Architecture Design Pattern - MVC

What’s the pros and cons? And what’s the next?

Jeff Wu

Tech + Dream = Future

Common problems when designing the architecture

A big ONE project

Basically having **ONE** project in a single solution which has ***all*** the code in there.

usually it's a large mess

Separation of concern

Deployment easiness

If you wish to make a single change to the website/application, *the entire application will go offline/restart*.

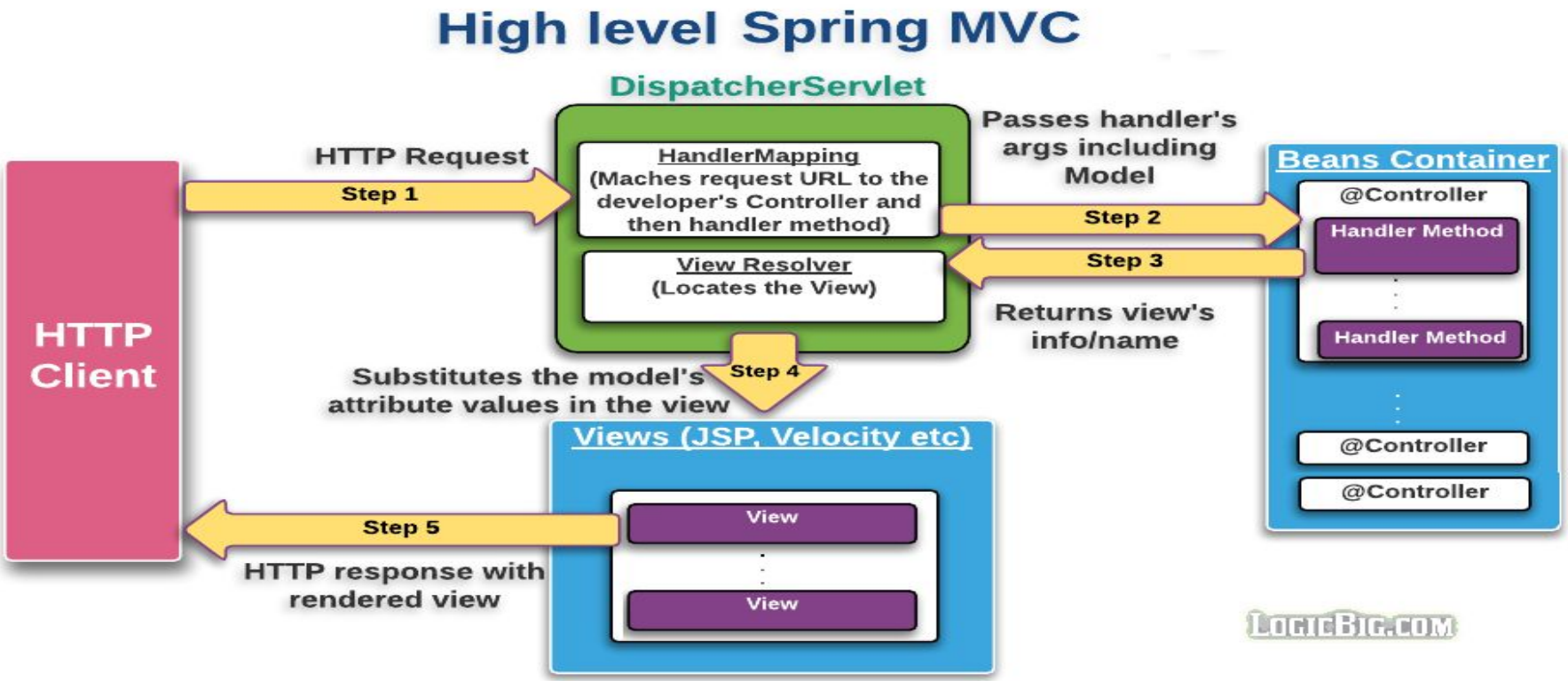
Significant downtime

Reusability

easier **reuse** of your various components into various *other* subsystems of your application

