



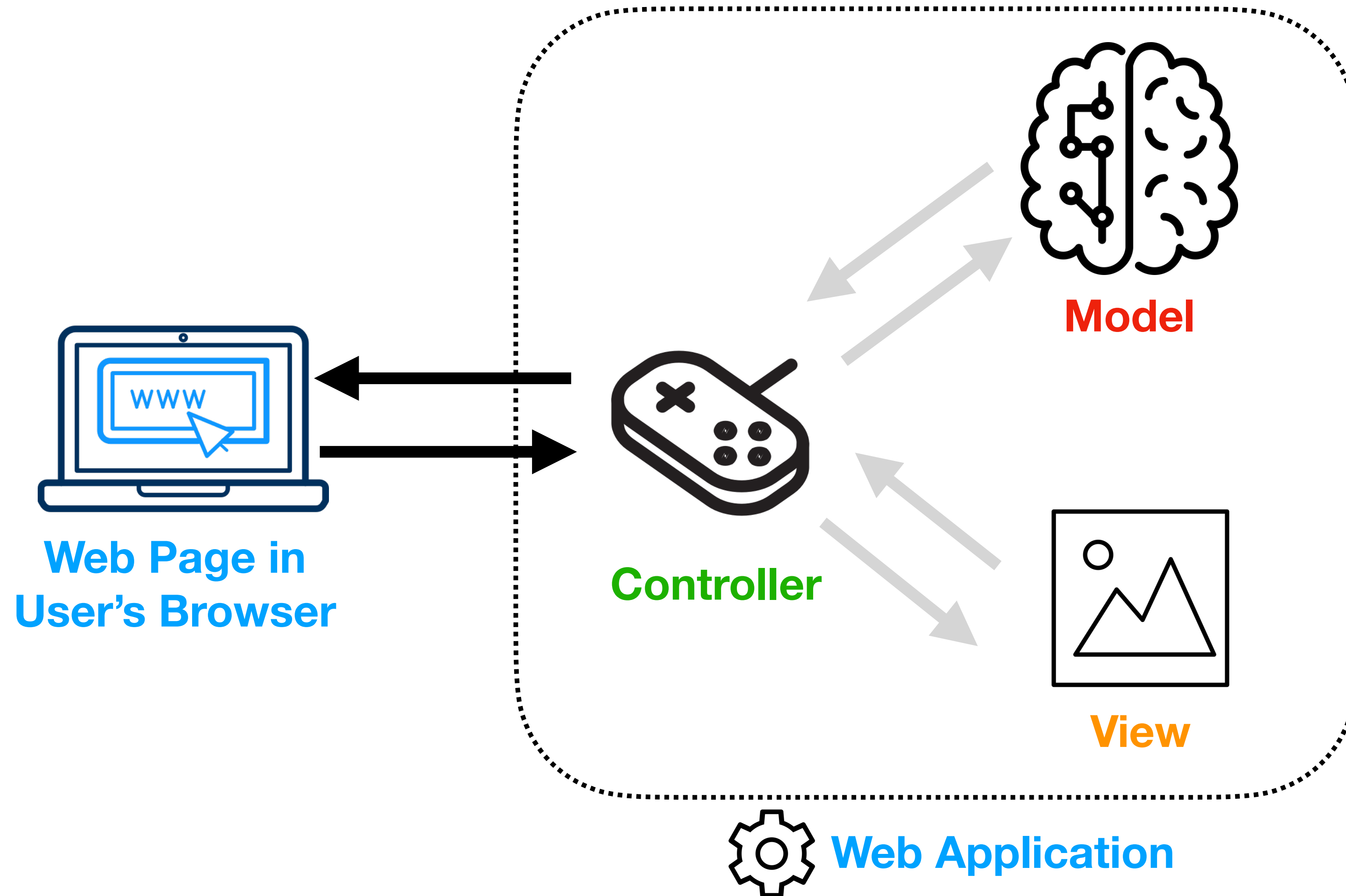
University of
Sheffield

COM1001 *Introduction to Software Engineering* SPRING
SEMESTER

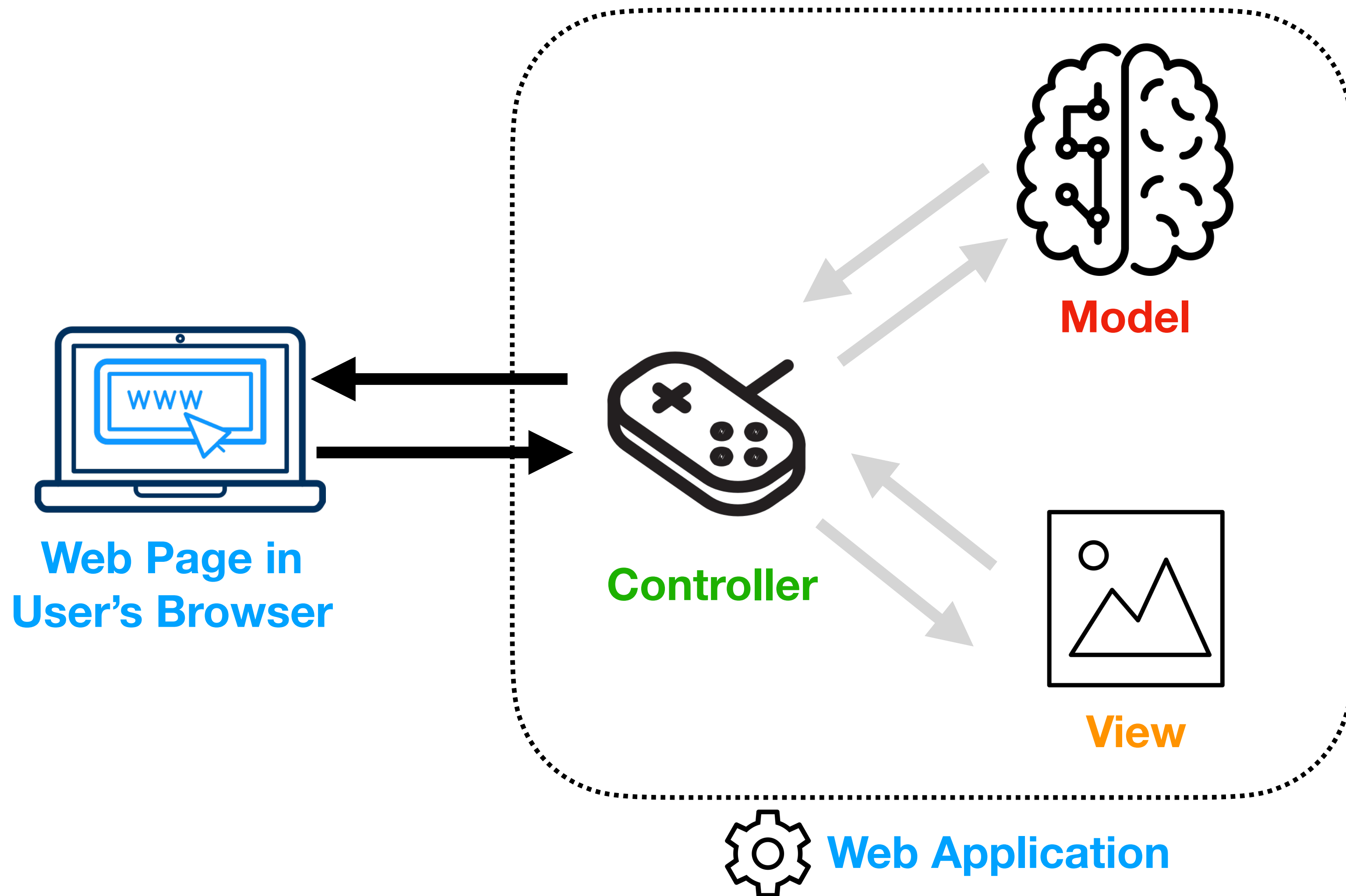
The Web Application Architecture: The Model-View-Controller (MVC) Design Pattern

Professor Phil McMinn

Model View Controller (MVC)



Model View Controller (MVC)

