Agile Software Development

Produced by

Eamonn de Leastar (edeleastar@wit.ie)

Department of Computing, Maths & Physics
Waterford Institute of Technology
http://www.wit.ie
http://elearning.wit.ie

Waterford Institute *of* Technology
INSTITIÚID TEICNEOLAÍOCHTA PHORT LÁIRGE

eLearning
support unit

Play Framework

Play 'Full Stack' Framework Set

Integrated HTTP server

Expressive HTTP interface (provides full access to HTTP features)		RESTful web services API	Integrated console and build system
High-performance template engine	Public asset compilation	Asynchronous I/O	
HTML form validation	Integrated cache	Akka	

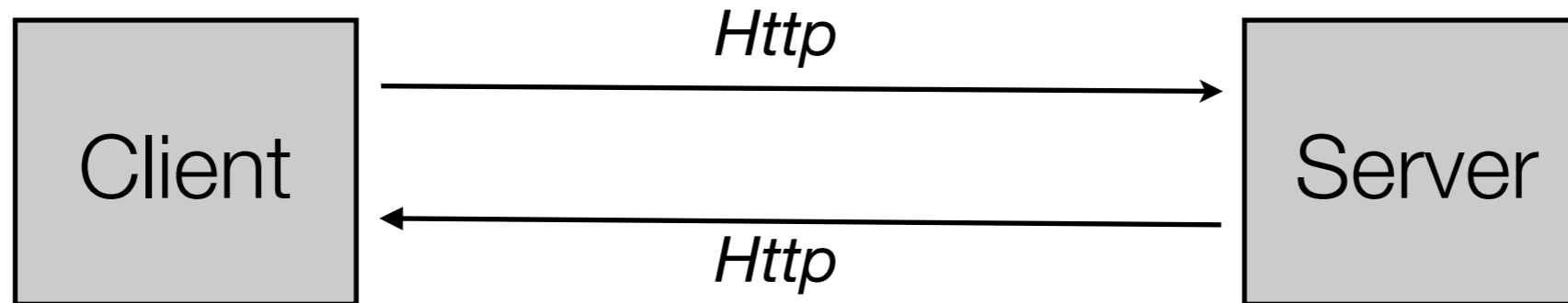| Datastore-agnostic model persistence |
|:---:|
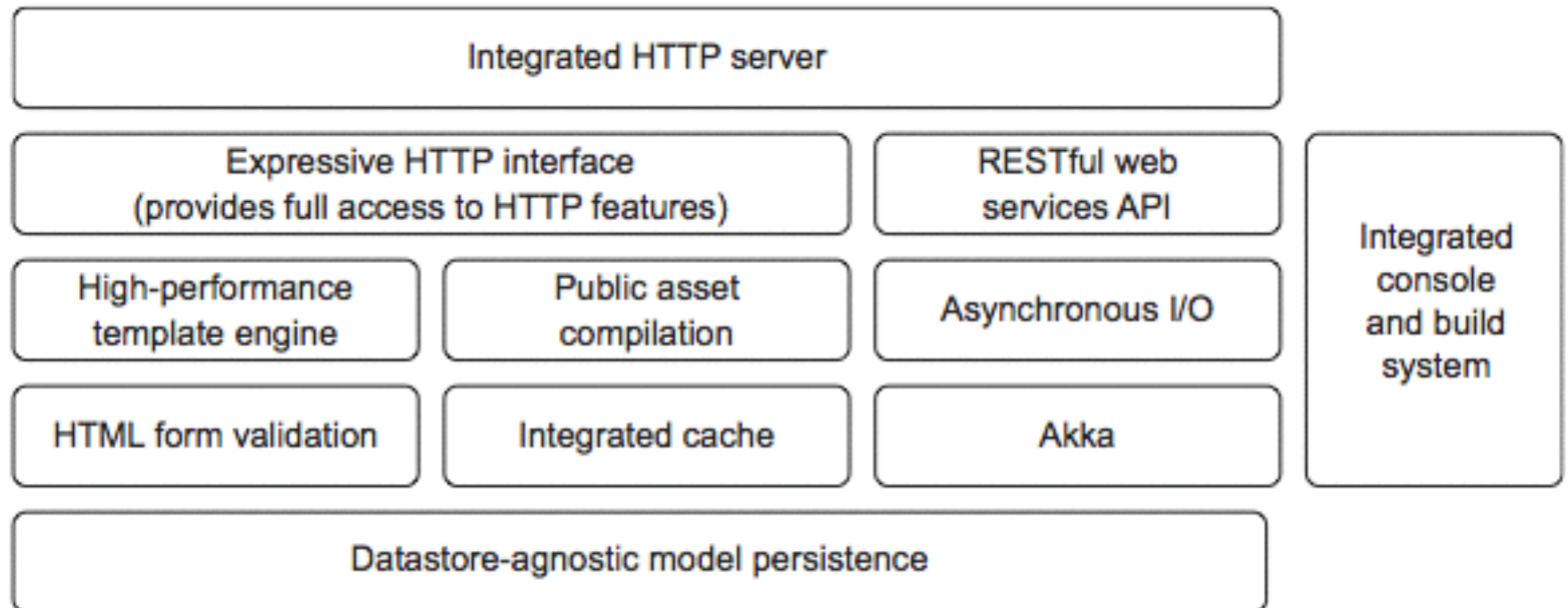
# Web Applications - Request/Response

