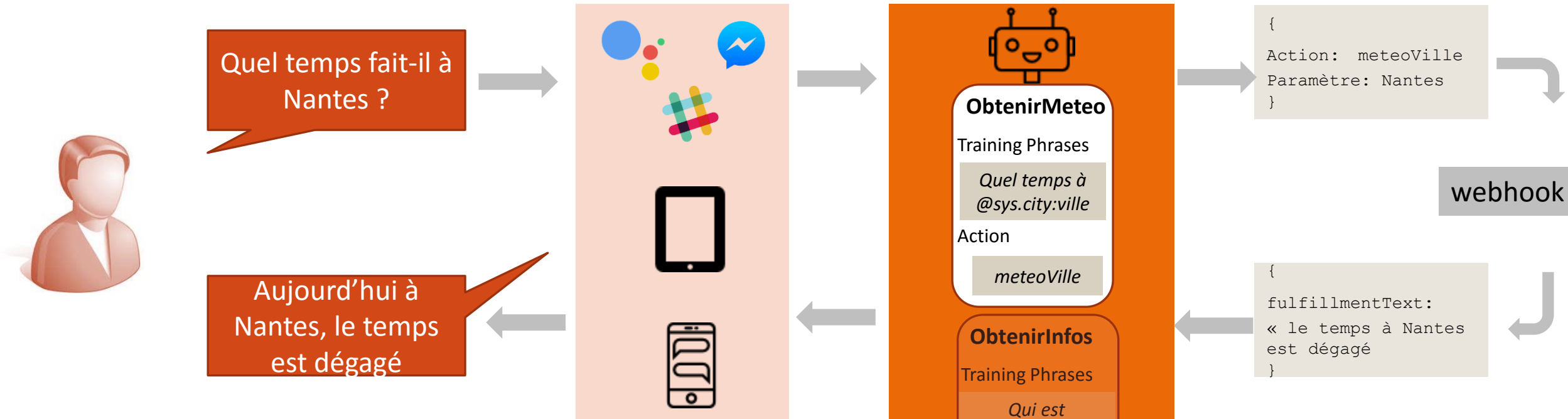


DialogFlow

Fonctionnalités de DialogFlow

Intégration avec une messagerie,
Webapp ou Mobile app

**Intégration avec
un backend
(Fulfillment)**

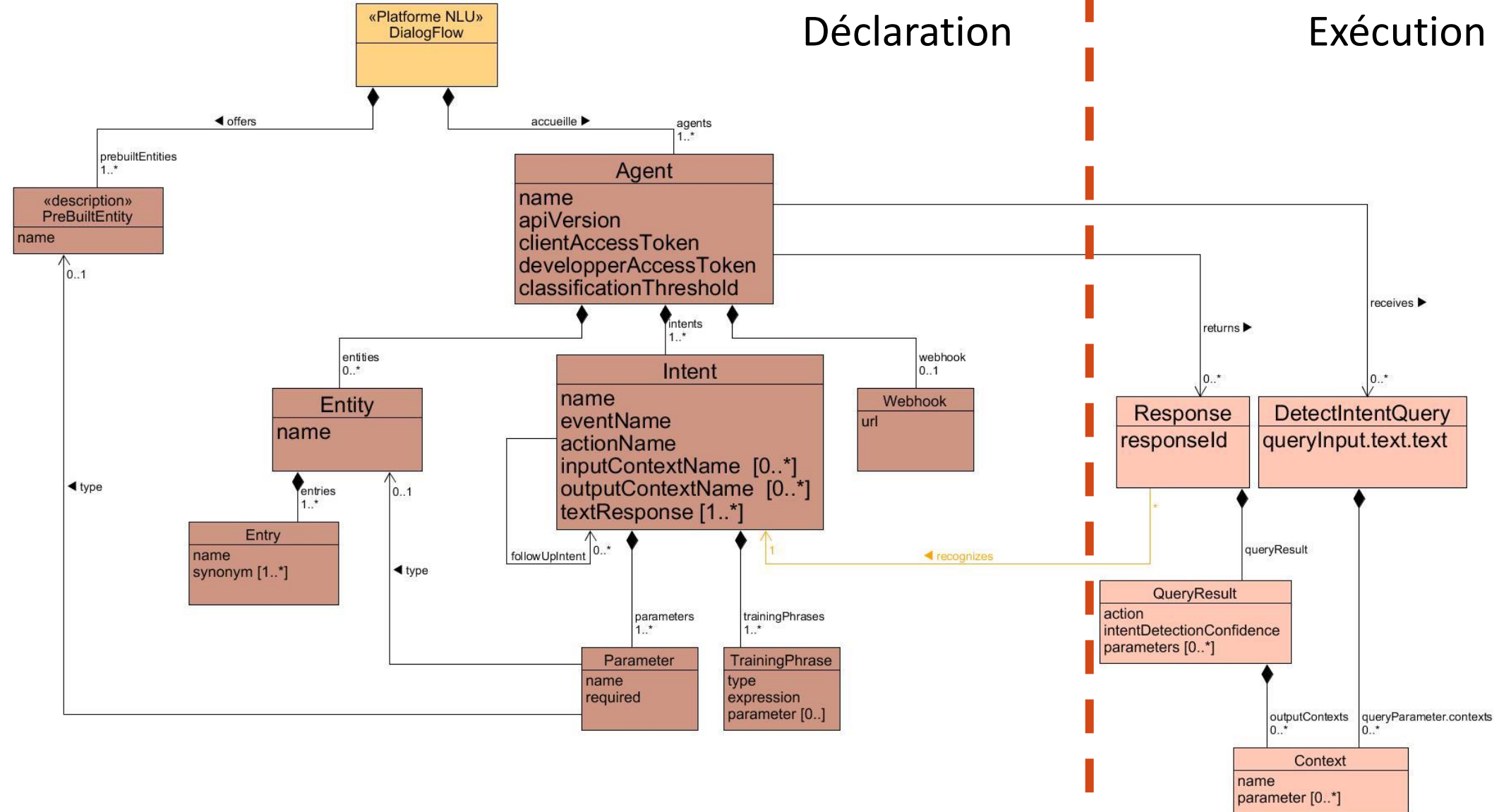


API de Gestion de l'agent
Change le comportement de l'agent

Vocabulaire de DialogFlow

Déclaration

Exécution



Agent

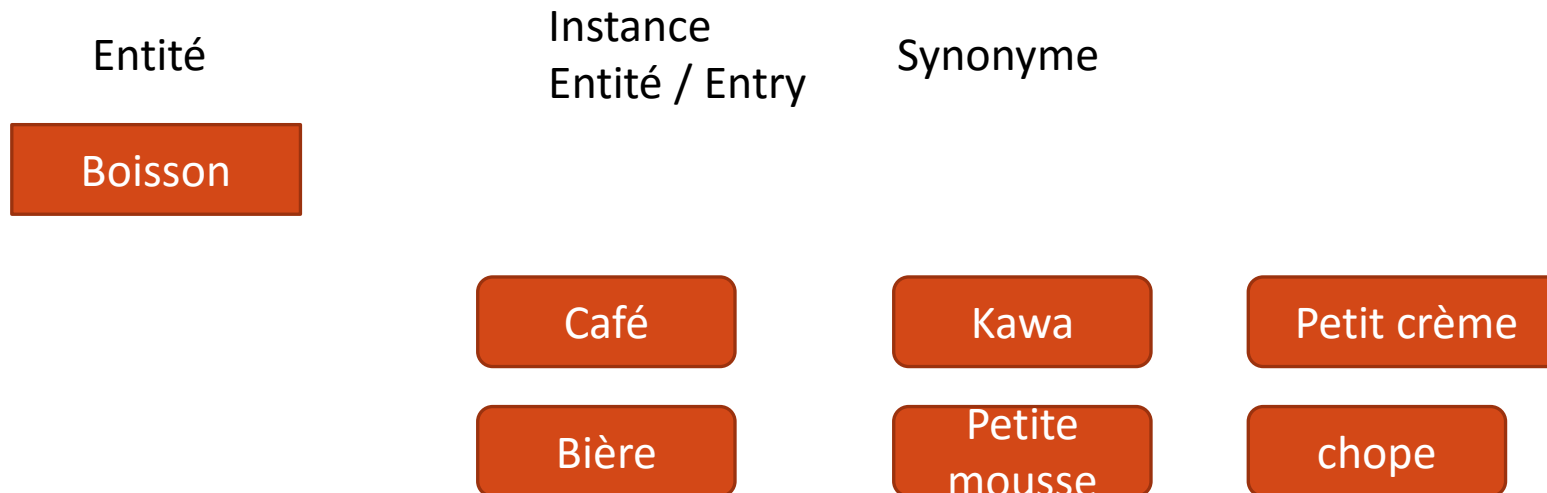
- ❖ C'est un composant ayant pour fonction la compréhension du langage naturel – il reçoit une phrase en Langage Naturel et renvoie une structure de données comportant
 - L'intention de l'interlocuteur (je commande un café)
 - Les paramètres éventuels sur lesquels l'intention porte (type du café long, court, serré)
- ❖ Il peut aussi gérer des fils de conversation en utilisant des concepts comme les contextes, les enchaînements d'intention, les remplissages de paramètres, etc.

Intention

- ❖ Une intention définit ce que l'agent doit comprendre de phrases en langage naturel fournies par l'utilisateur. C'est une correspondance entre des phrases et une action souhaitée par l'utilisateur avec les paramètres (entités) associées.
- ❖ Il y a 4 grandes parties à la définition d'une intention:
 - Les phrases d'entraînement qui sont des exemples ou des templates représentant ce que l'utilisateur pourra dire à l'agent – plus il y en a plus l'agent pourra s'adapter aux locutions de chaque interlocuteur – c'est la partie centrale de l'entraînement de l'agent
 - L'action qui est un libellé nommant l'action à effectuer – elle transmise par l'agent au webhook
 - Les réponses qui sont des phrases produites par l'agent en réponse à l'intention
 - Les contextes qui vont permettre de gérer des conversations composées de plusieurs successions d'intention et d'avoir une mémoire de la conversation.

Entité

