

Università Degli Studi di Ferrara

Corso di Laurea in Informatica - A.A. 2023 - 2024

Tecnologie Web

Moderne applicazioni Web

In questa lezione...

- Perché le applicazioni Web?
- Perché Laravel?
- Pattern e strumenti di Laravel 9.x

Perchè applicazioni sul Web?

Perchè sviluppare applicazioni Web invece di applicazioni Desktop?

- Disponibili ovunque
- Multipiattaforma
- Semplici da gestire

Perchè applicazioni sul Web?

Perchè sviluppare applicazioni Web invece di applicazioni Desktop?

- Disponibili ovunque
- Multiplatforma
- Semplici da gestire

... e sono anche belle.

Perchè Laravel?

Laravel è un framework scritto in linguaggio PHP che agevola lo sviluppo di applicazioni Web ed è il framework PHP più utilizzato dal 2014 ad oggi.

- Open Source
- Curva di apprendimento vantaggiosa
- Robusto, Sicuro, Maturo
- Tantissime funzionalità già implementate

MVC

Model - View - Controller

Il pattern MVC

È un modo di organizzare il nostro software che consente di separare i compiti delle varie parti di codice.

- Model
- View
- Controller

MVC - Model

Il Modello è il componente che “describe” i dati.

È anche il componente delegato alla gestione della persistenza e quindi di interazione con il database.

MVC - View

La View ha il compito di formare l'HTML che viene restituito al Client (Browser).

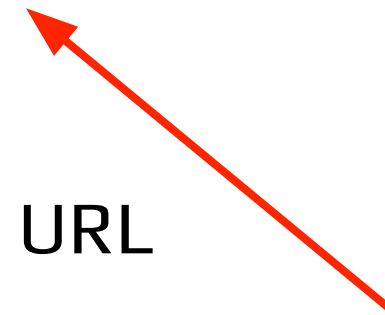
Nelle View è bene che non vengano elaborati i dati, la View ha il solo compito di “presentazione”

MVC - Controller

Il Controller è il componente che ha il compito di “intercettare” le richieste dell’utente ed elabora i dati provenienti dal Database e li prepara per essere presentati prima di passarli alla View.

