Coding Environment Install and Setup

Django

- An IDE is an Integrated Development Environment.
  - Larger software where you can code but also has many integrations (e.g. .NET apps, Source Control).
- An Editor allows you to open and edit files.
  - Lighter weight program that opens and supports editing of files.

Django

- Choosing an IDE or Editor is **highly personal** and there are many, many options available!
  - *NOTE:*
    - *Please feel free to use any IDE or editor you prefer and feel comfortable with!*

Django

- IDE or Editor Common Options:
  - PyCharm
  - Atom
  - Sublime Text Editor
  - Microsoft Visual Studio
    - *Important Note:*
      - *Notebook based editors like Jupyter won't work for Django!*

Django

- IDE for Django should ideally have:
  - Syntax Highlighting Support for multiple languages (HTML, CSS, Python).
  - Command Line Interface.
- Nice to haves:
  - Highly Customizable Styling and Themes
  - Web Browser in Editor

Django

- For this course we will use the free version of Microsoft Visual Studio Code.
  - *Note:*
    - *This is different than “Microsoft Visual Studio”!*

Django

- Microsoft Visual Studio Code
  - Free
  - Cross-Platform (Windows, MacOS, Linux)
  - Customizable
  - Supports Multiple File Types

Django

- Let’s download it at:
  - **https://code.visualstudio.com/**

Introduction to Django Framework

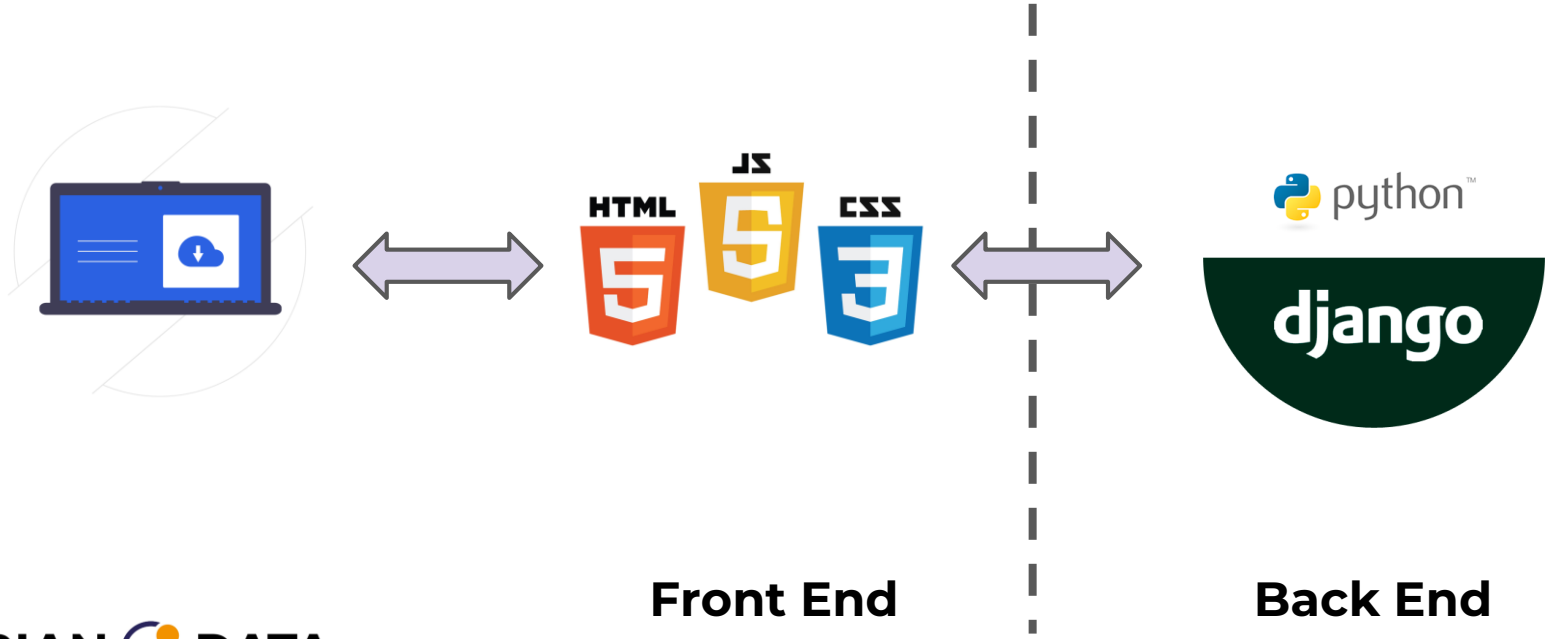
PIERIAN DATA

Django

- Let’s have a brief review:
    - What is Django Framework?
    - Why use Django?
    - Pros and Cons of Django
    - Who uses Django?
- Afterwards we’ll have a separate lecture on ***how*** the Django Framework works.

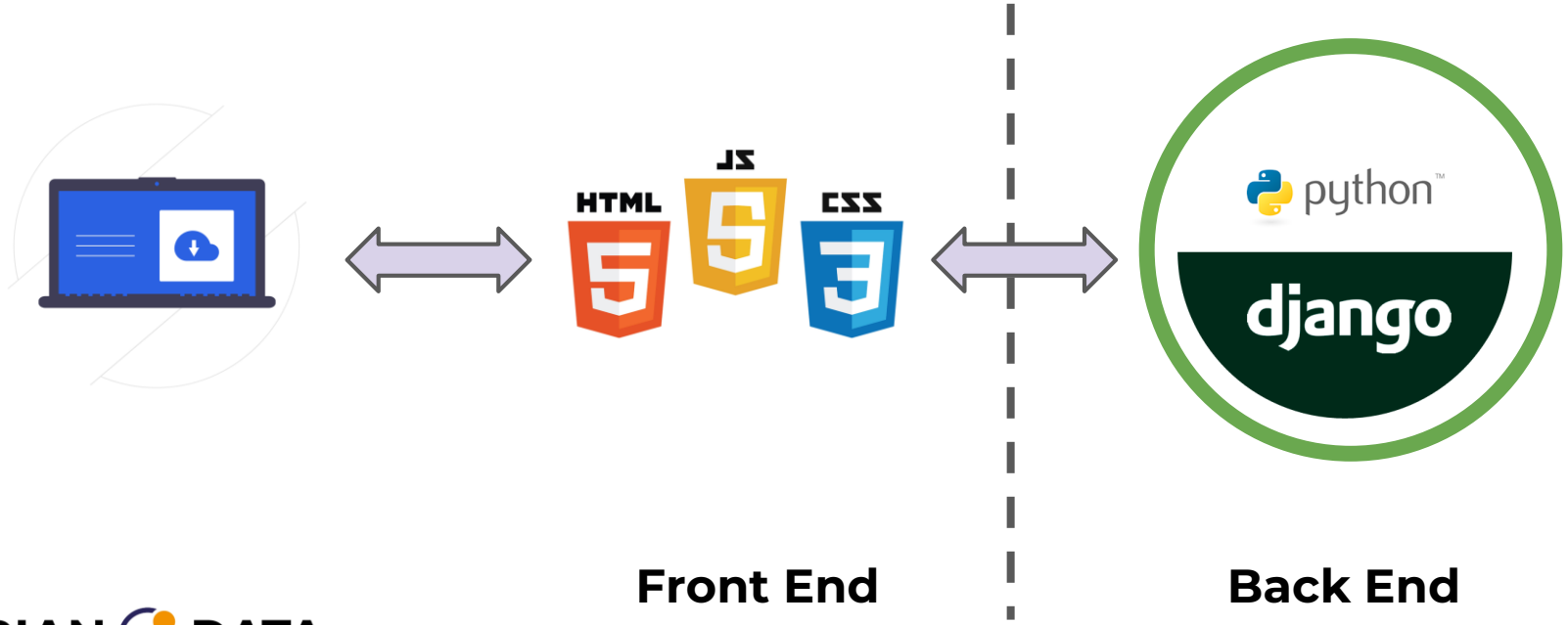
Django

- Website Process Review



Django

- Website Process Review



Django

- What is Django?

Django

- What is Django?
  - Created by Adrian Holovaty and Simon Willison in 2003 when they worked at the *Lawrence Journal-World* newspaper.
  - Released under BSD license in 2005 and in 2008 the non-profit Django Software foundation was created.

Django

- What is Django?
  - Take note of the fact that this project was started at a newspaper!
  - This has created a culture of **very** good online documentation for Django!
    - **www.djangoproject.com**

Django

- What is Django?
    - Named after Django Reinhardt

PIERIAN DATA

Django

- What is Django?
  - Python based framework for creating web applications.
  - Framework provides rules, structures, and functionality that allows us to use Python code and libraries on the back end of our web application.

Django

- What is Django?
  - Python is the programming language used to work with Django.
  - Django can then interact with our web applications to send information to the user of the web application.

Django

- Why use Django?

Django

- Why use Django?
  - Key Features:
    - Allows for fast development.
    - Many common features included.
    - Updated often and secure.
    - Very scalable.
    - Very versatile with Python.

Django

- Why use Django?
  - You may have heard Django has “batteries included” or is “fully loaded”.
  - This just means that is has a lot of built in Python modules that take care of really common web application features.

Django

- Why use Django?
  - Lots of built-in functionality:
    - Administration
    - Authentication
    - Database Interaction
    - Security

Django

- Why use Django?
    - One of the most important aspects of is that since it uses Python as its programming language, we get to use all the cool Python libraries available to use easily within Django!

Django

- Why use Django?
  - For example, if your favorite data analysis tool uses Python, we can easily integrate it into our code with the Django framework.
  - This allows us to expand Python based projects to become user interactive web applications.

Django

- Django Pros:
  - Lots of features already built-in.
  - Built with Python.
  - Scalable, versatile, and secure.

Django

- Django Cons:
  - Since it has so many features, there is sometimes a bit of a steep learning curve.
  - It may include many features you don’t intend on using.
  - Built with Python.

Django

- Who uses Django?
  - Instagram
  - Spotify
  - YouTube
  - Pinterest
  - DropBox
  - EventBrite
  - and many more!

Django

- We've only briefly given you an overview of the Django Framework.
- In the next lecture, let's dive into the specific details of how this Framework is designed, which will relate highly to the upcoming sections in which we learn about these core components!

How Django Works

Django

- Let’s discuss the key ideas behind how the Django Framework works.
- This will also lead to a nice roadmap of our learning journey for the rest of the course!

Django

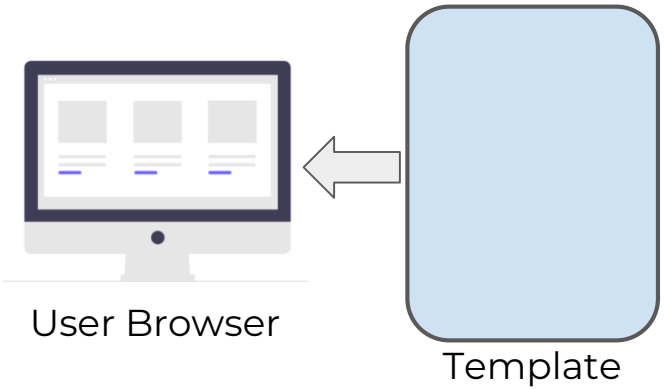
- Key Features of Django:
  - Model-Template-View (MTV) Structure
    - ORM - Object-relational Mapper
    - Models
    - URLs and Views
    - Templates