- Request - http request emitted by browser as a result ot url in address bar, link, button or form submission on page

- Response - web page returned from service to be presented in browser
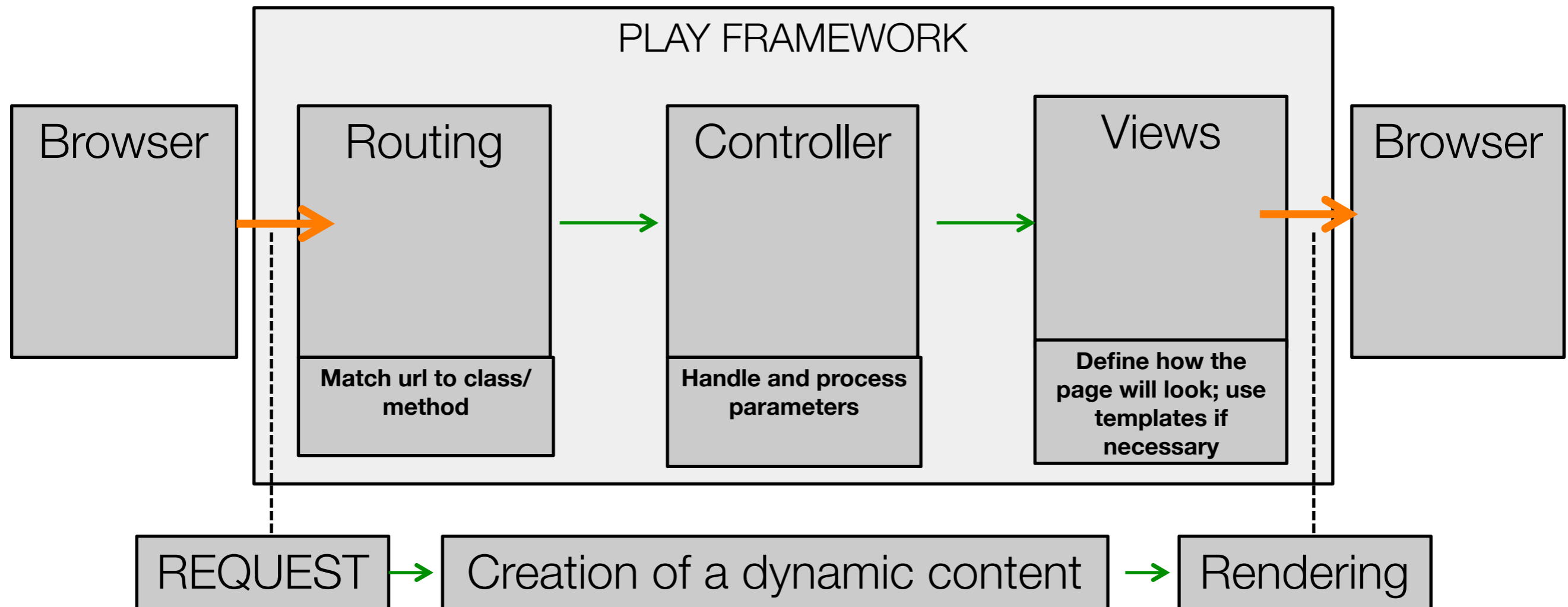
Web Applications - MVC