MVC

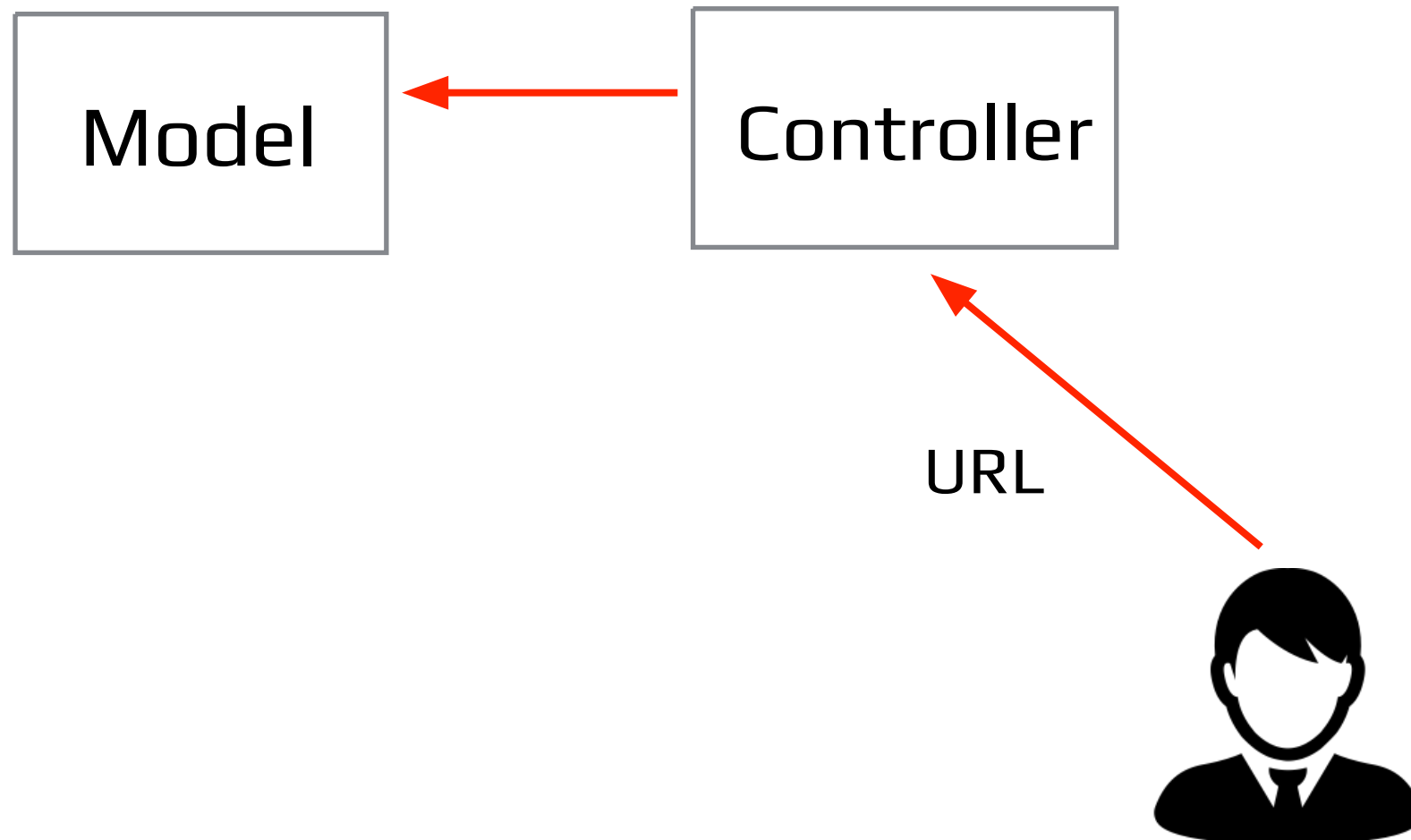
Controller



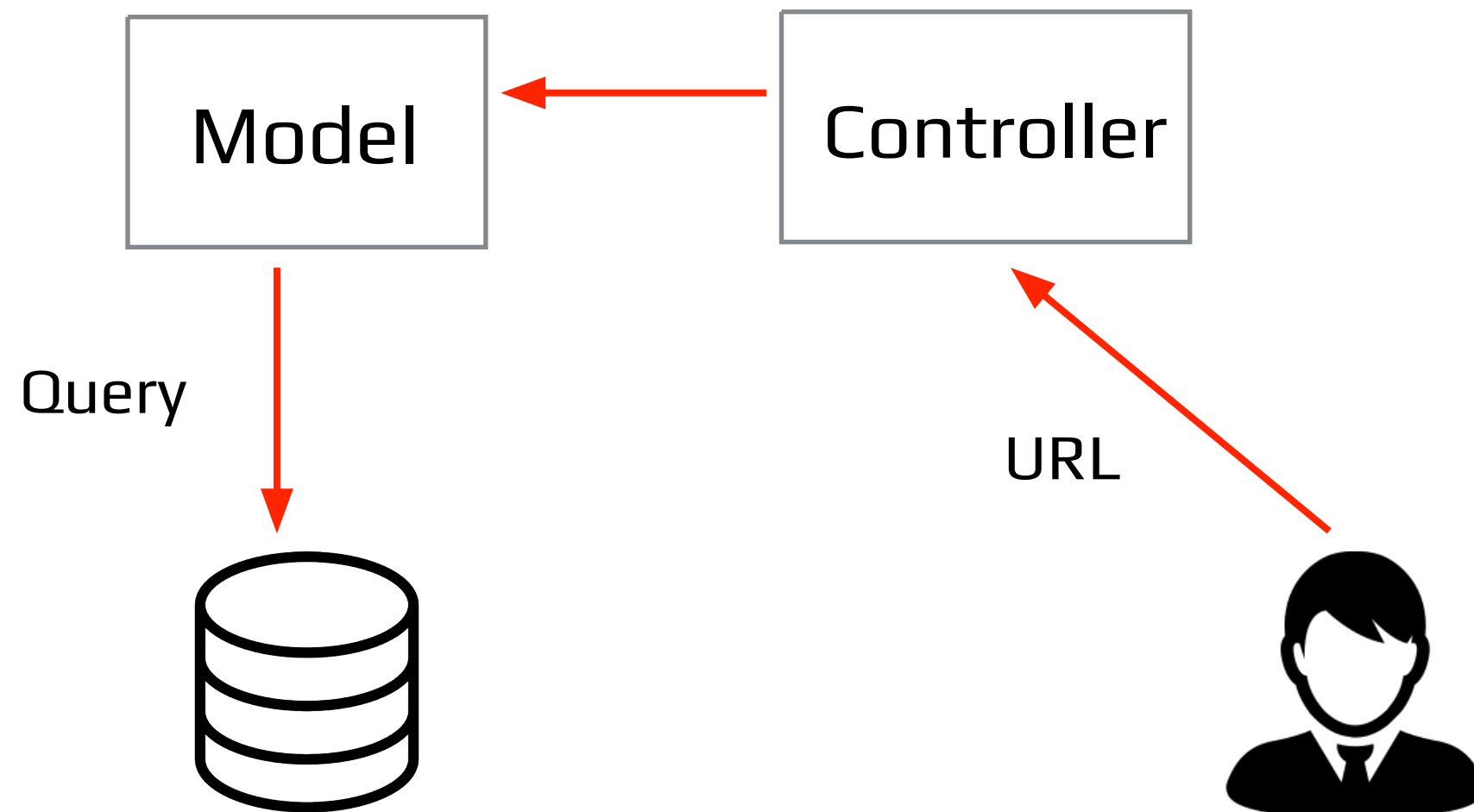
URL



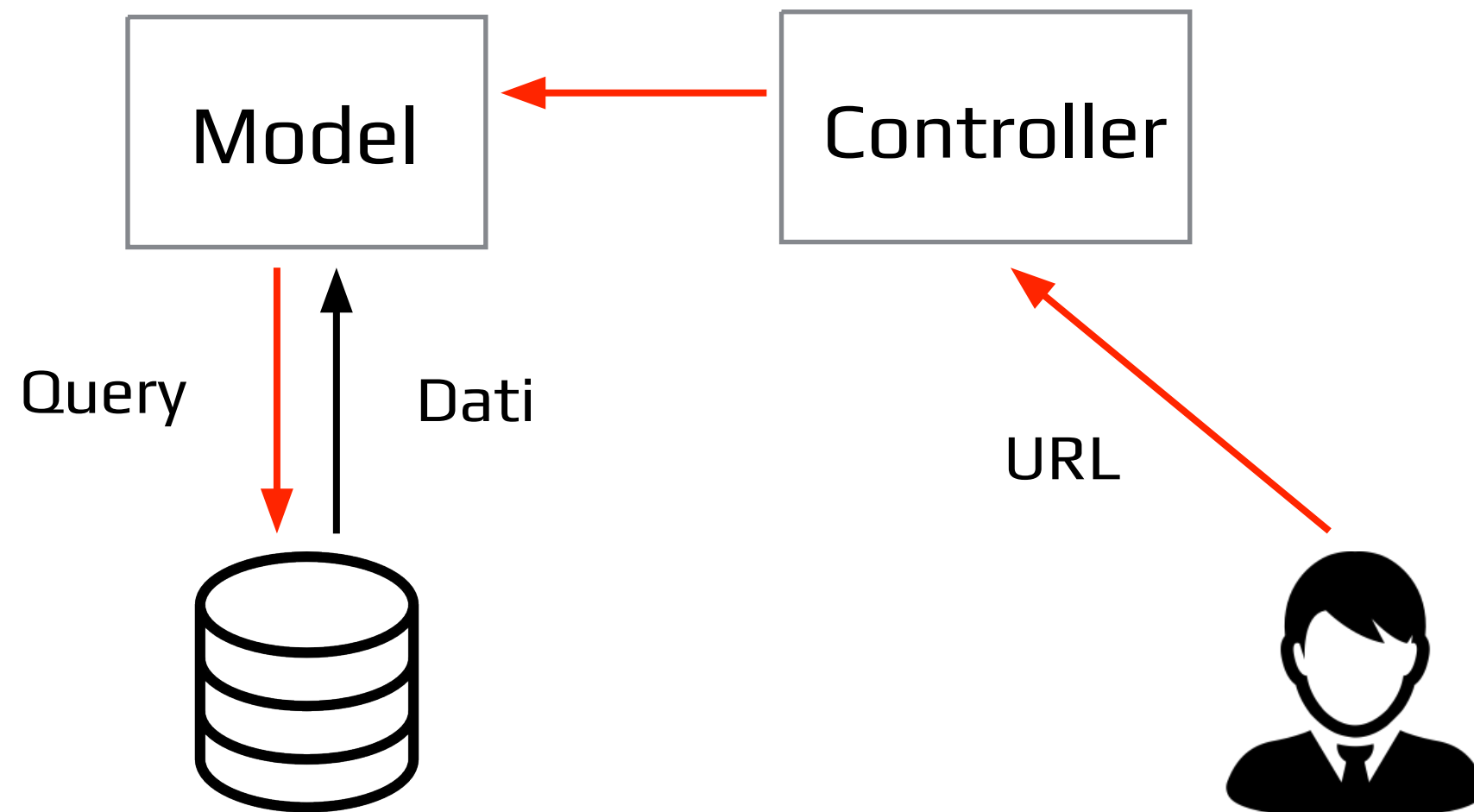
MVC



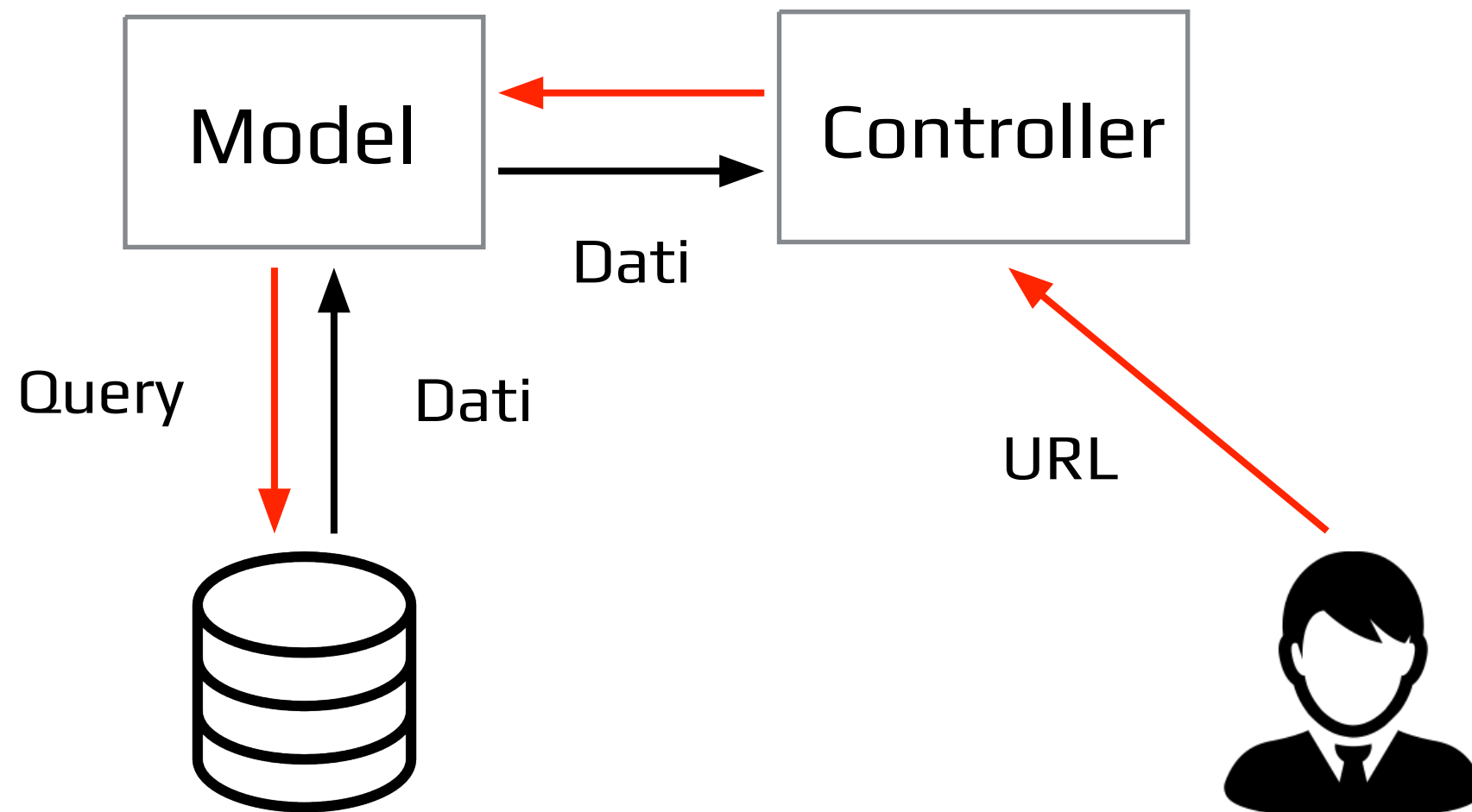
