Design Pattern, MVC

Sergio Revilla
@elreplicante

based on Martin Tschischauskas (@martynasc) slides

Software Design Patterns

- general reusable solution
- to commonly occurring problems
- description or template to solve a problem

MVC - Pattern

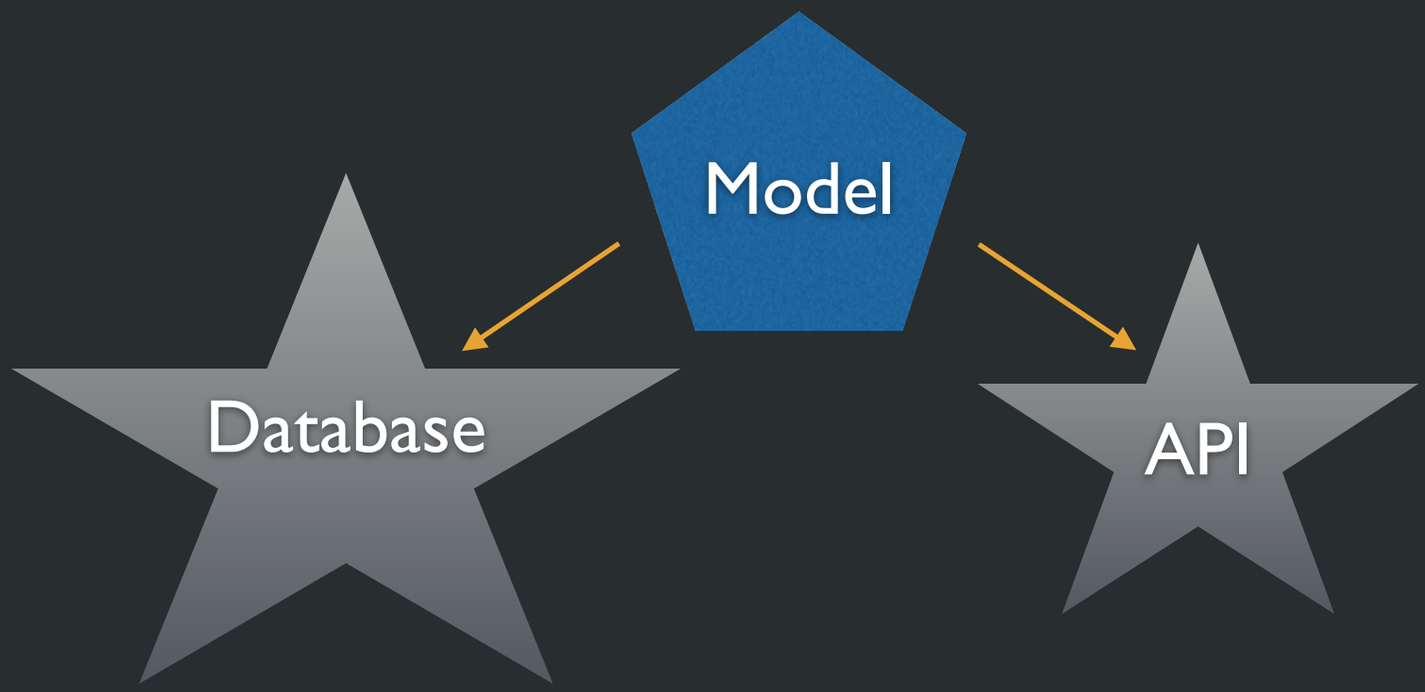
- Model
- View
- Controller

IRON HACK

Model

IRON HACK

Internal Representation of Logic



- application data
- business rules
- logic
- functions

IRON HACK

View

any output representation

View

- requests information from the model