Django

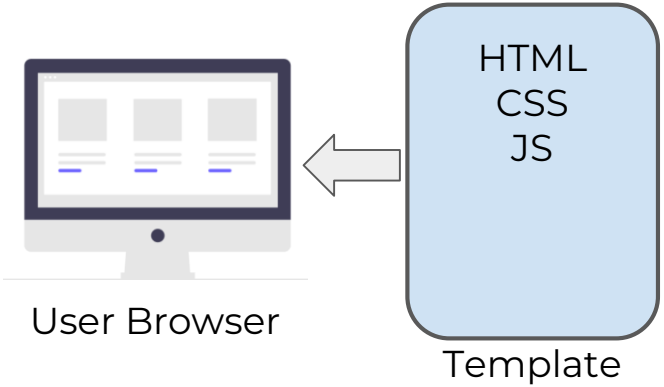
User Browser

Database

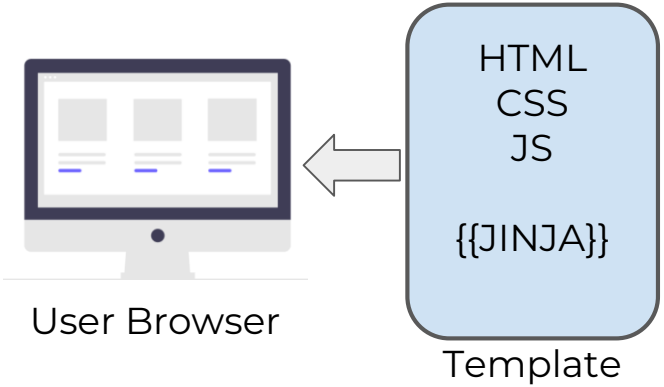
Django



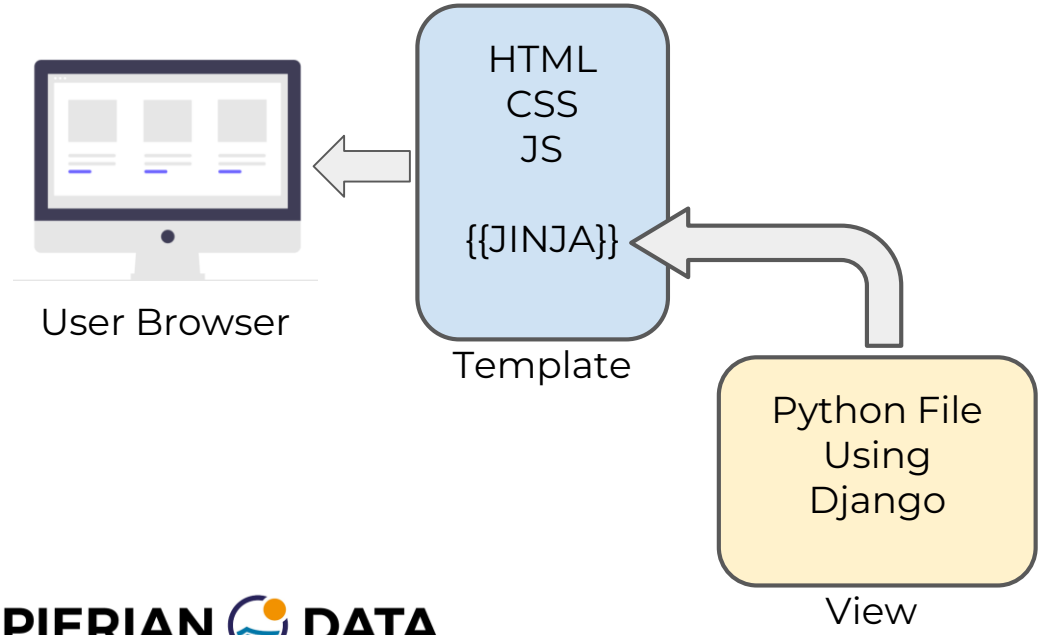
Django



Django

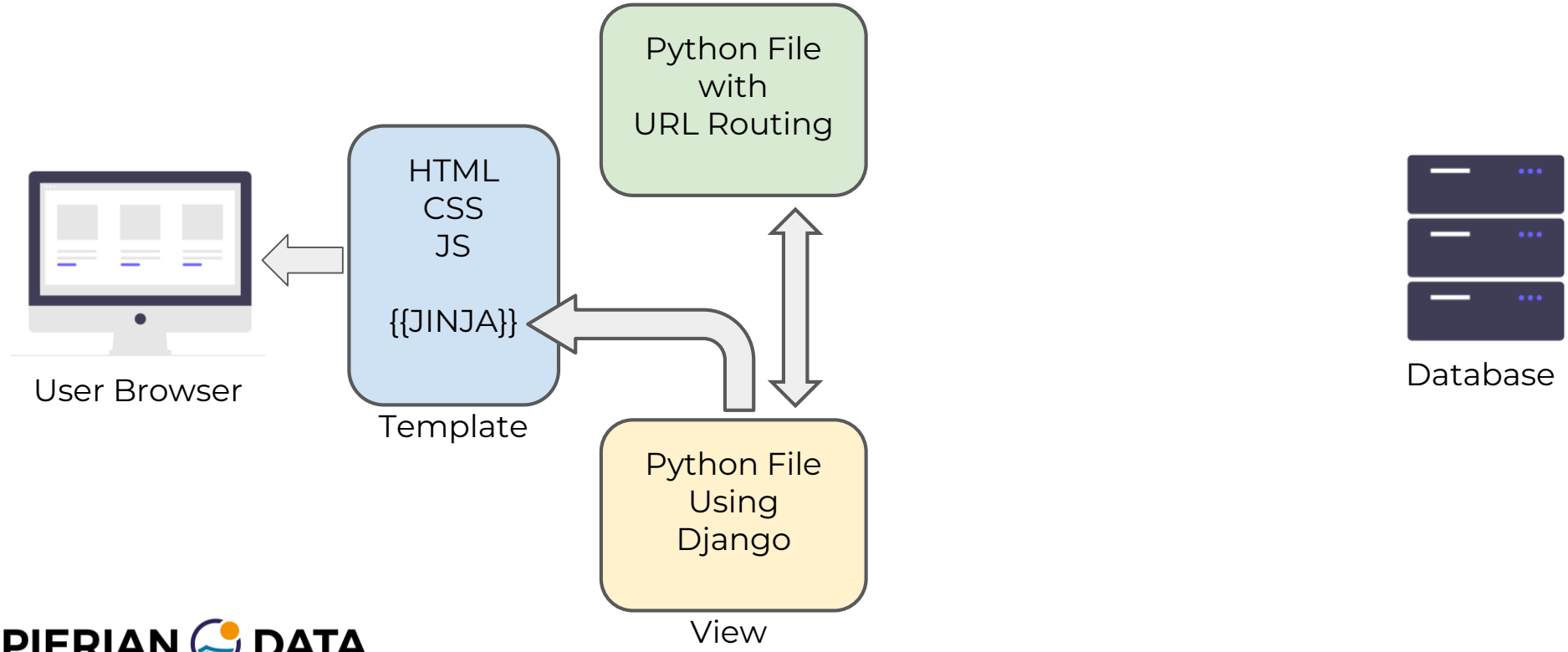


Django

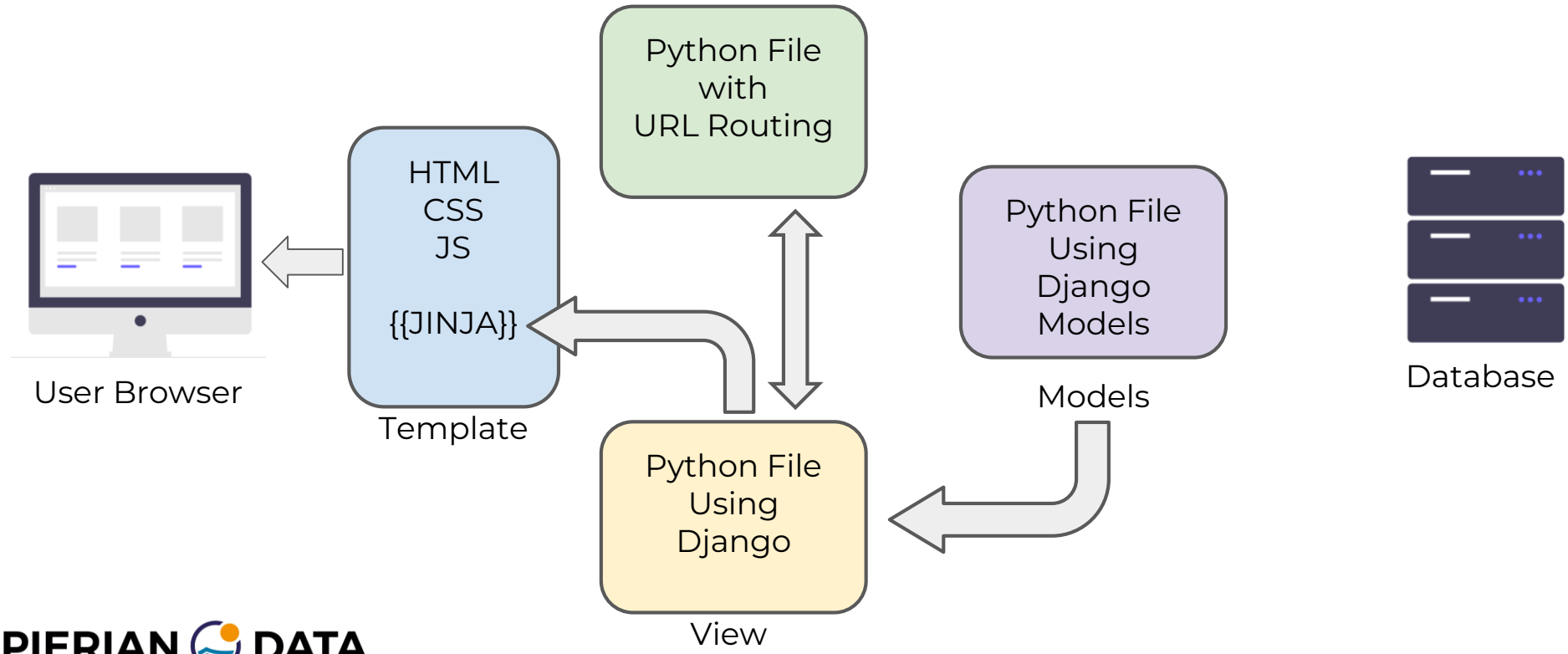


PIERIAN DATA

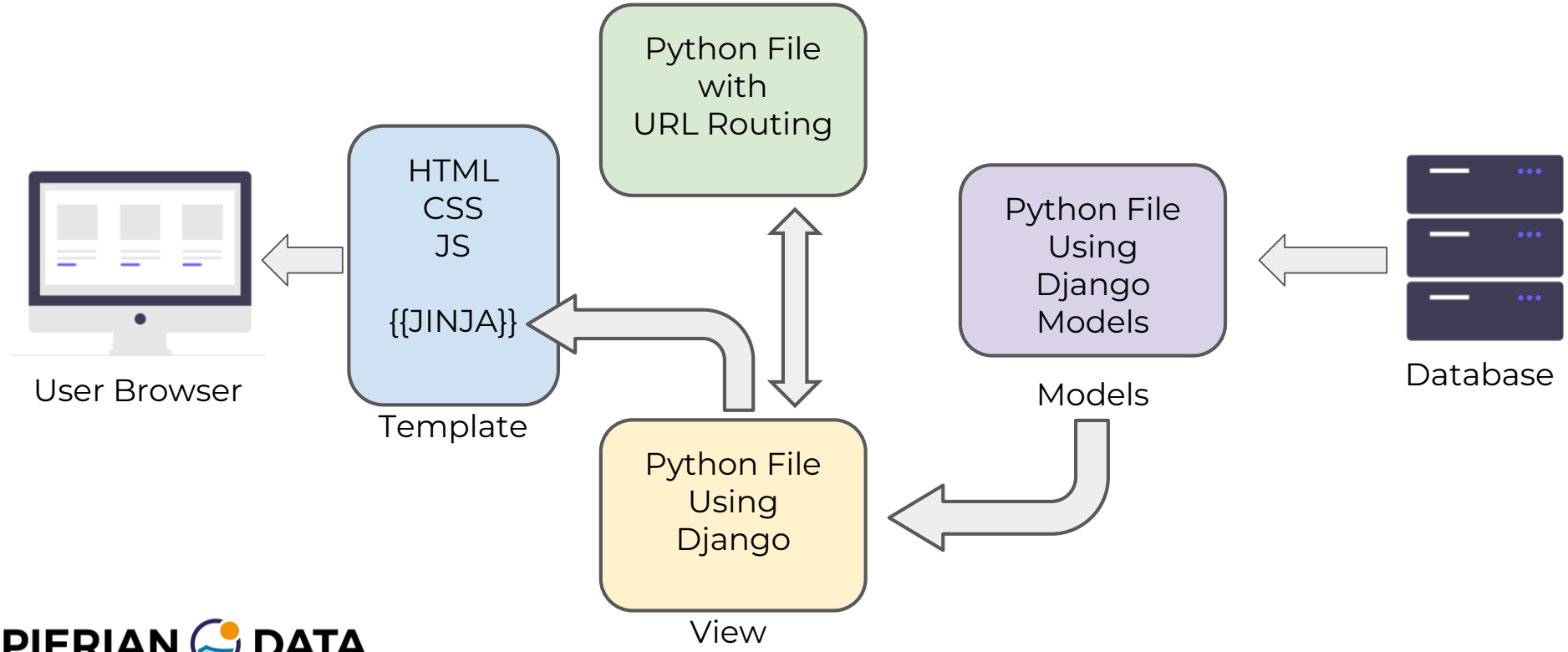
Django


Python File
with
URL Routing
HTML
CSS
JS
{{JINJA}}
Template
User Browser
Python File
Using
Django
View
Database

