

# Developing a virtual assistant for answering music related questions

Claudio SCALZO      Luca LOMBARDO

v1.0.0

# Contents

<b>1</b>	<b>Introduction</b>	<b>2</b>
1.1	Why a virtual assistant? . . . . .	2
1.2	Scope of the project . . . . .	2
1.3	Expected results . . . . .	2
<b>2</b>	<b>Natural Language Understanding</b>	<b>3</b>
2.1	What is NLU? . . . . .	3
2.2	How it works? . . . . .	3
2.3	Advanced techniques (Papers) . . . . .	3
2.4	State of art . . . . .	3
<b>3</b>	<b>Dialogflow</b>	<b>4</b>
3.1	What is Dialogflow . . . . .	4
3.2	Why Dialogflow . . . . .	4
3.3	How it works? . . . . .	4
<b>4</b>	<b>DOREMUS</b>	<b>5</b>
4.1	What is DOREMUS . . . . .	5
4.2	How is organized . . . . .	5
4.3	How to integrate it in our project . . . . .	5
<b>5</b>	<b>The bot</b>	<b>6</b>
5.1	The architecture . . . . .	6
5.2	Intents . . . . .	7
5.3	The flow . . . . .	7
5.4	The spell checker . . . . .	7

# Chapter 1

## Introduction

1.1 Why a virtual assistant?

1.2 Scope of the project

1.3 Expected results

## Chapter 2

# Natural Language Understanding

2.1 What is NLU?

2.2 How it works?

2.3 Advanced techniques (Papers)

2.4 State of art

# Chapter 3

## Dialogflow

3.1 What is Dialogflow

3.2 Why Dialogflow

3.3 How it works?

# Chapter 4

## DOREMUS

4.1 What is DOREMUS

4.2 How is organized

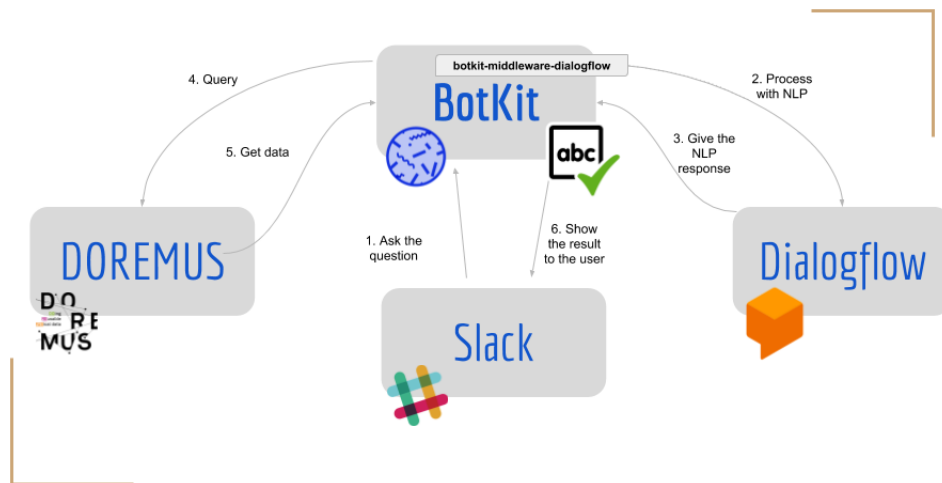
4.3 How to integrate it in our project

# Chapter 5

## The bot

### 5.1 The architecture

The architecture of the bot is divided in four categories:



First of all we can find the client, that in the case of our development, testing and validation process, has been *Slack*. Of course, it can be any of the clients which support the installation of the bot on it (*Telegram*, *Facebook Messenger*, etc.). The use of *Slack* let us to exploit the beautiful *Slack Cards*, to make the answers of our bot (works, artists, performances) prettier and easier to understand in a glance. A set of examples will be provided in the last chapter.

The second part of the architecture is represented by the NLP. In this case, as told in the previous chapter, we used *Dialogflow*, to exploit its advanced slot-filling techniques and its NLU power.

However, we didn't use *Dialogflow* on its own, exploiting the direct integration with *Slack*, but we used something that we placed in the middle of the two: *BotKit*. *BotKit* is a bot-making toolkit that aims to ease the building process of a bot, potentially exploiting different NLPs and/or different clients. In our case, *BotKit* has been deployed on a web server, and thanks to the *NodeJS* code we were able to come up with a series of features (like the spell checker) that would have been impossible to reach with a simple (direct) integration between *Slack* and *Dialogflow*. We'll talk more about that in the following paragraphs.

The last part of the architecture is of course represented by the data source: *DOREMUS*. We talked about that in the previous chapters, but from the architecture is important to notice how the knowledge base is queried: each query is dynamic, in the sense that according to the intent, the number of filters and the desired results wanted by the user, the query will have a different shape and a different content. The code in the web server is able to add different pieces of queries according to what *Dialogflow* is able to understand and to provide as output values of the API.

## 5.2 Intents

## 5.3 The flow

## 5.4 The spell checker