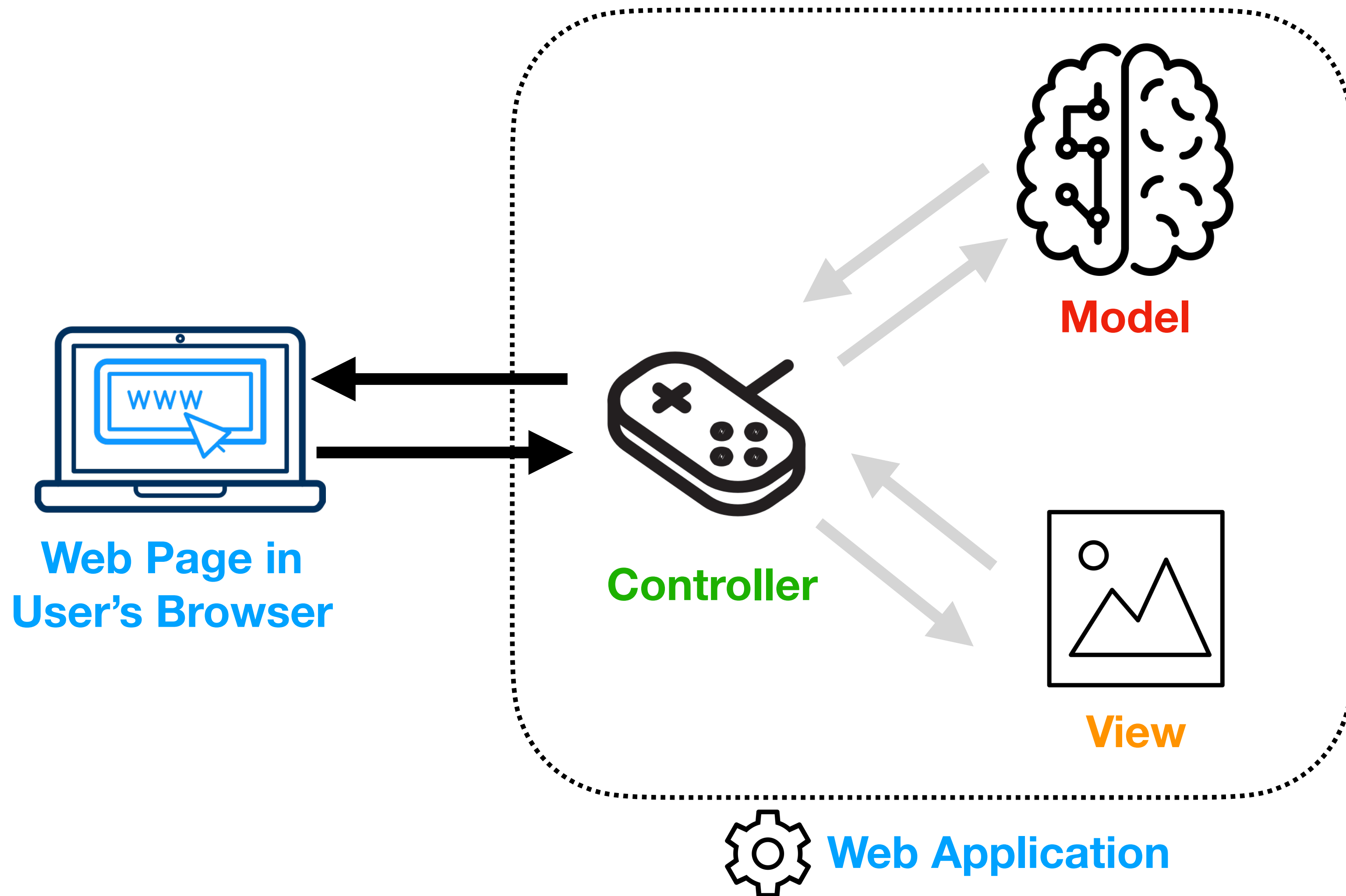


Models encode application-specific logic and manage data (in a persistent data store such as a database).

They form the “brain” of the web app.

(We won't talk about models much more in this lecture – but we'll be coming back to them later in the module.)

