MANDELROT APPS

OVERVIEW

Intro

The Mandelrot Apps backend is an ERP engine, fully portable and fully modular. Reading this document you will understand in a few minutes what all this means, and what makes this software different than others.

It's meant to be used in small or medium companies or organizations (a few hundreds of users would be sure ok, and the server could probably handle much more), and it's focused on the tech employees productivity and therefore the company costs. The system administration becomes dead simple and the apps development becames way faster and needing less people.

You will find all the updated info, including the guides and some basic example apps to see how the system works (don't miss those!), in the project GitHub repo:

https://github.com/mandelrot/mapps_backend

This software has been made by Jose Alemán / Mandelrot. You can download the code and use it freely, including changing whatever you want. In case you want to know more or contact me:

My blog: https://mandelrot.com (spanish)
My professional webpage: http://josealeman.info (english)

Concept

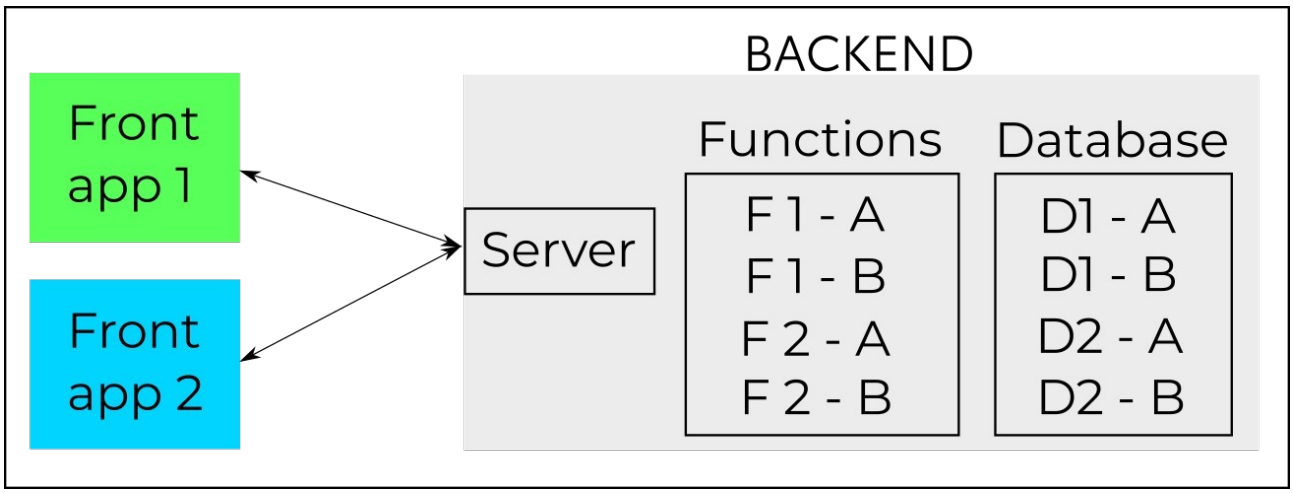
An ERP is basically an ecosystem where different apps work together. Let's suppose your ERP has an app that just manages clients data and stores it in the system database (let's call it "Clients"). And you have other different apps in the ERP called "Salespoint" and "Inventory". If a registered client comes to your shop and buys something, the Salespoint app will retrieve the client data from the DB (stored there by "Clients") and the purchase will update the inventory so when another employee checks the "Inventory" app they will have already synchronized.

So in an ERP, since the different aspects of your business manage and store the data in a common environment, all the frontend apps share information through the database and your employees can access everything easily. You don't need to send papers to Accountability to let them know a new sale has been made: they will already know because behind the scenes they are reading the same info you wrote.

This is the common ERP technical structure the software has. Some frontend apps ("Salespoint", "Inventory") the employee work with talk to a backend server, and the backend server proccesses the information and eventually stores in a common database.

