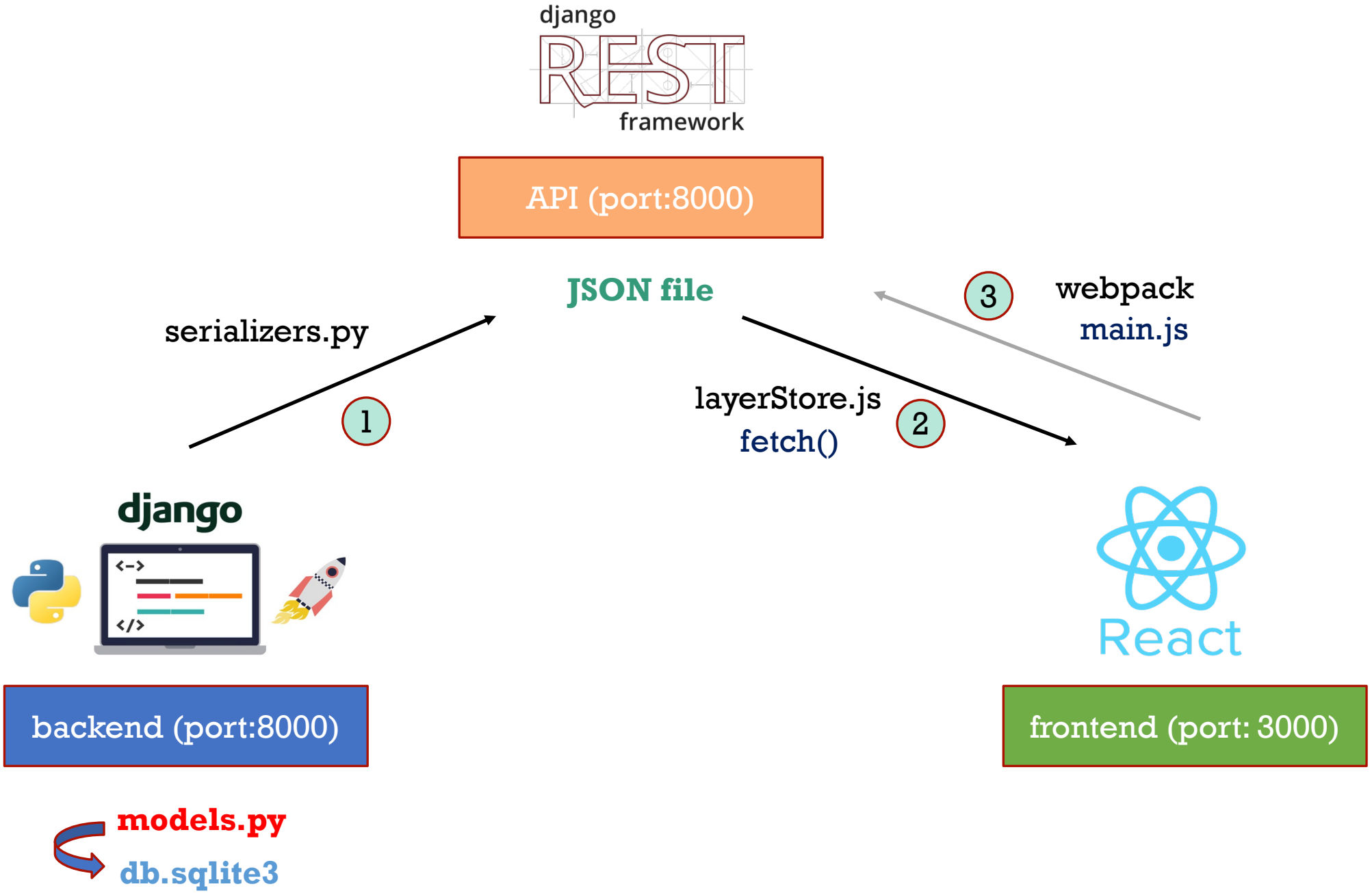
