

REC - DESCRIBING THE TECHNICAL SIDE

This document explains how the Rec works from a technological point of view. It explains which entities are part of it, which technologies they use and how it is deployed in the cloud. The aim is to explain how the platform works without going into detail on each of its entities.

REC's technical development has been divided into three parts in order to better explain the different stages the project has gone through

- STAGE 1. Initial development of REC connected to the blockchain
- STAGE 2. Scaling up the infrastructure
- STAGE 3. How the REC will work with the technology targets set for 2020

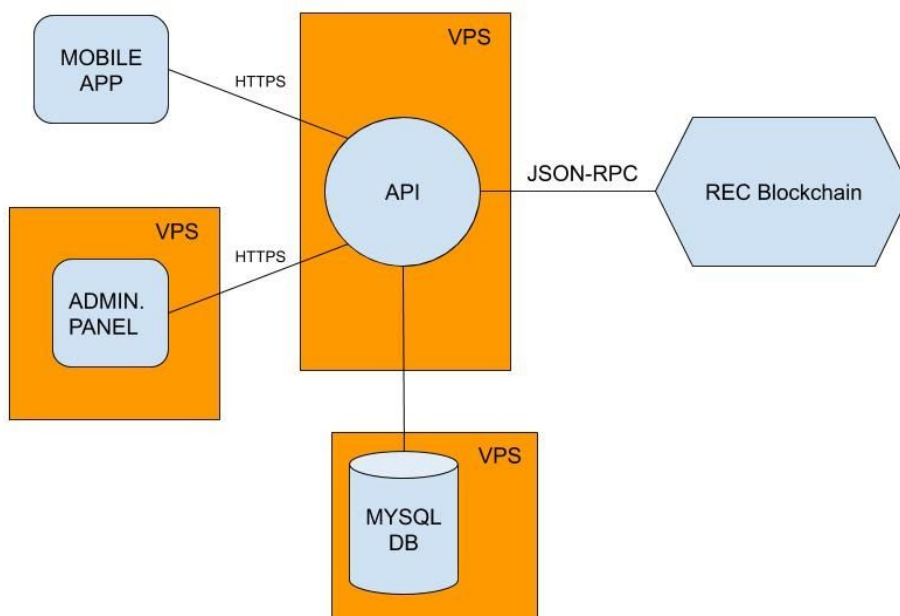
STAGE 1 - TRL7

In the first months that the REC platform was in production and working for the users, it worked with the blockchain described in the entities section, the API was the gateway between the users and the blockchain.

The blockchain is built using a fork of Litecoin with enhanced block size to store more transaction details. The nodes were deployed in the main REC supporting entities (UPC, NOVACT, QBIT).

With this architecture we used a couple of big servers in a french datacenter (OVH), both splitted into several VPS.

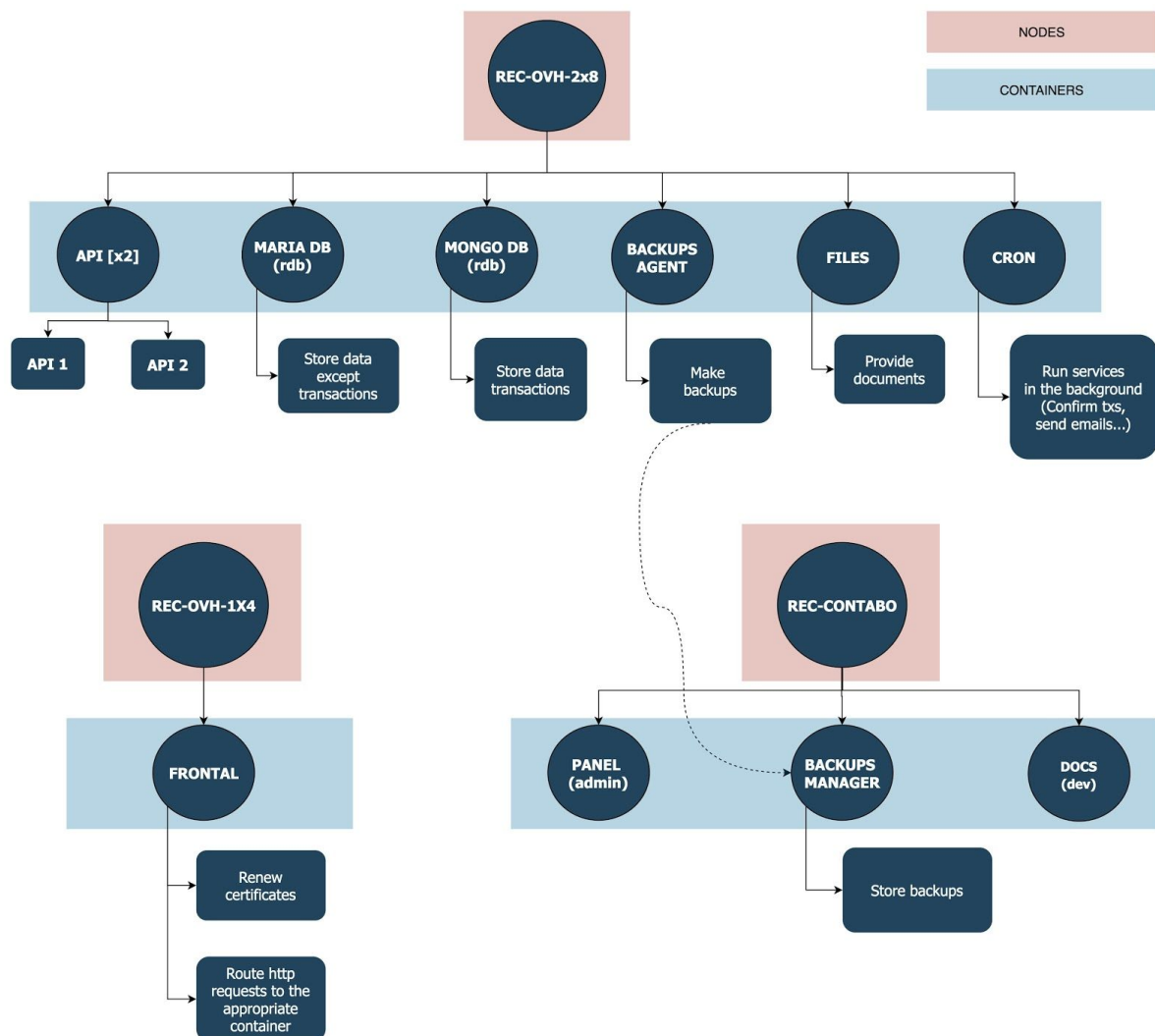
This graphics represents the initial architecture of our platform.



