

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>

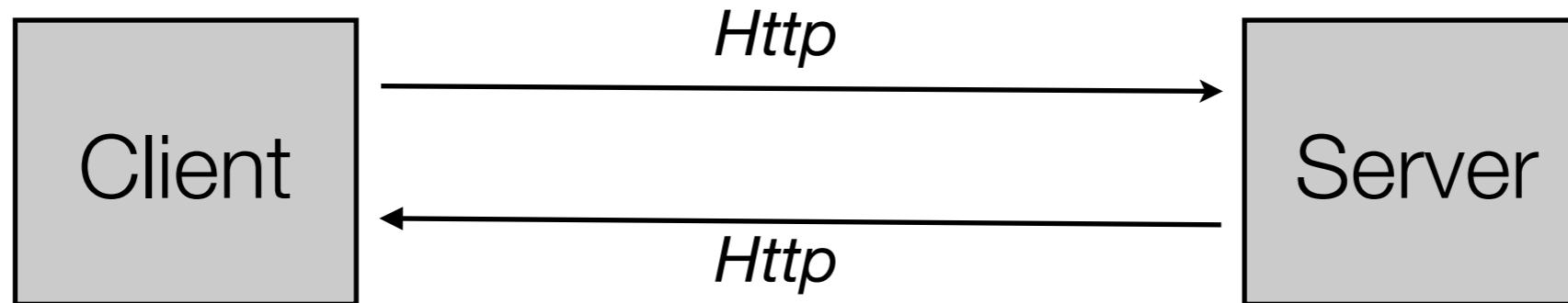


Waterford Institute of Technology
INSTITIÚID TEICNEOLAÍOCHTA PHORT LÁIRCE



Play 2

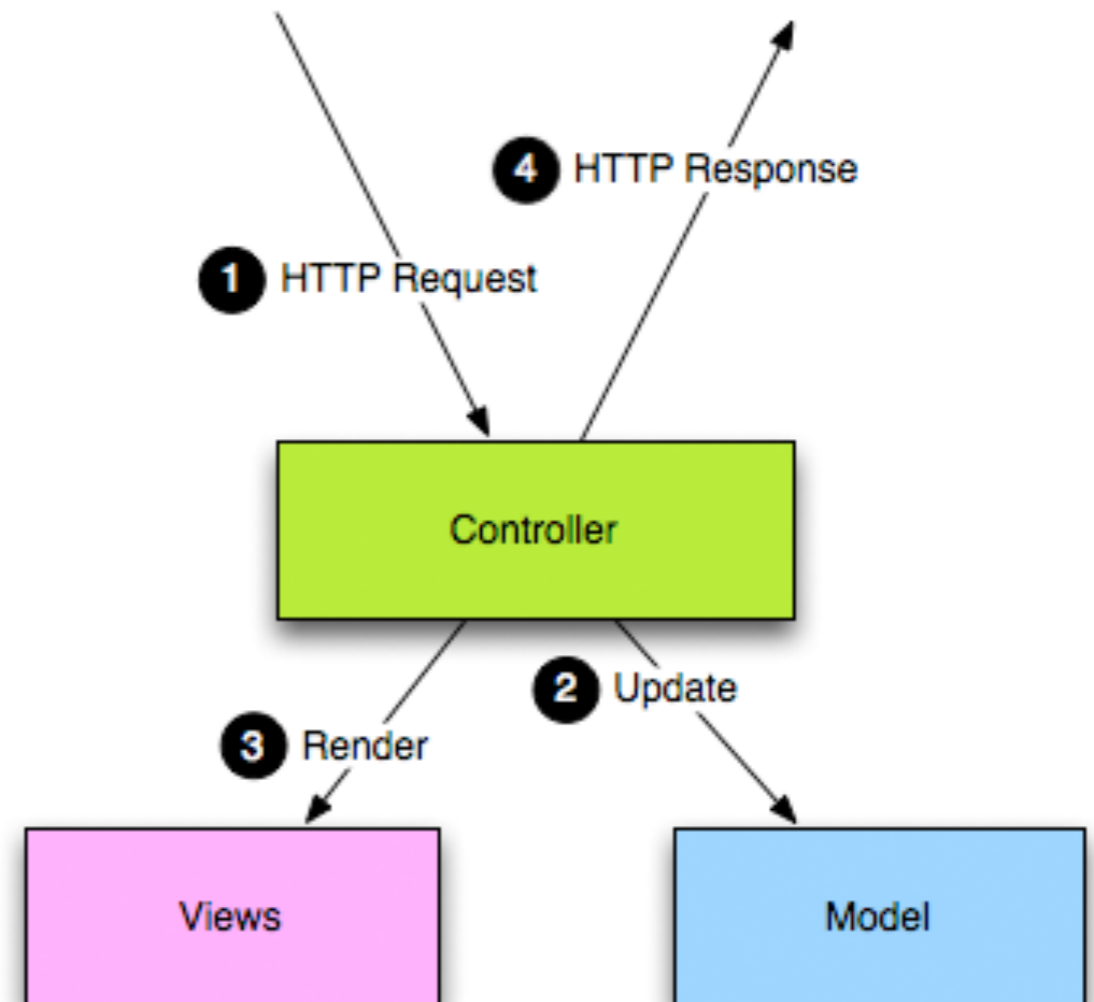
Web Applications - Request/Response



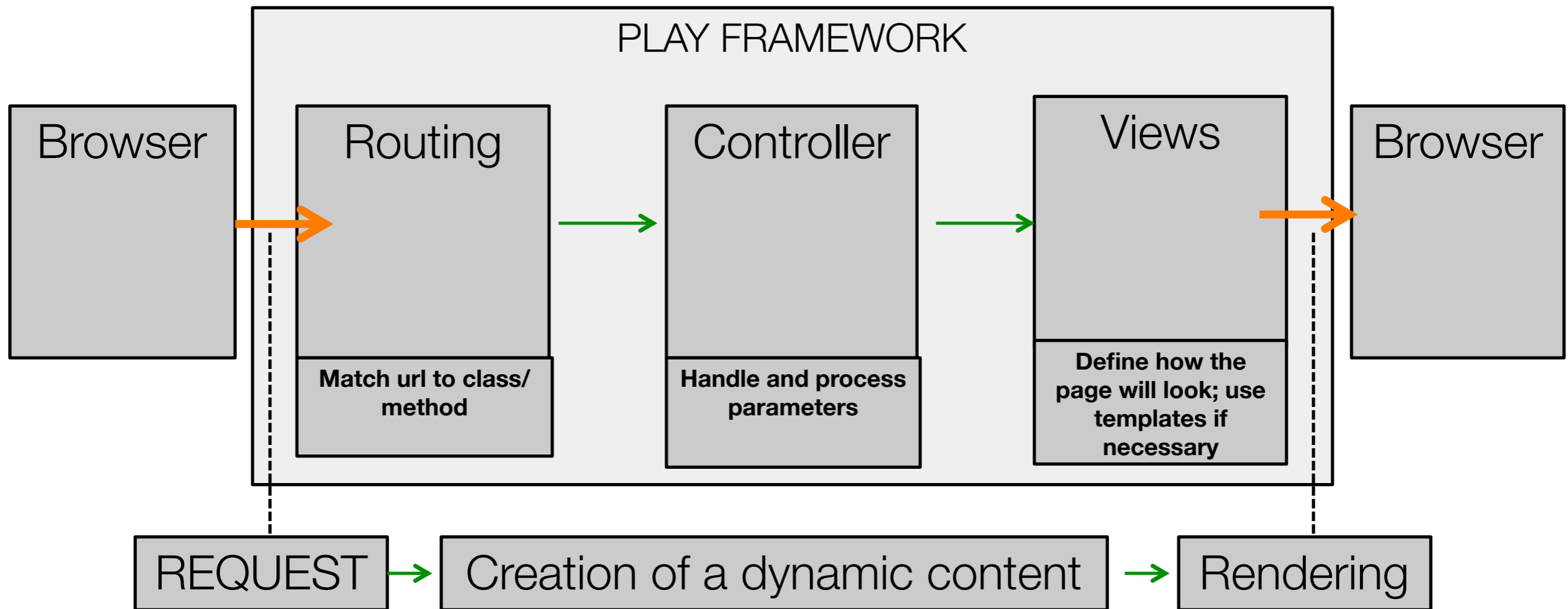
- Request - http request emitted by browser as a result of url in address bar, link, button or form submission on page
- Response - web page returned from service to be presented in browser

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
- **Controller:** Directly handle all requests, mediate with Model, build and assemble the response using the views