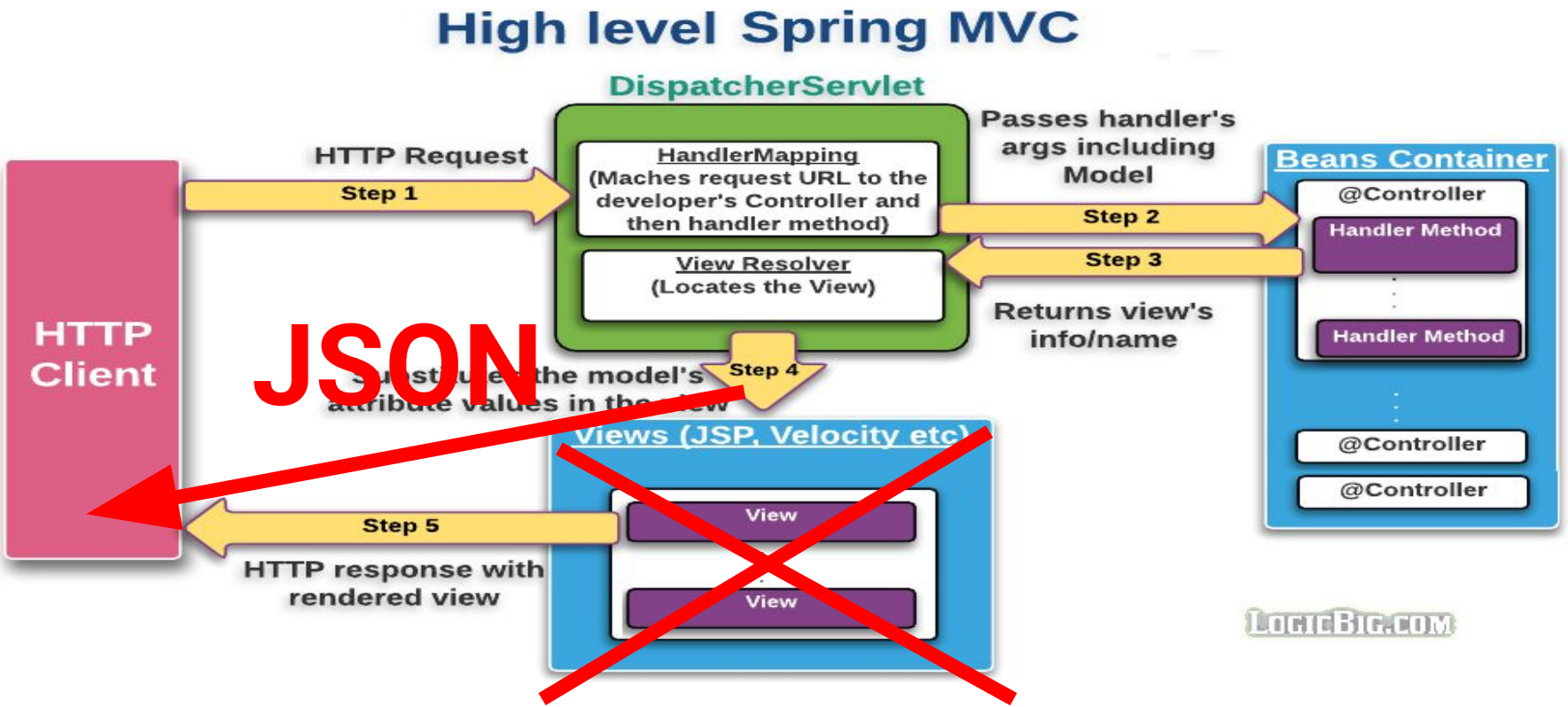
easier to maintain and support

Case1: How MVC works in a typical web application



Credit to LogicBig.com

Case2: How MVC works in a Restful API service



Credit to LogicBig.com

Is it still MVC in the case2 ?

Professor Lee: hey guys! What do you think about this?