- ❖ Les entités constituent les objets du vocabulaire du discours du bot – ils servent à typer les paramètres d'une intention.
- ❖ Chaque entité est définie par l'ensemble de ses instances (entry) et du nom donné à celles-ci.
- ❖ Pour chaque nom d'instance, on donne le maximum de synonymes susceptibles d'être employés par les utilisateurs





Interface principale DialogFlow – Lab1



mise à jour
agent

Console

Menu
Principal

Zone de
Travail

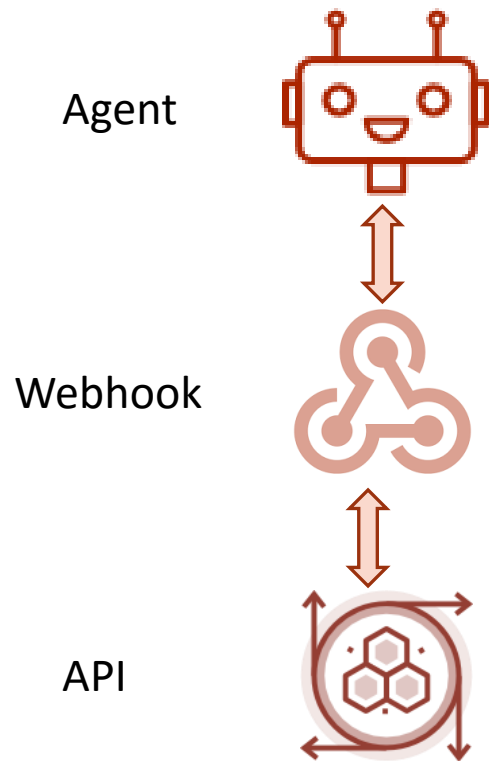
The screenshot shows the DialogFlow console interface for a project named "PolyTech-cours". The interface is divided into several sections:

- Left Sidebar (Menu Principal):** Contains navigation links for PolyTech-cours, Intents, Entities, Fulfillment, Integrations, Training, History, Analytics, Prebuilt Agents, Small Talk, Docs, Forum, Support, Account, and Logout.
- Top Bar:** Includes the Dialogflow logo, the project name "PolyTech-cours", a "SAVE" button, and a "Try it now" button.
- General Settings:** Includes tabs for General, Languages, ML Settings, Export and Import, and Share. The General tab is active, showing the project description, default time zone (GMT+2:00 Europe/Kaliningrad), and Google Project information (Project ID: polytech-cours-74f60, Service Account: dialogflow-mmjybs@polytech-cours-74f60.iam.gserviceaccount.com).
- API Version:** Shows the V2 API selected as the default for the agent.
- Beta Features:** Includes a toggle for "Enable beta features and APIs".
- API Keys (V1):** Displays client and developer access tokens.
- Log Settings:** Includes toggles for "Log interactions to Dialogflow" and "Log interactions to Google Cloud".
- Right Panel (Console):** Contains a "Try it now" button and a message: "Please use test console above to try a sentence." with a link to "See how it works in Google Assistant."