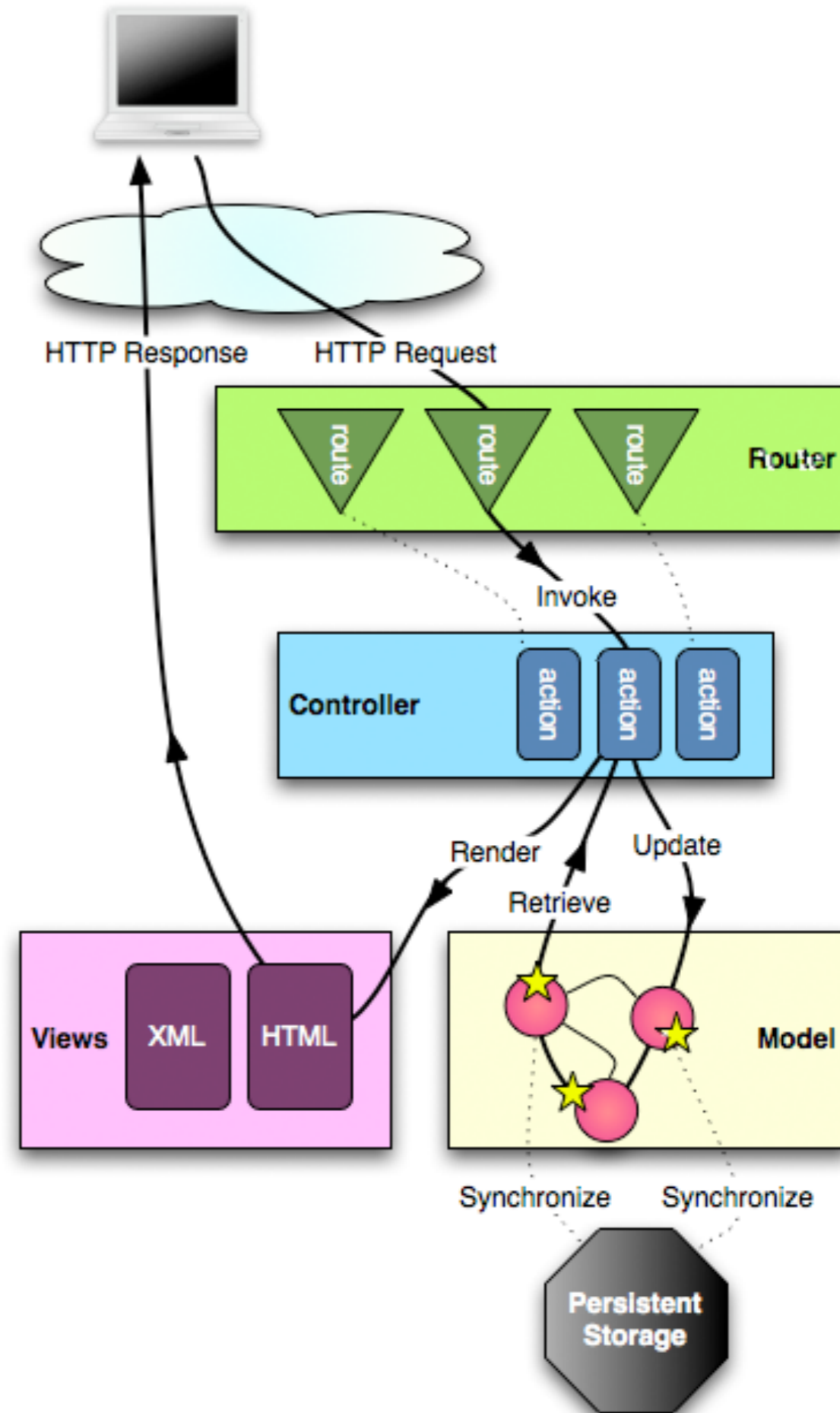


Request/Response Lifecycle



MVC in Play

- Router: examine incoming requests and match to corresponding Controller/Action
- Action: a method in the controller



Routes File Example (Pacemaker)

```
# 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()

# API

GET      /api/users      controllers.PacemakerAPI.users()
DELETE   /api/users      controllers.PacemakerAPI.deleteAllUsers()
POST     /api/users      controllers.PacemakerAPI.createUser()

GET      /api/users/:id  controllers.PacemakerAPI.user(id: Long)
DELETE   /api/users/:id  controllers.PacemakerAPI.deleteUser(id: Long)
PUT      /api/users/:id  controllers.PacemakerAPI.updateUser(id: Long)

GET      /api/users/:userId/activities controllers.PacemakerAPI.activities(userId: Long)
POST     /api/users/:userId/activities controllers.PacemakerAPI.createActivity(userId: Long)

GET      /api/users/:userId/activities/:activityId controllers.PacemakerAPI.activity(userId: Long, activityId:Long)
DELETE   /api/users/:userId/activities/:activityId controllers.PacemakerAPI.deleteActivity(userId: Long, activityId:Long)
PUT      /api/users/:userId/activities/:activityId controllers.PacemakerAPI.updateActivity(userId: Long, activityId:Long)

# Map static resources from the /public folder to the /assets URL path
GET      /assets/*file   controllers.Assets.at(path="/public", file)
```

