

# Wisdom Of the Krowd – Architecture and engineering of a crowdsourcing applications

Miguel Henrique Germano Garrido

Instituto Superior Técnico, Lisboa, Portugal

October 2019

## Abstract

This Project describes the development of *Wisdom Of the Krowd*, an application that supplies its users with tasks to answer. The responses are processed and the best response is returned. This application is made possible through the IST-Scope project, aiming to incorporate interdisciplinary collaboration across different engineer specialties to solve problems set by companies. It is complemented by the Capstone module, a new project approach that serves as a baseline for the creation of a thesis, culminating in academic and individual experience for the students. The Project goes into detail on the application's features, focusing on the Back-End.

The application consists of three services: The Facebook Handler, able to communicate with the Facebook users and send those tasks; The Client-Side Handler, able to supply the clients with a platform for managing its tasks; The Main Server, able to manage those tasks and responses. The application is hosted on Heroku and uses mongoDB databases to store each services data. It is programed in Python and uses a Flask-RESTful API and Requests module to communicate with the other the service.

**Key Words:** Co-Worker, service, user, client, communication, WOK

## Chapter 1 – Introduction

This project is a SCOPE project by *Instituto Superior Tecnico*, IST, aiming to incorporate interdisciplinary collaboration across different engineer specialties to solve problems set by companies. It is based on the CAPSTONE Model, a new project approach that serves as a baseline for the creation of a thesis, culminating in academic and individual experience for the students.

The project was made in collaboration with *Unbabel*, a Portuguese startup company pioneering in "Translation as a Service", combining neural Machine Translation, advance Quality Estimation

and a crowd of Editors to deliver professional quality translation at a massive scale.

The project is being developed by a group of four students: two, including myself, in software development, one in charge of the design & Branding and one managing the project.

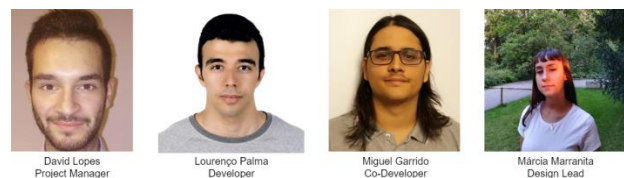


Figure 1: SCOPE Team and designated roles

### 1.1 – Motivation

A problem with outsourcing is the inability to guarantee that the price to pay is consistent with the value of the tasks requested. Some simple checks, like small translations issues or small problems, can be solved without the need of hiring an expert. Typically an expert is hired to solve many tasks for a price, but if those simple checks become most of the tasks they solve then the company could stand to lose funds that would be better applied elsewhere.

With this in mind we were told by Unbabel to develop a task solving application able to resolve complex problems through the usage of Wisdom of the Crowd. From this application we could assess the advantages and disadvantages of crowdsourcing those small problems compared to hiring an expert.

### 1.2 – Crowdsourcing

Crowdsourcing is a problem solving technique. It consists on taking a task usually performed by an employee or contractor and outsourcing it to a large group of people. It was coined by Jeff Howe and Mark Robinson to describe how businesses were using the Internet for outsourcing.

Crowdsourcing is not a simple procedure, having a multitude of approaches. When using it, one must consider the best way to achieve their objectives. Jeff Howe organizes those approaches into four basic strategies:

Crowdfunding – Tapping into the crowd's financial resources.

Crowdcreation – Using the crowd to create what you want to sell

Crowdvoting – Using the crowd to sift through things and vote

Crowdwisdom – Harnessing the collective intelligence of the crowd to solve complex problems. It is the strategy used in the development of WOK.

The Crowdsourcing Strategy, popularized as "Wisdom of the Crowd" by James Surowiecki, author of "The Wisdom Of Crowds" [1], states that the collective opinion of a group of individuals, usually less experienced, is more accurate than that of a single expert. Jeff Howe in his book "Crowdsourcing: Why the Power of the crowd is driving the future of Business" [2] describes the benefit of an inexperienced individual as "The untrained are also untainted. Their greatest asset is a fresh pair of eyes". It means that due to that inexperience he is able to view a problem or task in a new, possibly better angle than a trained expert, who would have more difficulty changing their normal perspective.