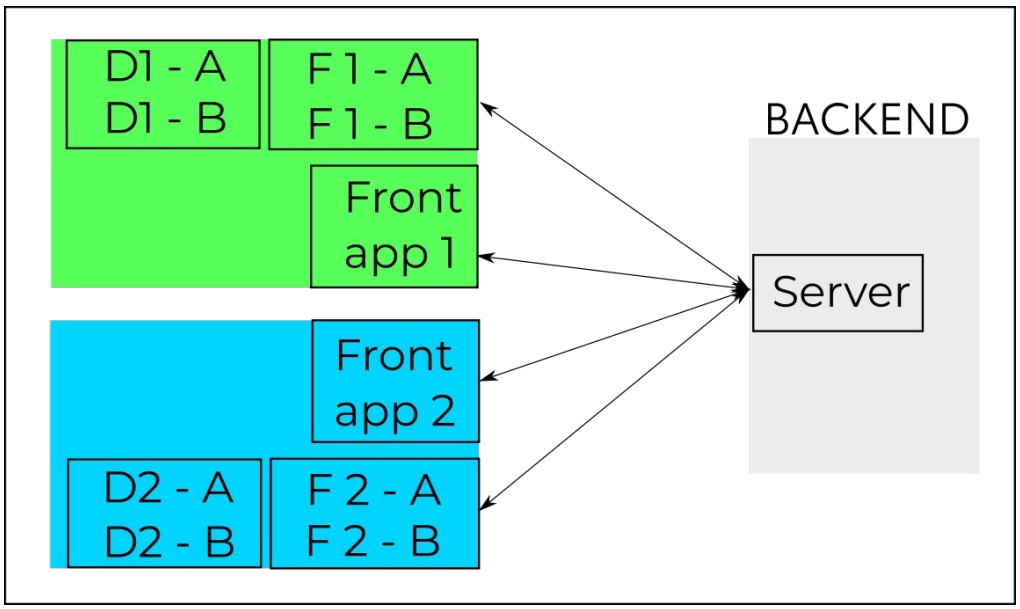
It's important to know that the backend structure and content may be adapted to which frontend apps are installed: this specific app requires some tools (functions) in the backend to proccess its information in a specific way, and some database lists of items (example: "Clients" will need a list of clients in the database, "Inventory" will need to have stored a list of products, and so on).

So this is what we have got so far:



So as we see, you can't create new frontend software without having to do things in the backend as well: this very likely implies than more than one person must be involved in the development/deployment (the front developer and the programmers who are familiar with the backend/db environment). Any change may became a big change.

Now, before getting into explanations, let's see the MApps concept:



The Mandelrot Apps backend is mainly a router (it does some other things but that's not the point here) that gets messages from the front apps, proccesses those messages, and trigger the targeted backend function (that eventually will store the data too).

There are three main differences here:

- The location of the backend elements involved with the frontend business logic. The frontend app "package" now is a folder that includes the front app itself, plus its own backend functions and eventually its own storage. Maintenance, backups and migrations are just a matter of copying or moving a folder... And, in this particular Mandelrot Apps case, the sysadmins won't even need to restart the server when doing admin changes.

- Now a front app can talk to its own backend functions... But also to other apps backend functions, directly. So Front app 2 could invoke a function of Front App 1 and viceversa (each developer easily decides which functions are private and which ones public). The message to the backend needs to say who is calling and who is the target, and the backend will know which function to invoke (details in the frontend developer guide in the Github project Docs folder).

- Development: the front package can be developed by only one person who understands the software business logic, and without affecting or having anything to do with the rest of the system. This design improves the way of working, prevent problems spreads and open totally new possibilities (see details in the frontend developers guide).

On the other hand, a very relevant feature is that the MApps backend doesn't need any instalation: the webserver and a in-file database system are already built in, hence the system admin will just have to copy the software folder to a server, run the executable file, and the ERP will be immediately working. This, by the way, is how your admin will do migrations: it's impossible to make it simpler.

This software has many other functionalities and details to boost the tech team produtivity and make your organization software better. If you want to see it working you can download an example app (Windows and Linux available) from the download link at the GitHub project page.

And if you have questions or comments do not hesitate to contact me.