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

Routes - API

```
# API

GET      /api/users           controllers.PacemakerAPI.users()
DELETE   /api/users           controllers.PacemakerAPI.deleteAllUsers()
POST     /api/users           controllers.PacemakerAPI.createUser()

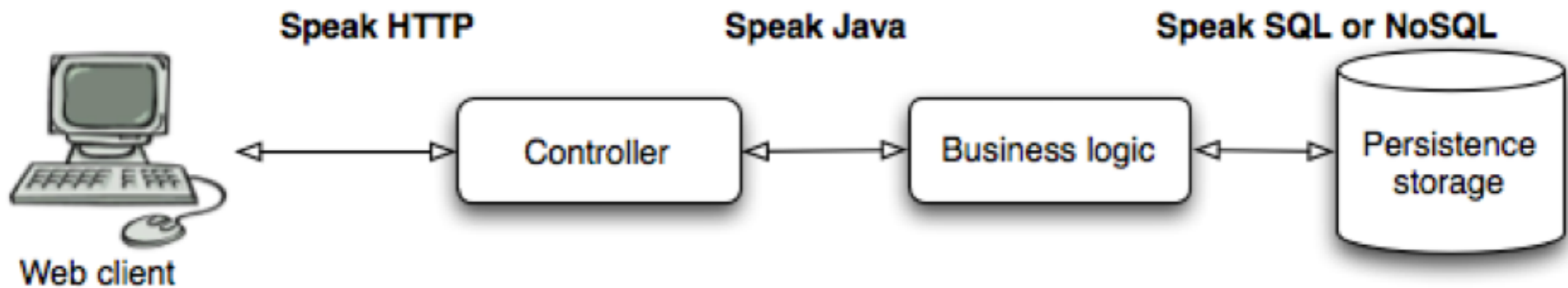
GET      /api/users/:id       controllers.PacemakerAPI.user(id: Long)
DELETE   /api/users/:id       controllers.PacemakerAPI.deleteUser(id: Long)
PUT      /api/users/:id       controllers.PacemakerAPI.updateUser(id: Long)

GET      /api/users/:userId/activities controllers.PacemakerAPI.activities(userId: Long)
POST     /api/users/:userId/activities controllers.PacemakerAPI.createActivity(userId: Long)

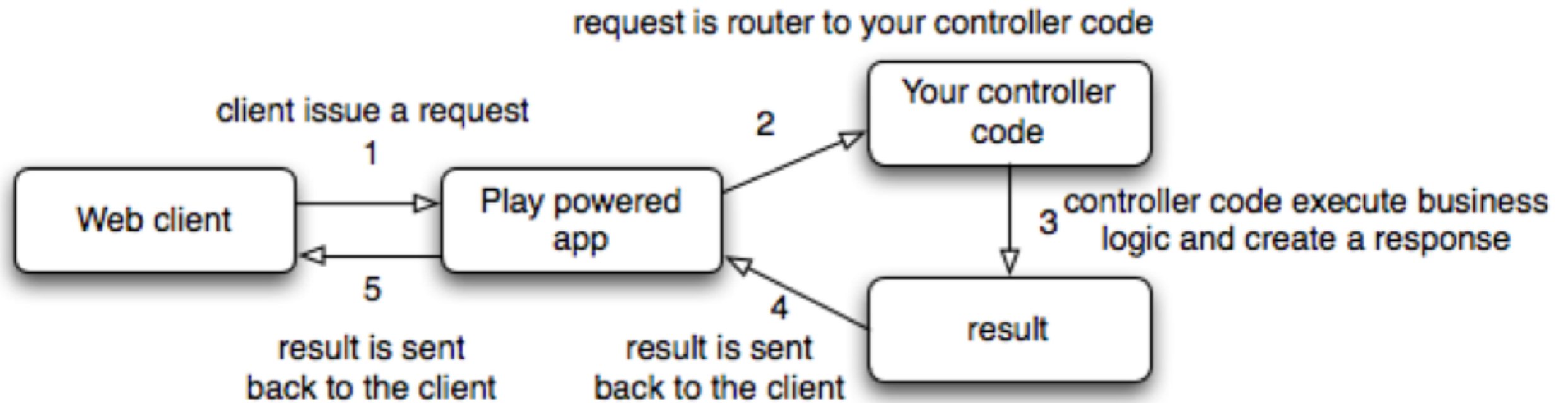
GET      /api/users/:userId/activities/:activityId controllers.PacemakerAPI.activity(userId: Long, activityId:Long)
DELETE   /api/users/:userId/activities/:activityId controllers.PacemakerAPI.deleteActivity(userId: Long, activityId:Long)
PUT      /api/users/:userId/activities/:activityId controllers.PacemakerAPI.updateActivity(userId: Long, activityId:Long)
```

- Routes to deliver API
- Each of these routes appears in views as an API Request
- Each of these actions generates and returns a JSON payload