Créer une intention - Lab 2

- Ajouter l'entité Pays avec les valeurs France, Pologne, Belgique
- Ajouter une intention pour demander des informations sur un pays
 - Que sais tu sur la Pologne ?
 - Que sais-tu sur la France ?
- Ajouter les réponses
 - Je n'ai pas encore de connaissances sur ce pays
 - Je ne connais pas le pays nommé [pays]
- Tester

Accès aux services - Webhook

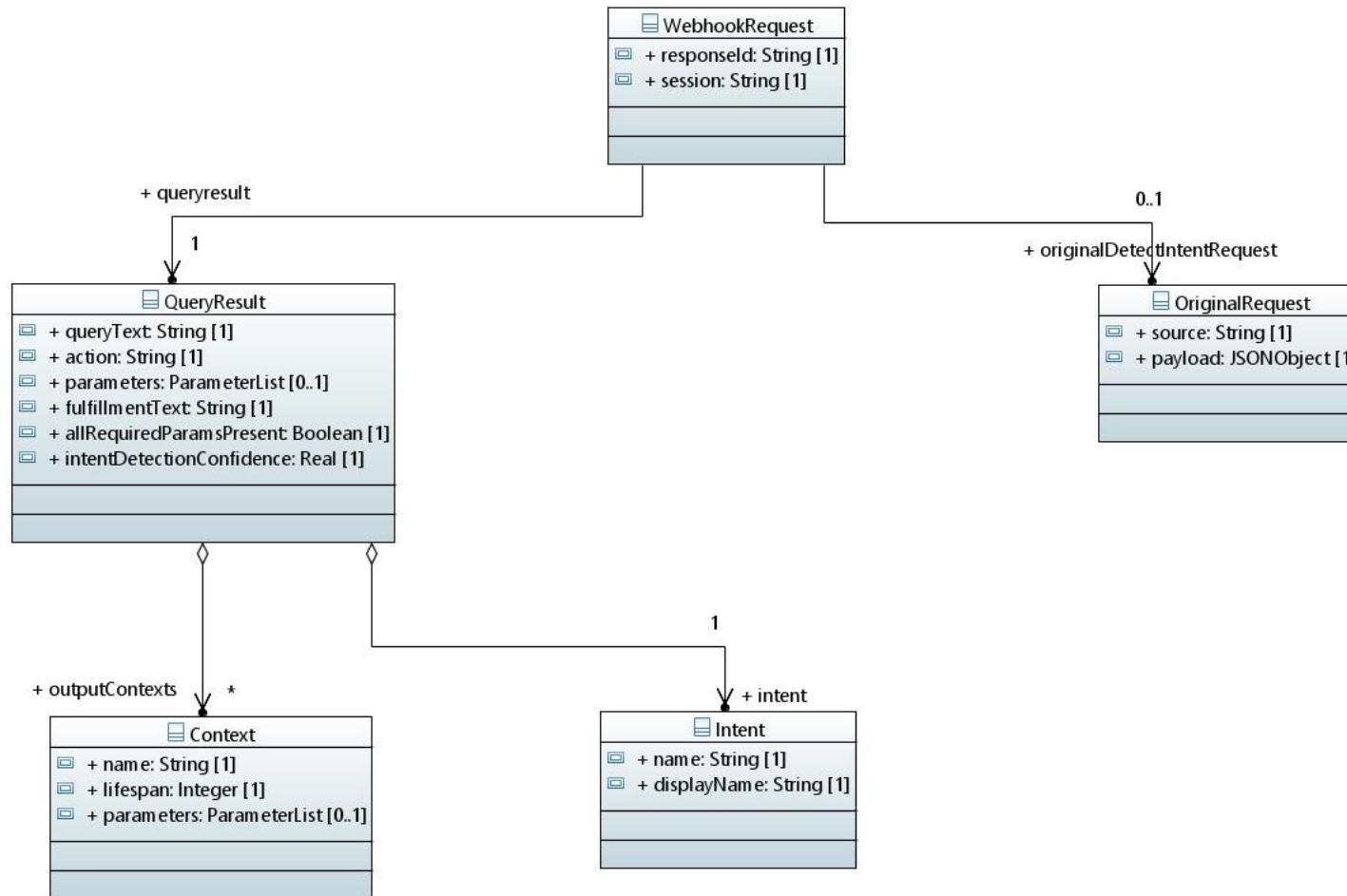


- Deux rôles pour le webhook :
 - Interface entre la partie NLU de l'agent et les services du back-end dont l'agent offre les fonctionnalités
 - Participer à la gestion d'une conversation complexe

Webhook

Structure de la requête d'appel au Webhook

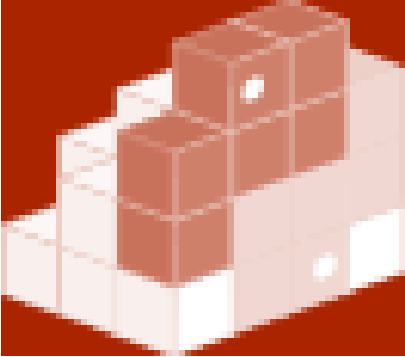
Format JSON



Webhook

- ❖ Un webhook est un service REST qui est appelé par l'agent sur les intentions déclarées comme l'utilisant.
 - Il doit interpréter en entrée le JSON correspondant à la réponse retournée par l'appel à l'opération */detectIntent* de dialogflow
 - Il doit retourner une réponse respectant le standard DialogFlow pour être utilisé par l'agent

```
{ « fulfillmentText » : « réponse à la requête »,  
  « fulfillmentMessages » : [{Message Object}],  
  « payload » : { Json data objet propre à une réponse interprétée par la plate-forme  
de messagerie visée},  
  « OutputContext »: [{« name »: « ctx1 », « lifespan »:2,  
                        « parameters »: {« p1 »: « valeur p1 »}}],  
  « followupEventInput » : {« name » : « eventName à déclencher »,  
                            « parameters »: {« param1 »: « valeur, etc.}}
```



Python

Webhook Python

Nécessite des librairies complémentaires

Flask : Infrastructure d'application web minimaliste – permet de créer un serveur en quelques lignes

