

md architecture

This project architecture is inspired by RVMS (Reactive Views, Managers, Services) , Clean architecture and tips from the clean code book

Documentation will be complete soon 😊

Digrams of Md Architecture



Why [RVMS \(Reactive Views, Managers, Services\)](#) ?

The most popular architectures for flutter are usually BLoC, MVC, or MVVM. But on second thought, as described by [Thomas Burkhardt](#), these almost feel unintuitive to the way flutter is built. Although it is practically possible to implement them, flutter follows a reactive approach where the UI reacts to the data flow.

BLoC accomplishes this but at the expense of boilerplate code and making the file structure more complex, while MVVM on the other hand is much less intuitive for the reactive nature of flutter as it is more suitable to Native android or Xamarian where the UI elements are in XML and need have a ViewModel Associated with, to represent and update the UI accordingly.

But Flutter widgets are self-responsible and can handle state without the need for any ViewModel, and they do not communicate via bindings like native android does to update its views, flutter widgets always rebuild instead.

The Basics

RVMS (Reactive Views, Managers, Services) is again, a reiteration of the previous architecture, RxVMS, RVMS simplifies the process by removing the Reactive part, streams with ValueNotifiers.

Services

Handle, incoming and outgoing requests with external services like a database, authentication, or geolocation service. They can be classes or interfaces that define the different types of requests needed by the app to fulfill its data requirements. They are not visible to the view and do not change any state.

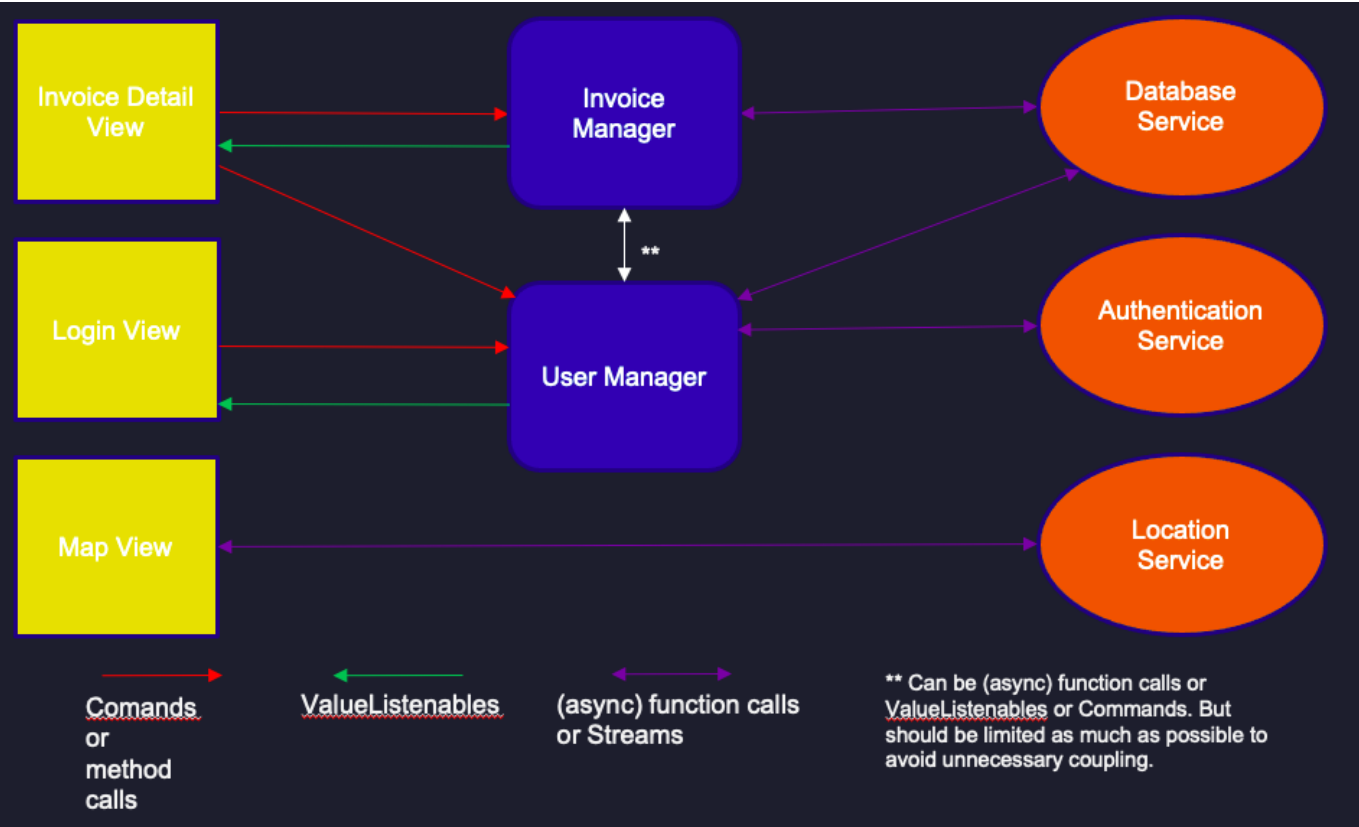
Managers

Managers are responsible for managing the business logic of the app and communicating incoming data to the Views via state updates. It groups together use cases or logic that multiple connected Views might use. Basically, managers act as the middle man between the Services and Views by sending data and transforming it, if required.

Views

Views are what a user sees on the screen, it describes how UI on the screen should look like, and what layout it should follow. Views consume the incoming data from the managers and send back any interaction that a

user makes. In flutter, Views are described using StatefulWidget or a StatelessWidget. Flutter’s declarative UI makes this simple.



RVMS by [Thomas Burkhart](#)

[Referenced](#)