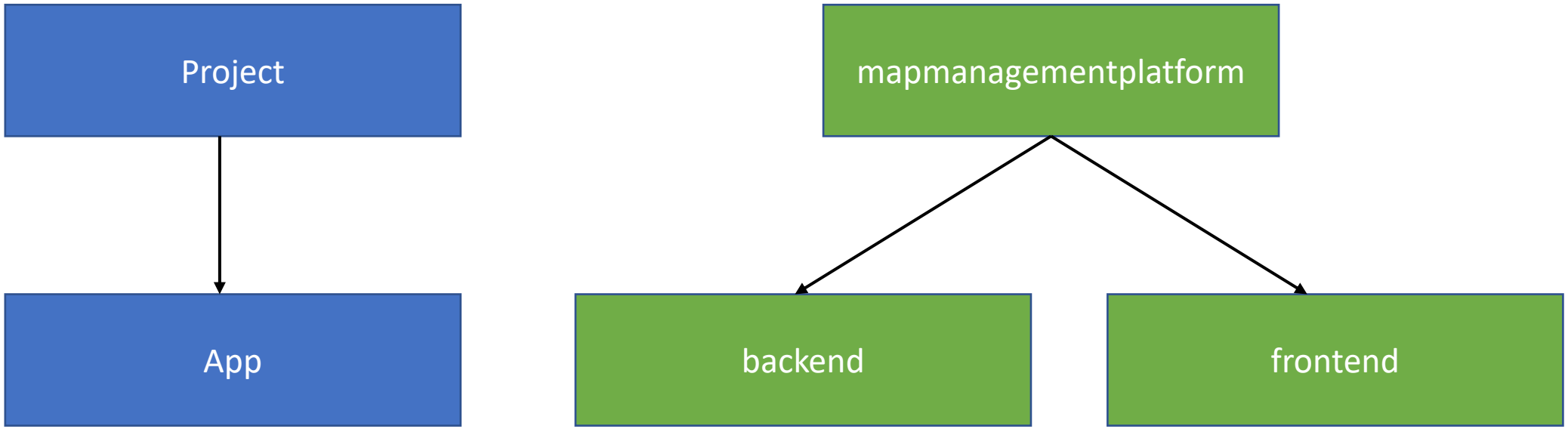