### 1.3 – Chatbot

A Chatbot is a computer program designed to simulate interactive human conversations through a chat interface by using predefined messages. They have gained a great amount of popularity due to the increasing usage of messenger apps like Slack, Facebook Messenger and others. Chatbots are not limited by time or location, always being available to its users. This makes it appealing to many businesses that desire to improve relations with its users and reduce costs.

A Chatbot can be developed with various levels of interactive complexities:

A contextual Chatbot uses machine learning to study the user's behavior and previous interactions to determine the best response. It is one the most complex implementations of a Chatbot;

A Keyword recognition Chatbot analyzes the messages received from the user and, based on specific keywords, determines the best reply;

A Buttons and Menu Chatbot is a glorified decision tree hierarchy where its actions are determined based on the buttons and menus accessed by the user. It is a simple implementation as every button corresponds to a specified action. This Chatbot

implementation is used in the development of our application.

More often than not the users of a Chatbot tend to become tired of the same responses and patterns. That is why when creating a Chatbot it is given a personality that appeals to its target audience, keeping them motivated by being fun and helping them when requested.

#### 1.4 – Microservice

An application using the microservice Architecture Style is formed by a collection of loosely coupled services which can be developed, deployed and maintained independently. Each service has its own responsibilities and can communicate with the other services through simple API's to solve a larger more complex business problem.

## Chapter 2 – WOK Application – General

### 2.1 – WOK Branding

The WOK logo is inspired by the traditional Chinese frying pan wok. Noodles are usually cooked in woks and in our case represent the “entanglement” of the tasks we have to solve. Our job is to “untangle” those tasks, making difficult problems easier to understand and be dealt with.

### 2.2 – Development Environment

By the suggestion of Unbabel, the application's services are programmed in Python 3.6.6 [3]. This language has the benefit of a large quantity of modules, files containing Python definitions and statements that facilitate the applications' incorporation with the hosting and database platforms.

### 2.3 – Hosting

To host the WOK application Heroku [4] was chosen, a Platform as a Service based on a managed container system. It was chosen due to previous experience with the platform, compatibility with the language desired for the creation of the application

and performing Load Balancing for the application's requests through its dynos for scalability.

In Heroku, the deployment of the application is fast and easy to do, using Git [5] to manage App deployments. Heroku also allows for the creation of Environment Variables that are used to manage the application's URLs and other values.

### 2.4 – Database

JSON, JavaScript Object Notation, Document-oriented databases were used to store the application's data due to previous experience with the format and, as the needed data was simple, there was no need for a more complex database . For that mLab [6] was chosen, a “Database-as-a-Service” Platform for MongoDB [7]. It allows for a quick and secure connection to the databases and supports the JSON Format.

### 2.5 – Tasks

Each task sent through the WOK application is assigned to a limited number of users and given a time limit to solve. There are currently two types of tasks supported by the application:

**Categorization task:** A user solving a category tasks must specify the task's category and subcategory. The client supplies all the possible categories used for identification and the message to identify.

**Quality Estimation:** Simple yes or no questions intended in estimating if the task's submitted objective was accomplished. The client supplies the original objective and extra data, one to two messages depending on the objective.

### 2.6 – User's Co-Workers

A new concept of the WOK application is the use of multiple Co-Workers that the user can choose from and bond with. Each Co-Worker has its own personality and interests studied to fit with the type of target audience that would use the application. This was done with the objective of establishing a

friendly environment with our community and to motivate the users to keep resolving tasks.

When developing the Co-Workers' personality a study of the demographic likely to use the WOK application was performed. It consisted on the creation of user examples based on the application's target audience to define the ideal behavior of the personalities, those users being determined to be between the ages of eighteen and thirty five years. Each user was defined with their own ethnicity, profession, hobbies and motives for using the application.

To define the Co-Workers' base personalities the Big Five Theory [8] was used. The Theory classifies peoples' personality by five personality traits: Extraversion, Agreeableness, Conscientiousness, Neuroticism and Openness.

From the theory and the user's data five Co-Workers were created, each focusing on a single personality trait and having behaviors that fit them.