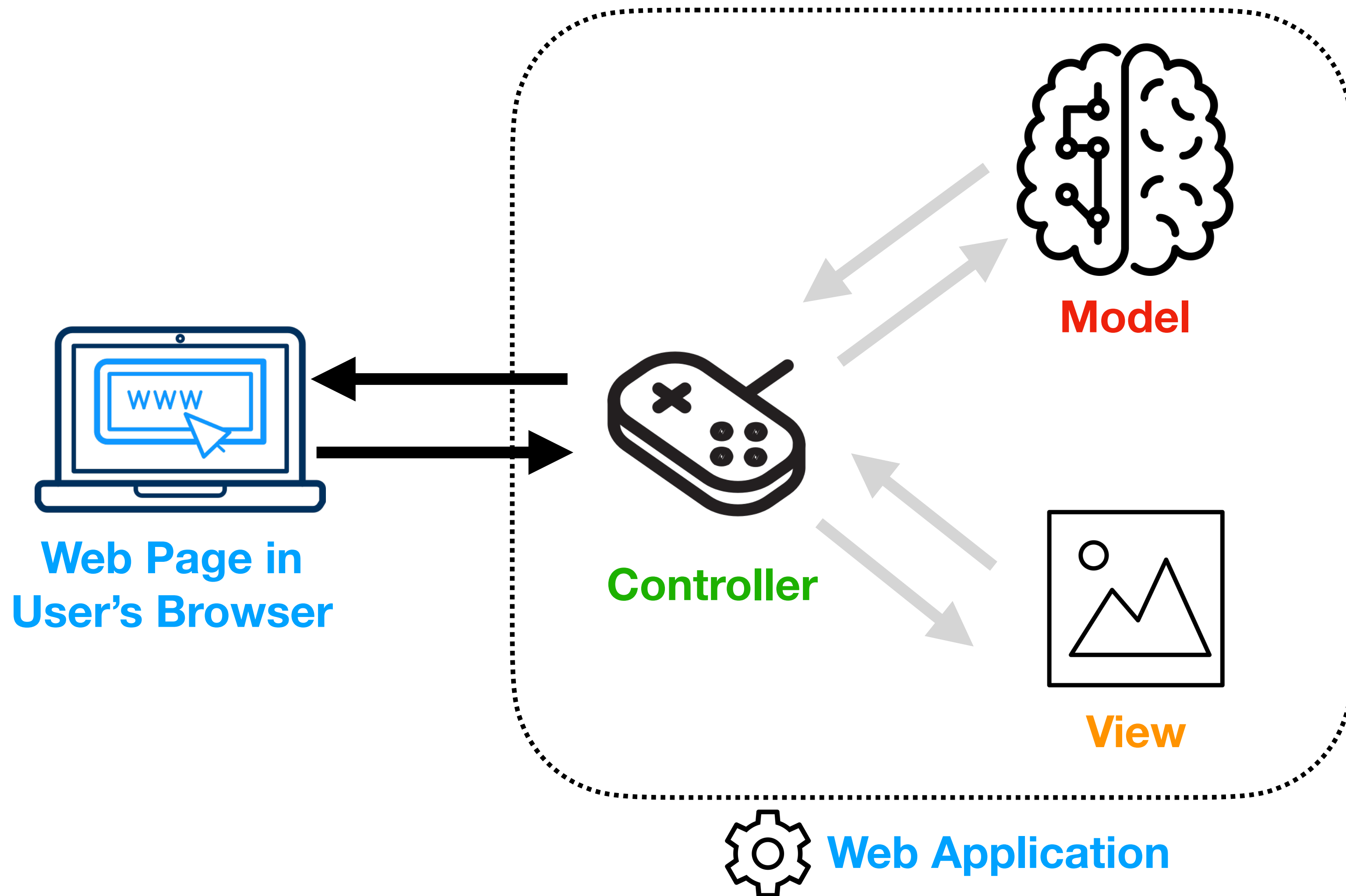
Model View Controller (MVC)



Views contain the code for rendering the front-end of the application, along with any presentation logic.

(We met Sinatra Views in the last lecture.)

Model View Controller (MVC)



Controllers take inputs (**HTTP requests**) and convert them to outputs (**HTTP responses, i.e. web pages**) by using models and views.