django
REST
framework
API (port:8000)
JSON file
serializers.py
1
layerStore.js
fetch()
2
3
webpack
main.js
django
React
backend (port:8000)
frontend (port: 3000)
models.py
db.sqlite3

Django Project & App



Project - mapmanagementplatform

```
├── manage.py
├── mapmanagementplatform
|   ├── __init__.py
|   ├── __pycache__
|   ├── asgi.py
|   ├── settings.py
|   ├── urls.py
|   └── wsgi.py
```

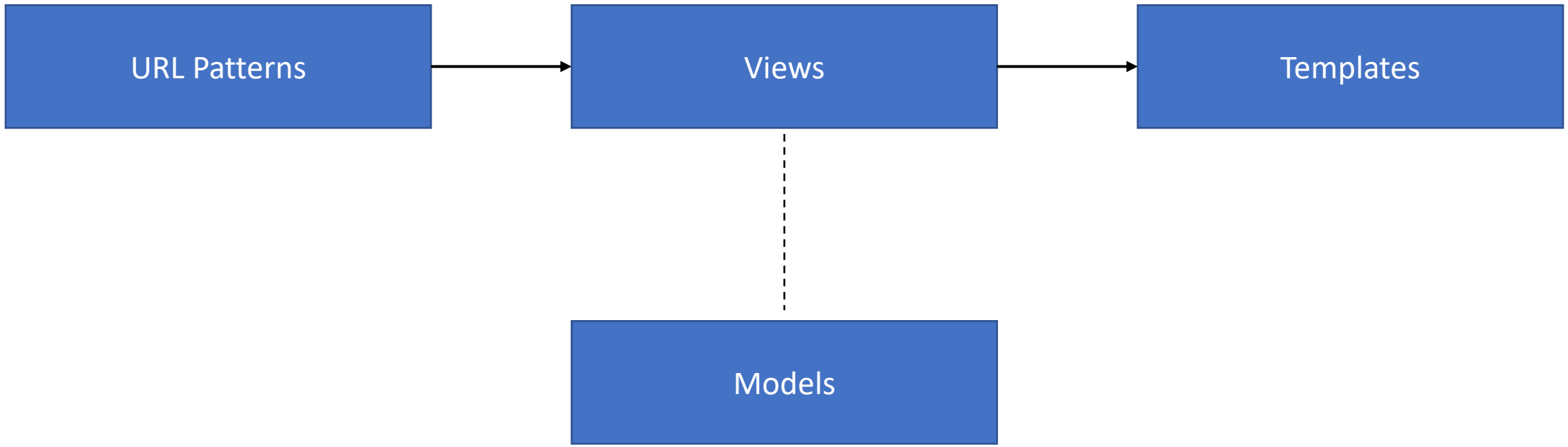
File	Role
manage.py	run commands
mapmanagementplatform/ __init__.py (or dunder anit)	tell Python that the folder contains Python code
mapmanagementplatform/ wsgi.py and mapmanagementplatform/ asgi.py	provide hooks for web servers when Django is running on a live website
mapmanagementplatform/ settings.py	configure the Django project
mapmanagementplatform/ urls.py	routes requests based on the URL

```
├── backend
|   ├── __init__.py
|   ├── __pycache__
|   ├── admin.py
|   ├── apps.py
|   ├── migrations
|   ├── models.py
|   ├── serializers.py
|   ├── static
|   ├── templates
|   ├── tests.py
|   ├── urls.py
|   └── views.py
├── db.sqlite3
├── frontend
|   ├── __init__.py
|   ├── __pycache__
|   ├── admin.py
|   ├── apps.py
|   ├── migrations
|   ├── models.py
|   ├── node_modules
|   ├── package-lock.json
|   ├── package.json
|   ├── public
|   ├── server
|   ├── src
|   ├── static
|   ├── templates
|   ├── tests.py
|   ├── urls.py
|   ├── views.py
|   └── webpack.config.js
```