JSON : transformer des dictionnaire en chaine Json

Requests : fournit un client HTTP simple pour effectuer les requêtes au back-end

PYOWM : fournit un client python du service Open Weather Map

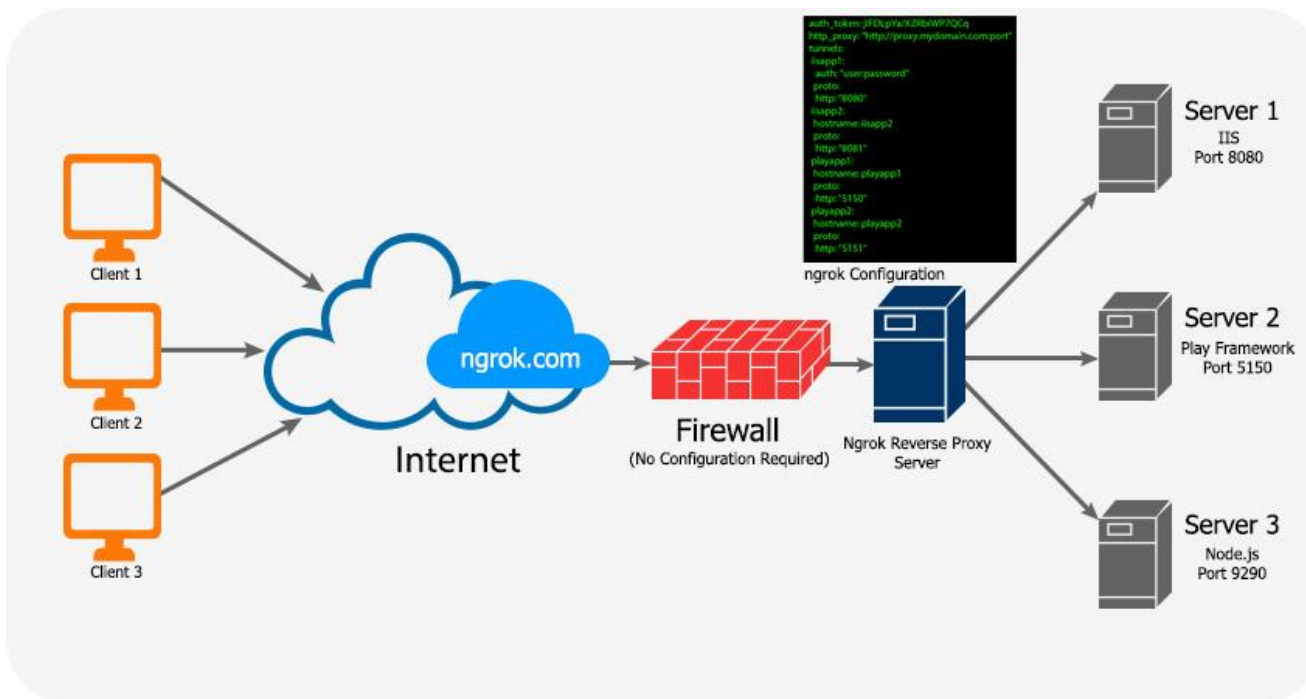
Initier le projet Python

- `pip install Flask`
- `pip install requests`

Squelette de webhook

```
# Librairie Dependences
from flask import Flask, request
import json
# intialisation de l'application
app = Flask(__name__)
# définition de la fonction de réponse sur l'URL <site>/webhook
@app.route('/webhook', methods=['POST'])
def webhook():
    if not request.json:
        print("requete sans json - erreur")
    req = request.get_json(force=True)
    # récupération du nom de l'action
    action = req.get('queryResult').get('action')
    if action == "testWebhook":
        # construction de la réponse
        reponse={
            "fulfillmentText": u"Le webhook est opérationnel"}
    return json.dumps(reponse)
```

Publier le serveur sur Internet NGROK



Note: All servers are in Internal Network running behind http proxy and not in DMZ

- ngrok http 9900
- lire l'adresse internet d'exposition du serveur localisé en 9900



Fournir l'URL reprise de
ngrok – mettre https

Dialogflow

PolyTech-cours

fr

Intents

Entities

Fulfillment

Integrations

Training

History

Analytics

Fulfillment

Webhook

Your web service will receive a POST request from Dialogflow in the form of the response to a user query match version enabled in this agent.

[Webhook example](#)

URL*

BASIC AUTH En

HEADERS En

En

[+ Add header](#)

DOMAINS ☐ Disable webhook for all domains

TestWebhook

Contexts ?

Events ?

Training phrases ?

Action and parameters ?

Responses ?

Fulfillment ?

☒ Enable webhook call for this intent

☐ Enable webhook call for slot filling

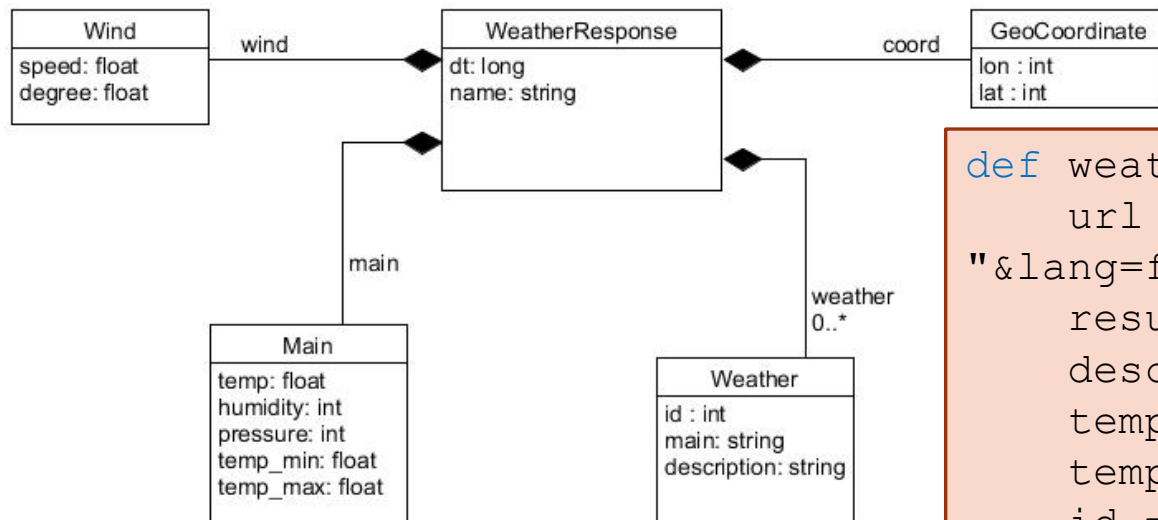
Indiquer l'usage du
webhook pour cette
intention

Tester le webhook



Lire la météo d'une ville – Lab 3

Se créer un compte openweathermap et noter son token



```
def weather(ville):  
    url = owmurl + "weather/?q=" + ville +  
    "&lang=fr&APPID=" + owmtoken  
    resultat = requests.get(url).json()  
    description = resultat['weather'][0]["description"]  
    temp_min = resultat['main']['temp_min'] - 273.15  
    temp_max = resultat['main']['temp_max'] - 273.15  
    id = resultat['weather'][0]['id']  
    ret= {"description": description,  
    "temp_min":temp_min, "temp_max":temp_max, "id":id}  
    return ret
```



Canal d'échanges : SLACK

Qu'est-ce qu'une messagerie instantanée ?