The Co-Workers are represented on Facebook by a name and avatar:

Gal: The default Persona associated to the user when he registers to the application. This Persona is identified as a genderless bot that gives long talks regarding the universe and life. This persona represents the Openness personality, transparent and curious.

Julian: A caring bot that loves to motivate others. This bot represents the Agreeableness personality, always willing to help others.

Eli: A very formal bot, work driven and focused on its goals. This bot represents the Conscientiousness personality, hard-working and reliable.

Jesse: An outgoing bot, very communicative and friendly. This bot represents the Extravert personality, being outgoing and social.

Sam: A shy bot, mostly keeping to himself. This bot represents the Neurotic personality, being very self-conscious.

## 2.7 – Notifications

The notification system helps the WOK users stay up to date on task solving by sending them a message reminding them of incomplete, pending and/or new tasks to solve. It also informs the user if the group of tasks submitted has expired by reaching its solving time limit.

The system sends the messages to the chat using the Co-Workers, each having their own way of notifying the users on any events.

## Chapter 3 – WOK Application – User Interaction

### 3.1 – Facebook

The Facebook platform [9] was chosen owing to having the API's and features that allowed the application to become more interactive, such as the Messenger Sender Action [10] using the Send API [11], able to inform the user if the bot had seen its message and if it's responding; The Persona API [12], also using the Send API, used to create the various Co-Workers; The Messenger Webview [13] used to load web pages inside of the Messenger.

### 3.2 – Facebook Messenger

For a user to access the WOK application he must first go to its Facebook page, Wisdom Of the Krowd [14], and use the embedded messenger.

The interaction is done by supplying the user with all the choices it can make either by messenger Buttons [15] or by the use of the Persistent Menu [16], an easily accessible and simple dropdown menu that provides the top-level bot actions to the users.

### 3.3 – Registering

When first using the messenger, the user will be sent to a web page through the Webview and will be

prompted to fill a Registration Form in order to use the service. It consists on submitting a nickname, and their Phone Number. The user will receive a message from Gal, our default Co-Worker, greeting the user and showing him the possible actions it can take.

### 3.4 – Settings

After registering, the user can click on the Settings button presented by the Co-Worker to open a web page with all the options it can change. These options allow the user to change between the available Co-Workers, modify the user's Nickname and enable or disable the notification messages for the user.

### 3.5 – Request Tasks

When the Request Tasks action is taken by the user, the WOK application will process and submit through a Button and the Webview a group of tasks for him to solve. After the tasks are completed or if the time limit for the completion of the tasks is reached, the Co-Worker will inform the user of the quantity of coins gained through the solved tasks and will prompt him to continue solving tasks by sending him the Buttons with the next possible actions it can take.

## **Chapter 4 – WOK Application – Client Interaction**

### 4.1 – Client Side Handler

The Client-Side Handler creates and manages a Web Page [17] accessible only to the clients, where they can add new tasks and check on the progress of their existing tasks. The site allows the client to filter according to the information they desire to see. They can change between seeing the Category or QE tasks and filter through the time data, description data and response data:

### 4.2 – Exporting Tasks

The client has the option of exporting its tasks, be them pending and/or solved. When doing so he can

choose from exporting all the tasks, which will export two files, one for each task type, each containing all the data from every filter, or export only the filtered content displayed in the web page, exporting a single file containing only the previewed task type and filters. The supported formats are Comma Separated Value, CSV, Tab Separated Value, TSV and JSON.

### 4.3 – Uploading New Tasks

By clicking the “New Tasks” or by going to our Upload Web Page [18], the client can upload new tasks to the service. Those tasks will then be processed, sent to the users and solved. The Categorization Tasks can only be uploaded in JSON Format while the Quality Estimation Tasks support uploading in JSON, TSV and CSV.

## **Chapter 5 – WOK Application – Work Developed**

I was responsible for the back-end of the project, setting up the services and managing the communication between the databases and those services. I also managed the communication between the service and the Facebook Messenger.