STAGE 2 - TRL8

When we tried to scale up the system, we found some problems in the blockchain side that caused a bottleneck for the development, due to this problems, the low budget and the hurry to solve some production issues, we had to deactivate temporarily the blockchain and store the transactions only in the database. This model had to offer an easy implementation, allowing to store a great amount of transactions and somehow to be able to extract from it all the transactions to replicate them in the future re-add the blockchain side, so the new propose was implement mongodb to store the transactions.

Now the system is deployed with Docker Swarm as a container orchestrator, using a private cloud with servers across multiple datacenters in europe. This new architecture allows to scale easier the system and makes it more robust and secure by decoupling the services.



STAGE 3 - TRL9

[Work in progress..]

For this stage to happen, the rec needs economic capital and this would be possible if we win the prize with gmerits. The aim of this stage is to reconnect the blockchain and have it control everything related to the transactions instead of being the API. In this stage we would also adapt and improve the api together with the architecture of the platform to be able to easily replicate the rec in other cities or communities, improve the security of payments, pay security audits and be able to offer some of our entities to the open source community.

ENTITIES

- ADMIN. PANEL

A control panel to administrate the differents elements of the rec system. This elements are users, accounts, kyc, documents, transactions and special actions.

Technology: Angular
Available for: Rec Administrators
Security: Basic auth + password + 2FA

- API

It's the core, it manage all requests from the other entities and get a response.

Technology: Symfony (PHP Framework)
Available for: Public, users, administrators.
Authentication: Oauth 2.0
Documentation: <https://dev.rec.barcelona/> (Needs an update)

- MOBILE APP.

This is the tool from which users can register and interact with the REC platform; performing operations, managing their accounts, consulting the map of shops...

Technology: Ionic Framework (mobile dev framework)
Available for: Android and iOS
Security: Password and PIN to make payments

- MYSQL DATABASE

It is our relational database, it stores all information that is not related with recs transactions.

Technology: MariaDB
Available for: REC Developers
Security: Basic auth + strong password

- MONGODB DATABASE

Since the blockchain was deactivated this is our no-relational data base to store all the transactions that are made on the platform.

Technology: Mongodb
Available for: REC Developers
Security: Basic auth + strong password

- BLOCKCHAIN

The blockchain is built using a fork of Litecoin (creativechain) with enhanced block size to store more transaction details. The nodes were deployed in the main REC supporting entities (UPC, NOVACT, QBIT).

DEVELOPMENT TOOLS

- Github to store code changes, continuous integration (with github actions) and project management.
- PhpStorm to develop the API
- Visual Studio Code to develop the app and admin panel
- Ionic Framework for the APP
- Symfony 3.4 LTS for the API
- PHPUnit to test the API
- docker-compose to setup the local development environment
- PostMan to test the API

DEPLOYMENT TOOLS

We use **docker swarm** as a orchestrator to deploy the system into the cloud, It allows to easier scale the system by adding more replicas on-demand.

We also use Github Actions for continuous delivery, when a release is published, it takes the code and sends it to the **swarm**.