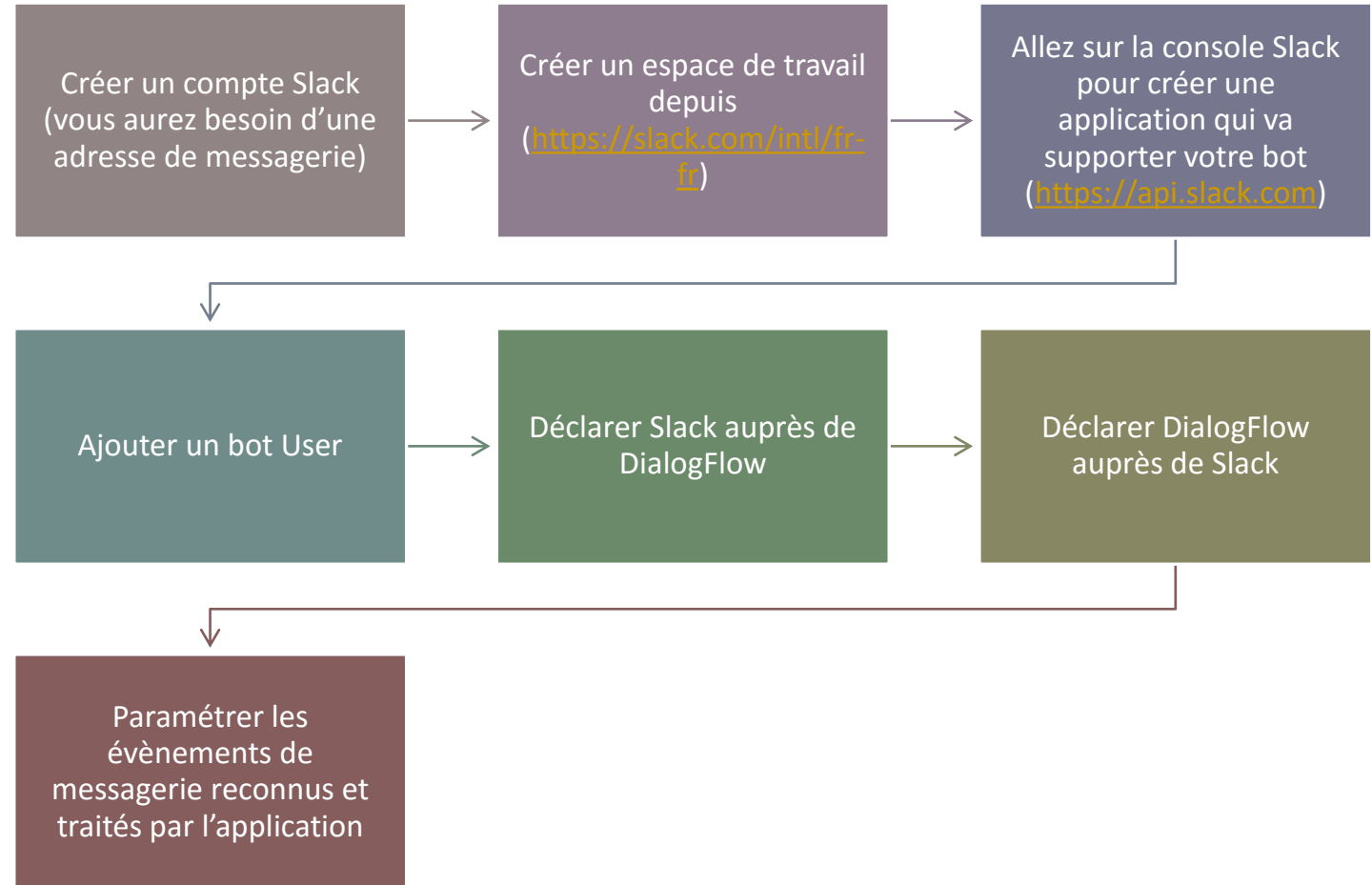
- Email -> paradigme du courrier
- Messagerie -> paradigme du téléphone
 - Permet d'échanger entre individu de manière instantanée à la différence du mail qui est plutôt asynchrone et de désengorger les boîtes mail
 - Échanges de manière textuelle avec suivi des échanges
 - Possibilité d'échanger des fichiers
 - Partage d'écran
 - Peut s'étendre vers le téléphone et la vidéo assez naturellement
 - S'étends de plus en plus comme un espace de collaboration qui intègre des outils -> Slack, Teams, Hang out, etc.

Les fonctionnalités de Slack

- Plate-forme de communication collaborative sur ordinateur et smartphone
 - Messagerie organisée
 - Par chaîne (sur une thématique)
 - Par groupe privé (seuls les membres peuvent y participer et la consulter)
- Par fil de discussion direct avec un autre membre
- Espace de partage de document de tout type (texte, vidéos, pdf, etc.)
- Toutes les conversations sont archivées et recherchable
- Intégration avec de nombreux outils (Google drive, GitHub, Jira, Skype, etc.)
- Construire des Application Slack et ... des bots




Déployer votre bot sous Slack – Lab 4



Ajouter une
application Slack
liée à votre
espace de travail

Create a Slack App



Interested in the next generation of apps?
We're improving app development and distribution. Join the API Preview period for workspace tokens and the Permissions API.


App Name


PolytechApp


Don't worry; you'll be able to change this later.


Development Slack Workspace

Development Slack Workspace

 CoursPolytech

 CRA-BOT-TEST

 CoursDialogFlow

 Sign in to another workspace

Basic Information

Collaborators

Install App

Manage Distribution

Features

Incoming Webhooks

Interactive Components

Slash Commands

OAuth & Permissions

Event Subscriptions

Bot Users

User ID Mapping

Building Apps for Slack

Create an app that's just for your workspace (or build one that can be used by any workspace) by following the steps below.

Add features and functionality

Choose and configure the tools you'll need to create your app (or review all [our documentation](#)).

Incoming Webhooks

Post messages from external sources into Slack.

Interactive Components

Add buttons to your app's messages, and create an interactive experience for users.

Slash Commands

Allow users to perform app actions by typing commands in Slack.

Event Subscriptions

Make it easy for your app to respond to activity in Slack.

Bots

Add a bot to allow users to exchange messages with your app.

Permissions

Configure permissions to allow your app to interact with the Slack API.

Discard Changes

Save Changes

Ajouter un
bot User

Gérer votre application Slack

Déclarer
DialogFlow

Récupérer les
identifiants
Slack



Launch

To make your bot public, take the following steps:

- Create a new [Slack app](#).
- In the Slack app, go to the 'Bot users' section and click the 'Add a Bot User' button.
- Enable events in the 'Event Subscriptions' section.
- Select necessary events in 'Subscribe to Bot Events' section for example message.im, message.group, message.channel, im_created. These events will define which message types (direct, in channel, etc.) your bot will react to. Save changes.
- In the Slack app 'Basic Information' section, copy the 'Client ID' and 'Client Secret' and paste their values into the respective fields below.
- Start the bot.
- Copy the 'OAuth URL' value below and paste it into the 'Redirect URL(s)' field from the 'OAuth & Permissions' section of your Slack app settings. Save URLs.
- Copy Events Request URL from the field below and paste it to Request URL in the 'Events Subscription' section of your Slack app.
- In the Slack app settings, go to 'Manage Distribution' and add the bot to your Slack team using the 'Add to Slack' button.
- To use [rich messages](#), copy the 'Events Request URL' value below and paste it into the 'Request URL' field from the Interactive Messages section of your Slack app settings.

