

Project re-structuration

Table content

- Current structure
- File types proposal
- Functional based structure
- App based structure

Current structure

API

- File1-module
- File2-module

MODELS

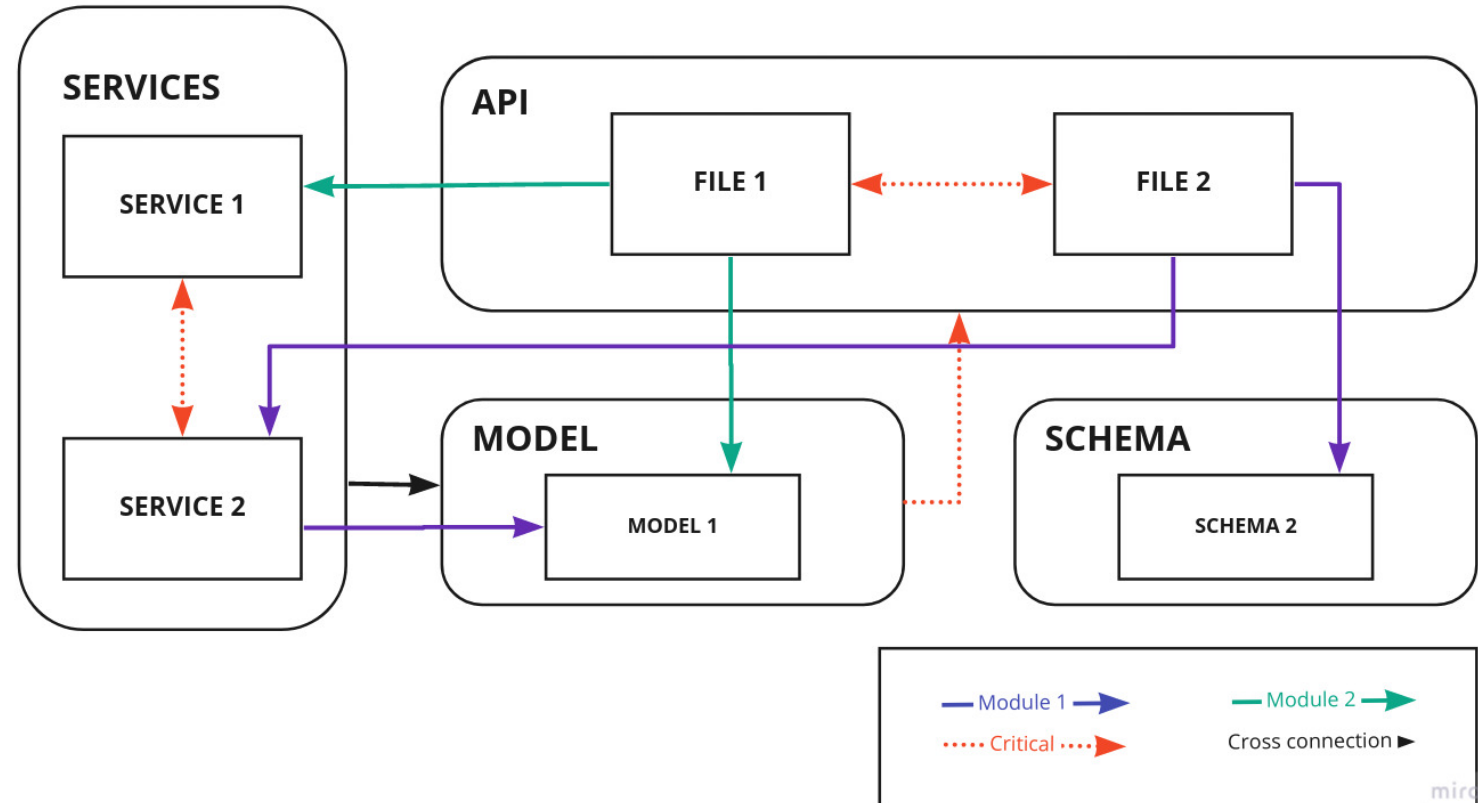
- Model1
- Model2

SERVICES

- File1-module
- File2-module

SCHEMA

- File1-module
- File2-module



File types proposal

File	Description
Models	Database models
Services	Business logic
Selectors	Data access orm services
Schema	Validate Python data structures
Views	Logic that respond to requests to the application