- Model View Controller is a generally accepted pattern or separation of concerns within the server

- **Model:** Core application domain model + database persistence

- **View**: User Experience

- **Controlle**r: Directly handle all requests, mediate with Model, build and assemble the response using the views
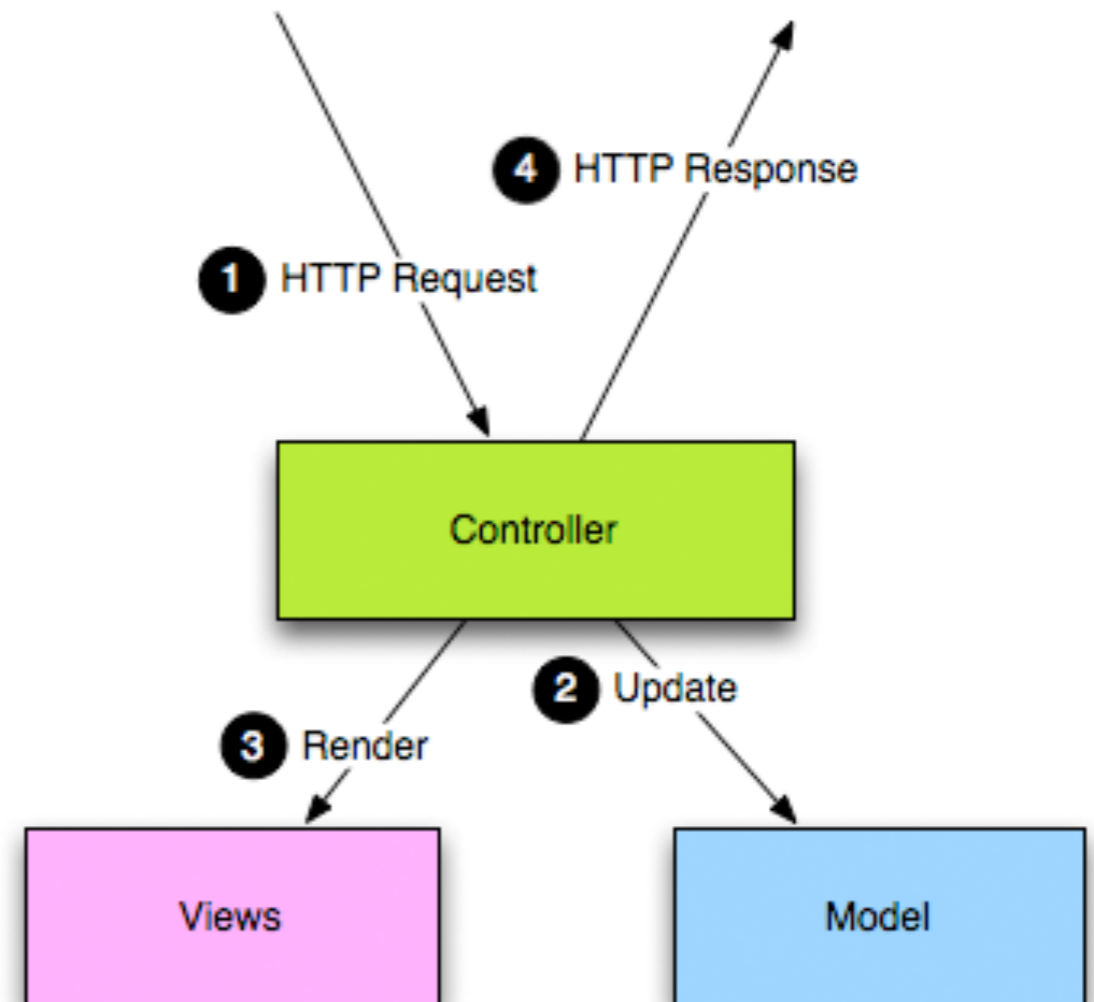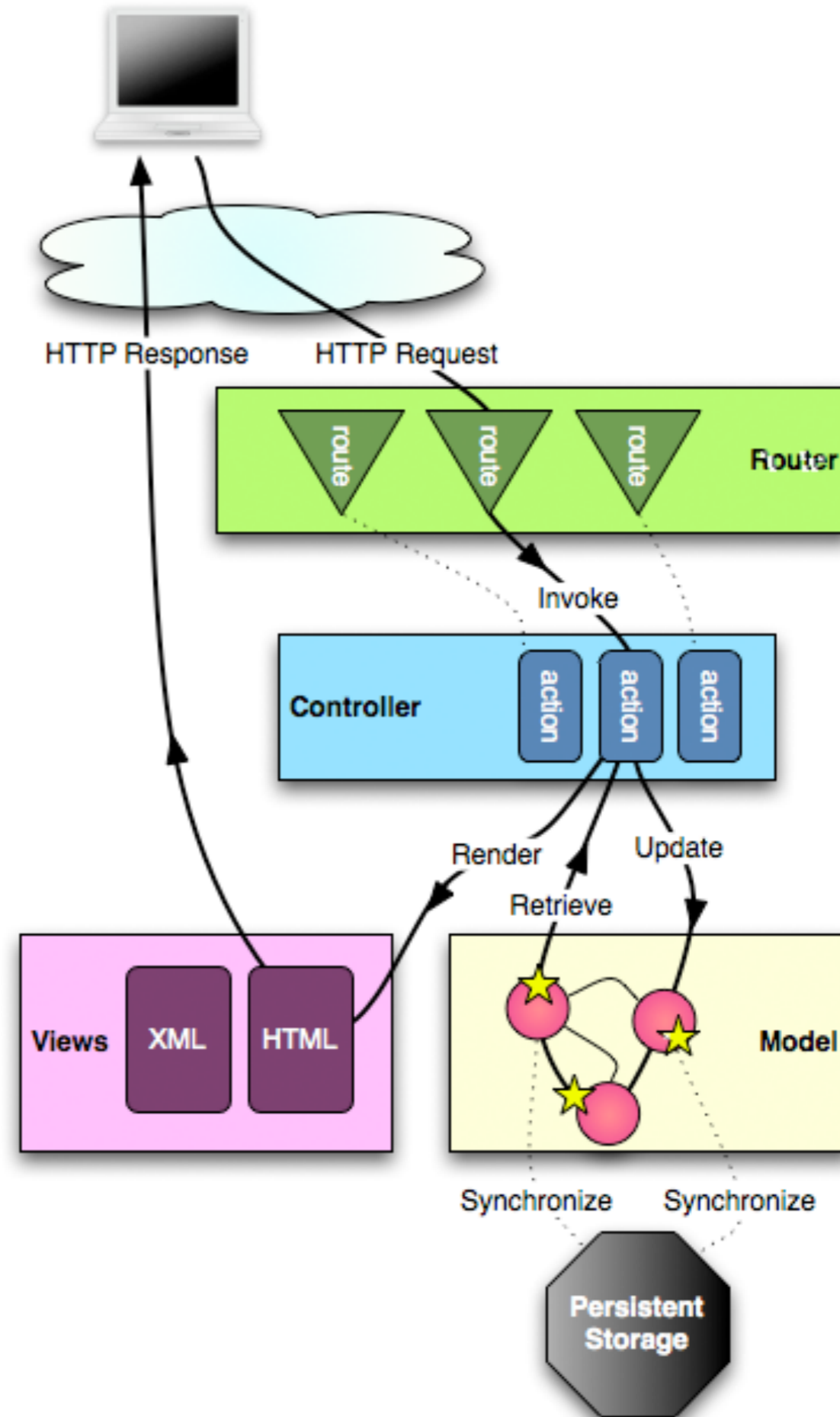
Request/Response Lifecycle