### 5.1 – Architecture

The WOK application is composed of three services, working together to form the application:

The Facebook Handler, responsible for the communications with Facebook, supplying the users with their requested tasks, managing their settings and responses and keeping them up to date by the use of the notification system;

The Client-Side Handler, in charge of the communications between the client web page and their clients, supplying them with the options for exporting or importing tasks to the application. Those imported tasks are sent to the Main Server to be used by the application;

The Main Server, responsible for the management of the tasks and user responses, receiving the tasks

from the Client-Side Handler, supplying those tasks to the Facebook user through the Facebook Handler and processing and evaluating all the task's responses in order to send them to the clients through the Client- Side Handler.

Each service has its own API, developed using Flask-RESTful [19], an extension of the Flask [20] module, a lightweight WSGI web application framework for python. It allows to quickly build an API following the best practices and with minimal setup.

To send messages to the services it uses the standard library Requests, for there was no need for a more complex library to deal with the simple requests sent between the services and Facebook.

The communication between the databases and the services uses pyMongo [21], a module containing the tools needed to easily communicate with the MongoDB databases created for the application. It is one of the most recommended and simple libraries used to communicate with the MongoDB API.

## 5.2 – Facebook Messenger

To allow the Facebook Handler to communicate with the Facebook Page messenger a Facebook Bot was created by the name of "WOK Here4U". It listens to the users requests through a Webhook and sends them on to the application.

When a request for a web page is sent from the messenger the application generates a html from templates using Jinja2 [22], a fast and secure tempting language, and sends it to the Facebook Messenger's Webview. To manage the behavior and responses processed through the Webview the Webview Messenger Extension SDK [23] is used.

## 5.3 – Facebook Handler

The Facebook Handler's Backend is responsible for managing the user's interactions with the Co-Workers and Tasks and keeping them notified. It receives user requests from the Facebook

messenger and alters the user's data depending on the interaction. It also sends and receives data from the Main Server either to accommodate the user's changed data or for complementing the rendering of the html task templates, using Jinja2, sent to the Facebook's Webview.

When the user presses a button or uses the persistent menu in Facebook those requests are sent to the root of the Facebook handler's API to be processed and replied.

Facebook Handler contains a small database focused on keeping track of the users and Co-Worker's interaction data. It stores the user's task progression and contains the user id used to communicate with the Main Server. The separate Id's between the handler and the Main Server allows for multiple handlers to share the same user data.

## 5.4 – Main Server

The Main Server is the Core of the system, containing the important data of both the users and clients. It is responsible for keeping track of the user's task solving activities and the client's task submissions.

The Facebook Handler sends task requests to the server. It randomly chooses from the available tasks and sends them to the handler. It associates the job of solving that task to the user that requested it, so that even if the user does not solve the task immediately it can pick it up when possible or be reminded through the notification system.

The Database of the Main Server contains the tasks information and their user responses. It also contains the users' task completion and pending data.

## 5.5 – Client-Side Handler

This handler generates a web page used by the application's clients to send new tasks to the Main Server and receive information regarding its task's completion.

When the Tasks web page is accessed by a client the Client-Side Handler obtains through a request to the Main Server all the tasks that the client submitted. To do so, it identifies and sends through the payload the client's id. If a task is incomplete the content displayed is adjusted to the available data, showing either the partial analysis made by the Main Server or not available ("N/A") due to insufficient data.

## Chapter 6 – Conclusion

The application Wisdom of the Krowd currently allows a client to submit their Quality Estimation tasks and Categorization tasks and view their resolution progress. It also allows for Facebook users to subscribe and request tasks to solve.

While the user testing feedback was mostly positive, new user tests on a larger scale are needed to assess the viability of the application on the market.

The next step in the development of the application is to identify the best payment method and the price value.

A possible future improvement to the application would be to the communication between the user's and the Co-Workers. This could be arranged by adding a trivia system to the Co-Workers where, after a user solves a number of tasks, the Co-Worker sends a message containing some trivia based on their personality and interests. The Co-Workers could also change their messages depending on events or time of day.

An improvement to the communication between the client and the application would be the addition of a priority and difficulty system to the tasks. The clients could define the priority and difficulty of each task or it could be defined by the use of an algorithm using the task's submitted time and/or similar previous tasks as a base. The higher priority and difficulty tasks could reward the users with extra coins.