Functional based structure

VIEWS

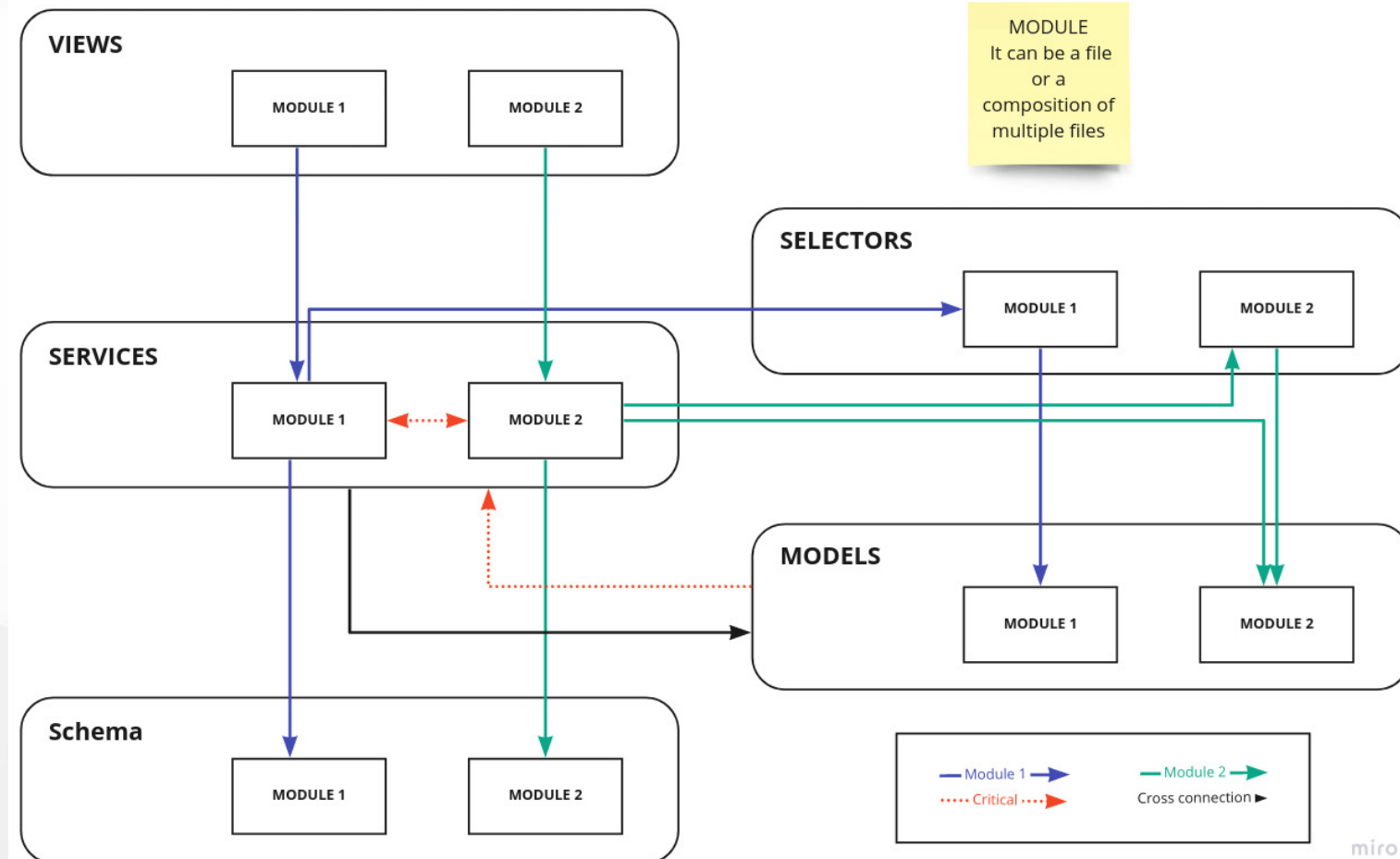
- Settings
 - File1-module
- Company
 - File1-module