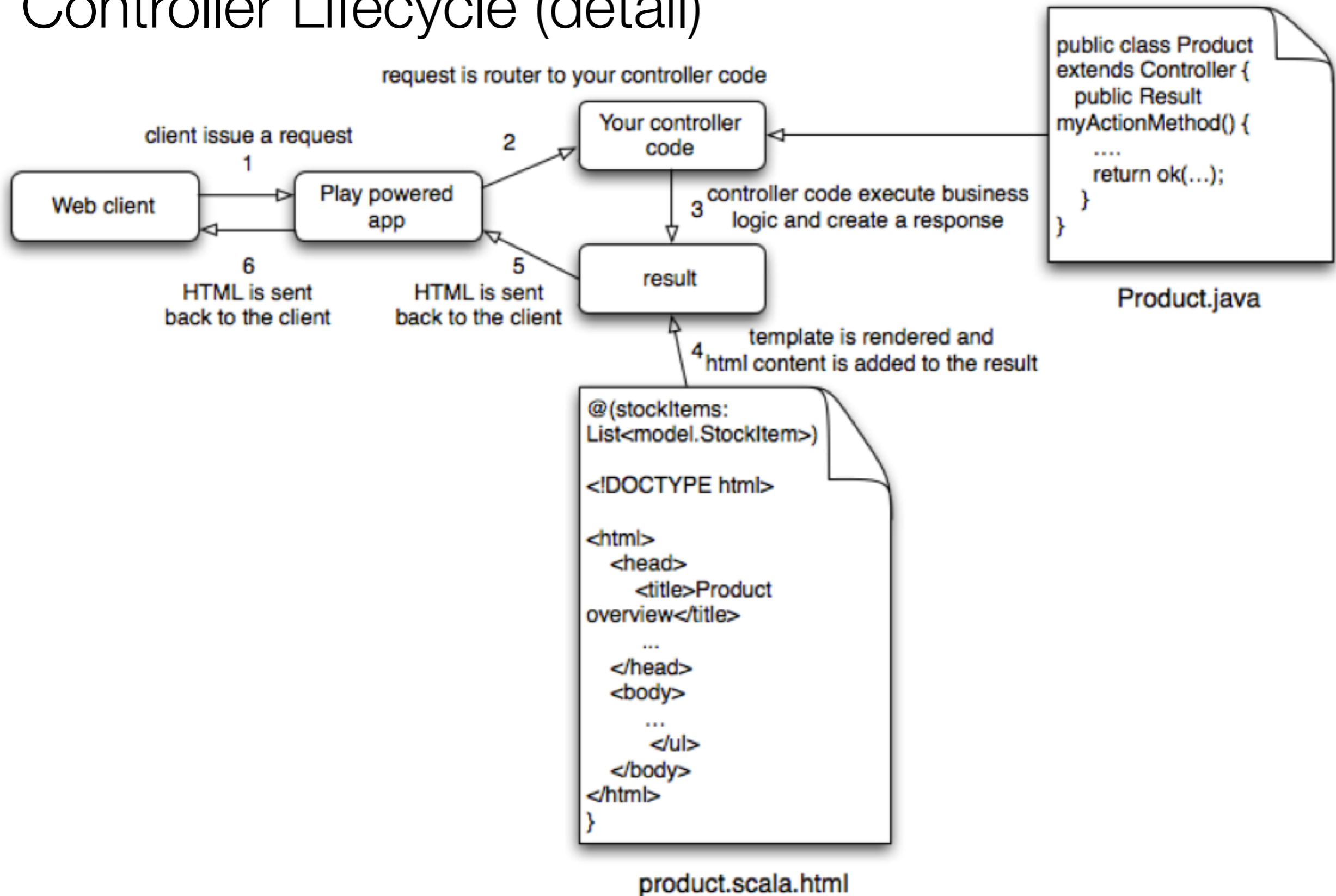
Role of Controller



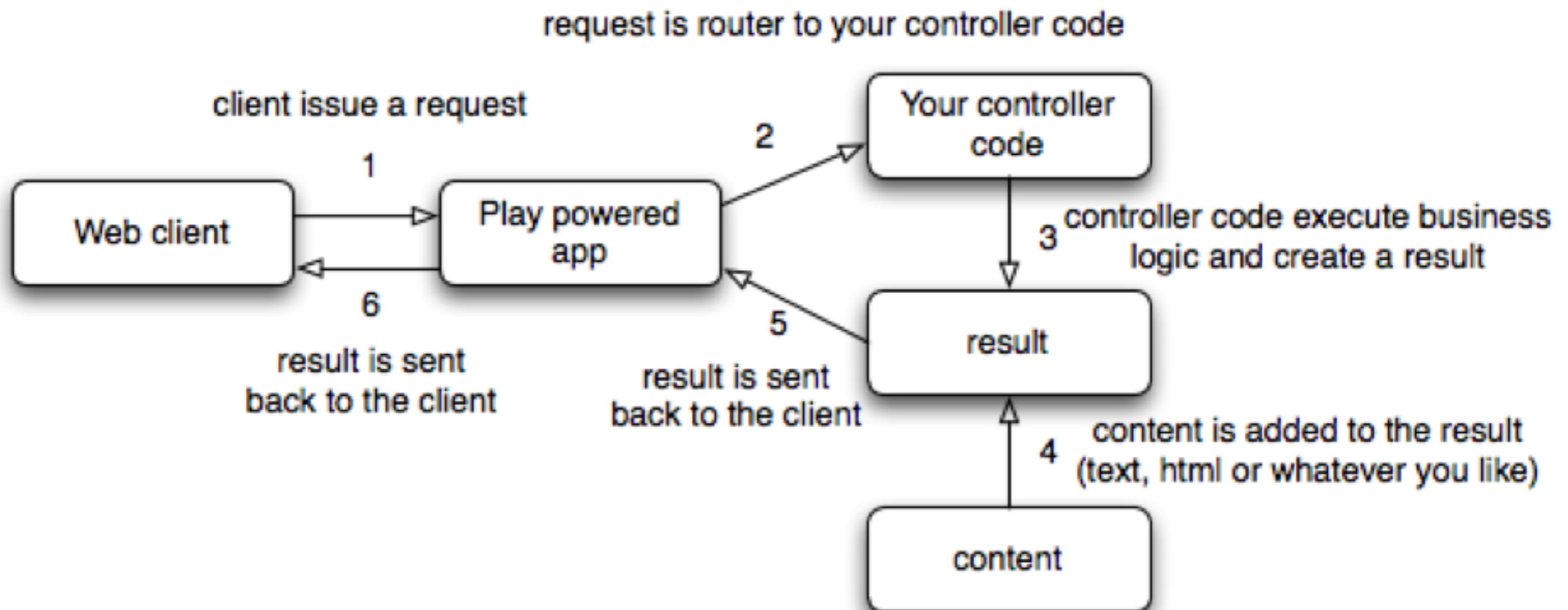