Other improvements would be to the Security and performance of the application, optimizing task management for multiple handlers and identifying

users using the application through various platforms and multiple clients.

This project was a challenging but enriching experience. It was my first time working in a business environment. It was also the first time in a group project where I collaborated with colleagues from other specialties.

## Acknowledgements

I would like to express my thanks to my supervisor and co-supervisor for the opportunity of experiencing the business environment through this project, and all the support they gave during its development: supervisor Nuno Jardim Nunes from IST and co-supervisor Sónia Rafael from Faculdade de Belas Artes da Universidade de Lisboa;

I would also like to thank Unbabel for supplying us with the necessary tools to complete the project, for their many ideas and tweaks and for their help during various phases of the project, especially the User Testing phases.

I would also like to thank the experts involved during development for all their support, ideas and suggestions: João Ferreira from Faculdade de Belas Artes and Pedro Oliveira, Anthropologist;

To my team for their dedication and perseverance I would like to give big thanks: Lourenço Palma; David Lopes; Márcia Marranita;

For volunteering to our User Testing and giving us very useful suggestions I would like to thank the other IST-Scope Team: Carlos Silva; Diogo Lopes, Miguel Pinto; Ricardo Monteiro;

Finally I am very grateful to all the endless unconditional support from my family and friends.

## Bibliography

- [1] Surowiecki, J. (2005) The Wisdom Of Crowds
- [2] Howe, J. (2009) CrowdSourcing: Why the Power of the Crowd Is Driving the Future of Business
- [3] Python 3.6.6 -  
<https://www.python.org/downloads/release/python-366/> - Last Seen 08-07-2019
- [4] Heroku - <https://www.heroku.com/> - Last Seen 09-07-2019
- [5] Git - <https://git-scm.com/> - Last Seen 09-07-2019
- [6] MLab - <https://mlab.com/> - Last Seen 09-07-2019
- [7] MongoDB - <https://www.mongodb.com/what-is-mongodb> - Last Seen 09-10-2019
- [8] Srivastava, J. (1999) Big Five Inventory (BFI)
- [9] Facebook Platform - <https://www.facebook.com/> - Last Seen 09-10-2019
- [10] Sender Actions -  
<https://developers.facebook.com/docs/messenger-platform/send-messages/sender-actions/> - Last Seen 09-07-2019
- [11] Send API -  
<https://developers.facebook.com/docs/messenger-platform/reference/send-api> - Last Seen 09-07-2019
- [12] Persona API -  
<https://developers.facebook.com/docs/messenger-platform/send-messages/personas/> - Last Seen 09-07-2019
- [13] Webview -  
<https://developers.facebook.com/docs/messenger-platform/webview/> - Last Seen 08-07-2019
- [14] Wisdom Of the Krowd Facebook Page -  
<https://www.facebook.com/Wisdom-Of-the-Krowd-1072603452901711/> - Last Seen 08-07-2019
- [15] Facebook Buttons -  
<https://developers.facebook.com/docs/messenger-platform/send-messages/buttons-> Last Seen 08-07-2019
- [16] Persistent Menu -  
<https://developers.facebook.com/docs/messenger-platform/reference/messenger-profile-api/persistent-menu/> - Last Seen 08-07-2019
- [17] Client-Side Handler Page - <http://wokhere4u-client.herokuapp.com/tasks/> - Last Seen 09-10-2019
- [18] Client-Side Handler Import Page -  
<http://wokhere4u-client.herokuapp.com/upload/> - Last Seen 09-10-2019
- [19] Flask-RESTful - <https://flask-restful.readthedocs.io/en/latest/> - Last Seen 08-07-2019
- [20] Flask - <http://flask.pocoo.org/> - Last Seen 08-07-2019
- [21] PyMongo -  
<https://api.mongodb.com/python/current/> - Last Seen 09-07-2019
- [22] Jinja2 -  
<https://jinja.palletsprojects.com/en/2.10.x/> - Last Seen 12-10-2019
- [23] Messenger Extension SDK -  
<https://developers.facebook.com/docs/messenger-platform/webview/extensions> - Last Seen 12-10-2019