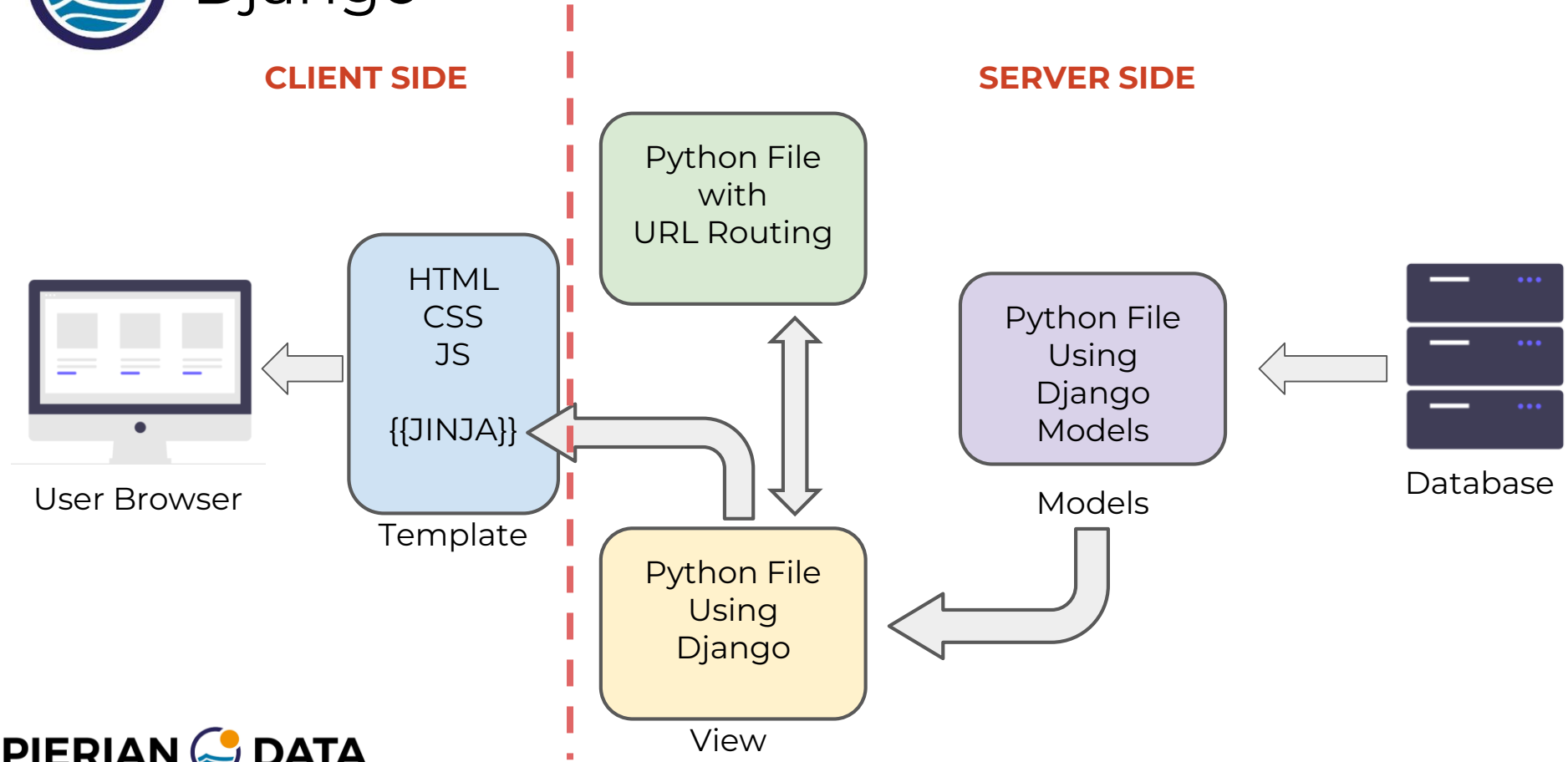
Django


Python File
with
URL Routing
HTML
CSS
JS
{{JINJA}}
Template
User Browser
Python File
Using
Django
Models
Models
Database
Python File
Using
Django
View

Django

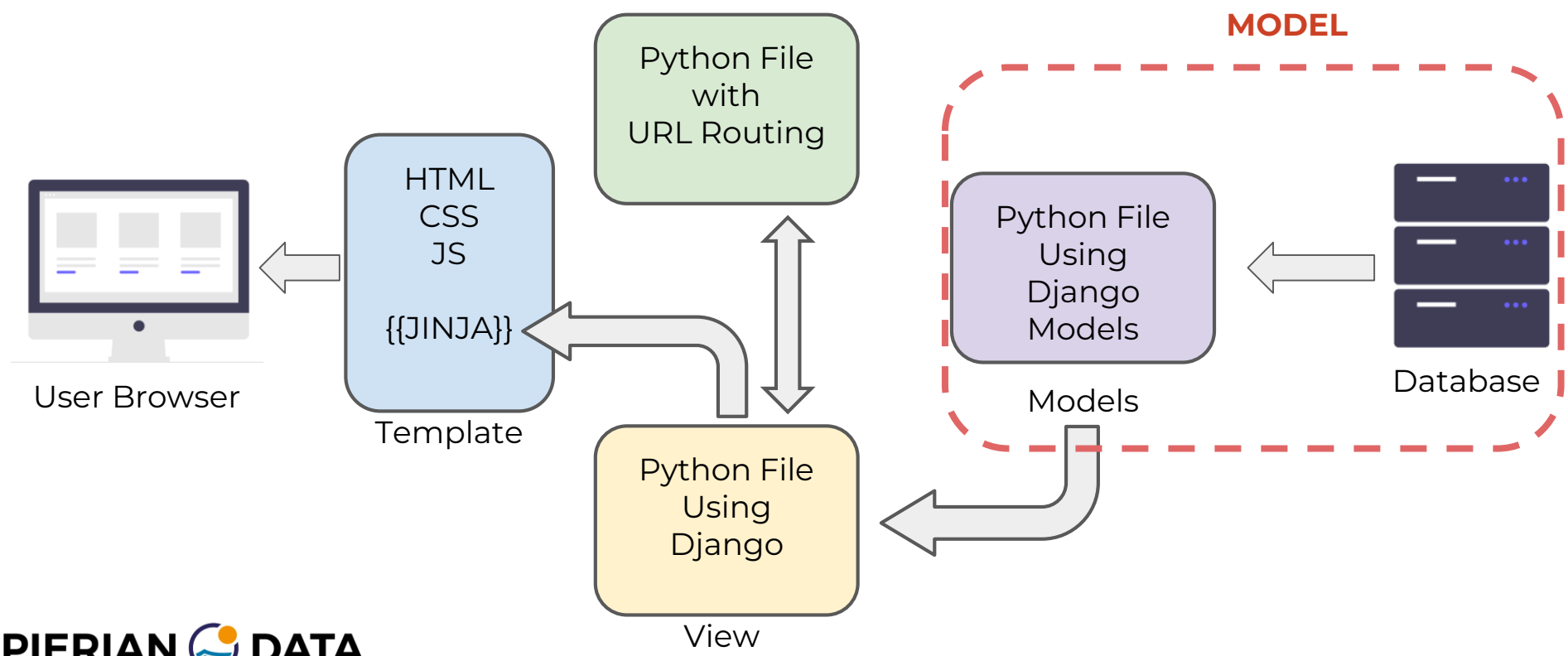

Python File
with
URL Routing
HTML
CSS
JS
{{JINJA}}
Template
User Browser
Python File
Using
Django
Models
Models
Database
Python File
Using
Django
View