MODELS

- Settings
 - File1-module
- Company
 - File1-module

SCHEMAS

- Settings
 - File1-module
- Company
 - File1-module



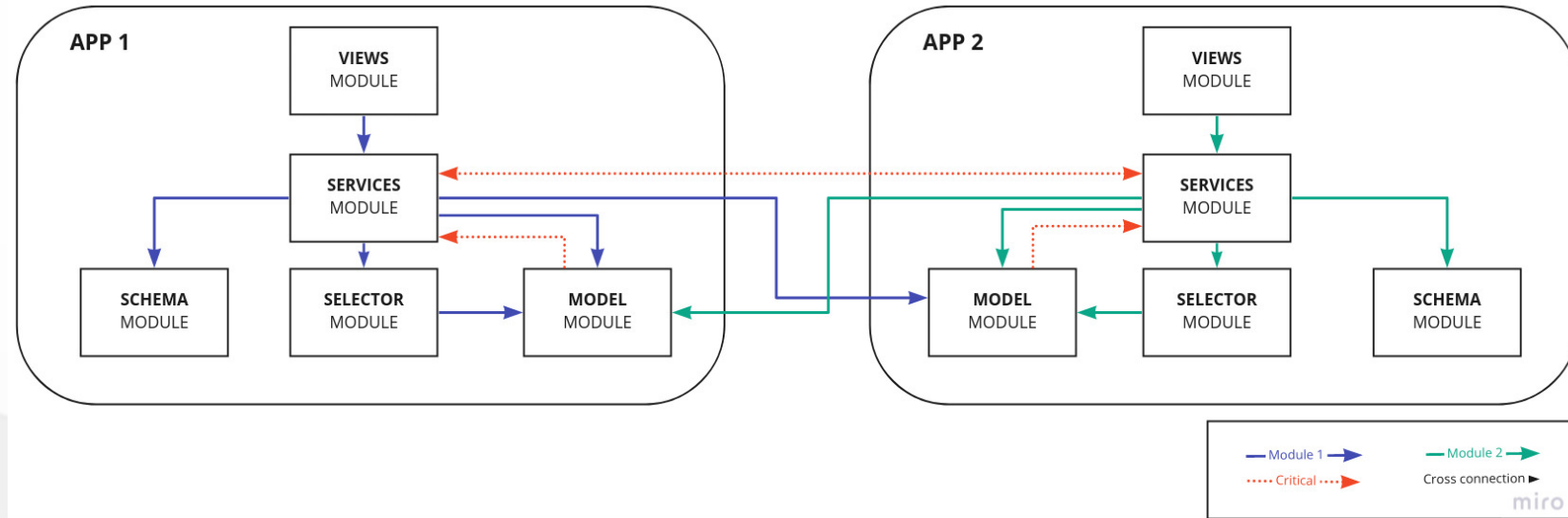
App based structure

SETTINGS

- services.py
- views.py
- selectors.py
- schema.py
- models.py

COMPANY

- services.py
- views.py
- selectors.py
- schema.py
- models.py



Analysis

FUNCTIONAL BASED VS APP BASED

Questions

- What to do with the classes that are highly coupled?
- When has a file to be splitted?
- How should be grouped the files which are part of the same category? EX: services -> File1/File2

Considerations

- **Test** are going to **keep apart from the other files** as it is today, but it should adopt internally the **same structure of the main folder**
- Each function needs its own **test class** if we are doing **unittesting**

The background is a stylized illustration of a mountain landscape. In the foreground, a series of white, rectangular steps lead up a slope towards the right. At the top of this slope, a black flag with a white arrow pointing right is planted. The mountain's peak is brown. The sky is a light teal color with several white, fluffy clouds. In the background, there are more brown mountains of varying heights. The overall style is flat and modern.

Small steps **add up**