- generates output representation

IRON HACK

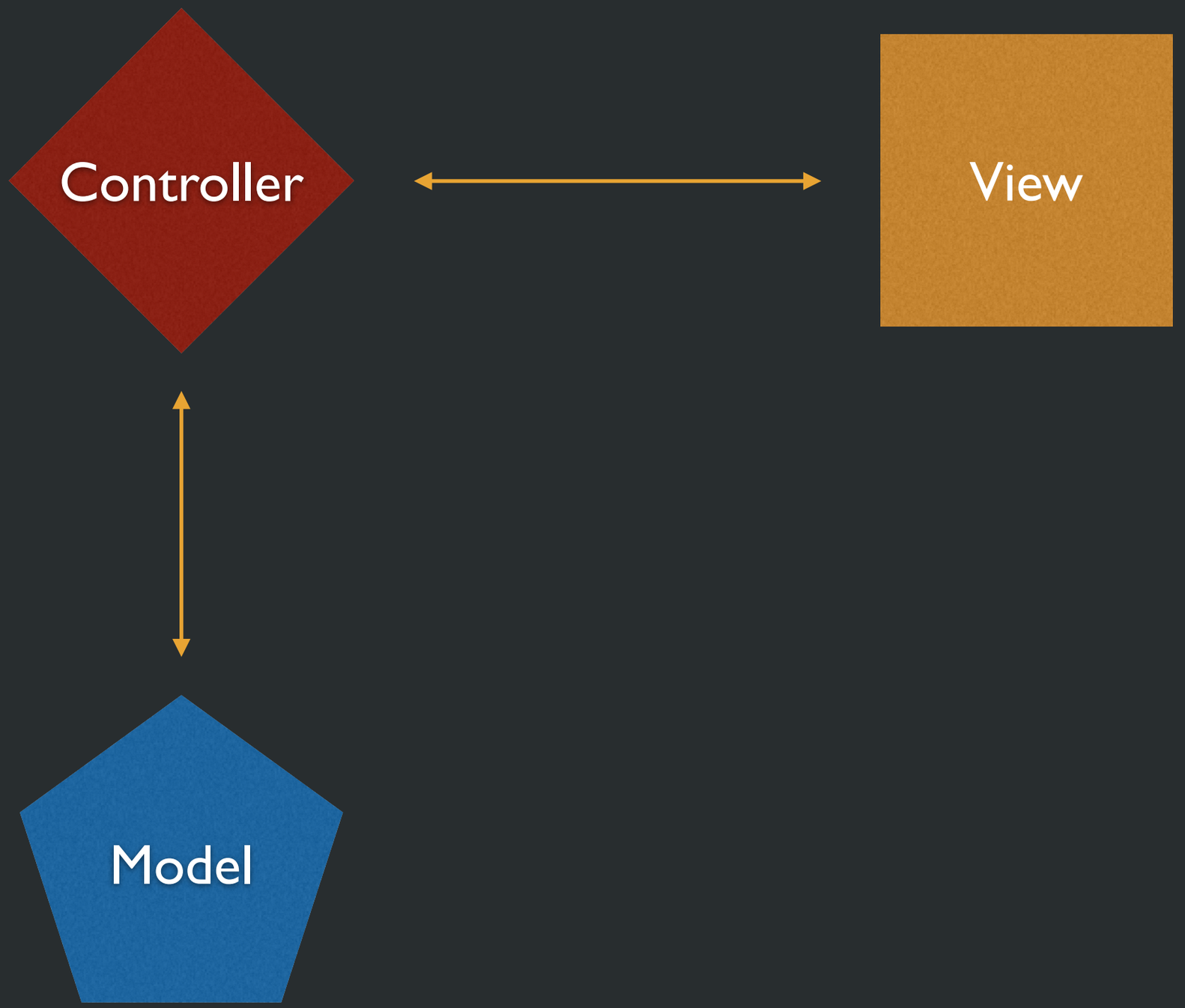
Controller

IRON HACK

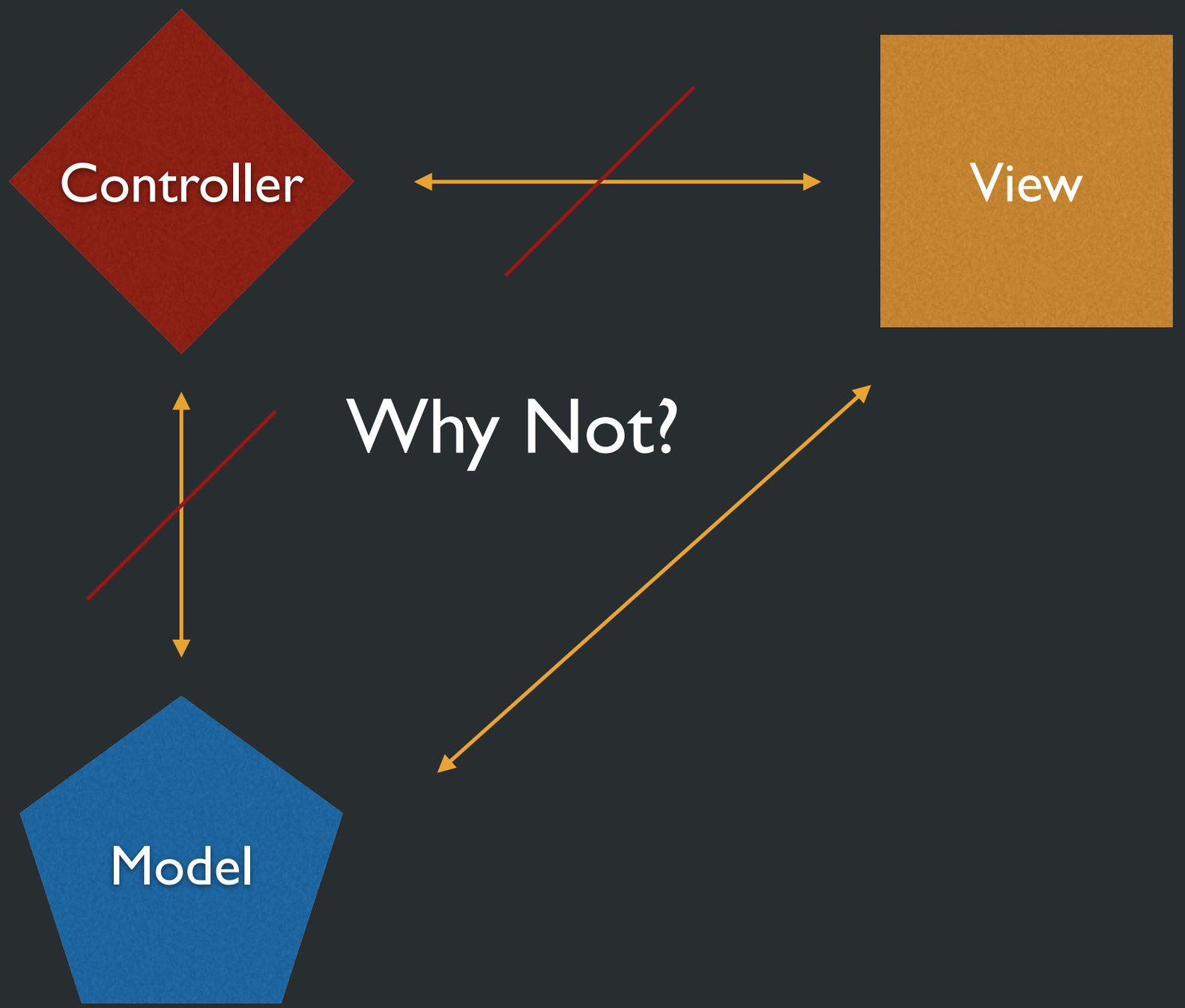
interaction between model and view

- accepts input
- converts it to commands
- for the model or view

IRON HACK



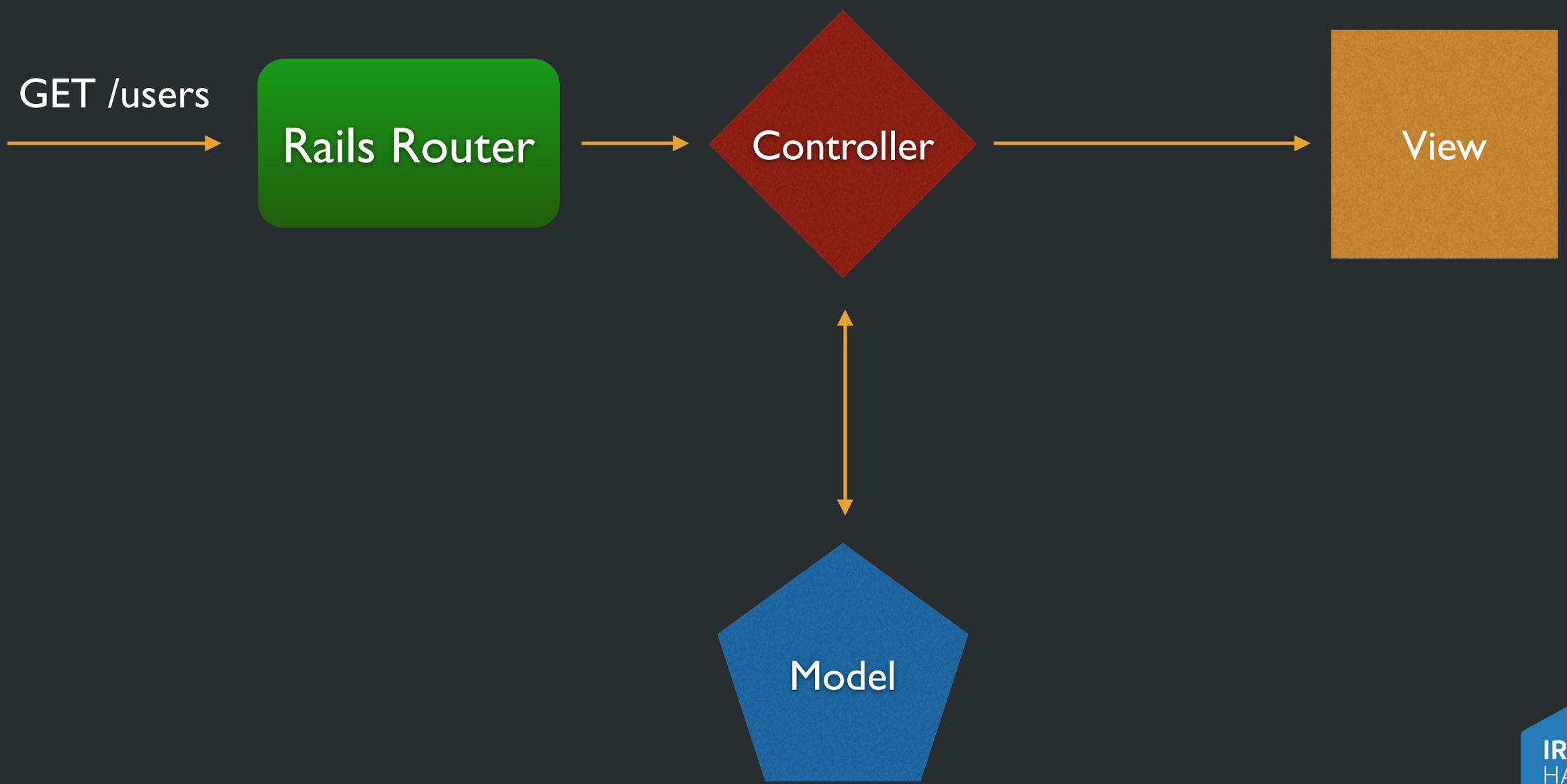
IRON HACK


Controller
View
Why Not?
Model

Why not MV?

- avoids coupling
- easy to exchange parts
- view can be “stupid”

In Ruby on Rails



First Controller

```
$ rails g controller home
```

app/controllers/home_controller.rb

app/controllers/home_controller.rb

```
class HomeController < ApplicationController
  def welcome
    render "welcome"
  end
end
```

config/routes.rb

```
get 'home' => 'home#welcome'
```

IRON HACK

app/views/home/welcome.html.erb

```
<h1> Welcome to MeetMe </h1>
<b>your place to tell your friends where they can meet you!</b>
```

Run the server

```
$ rails s
```

Open your Browser

http://localhost:3000/home

Exercise #1

Implement a contact page which will have your contact details

1. Use the Home Controller
2. The page should be reachable under /contact

Exercise #2

Try to remove the execution of render in the home controller

Does the page still work?
Why?
Examine the logs, which templates are rendered?

Exercise #3

Change the routes so that the welcome page is the initial page… under

/

Hint: See comments in routes.rb

Naming Conventions

- Controller Name = view folder name
- Action Name = template file name

app/views/home/welcome.html.erb

app/controllers/home_controller.rb

```
class HomeController < ApplicationController
  def welcome
  end
end
```