Controller Lifecycle



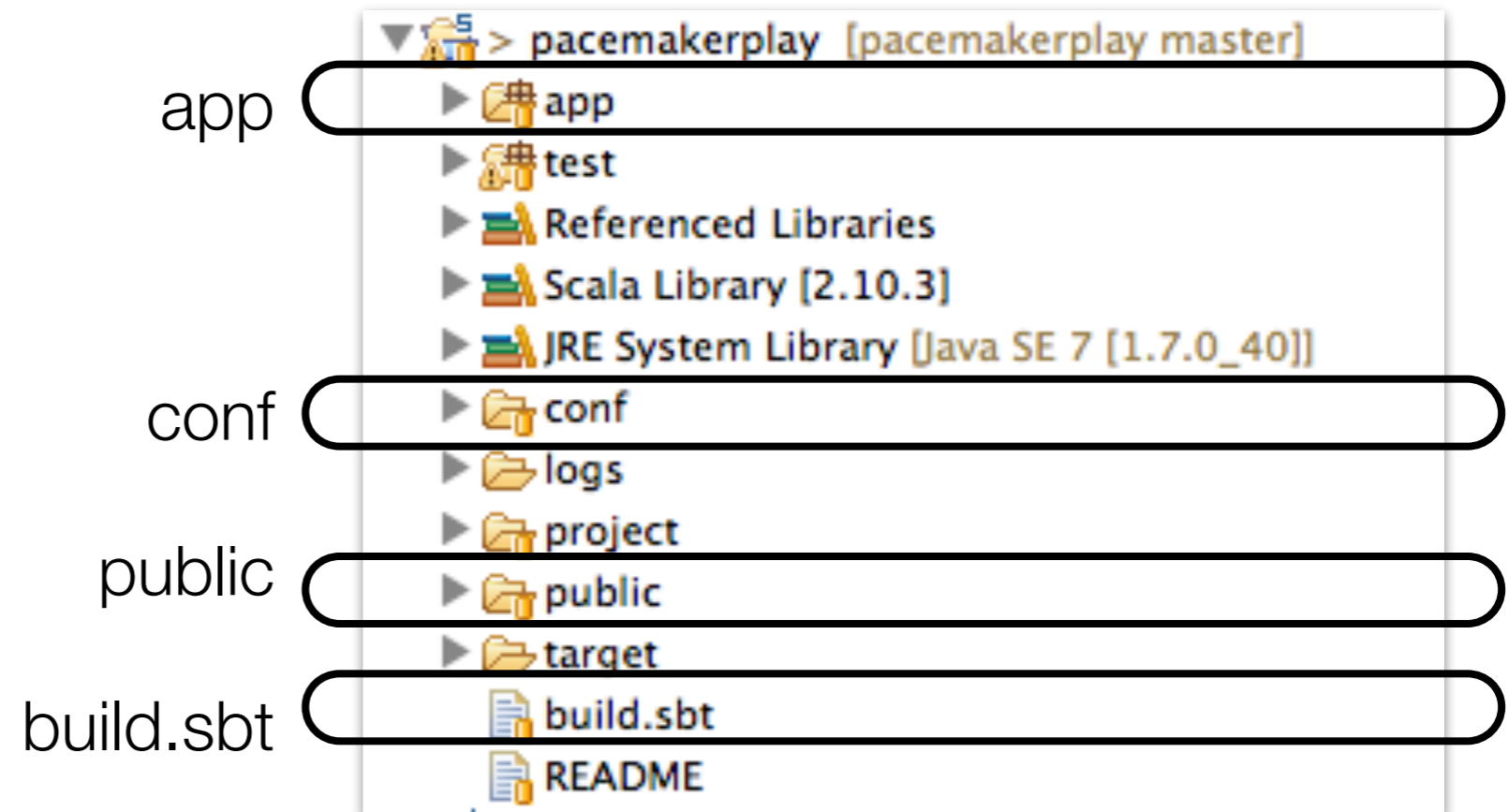
Controller Lifecycle (detail)