MVC in Play

- Router: examine incoming requests and match to corresponding Controller/Action

- Action: a method in the controller

Routes - UI

```
# UI

GET     /                                    controllers.Accounts.index()
GET     /signup                              controllers.Accounts.signup()
GET     /login                               controllers.Accounts.login()
GET     /logout                              controllers.Accounts.logout()
POST    /register                            controllers.Accounts.register()
POST    /authenticate                        controllers.Accounts.authenticate()

GET     /dashboard                           controllers.Dashboard.index()
GET     /upload                              controllers.Dashboard.uploadActivityForm()
POST    /submitactivity                      controllers.Dashboard.submitActivity()
```
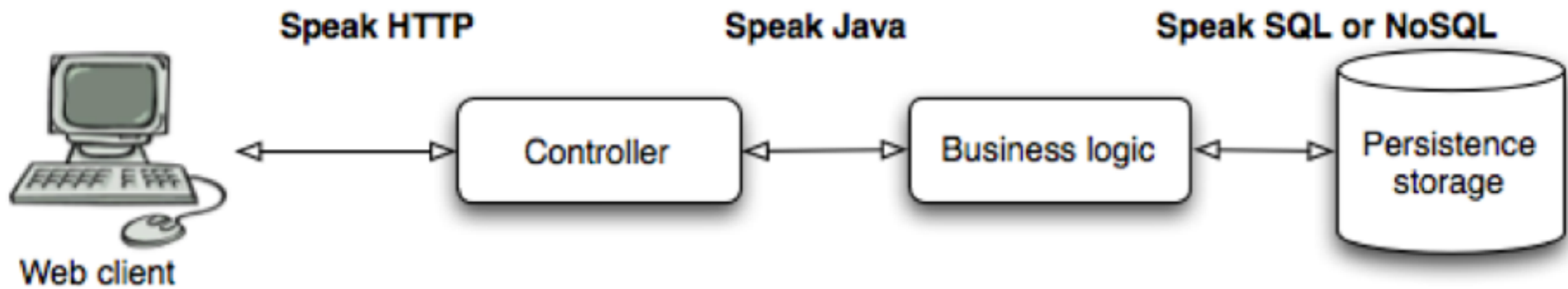
- Routes to deliver UI