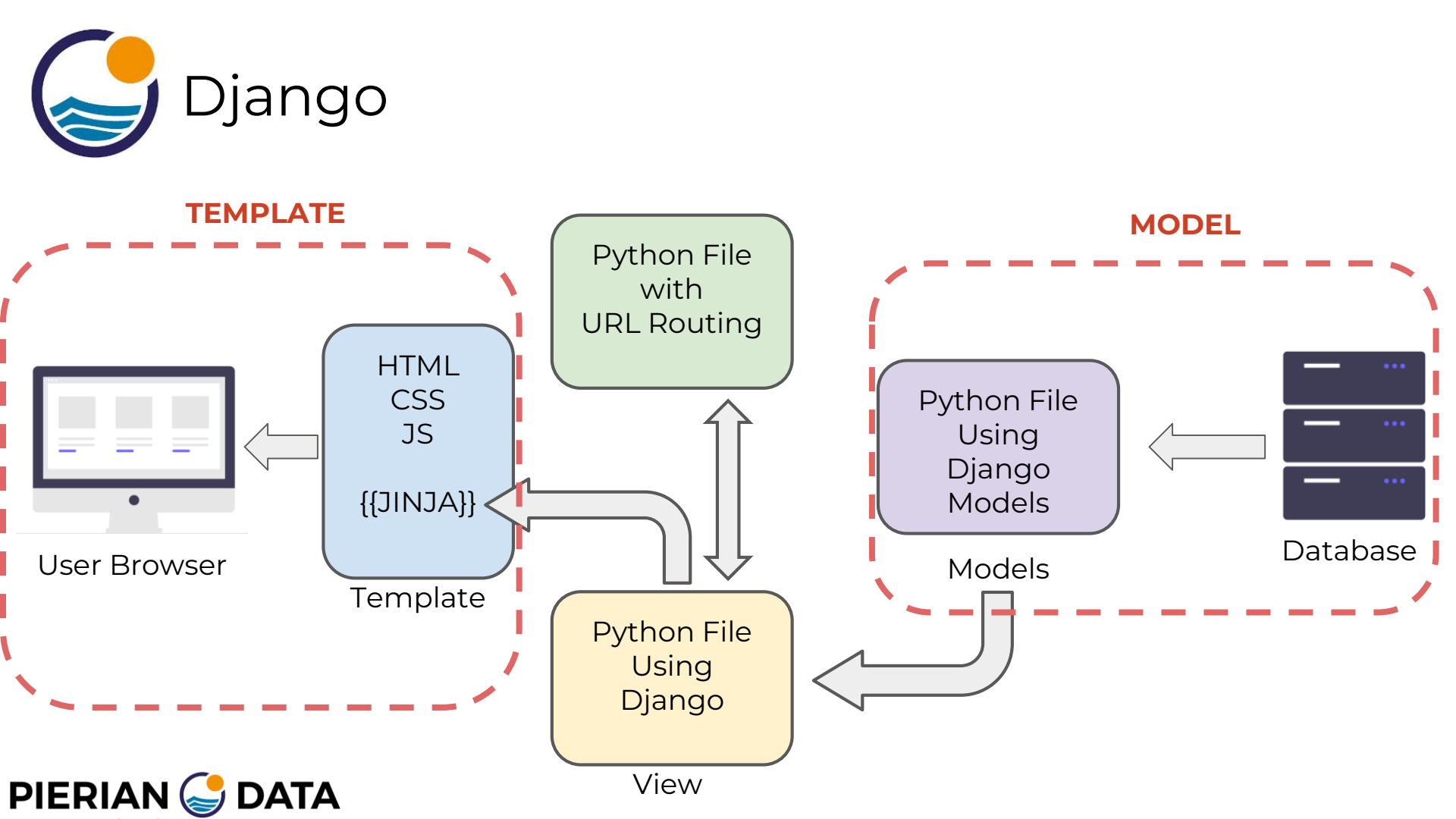
Django

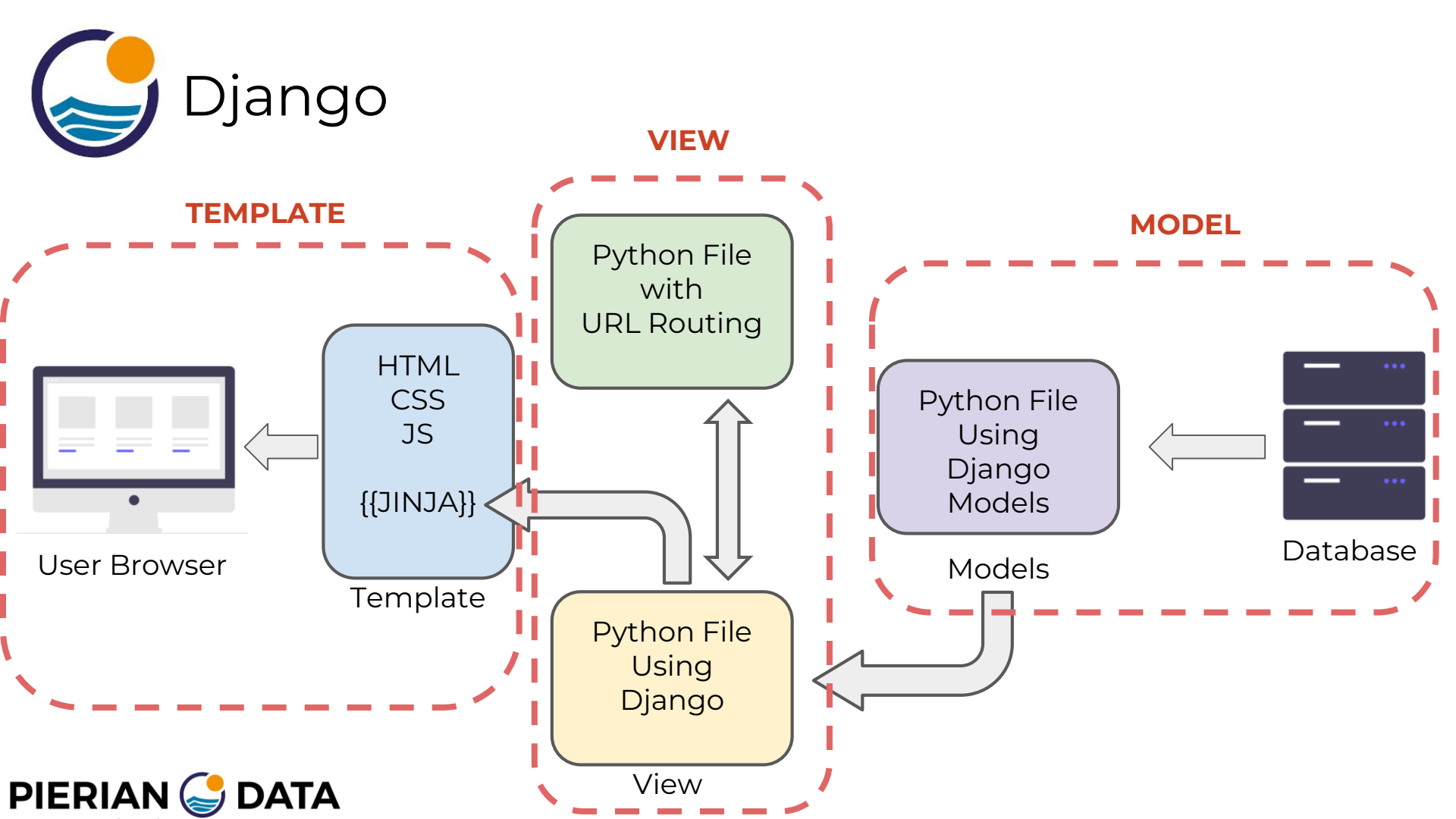


PIERIAN DATA

Django


MODEL
Python File with URL Routing
HTML CSS JS
{{JINJA}}
Template
User Browser
Python File Using Django Models
Models
Database
Python File Using Django
View


Django
TEMPLATE
MODEL
Python File
with
URL Routing
HTML
CSS
JS
{{JINJA}}
Template
User Browser
Python File
Using
Django
Models
Models
Database
Python File
Using
Django
View
PIERIAN DATA


Django
VIEW
TEMPLATE
MODEL
Python File
with
URL Routing
HTML
CSS
JS
{{JINJA}}
Python File
Using
Django
Models
Database
User Browser
Template
Models
Python File
Using
Django
View
PIERIAN DATA

Django

User Browser

HTML
CSS
JS

{{JINJA}}

Template

Python File
with
URL Routing

Python File
Using
Django

View

Python File
Using
Django
Models

Models

Application
Logic
in Python

Database

PIERIAN DATA

Django

User Browser

HTML
CSS
JS

{{JINJA}}

Template

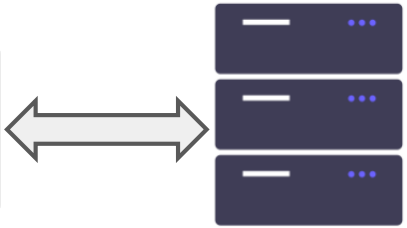
Python File
with
URL Routing

Python File
Using
Django

View

Python File
Using
Django
Models

Models



Database

Application
Logic
in Python

Django

User Browser

HTML
CSS
JS

{{JINJA}}

Template

Python File
with
URL Routing

Python File
Using
Django
Models

Database

Models

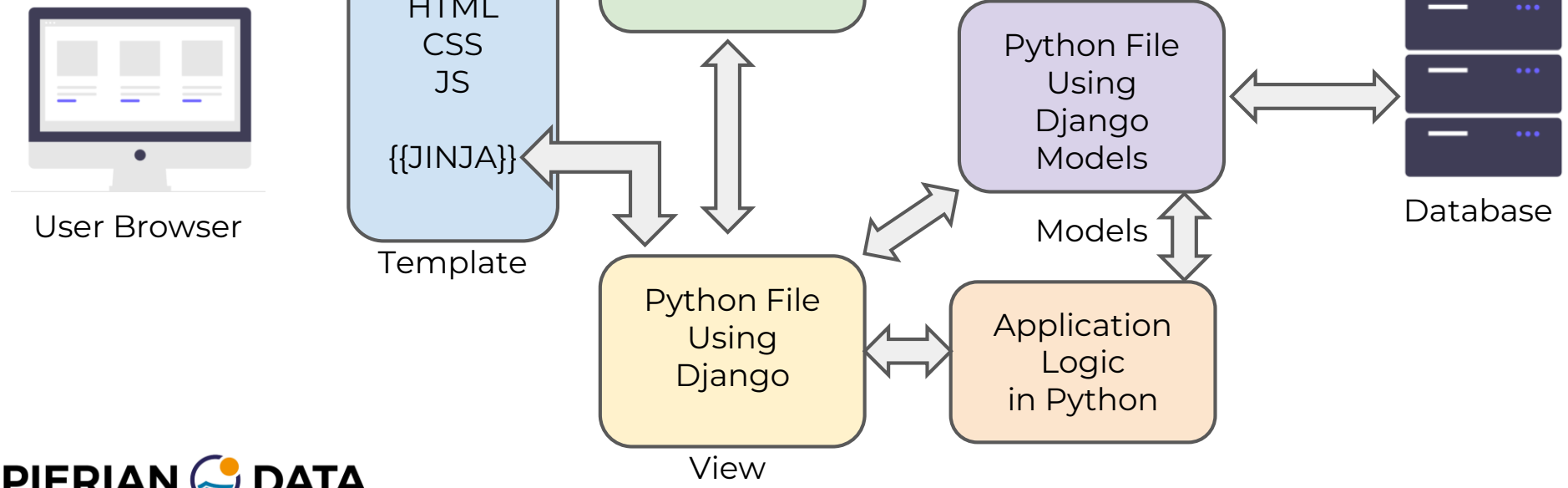
Python File
Using
Django

View

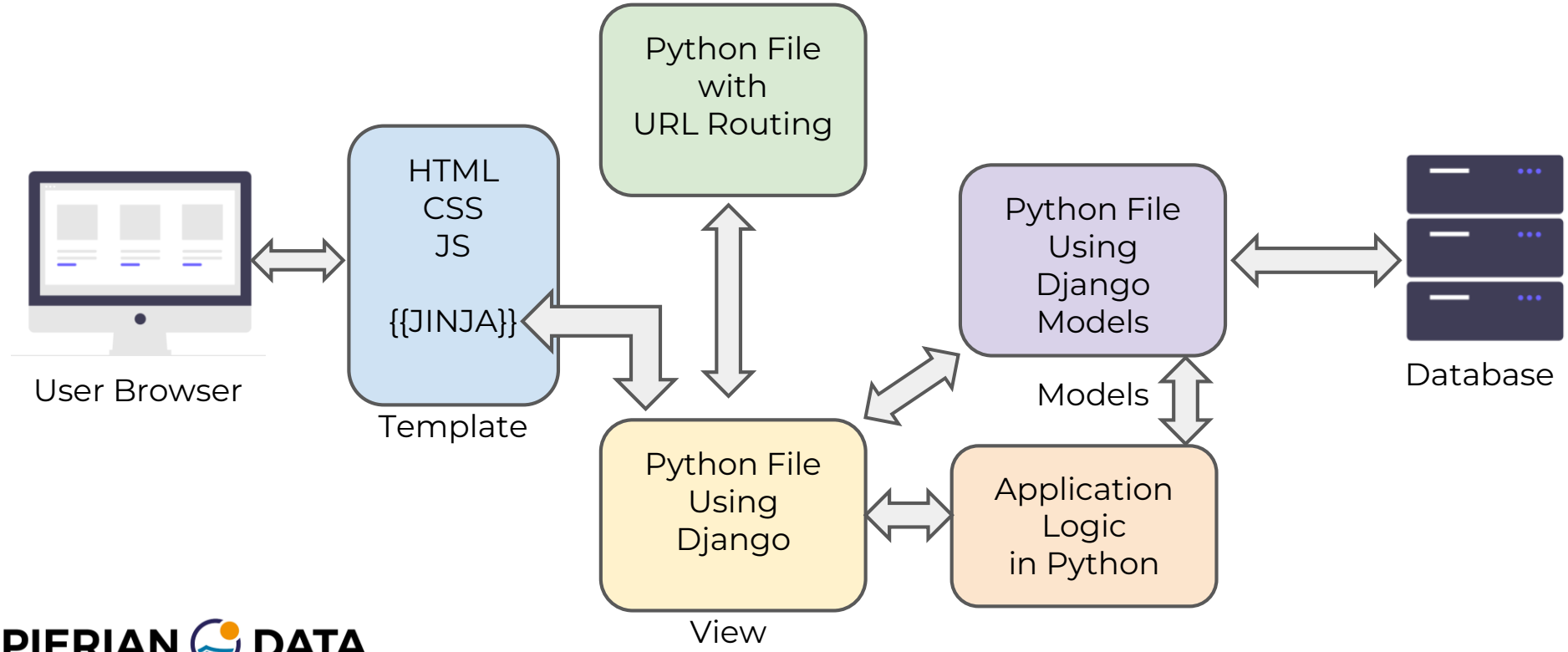
Application
Logic
in Python

PIERIAN DATA

Django


Python File
with
URL Routing
HTML
CSS
JS
{{JINJA}}
Template
User Browser
Python File
Using
Django
Models
Database
Models
Python File
Using
Django
View
Application
Logic
in Python

Django


Python File
with
URL Routing
HTML
CSS
JS
{{JINJA}}
Template
User Browser
Python File
Using
Django
Models
Models
Database
Python File
Using
Django
View
Application
Logic
in Python

Django

- Django structure will also have many more features not shown in this MTV diagram, such as authentication and administration access.

Django

- Django Drawbacks?

Django

- Django Drawbacks:
  - Heavily reliant on idea of Model.
  - The model is a Python/Django representation of a table in a database.
  - This makes it very easy to work with querying data, but does add the requirement of understanding Models and setting them up for views.

Django

- Let’s continue our understanding of Django by beginning to learn about these key components separately and then slowly adding on complexity to a fully functioning website!

First Django Project

PIERIAN DATA

Django

- Let’s explore how to launch a project with Django.
- This is done easily at the command prompt with the django-admin tool.
- Django will automatically create a set of subdirectories and files for us.

Django

- At the command prompt, navigate to your desired location and type:
  - **django-admin startproject my_site**

Django

- At the command prompt, navigate to your desired location and type:
  - **django-admin startproject my_site**

Special command installed by django. Many sub-commands can be called from django-admin.

Django

- At the command prompt, navigate to your desired location and type:
  - **django-admin startproject my_site**

Subcommand to create a new project directory. Automatically creates a set of directories used in most Django projects.

Django

- At the command prompt, navigate to your desired location and type:
  - **django-admin startproject my_site**

Your chosen name for the project and top level project directory. Feel free to call this anything you want, but use common sense!

Django

- This creates the following files and folders:
  - **my_site**
    - **my_site**
    - *manage.py*

Django

- This creates the following files and folders:
  - **my_site**
    - **my_site**
    - *manage.py*

Django

- This creates the following files and folders:
  - **my_site**
    - **my_site**
    - *manage.py*

Django

- This creates the following files and folders:
  - **my_site**
    - **my_site**
    - *manage.py*

Django

- **my_site**
  - **my_site**
    - *__init__.py*
    - *settings.py*
    - *urls.py*
    - *asgi.py*
    - *wsgi.py*
  - *manage.py*

Django

- **my_site**
  - **my_site**
    - *__init__.py*
    - *settings.py*
    - *urls.py*
    - *asgi.py*
    - *wsgi.py*
  - *manage.py*

Django

- **my_site**
  - **my_site**
    - *__init__.py*
    - *settings.py*
    - *urls.py*
    - *asgi.py*
    - *wsgi.py*
  - *manage.py*

Django

- **my_site**
  - **my_site**
    - *__init__.py*
    - *settings.py*
    - *urls.py*
    - *asgi.py*
    - *wsgi.py*
  - *manage.py*

Django

- **my_site**
  - **my_site**
    - *__init__.py*
    - *settings.py*
    - *urls.py*
    - *asgi.py*
    - *wsgi.py*
  - *manage.py*

PIERIAN DATA

Django

- Let’s run this ourselves and create our first Django Project!