More in the [documentation](#).

Slack Client ID *

391339120343.390433140021

Slack Client Secret *

72a2b1d13aaf72211df21b25aaf59d9a

Slack Verification Token *

IfyHvIMPqSEaqA341pc0IVBu

Success page (Optional)

[CLOSE](#)

[STOP](#)

Marche à suivre

Id Slack

Identifiants DialogFlow

Intégrer Slack dans DialogFlow

Connexion Slack - DialogFlow

SLACK

- ☐ Créer une application Slack et lui ajouter un bot user (mettez le toujours en ligne)
- ☐ Récupérer les App credentials
 - ☐ Client ID, client secret, verification token
- ☐ Ajouter la Redirect URL dans Oauth & Permissions en prenant Oauth URL
- ☐ Ajouter l'Event Request URL en la reprenant de Dialog Flow
- ☐ Souscrire aux évènements du canal (message.im, message.groups.message.channels, im_created)
- ☐ Rajouter l'application à votre équipe avec laquelle le bot va converser (Manage Distribution)

DIALOGFLOW

- ☐ Aller dans l'onglet Integration de l'agent DialogFlow
- ☐ Sélectionner Slack
 - ☐ Recopier les client ID, client Secret et Verification token de l'application slack
- ☐ Faites Start pour démarrer le bot
- ☐ Copier les deux adresses DialogFlow qui sont fournies nécessaires pour Slack Oauth URL et Event Request URL

CoursPolytech

hve

Fils de discussion

Chaînes

aléatoire

général

Messages directs

slackbot

hve (vous)


+ Inviter des personnes

Applications

PolytechBot

☆ PolytechBot

Messages À propos de




PolytechBot

Ceci est le tout début de votre historique des messages directs avec @PolytechBot.

[Comment fonctionne PolytechBot ?](#)


Aujourd'hui



hve

11 h 30


bonjour



PolytechApp

11 h 30


Bonjour !



hve

11 h 30

quel temps fait-il à nantes



PolytechApp

11 h 30

Aujourd'hui, à NANTES le temps est de type :ciel dégagé

+

Envoyer un message à @PolytechBot

Tester

HUGUES VAN-EYLEN

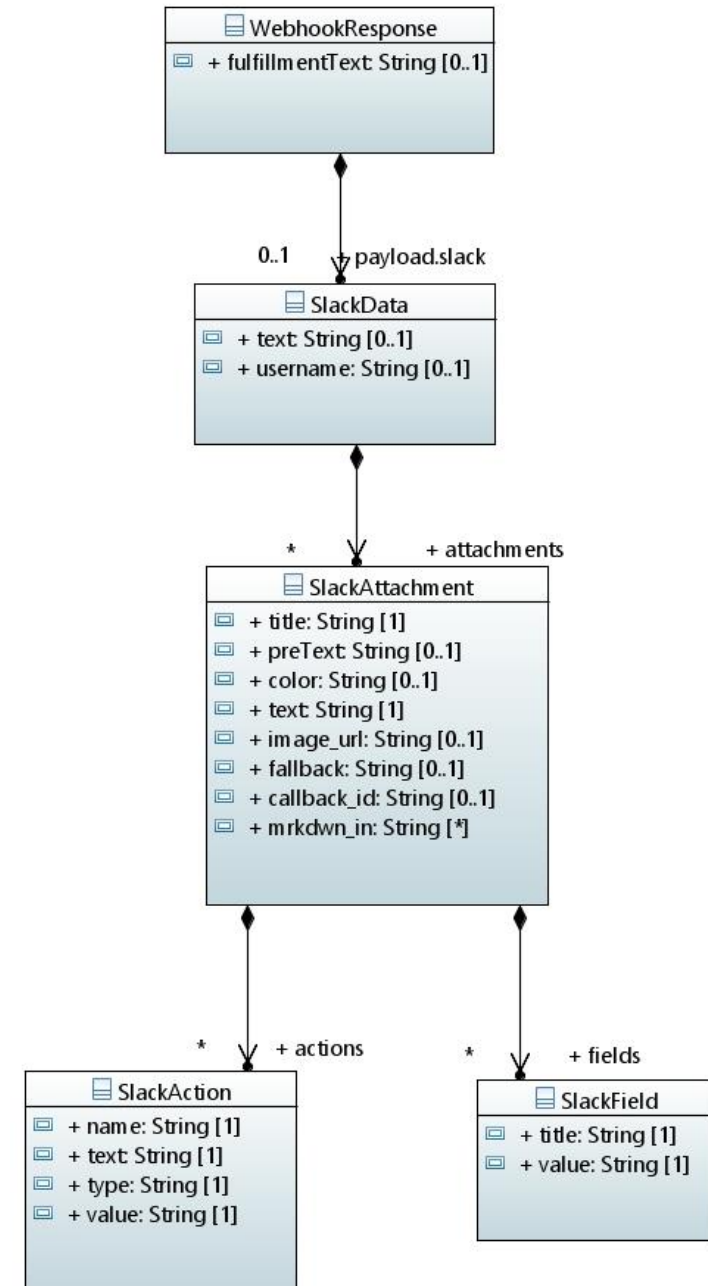
11/17/2020

58

Structure réponse du webhook pour Slack

slack.text est le champs de réponse simple

slack.attachment permet de construire une réponse plus complexe avec champs et bouton.

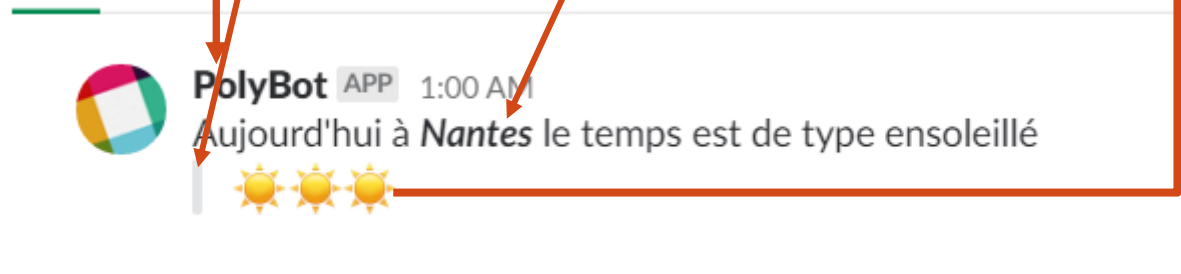


Formatage de base

CODE JSON

```
{  
  "text": "Aujourd'hui à *_Nantes_* le temps est de type  
ensoleillé\n> :sunny::sunny::sunny:",  
  "username": « PolyBot »  
}
```