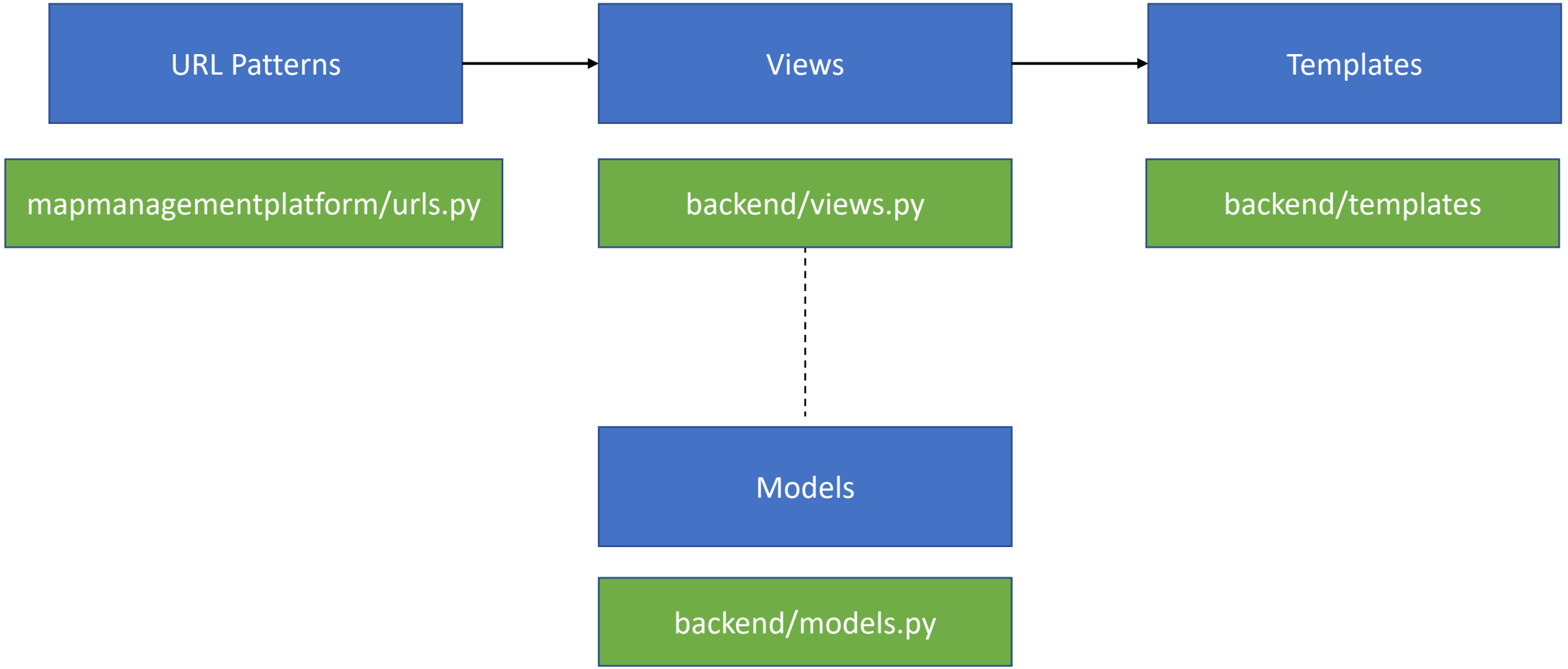
Apps - backend & frontend

File or Folder	Role
admin.py	define an administrative interface for the app that will allow us to see and edit the data related to this app
apps.py	control settings specific to the app
migrations/	hold files which Django uses to migrate the database as we create and change our database schema over time
models.py	provide the data layer, which Django uses to construct the database schema and queries
tests.py	used for writing unit tests for the functionality of the app
urls.py	used for URL routing specific to the app
views.py	define the logic and control flow for handling requests and define the HTTP responses that are returned
db.sqlite3	SQLite database

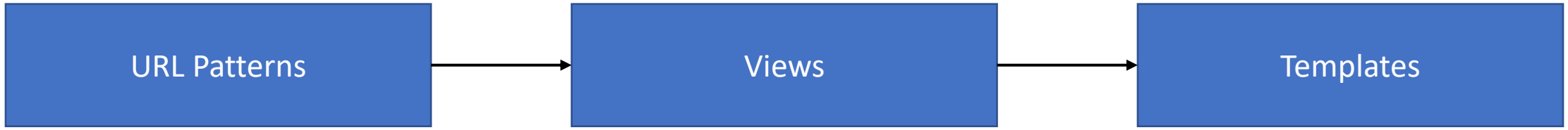
MVC Architecture



MVC Architecture

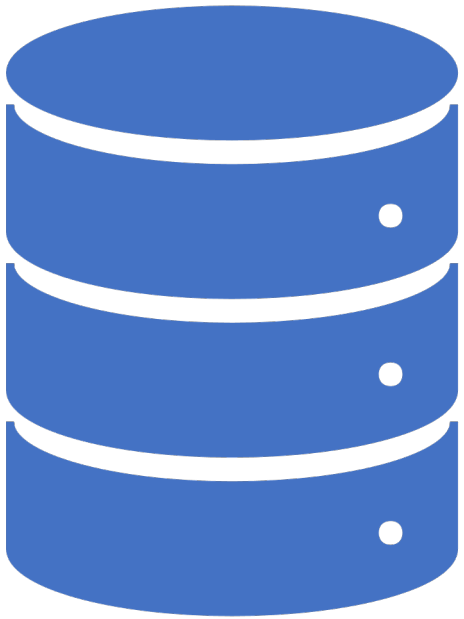


Flow of Control - Examples



`http://127.0.0.1:8000/`

`(empty)`



Django Models

- Create the data layer of a Django app
- Define database structure
- Allow us to query the database

models.py

The models.py file contains the set of models for its Django app.

Model

A model is a class that inherits from django.db.models.Model, and it defines fields as class attributes.

UML class diagram from Django models