Django Applications

Django

- Django Projects can have separated components called “apps”.
- *Don’t get confused by this nomenclature!*
  - Typically a “web app” describes the full website or mobile application on the web.
  - A “Django app” is a sub-component of a single Django Project (web application).

Django

- Often it becomes much easier to organize your code through the use of apps.
- Each app should cover a different key functionality for your website.
- Also keep in mind that if you’re beginning as a solo developer with a simple website, it may make more sense to put everything under a single Django app.

Django

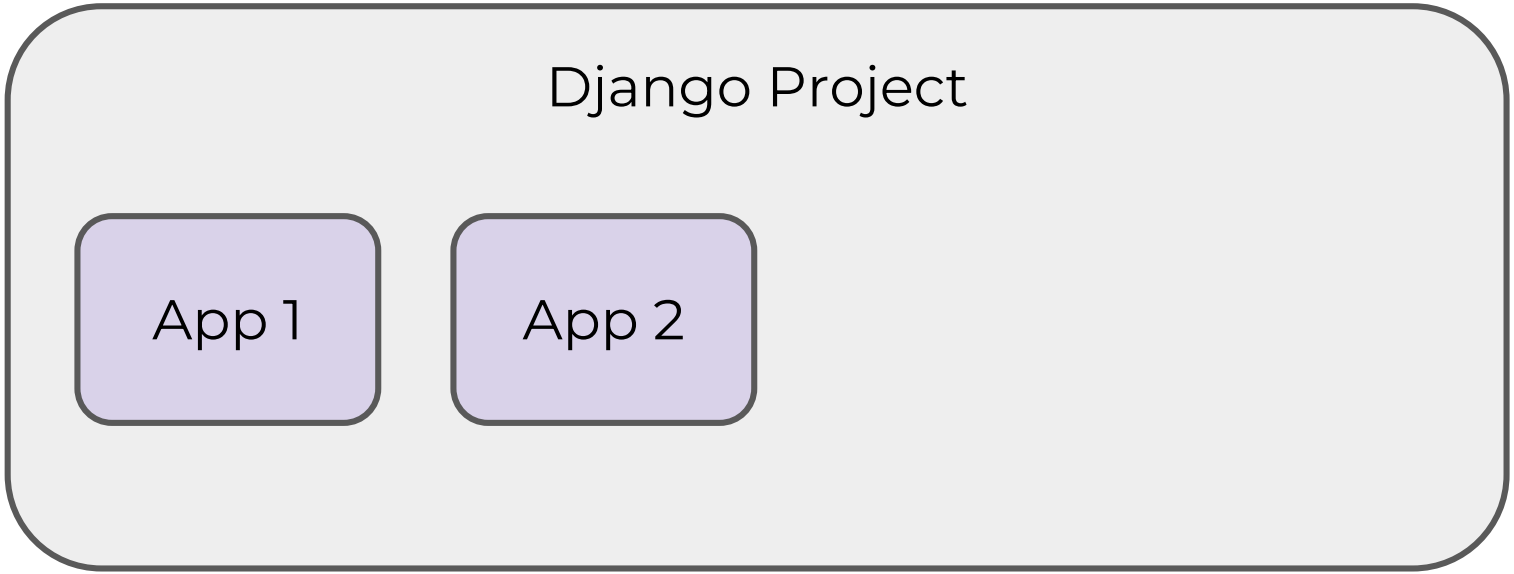
- Django Project Structure

Django

- Django Project Structure

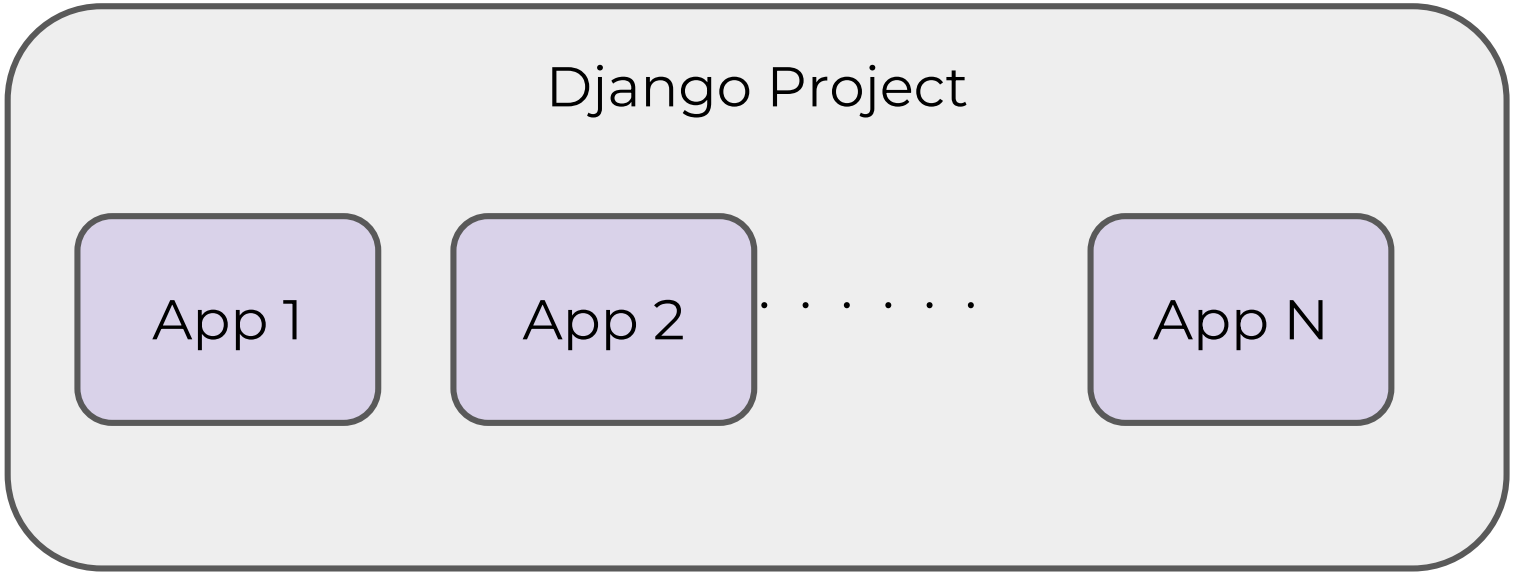
Django

- Django Project Structure



Django

- Django Project Structure



PIERIAN DATA

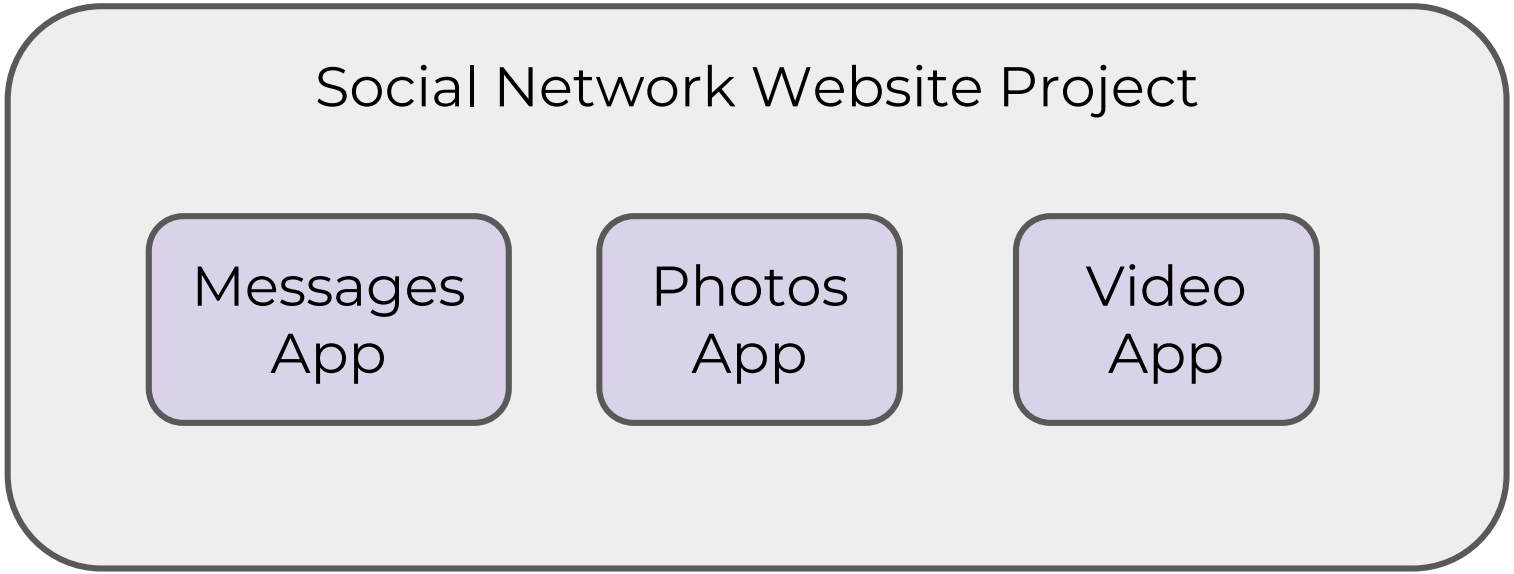
Django

- Django Project Structure

Social Network Website Project

Django

- Django Project Structure



PIERIAN DATA

Django

- To create a new app, we use the manage.py file created by the project:
  - **python manage.py startapp app_name**

Calls the manage.py file to run a command line execution.

Django

- To create a new app, we use the manage.py file created by the project:
  - **python manage.py startapp app_name**

Specific command to run from manage.py file.

PIERIAN DATA

Django

- To create a new app, we use the manage.py file created by the project:
  - **python manage.py startapp app_name**

Custom name for app, feel free to change so its relevant to scope of application.

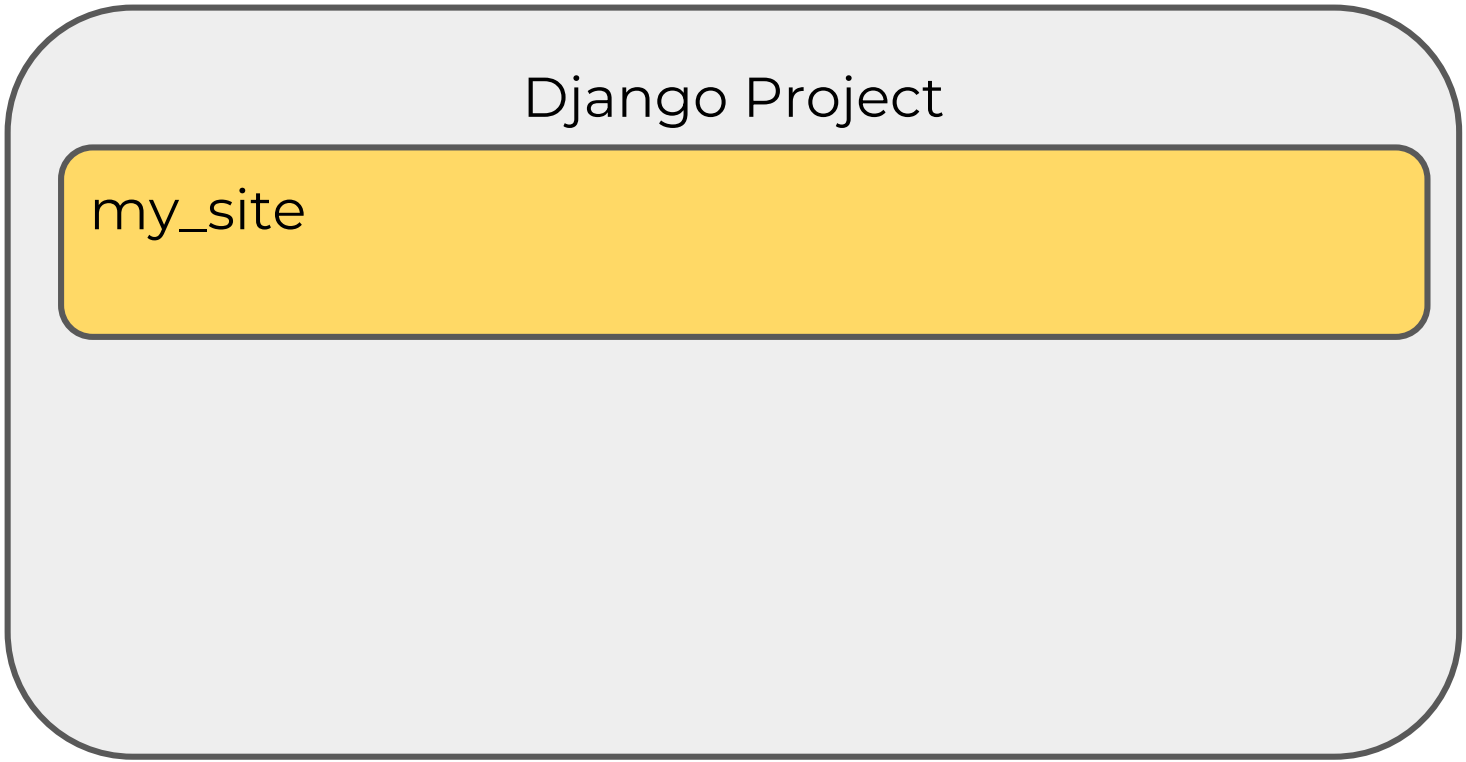
Django

- Let’s create an example app and explore how to connect it to a URL view.
- That will conclude this section and we’ll revisit all of this in the new section by starting a new project from scratch.