- Each of these routes appears in views

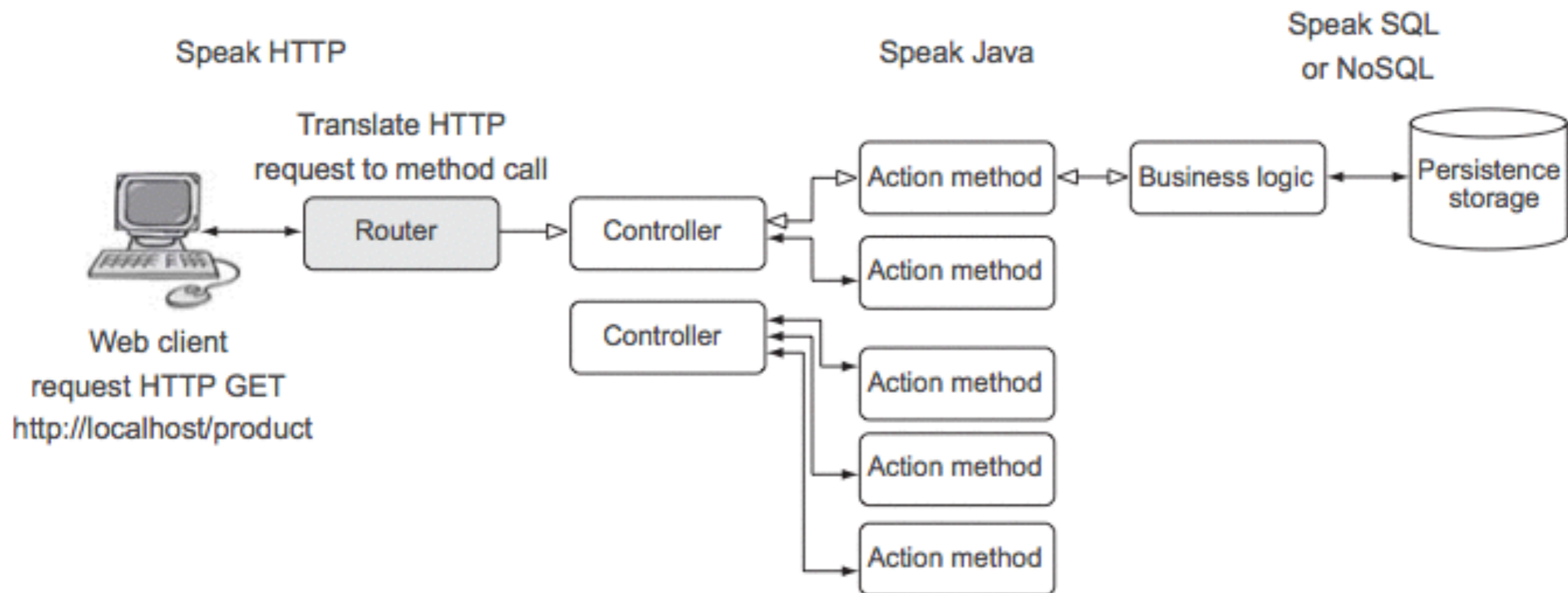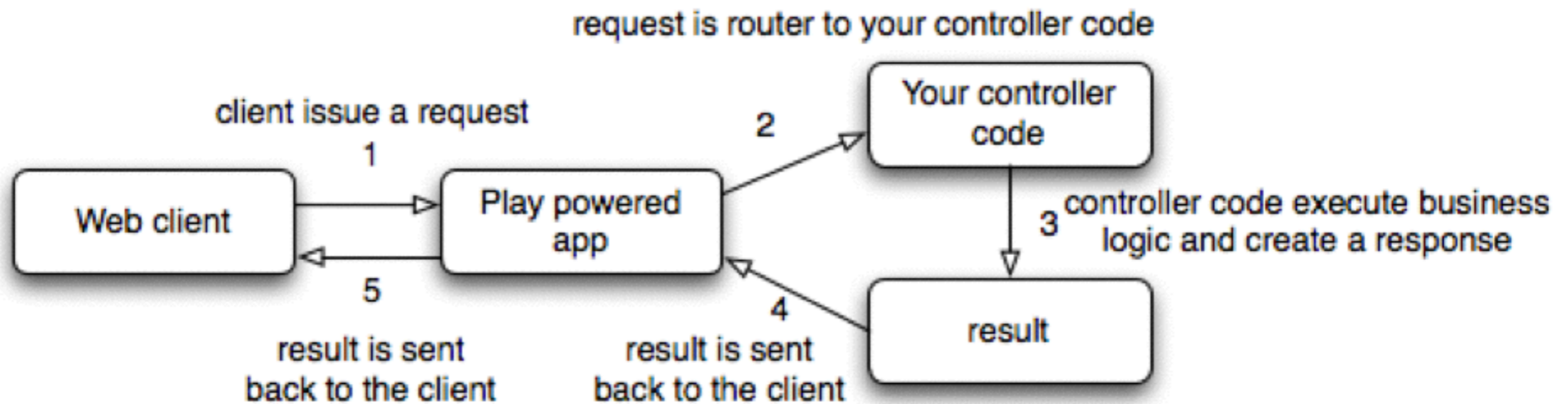- Each of these actions generates and returns a complete HTML page