AFFICHAGE



Formatage avancé

```
{ "text": "Démonstration des *possibilités* de *Slack*",
  "username": "DemoBot",
  "attachments": [
    { "title": "Attachement simple",
      "color": "ff0000",
      "pretext": "_Titre de l'attachement_",
      "text": "le couleur de la barre est rouge",
      "mrkdwn_in": ["text", "pretext"]
    },
    { "text": "Attachement avec champs",
      "color": "3333ff",
      "fields": [
        { "title": "champs 1",
          "value": "valeur du champs 1",
          "short": true
        },
        { "title": "champs 2",
          "value": "valeur du champs 2",
          "short": true
        }
      ]
    }
  ]
}
```



DemoBot APP 1:00 AM

Démonstration des **possibilités** de **Slack**

Titre de l'attachement

Attachement simple

le couleur de la barre est rouge

Attachement avec champs

champs 1

valeur du champs 1

champs 2

valeur du champs 2



Formatter votre réponse Slack – Lab 5

- Construire une réponse Slack qui
 - Mette en gras le nom de la ville
 - Utilise deux champs pour indiquer les températures minimum et maximum
 - Affiche un émoji en rapport avec le type de temps
 - <https://www.webpagefx.com/tools/emoji-cheat-sheet/>
 - Catalogue complet : <https://unicodey.com/emoji-data/table.htm>
 - sunny, umbrella, cloud, partly_sunny
- Consulter la page : <https://openweathermap.org/weather-conditions> pour déterminer le type de temps (json.weather[0].id).

Capacités avancées

- Les entités systèmes
- Le slot-filling
- Les contextes et la gestion d'une mémoire de conversation
- Entité système sys.any

Entités système et Slot-filling



EN	DA	DE	ES	FR	HI	ID	IT	JA	KO	NL	MORE ▾
Entity Name	Description	Examples	Returned Data Type	Returned Object Example							
@sys.date-time	Matches date, time, intervals or date and time together	14h30 13 juillet avril Matin demain à 16h30 demain dans l'après-midi	String in ISO-8601 format or object: Strings in ISO-8601 format	"2018-04-05T14:30:00-06:00" "2018-07-13T18:00:00-06:00" { "startDate": "2018-04-01T12:00:00-06:00", "endDate": "2018-04-30T12:00:00-06:00" } { "startTime": "2018-04-06T08:00:00-06:00", "endTime": "2018-04-06T12:00:00-06:00" } { "date_time": "2018-04-06T16:30:00-06:00" } { "startDateTime": "2018-04-06T12:00:00-06:00", "endDateTime": "2018-04-06T16:00:00-06:00" }							
@sys.date	Matches a date	demain	String in ISO-8601 format	"2018-04-06T12:00:00-06:00"							
@sys.date-period	Matches a date interval	avril	Object: Strings in ISO-8601 format	{ "endDate": "2018-04-30T12:00:00-06:00", "startDate": "2018-04-01T12:00:00-06:00" }							
@sys.time	Matches a time	16h30	String in ISO-8601 format	"2018-04-05T16:30:00-06:00"							
@sys.time-period	Matches a time interval	dans l'après-midi	Object: Strings in ISO-8601 format	{ "startTime": "2018-04-05T12:00:00-06:00", "endTime": "2018-04-05T16:00:00-06:00" }							

Entité système Temps

EN	DA	DE	ES	FR	HI	ID	IT	JA	KO	NL	MORE ▾
Entity Name	Description					Examples		Returned Data Type	Returned Object Example		
@sys.address	France, Canada (French part) full address					8 Rue de Londres-15-15 Bis Rue de Clichy 75009 Paris France		String in French	"8 Rue de Londres-15-15 Bis Rue de Clichy 75009 Paris France"		
@sys.street-address	Part of address without city, state, country or zip code.					Rue de Londres-15-15 Bis 8 Rue de Londres-15-15 Bis 8 Rue de Londres-15-15 Bis app. 22B		String in French	"Rue de Londres-15-15 Bis" "8 Rue de Londres-15-15 Bis" "8 Rue de Londres-15-15 Bis app. 22B"		
@sys.zip-code	France, Switzerland, Canada postal codes					75009 8002 M5H 2G4		String	"75009" "8002" "M5H 2G4"		
@sys.geo-country	Short and full names of country					États-Unis Royaume-Uni		String in French	"États-Unis" "Royaume-Uni"		
@sys.geo-city	Major cities					New York Paris		String in French	"New York" "Paris"		
@sys.geo-city-fr	France cities					Échirolles Biarritz		String in French	"Échirolles" "Biarritz"		

Entité système Géographie



Donner la météo de toutes les villes – Lab 6

- Ajouter une intention « obtenirMeteoDesVilles »
- Saisir des phrases d'entraînements
 - Quel temps fait-il à Moscou ?
 - Quelle est la météo à Pékin ?
- Observer le comportement de DialogFlow
- Que faut-il faire pour terminer cette intention et tester la météo à Moscou ?
- Saisissez la phrase de test suivante : « quel est la météo ? » et observer ce qui se passe ?
- Cliquer « required » sur le paramètre Ville et saisissez la réponse en cas de paramètre absent
- Tester
- essayez la phrase : « quel temps fait-il à Pékin ? » - observer et expliquer - corriger

Contexte et mémoire

- Objectif d'un contexte
 - transmettre des informations d'un échanges sur l'autre
 - Informations rangées en général dans un dictionnaire
- Implémentation dans DialogFlow
 - Contextes d'entrées d'une intention : celle-ci ne peut être déclenchée que si un de ces contextes est présent
 - Contextes de sortie d'une intention : celles-ci sont produites si cette intention est déclenchée et les paramètres de l'intention sont ajoutés au contexte
- Exemple : tester la requête précédente de slot-filling dans la console DialogFlow et regarder les contextes utilisés pour réaliser ce mécanisme.