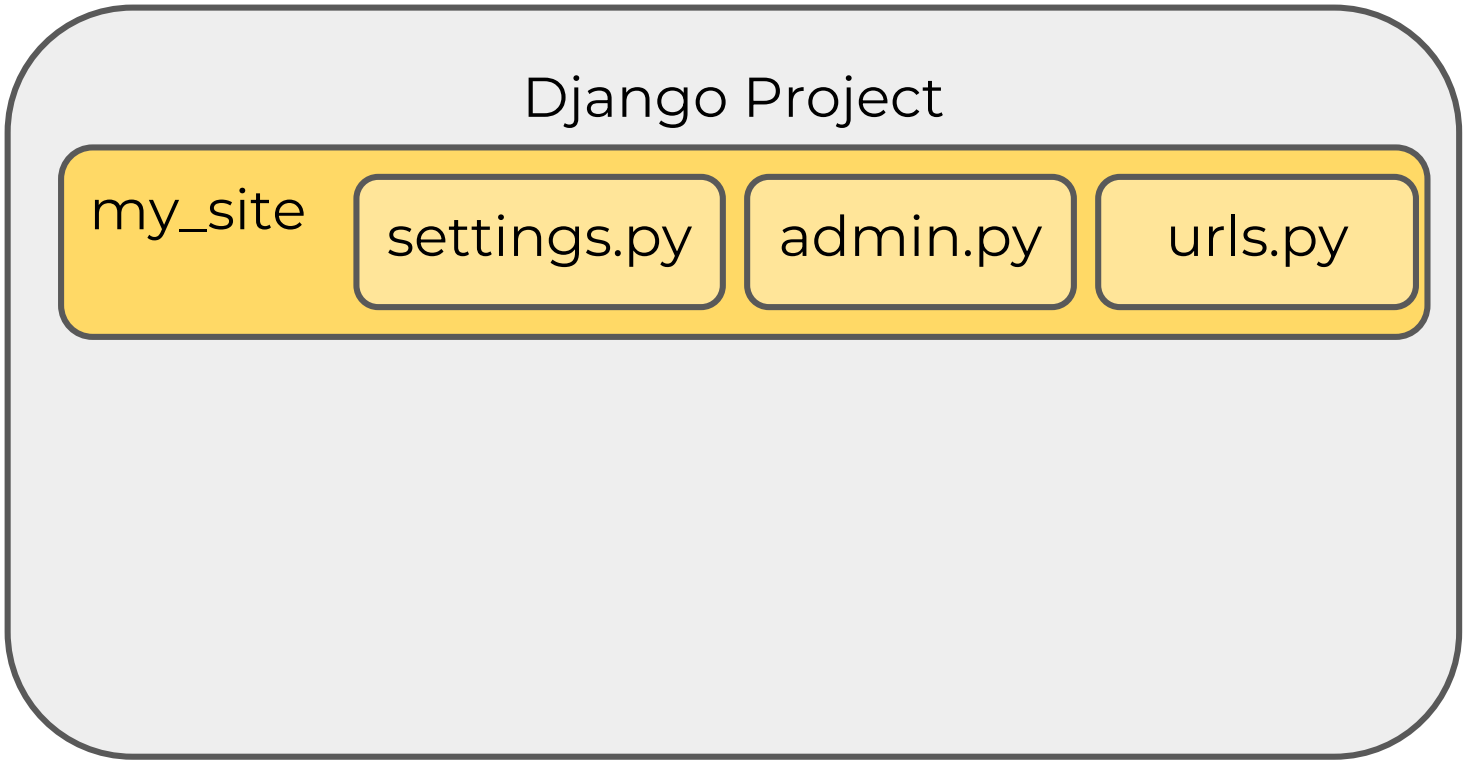
Django

- To create a new app, we use the manage.py file created by the project:
  - **python manage.py startapp app_name**