Role of Controller

Role of Controller / Actions

Speak HTTP

Speak Java

Speak SQL
or NoSQL

Translate HTTP
request to method call

Web client
request HTTP GET
http://localhost/product

Router

Controller

Controller

Action method

Action method

Action method

Action method
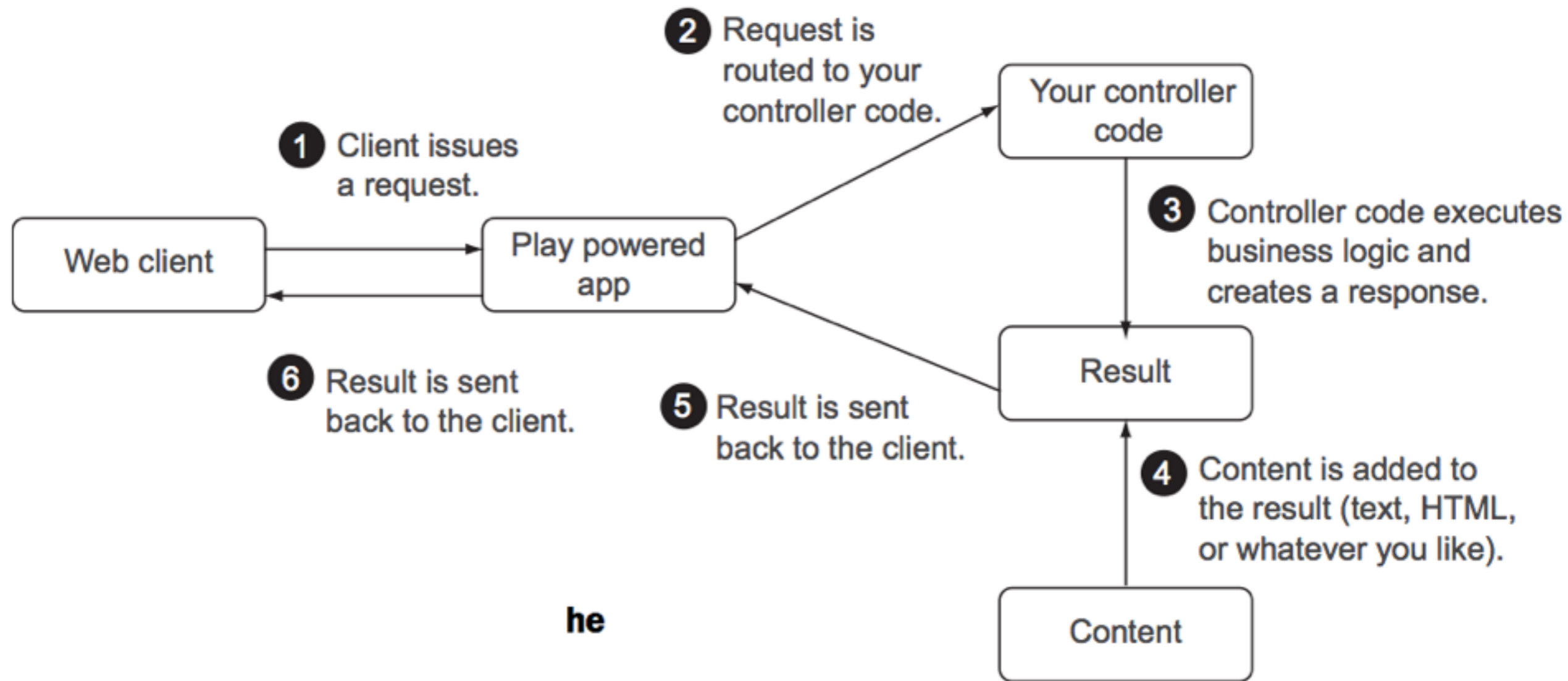
Action method

Business logic

Persistence
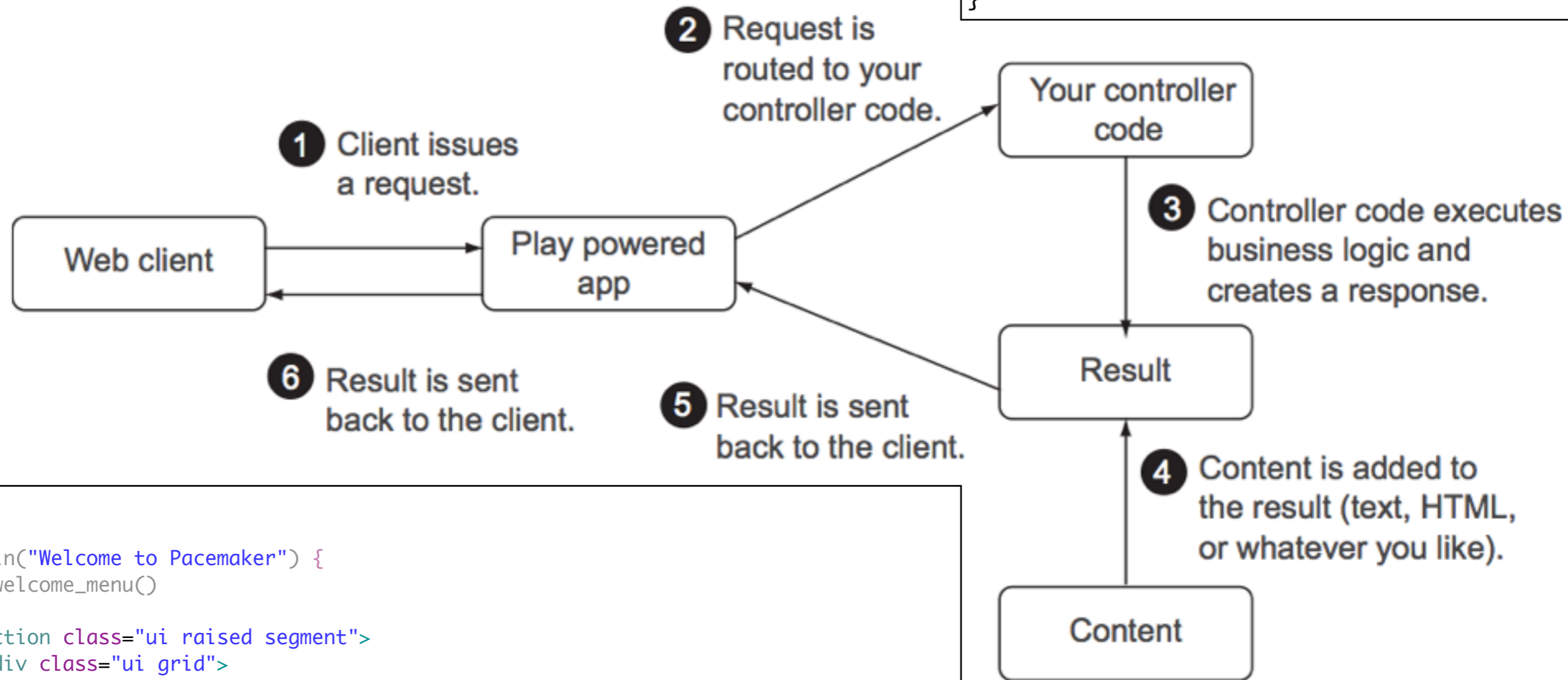storage

Controller Lifecycle

Controller Lifecycle with Content

Controller Lifecycle with Content

```java
public class Accounts extends Controller
{
  public static Result index()
  {
    return ok(welcome_main.render());
  }
...
}
```

2 Request is routed to your controller code.

1 Client issues a request.

Web client

Play powered app

Your controller code

3 Controller code executes business logic and creates a response.

6 Result is sent back to the client.

5 Result is sent back to the client.

Result

4 Content is added to the result (text, HTML, or whatever you like).

Content

```html
@()

@main("Welcome to Pacemaker") {
  @welcome_menu()

<section class="ui raised segment">
  <div class="ui grid">
    <aside class="six wide column">
      <img src="@routes.Assets.at("images/pacemaker.jpg")" class="ui medium image">
    </aside>
    <article class="ten wide column">
      <h1 class="ui  header"> Sign up for Pacemaker  </h1>
      <p> No Bitcoins accepted! </p>
    </article>
  </div>
</section>
}
```

Waterford Institute *of* Technology
INSTITIÚID TEICNEOLAÍOCHTA PHORT LÁIRGE

eLearning
support unit