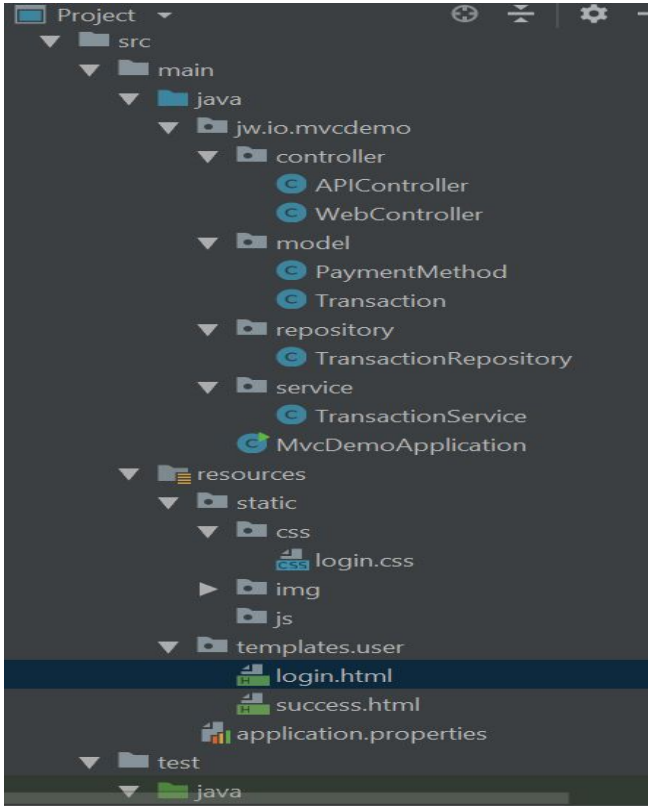
Student Adam: MVC is definitely obsolete. We should consider adopting a new pattern

Student Emily: It's still there, but apparently something happened. I think the VIEW is changed to a JSON response now.

Anyway! Let’s focus on MVC itself

- Spring Boot 2 Implementation

Project Structure



Main Components:

- Controller
- Model
- Repository
- Service
- Static resources
  - Images
  - Js
  - css

- Template engine - ThymeLeaf

Source: https://github.com/99887710/mvc-demo

Web Application

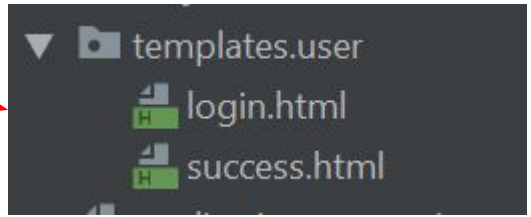
```java
@Controller
public class WebController {

    @RequestMapping("/")
    public String login() {
        return "user/login";
    }

    @RequestMapping("/success")
    public String login_success() {
        return "user/success";
    }

}
```

View



Restful API service

```java
@RestController
@RequestMapping("/transactions")
public class APIController {

    @Autowired
    TransactionService transactionService;

    @RequestMapping("/list")
    public List<Transaction> listTransactions() { return transactionService.li
}
```

JSON


```
[{"id":"5eabb37851be1fa37313f983","customerFirstName":"Hanson","customerLastName":"Bryan","merchantName":"voluptate","paymentMethod":
{"cardScheme":"VISA","cardNumber":4539466317346687,"issuer":"HSBC","transactionDateTime":"2020-01-16T18:53:05.126Z","amount":1289.34}},
{"id":"5eabb378f3bdaf1ddbee9d6f","customerFirstName":"Isabella","customerLastName":"Burke","merchantName":"dolor","paymentMethod":
{"cardScheme":"VISA","cardNumber":4539466317341837,"issuer":"Citi","transactionDateTime":"2020-04-27T11:26:01.119Z","amount":250.56}}]
```

@Controller v.s. @RestController

```
@Target({ElementType.TYPE})
@Retention(RetentionPolicy.RUNTIME)
@Documented
@Controller
@ResponseBody
public @interface RestController {
    @AliasFor(
        annotation = Controller.class
    )
    String value() default "";
}
```

Architecture is evolving...

Monolith

- **Single**
- **All-in-one**
- **Highly-coupled**

...

N Tier

- **Separation of concern**
- **Multi-tier**
- **Reusability**

....

Microservices

Light-weight

Autonomous

Heterogeneous

Has its own database

...

Where we are ?

Who is/has using/used Microservices?

Goldman Sachs

monzo

Thank You !