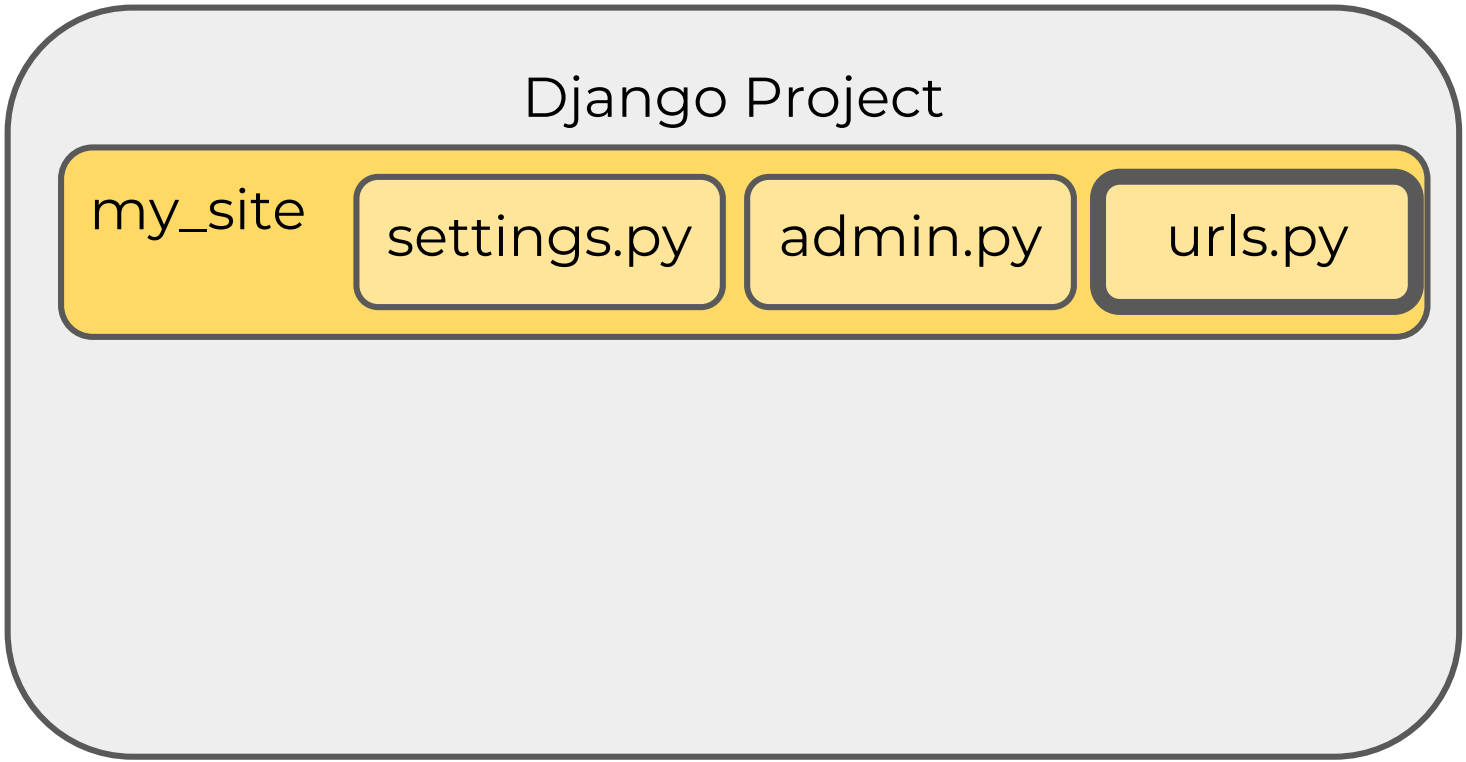
Django



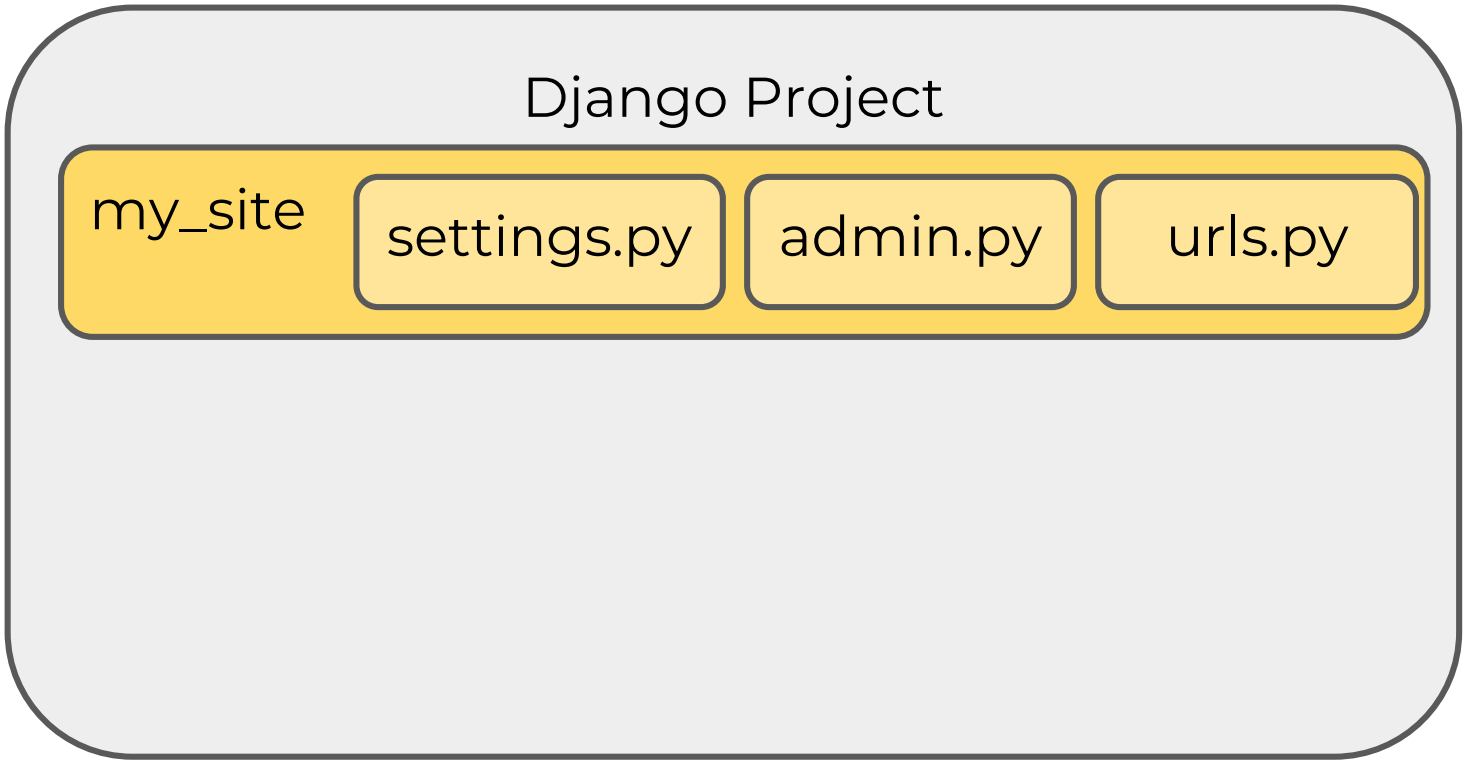
Django



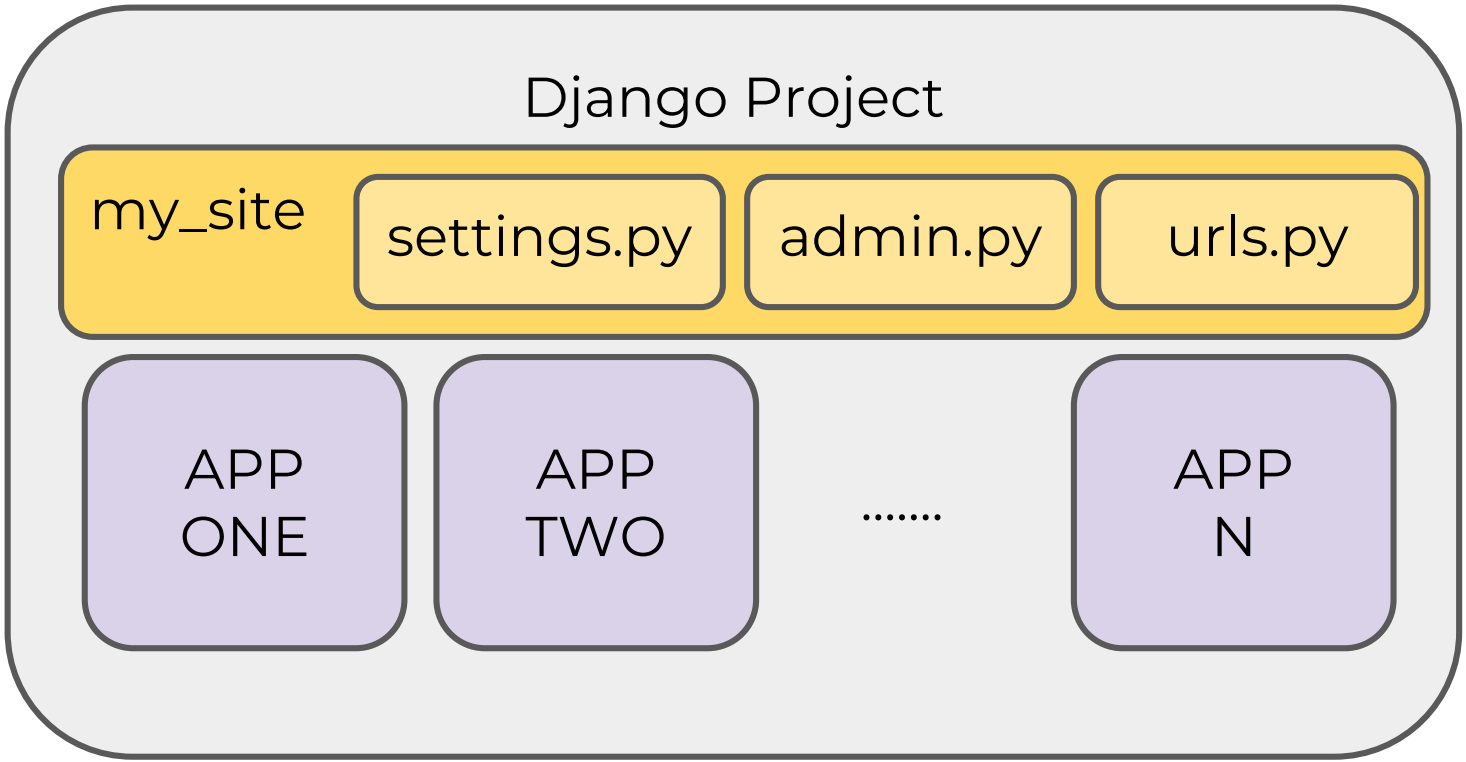
Django



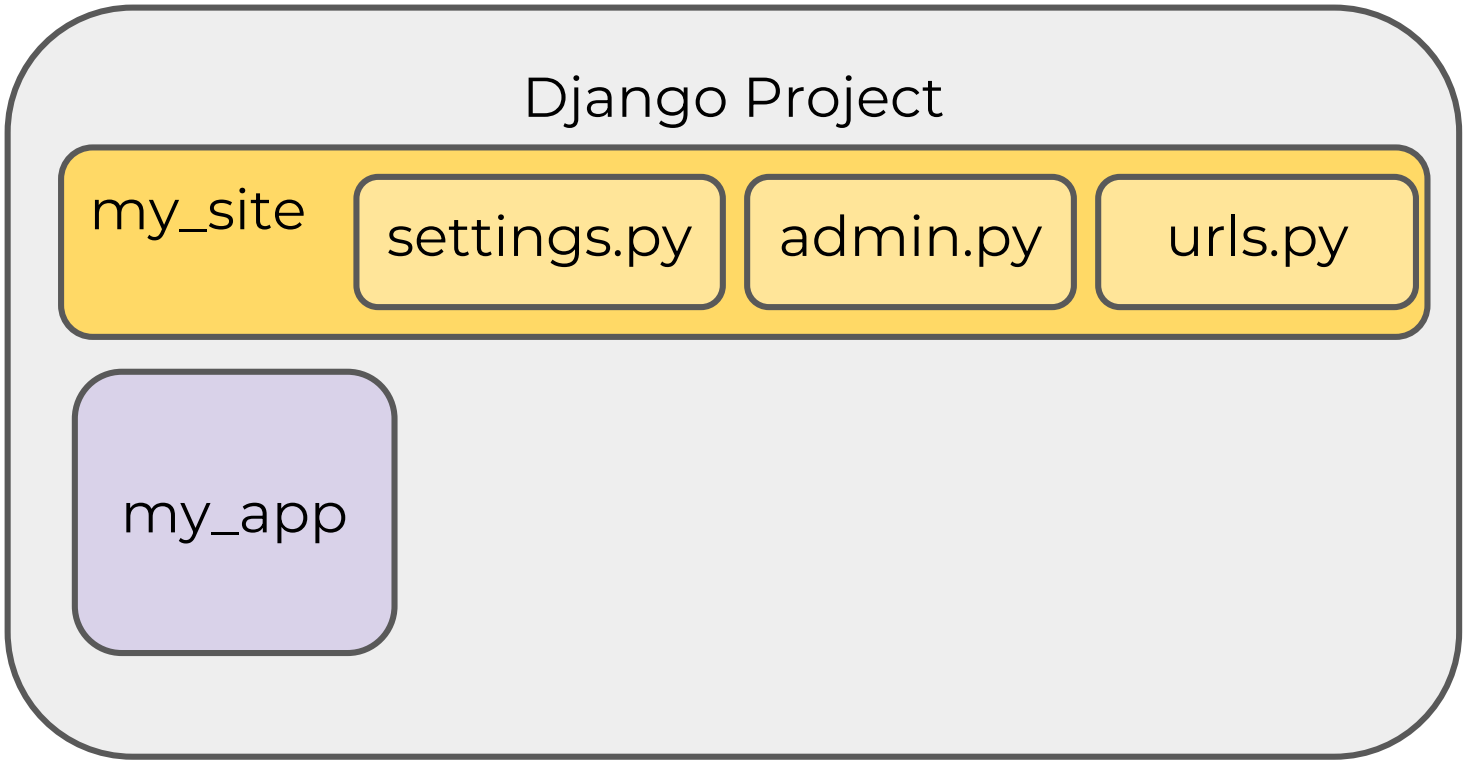
Django



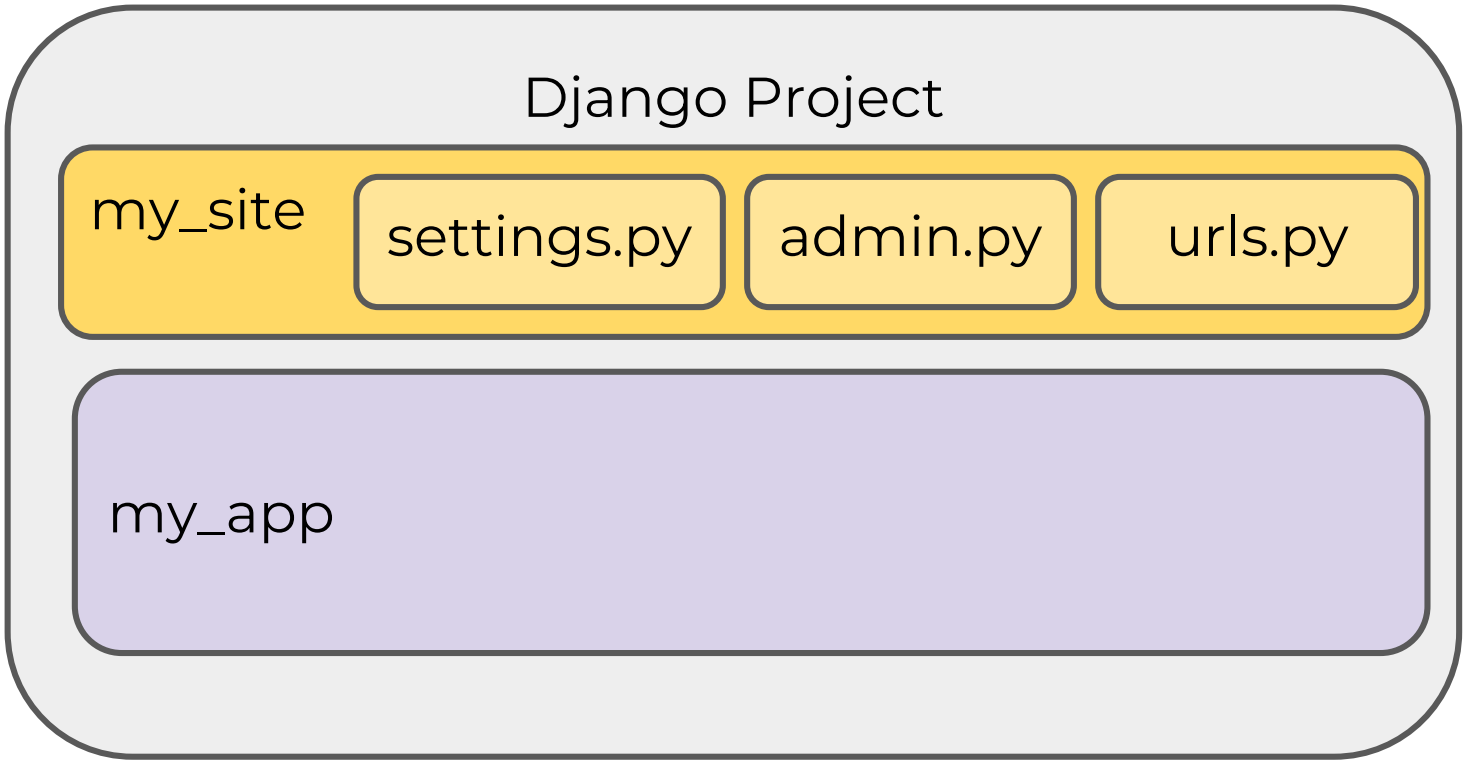
Django


Django Project
my_site
settings.py
admin.py
urls.py
APP ONE
APP TWO
.....
APP N

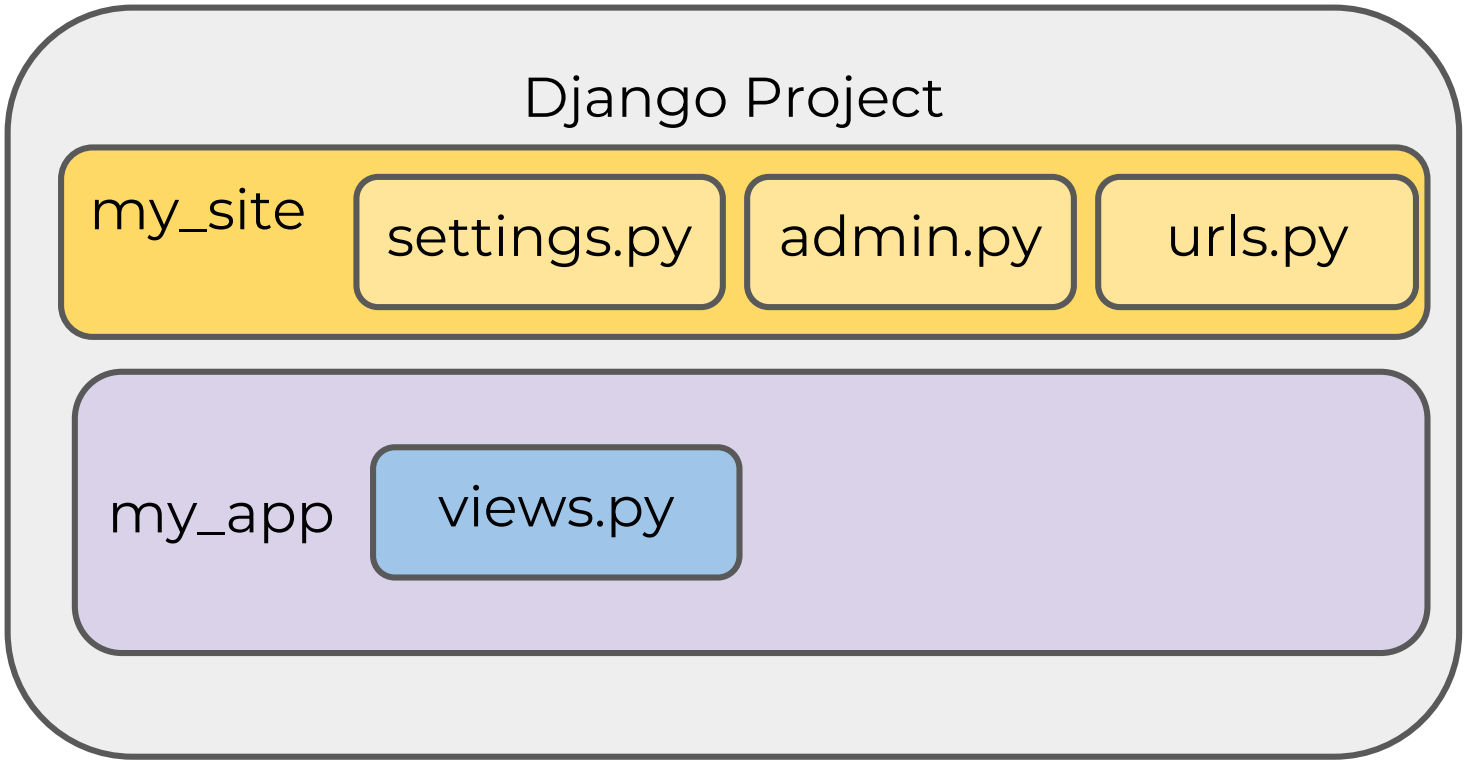
Django



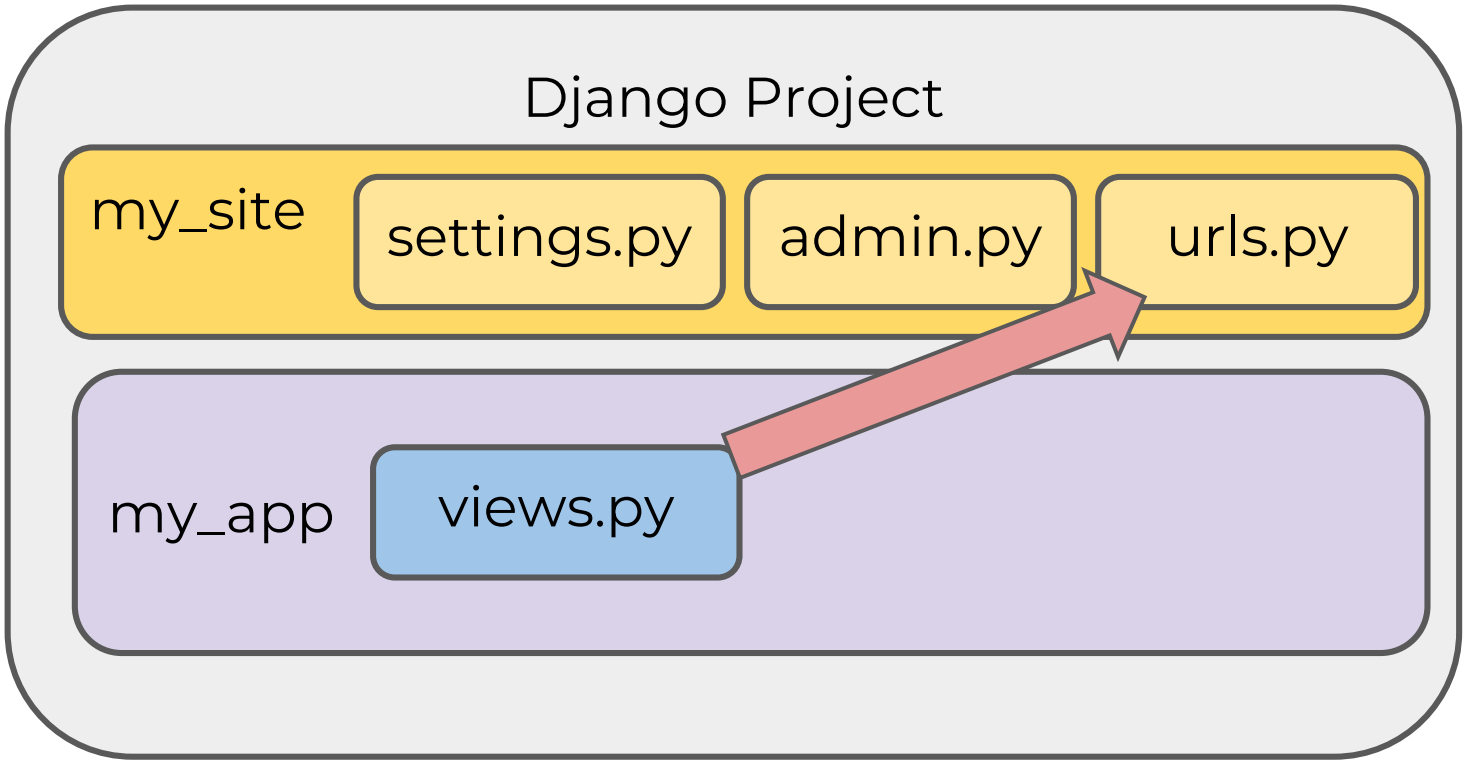