Controller Lifecycle with Content



Play Project Anatomy

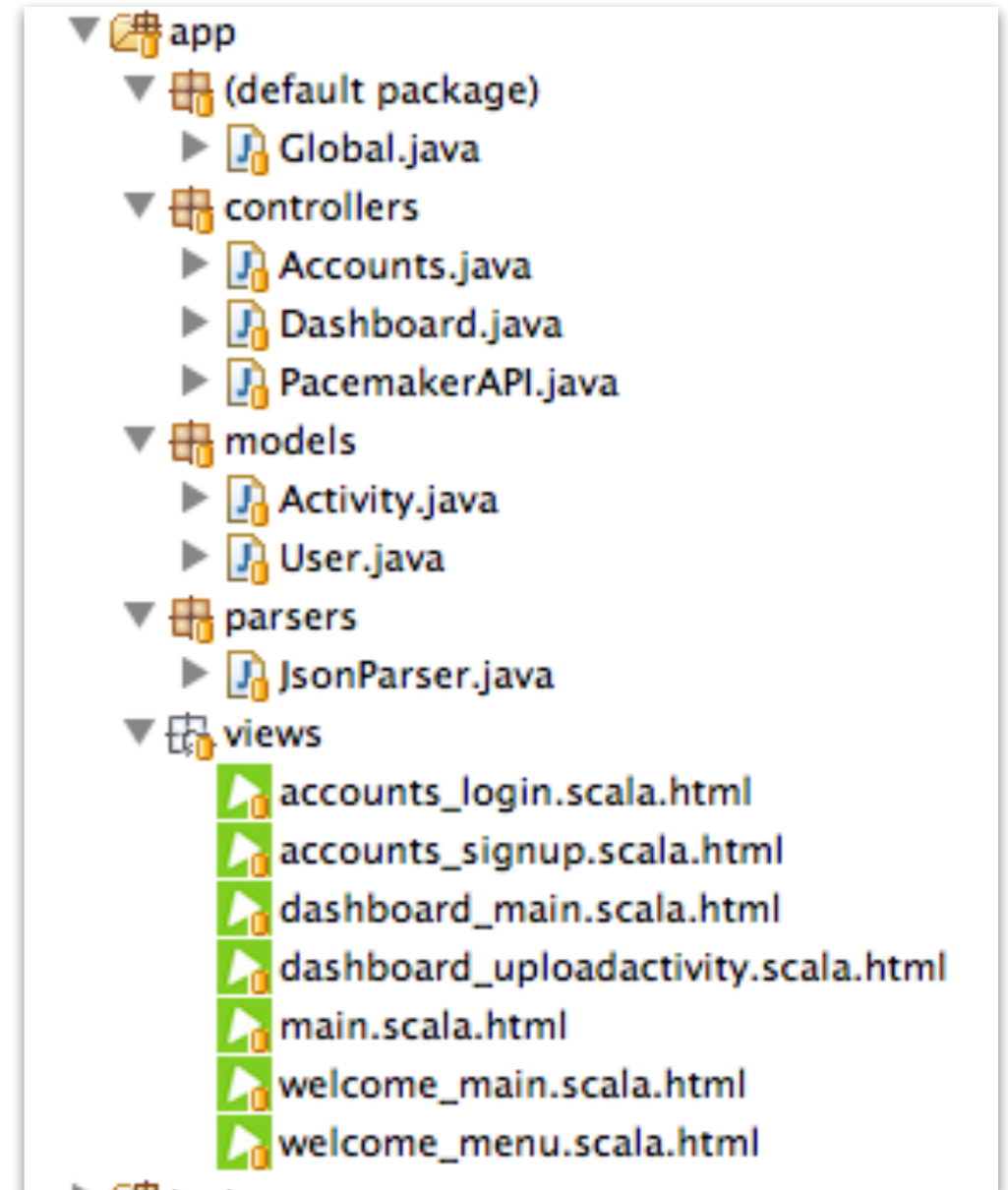


App

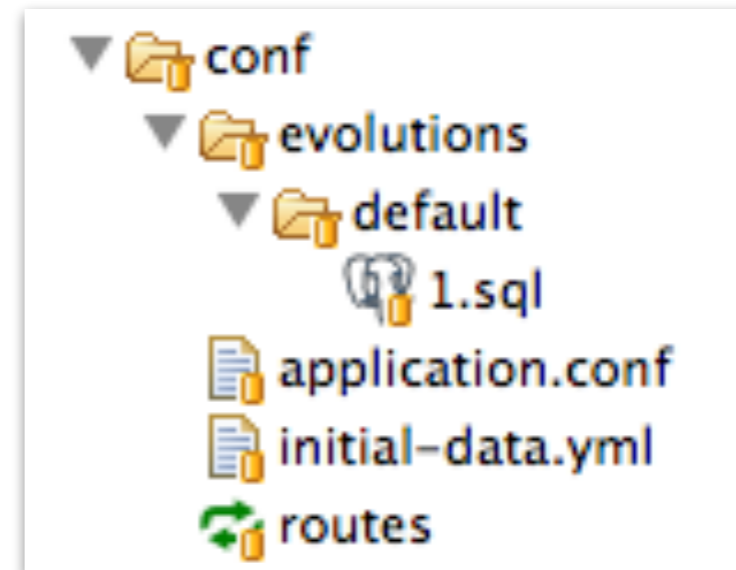
controllers

models

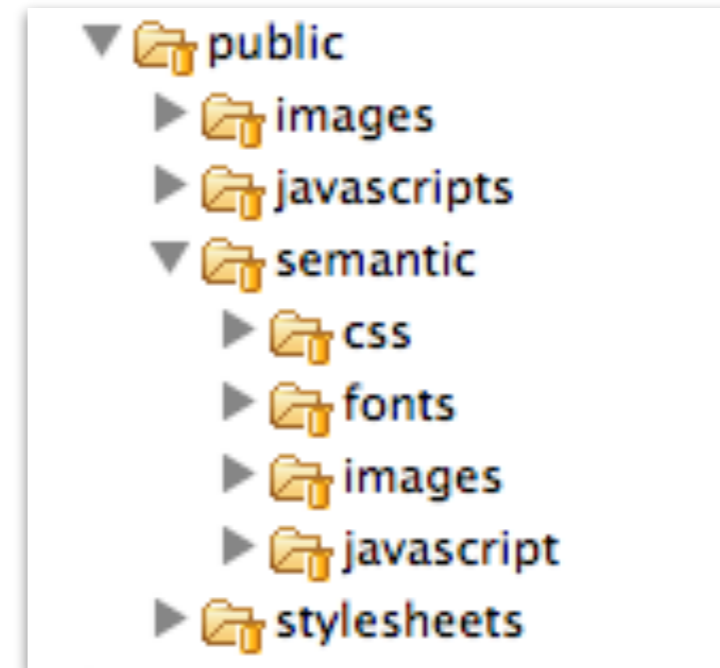
views



conf



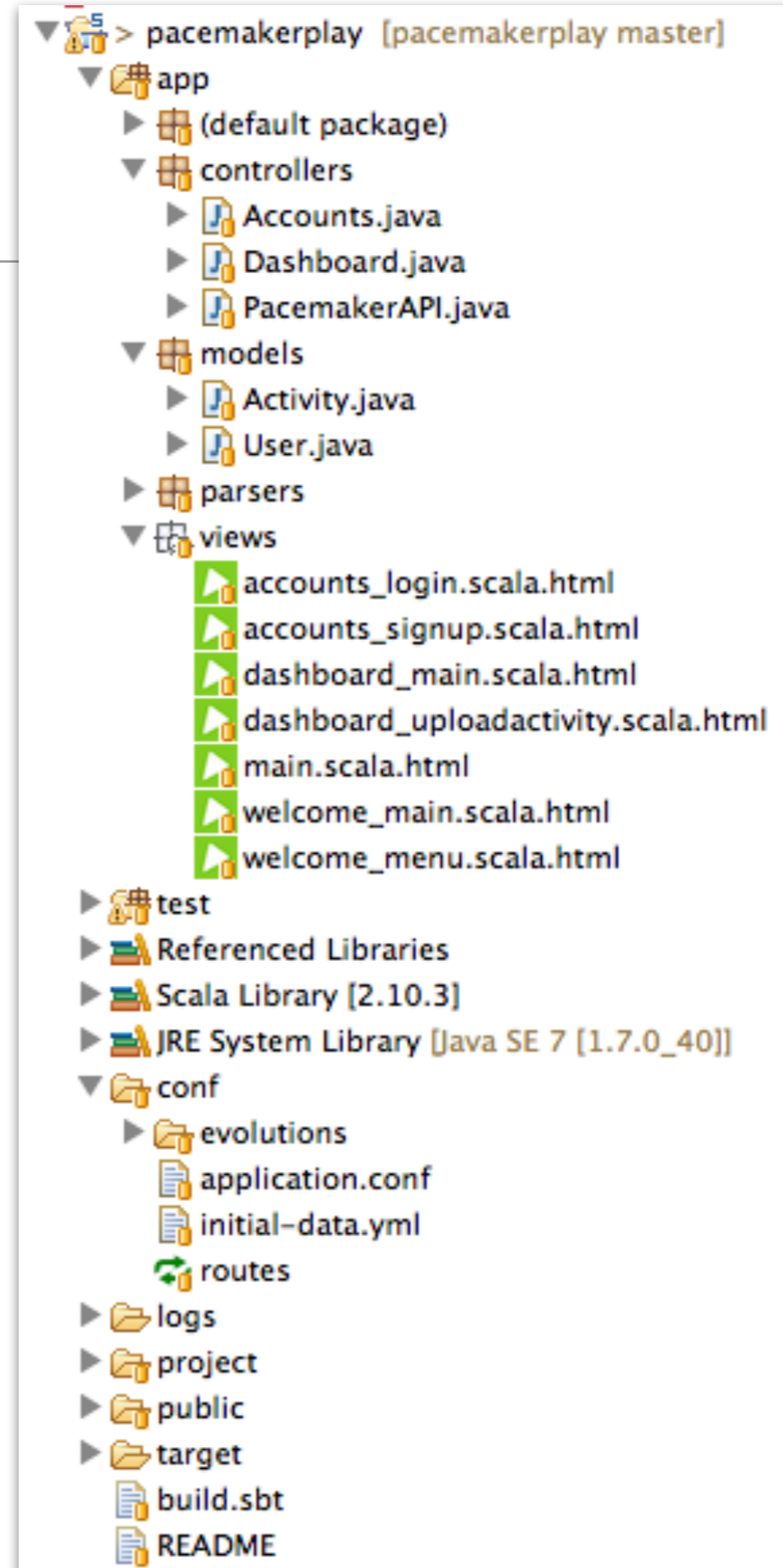
public

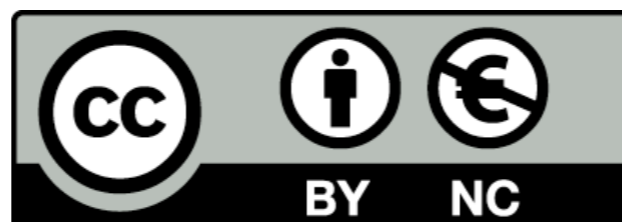


build.sbt

```
name := "pacemakerplay"
version := "1.0-SNAPSHOT"
libraryDependencies += Seq(
  javaJdbc,
  javaEbean,
  cache,
  "net.sf.flexjson" % "flexjson" % "3.1",
  "postgresql" % "postgresql" % "9.1-901-1.jdbc4"
)
play.Project.playJavaSettings
```

pacemakerplay





Except where otherwise noted, this content is licensed under a Creative Commons Attribution-NonCommercial 3.0 License.

For more information, please see <http://creativecommons.org/licenses/by-nc/3.0/>



Waterford Institute of Technology
INSTITIÚID TEICNEOLAÍOCHTA PHORT LÁIRCE