We've already met controllers in the form of **routes** that return strings from Sinatra blocks.

Why use MVC?

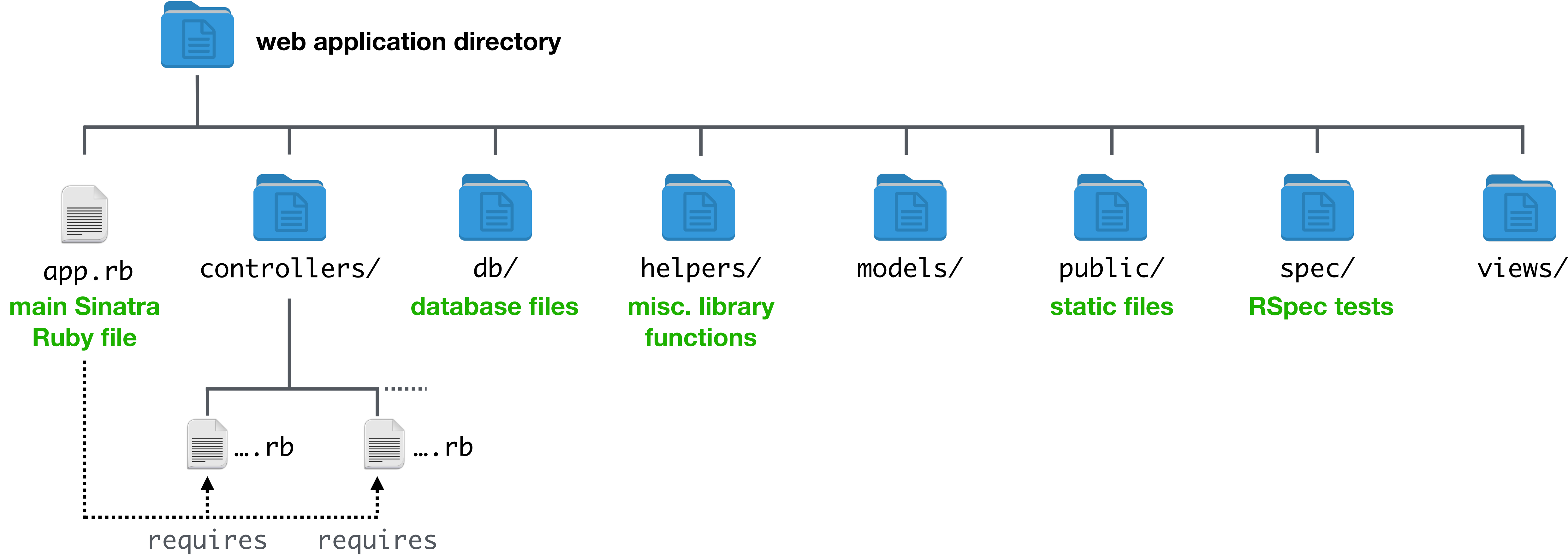
Sinatra does not enforce the use of MVC, **however it is good practice.**

MVC is based on the software engineering principle called **“separation of concerns”**.

Unrelated code is not jumbled up, so that each individual **code unit** (method, file) is more:

- **Cohesive**. Each unit does fewer things, more simply.
- **Modular and reusable**. We don't have to re-write the same logic several times in different places in our application, obeying the software engineering maxim **“don't repeat yourself”**.
- **Easier to maintain and understand**. We know where a piece of code should live, and we can get to know what it does more quickly.

Suggested File Structure of a Sinatra Application



The Main Sinatra File as an “Entry Point”

```
require "sinatra"
require "require_all"
require_rel "controllers"
```

controllers/multi-controller/app.rb

(main Sinatra Ruby file run from the terminal)

Loads all the Ruby files in a specified directory relative to the current Ruby file. Needs the “`require_all`” gem.

```
get "/" do
  # ...
end

get "/list-products" do
  # ...
end
```

controllers/multi-controller/
controllers/user_facing.rb

```
get "/admin/login" do
  # ...
end

get "/admin/logout" do
  # ...
end
```

controllers/multi-controller/
controllers/admin_system.rb