Lab 7 – Se rappeler un prénom

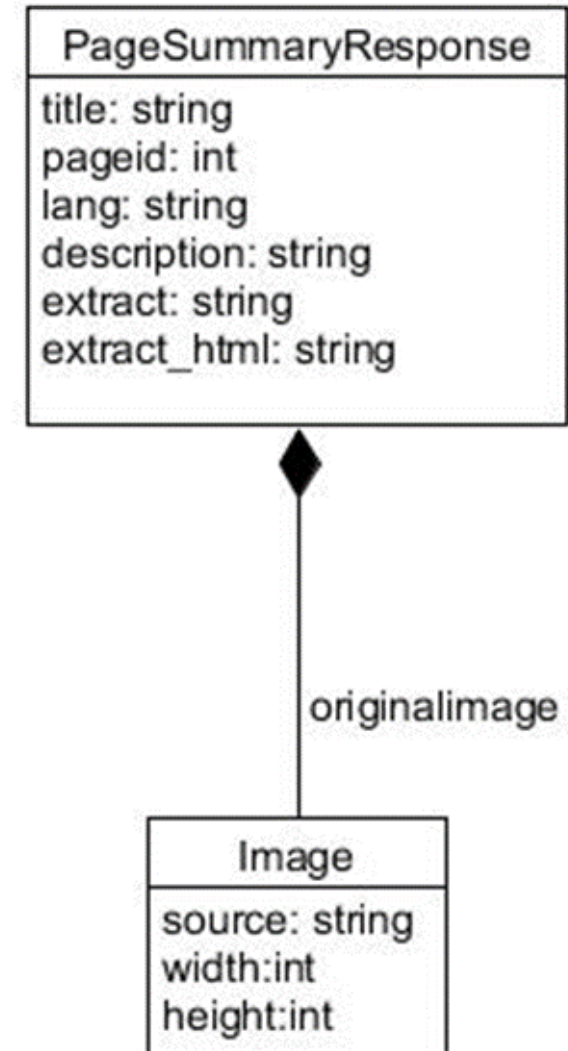
- Objectif:
 - Si on se présente, alors le bot pourra utiliser le prénom donné dans la réponse météo : « Hugues, aujourd’hui à Nantes, le temps est de type : ciel dégagé »
- Procédure :
 - Ajouter une intention fournirPrenom
 - Utiliser le type sys.any et le mode template pour rédiger les phrases d’entraînement : « Bonjour, je m’appelle @sys.any:prenom »
 - Ajouter un contexte de sortie nommé « mémoire »
 - Modifier le webhook pour que le traitement de la météo recherche le prénom.

Lab 8 (optionnel)

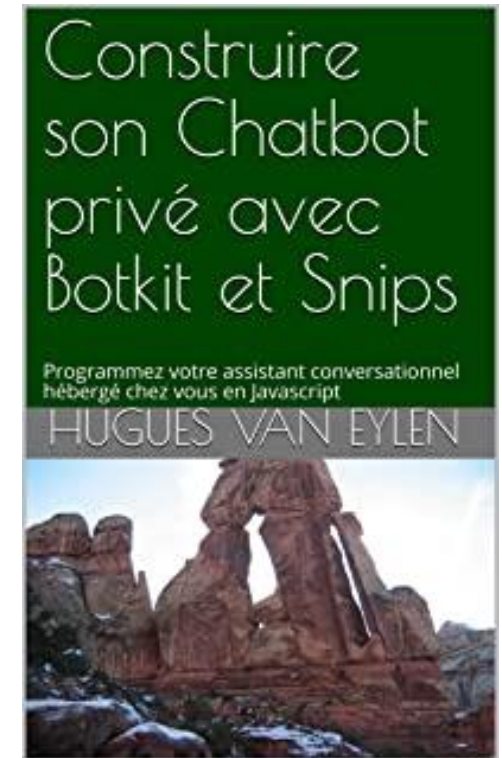
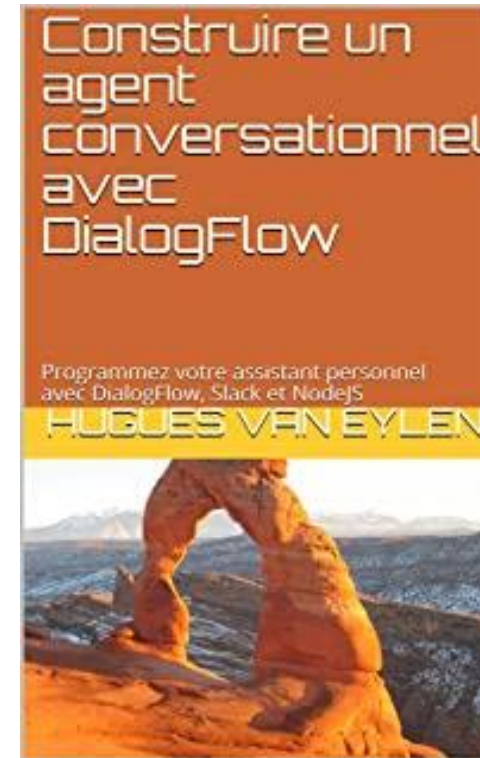
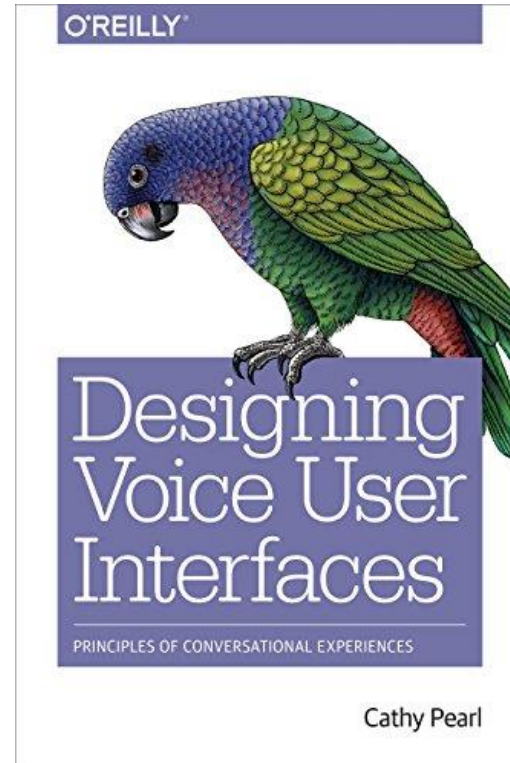
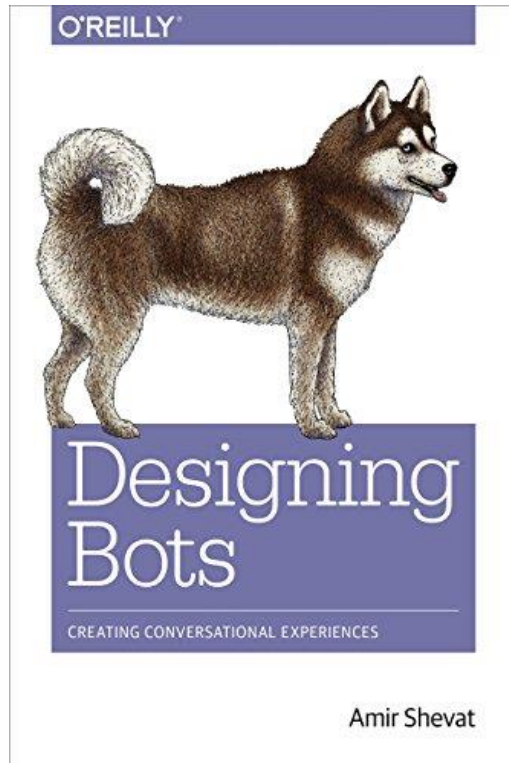
Ajouter la possibilité d'interroger wikipedia pour demander des informations sur une personnalité : « qui est Usain Bolt ? »

Requête

https://fr.wikipedia.org/api/rest_v1/page/summary/<sujet>



Livres



Livres