Django



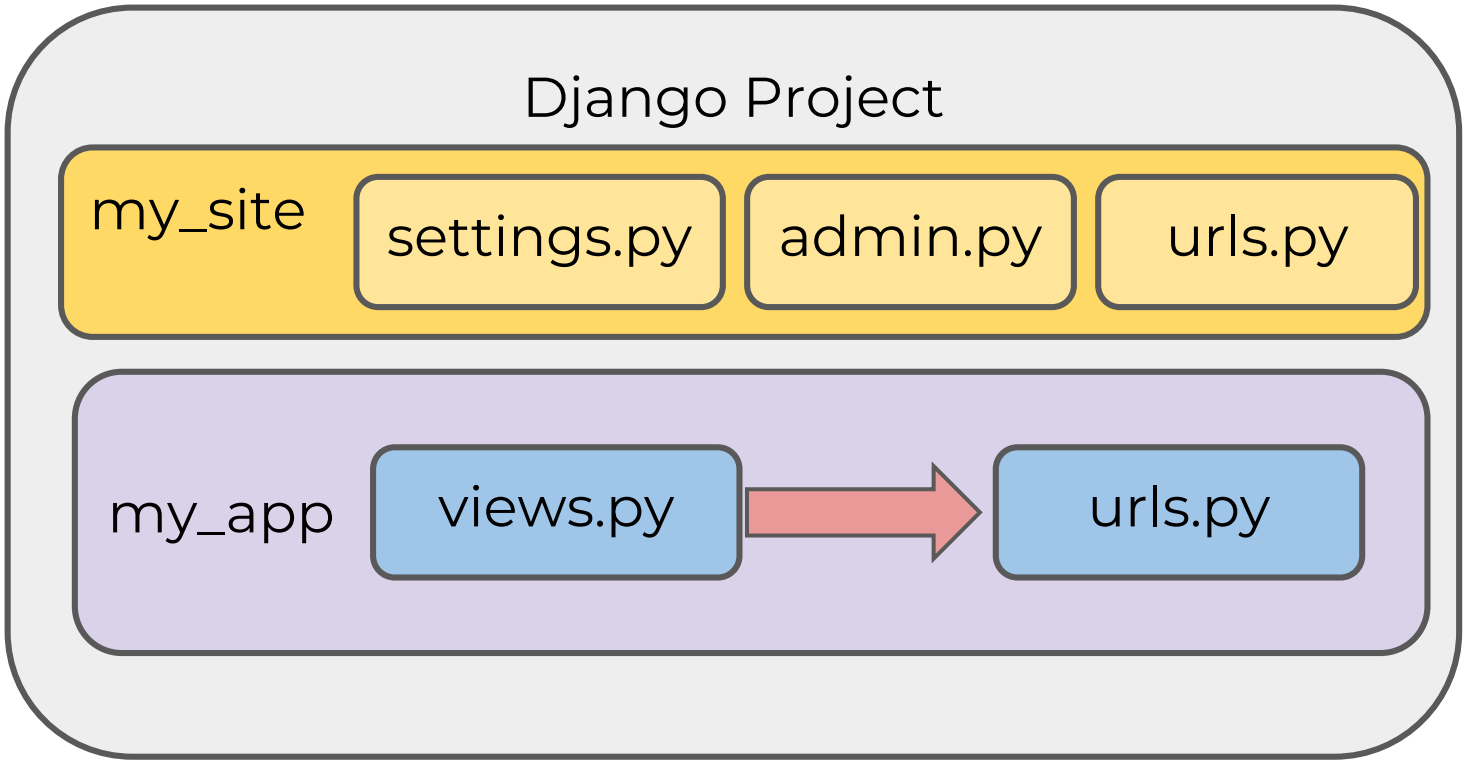
Django


Django Project
my_site
settings.py
admin.py
urls.py
my_app
views.py

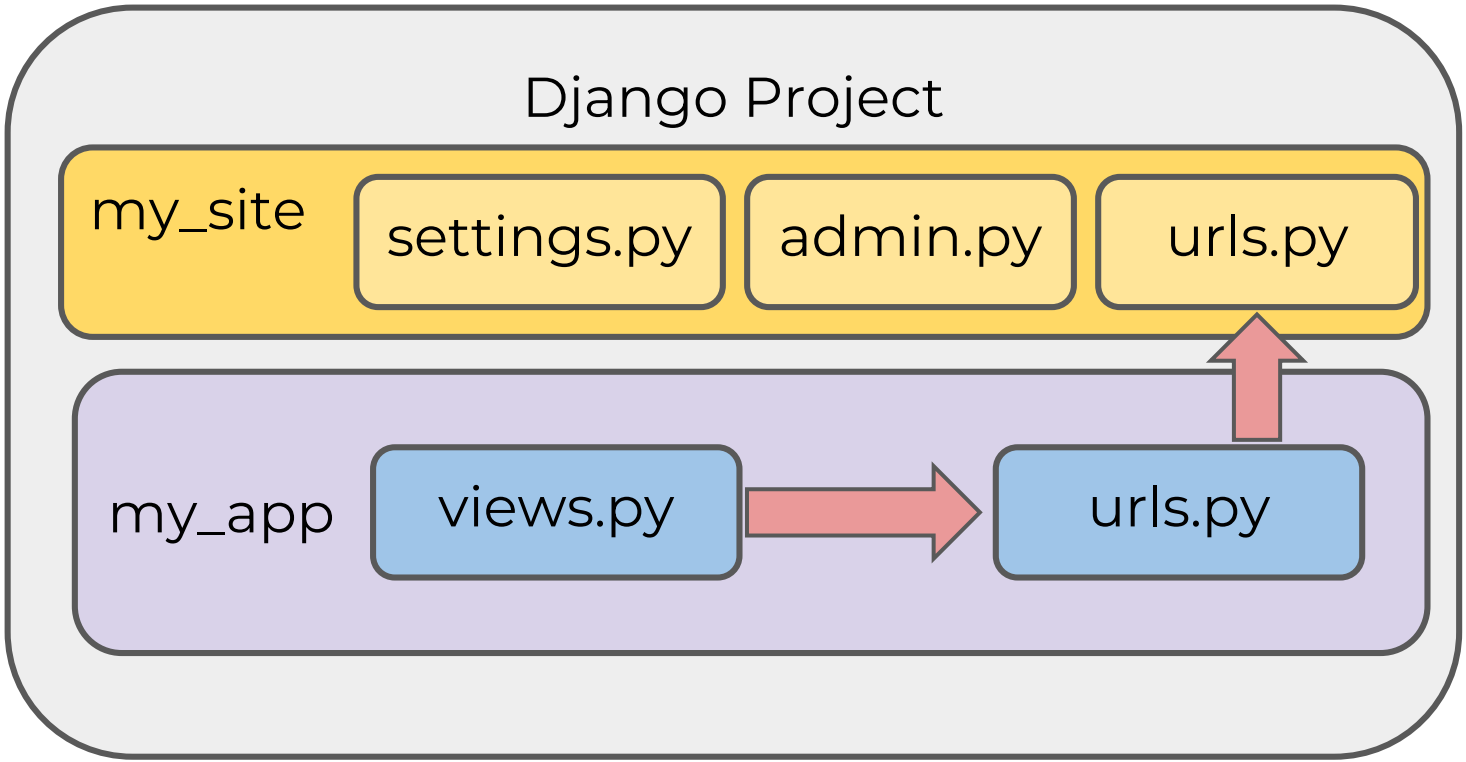
Django


Django Project
my_site
settings.py
admin.py
urls.py
my_app
views.py

Django


Django Project
my_site
settings.py
admin.py
urls.py
my_app
views.py
urls.py

Django


Django Project
my_site
settings.py
admin.py
urls.py
my_app
views.py
urls.py