

Wyndo

Software Architecture and Stack Proposal

2023/06/29



Introduction

The purpose of this document is to present a proposed architecture and development stack for the Wyndo project. Wyndo is a web application that allows access to a catalog of products previously registered from a POS (Point of Sale) by retailers. Wyndo users can access inventory and reserve products for future purchases at a local business.

This project is in the MVP phase, so we have selected development technology (to be discussed in more detail later) that will allow us to speed up the development of functionality and at the same time have enough maturity and robustness to scale in the future when the software grows.

In this regard, a diagram of the proposed solution for this application and a detailed description of the technological stack that fits the needs of the project have been elaborated.

Architecture Overview

Below is the diagram that outlines the software components involved in this development and how they will interact with each other:

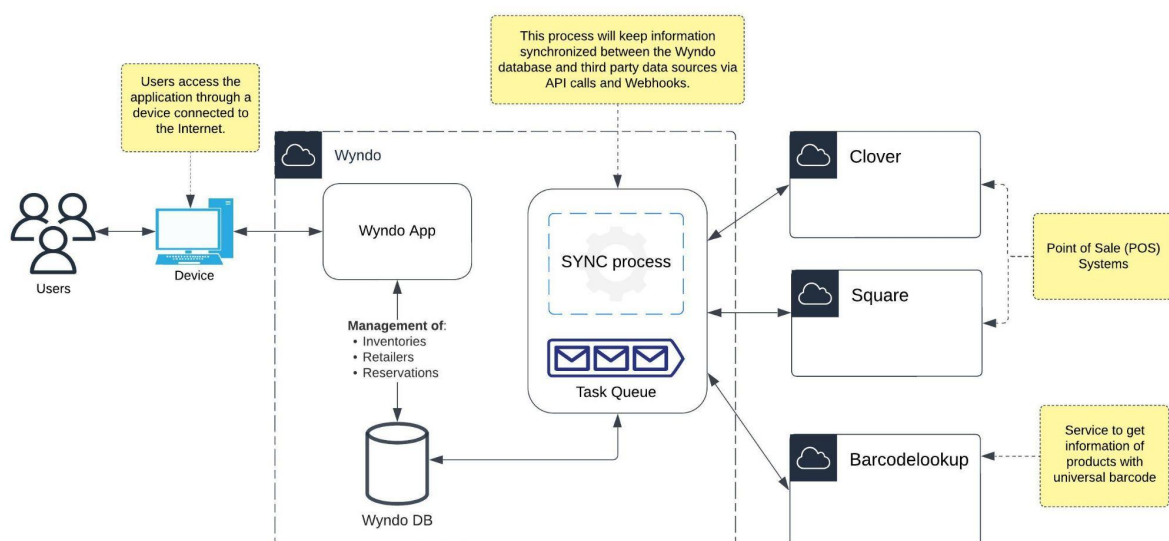


Figure 1. Diagram of application components.

1. **Wyndo App:** It is the software with which the end user will interact. Here you can find all the functionalities agreed for this project such as: The product page, The retailer page, Reservations, etc. As well as the administration panel of the application.
2. **Wyndo DB:** There will be an internal application database where not only its resources can be managed. It will also be populated with data from integrations with third parties such as inventories. Thus facilitating the ease of immediate queries and operations within the application.
3. **SYNC process:** This is the most complex component of the system. It will be responsible for Processing, Homologating, Transforming, Storing and Synchronizing the data coming from the third party platforms in the internal database of the Wyndo App. We know that each platform has different ways of structuring their data so a process that consolidates this information into a single structure is necessary. It will also maintain the synchronization of shared data between the Wyndo App and the POS such as inventories through the use of [webhooks](#) and recurring queries to data sources.
4. **POS Systems:** These are the points of sale (Clover and Square) with which Wyndo will be integrated. It is the place where retailers will manage, sell and offer the products in their inventory from the physical store.
5. **Barcodelookup:** It is a platform that offers information on products in general that have a universal barcode. Will be used to obtain the name, description and photos of those products that have UPC.

Tech Stack proposal

Considering the data synchronization process and the multiple integrations with third parties, Python will be the chosen core language for this project. It is a mature and robust programming language well known in the field of data

processing. Taking the above into consideration, the following tools have been selected:

1. **Django:** [Django](#) is a high-level, open source web framework written in Python. It stands out for its focus on simplicity and efficiency in web development. Some of its key features are. One of the advantages of Django is its admin, a powerful tool that allows you to perform CRUD (Create, Read, Update and Delete) operations on the models created in the framework's database without investing time in the development of a graphical user interface. Basically Django has all the necessary tools to develop a robust, efficient and scalable product in the shortest possible time. It also has a large community, documentation and implementations since its beginnings. Django will provide the fullstack tools with which the Wyndo application will be developed. Its approach is Server Side Rendering although it can be modified to an approach where the logic, persistence and presentation layer are totally separated using Django REST framework.
2. **Celery:** [Celery](#) is an open source library for asynchronous and distributed task execution in Python applications. It allows decoupling long-running or resource-intensive tasks from the main application, improving scalability and responsiveness. Celery will be the task queue manager for ETL processes and data synchronization, ensuring a scalable architecture from the start for multiple integrations with heavy background processes.
3. **Redis:** [Redis](#) is an open source, high-performance in-memory data storage system. Its purpose in Wyndos is to serve as a message queue broker to manage the tasks loaded by Celery in ETL and synchronization processes.
4. **Postgres:** [Postgres](#) is a robust relational database widely known in the software development market for its speed and efficiency. It will store the data to be managed by the Wyndo application.