MVC



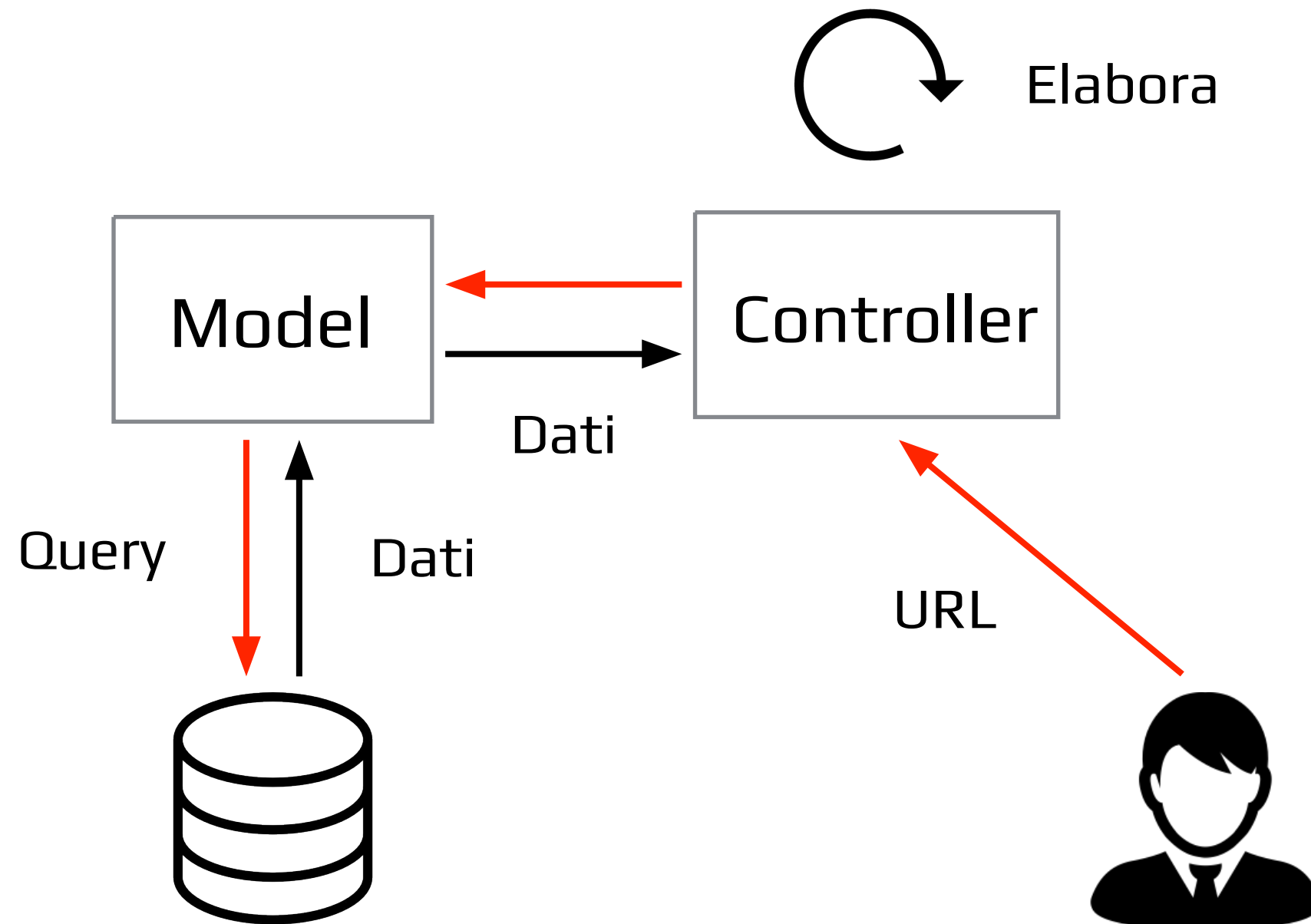
MVC



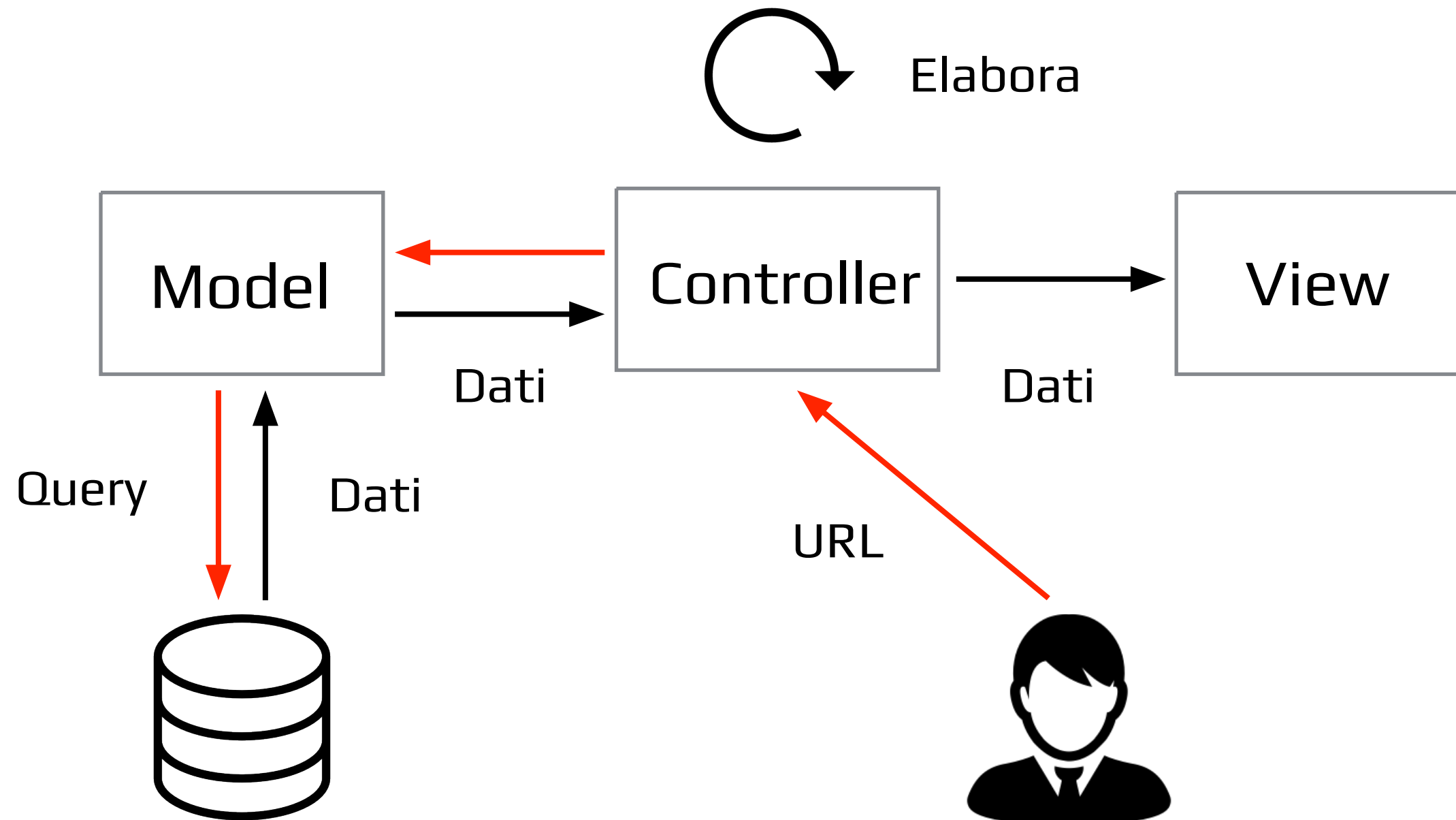
MVC



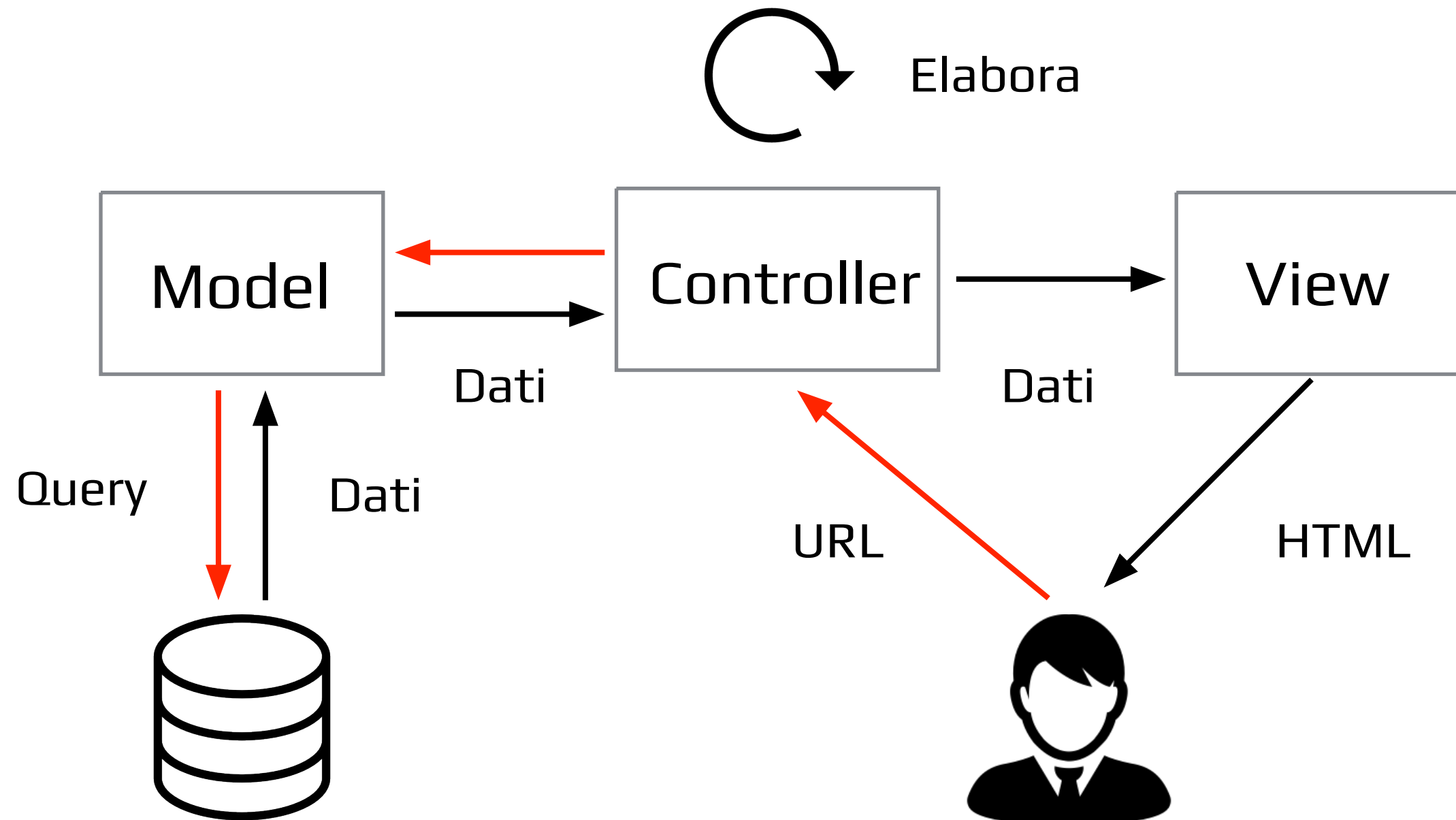
MVC



MVC



MVC



REST

REpresentational State Transfer

REST

L'acronimo REST sta per: **RE**presentational **S**tate **T**ransfer

Non è uno standard, è un modello architetturale introdotto per la prima volta nel 2002 da Roy Fielding nella sua tesi di Laurea alla Università della California.

REST

È un insieme di linee guida per la realizzazione di una “architettura di sistema” in grado di realizzare dei servizi Web (Web Services) per mettere in comunicazione e permettere lo scambio di dati tra Client e Server.

Quando realizziamo un sistema Server in grado di comunicare con i Client attraverso l'utilizzo di REST parliamo di un sistema RESTful.

URL REST

Le caratteristiche di un sistema RESTful sono 4:

- Identificazione univoca delle risorse
- Utilizzo esplicito dei metodi HTTP
- Comunicazione senza stato (Stateless)
- Utilizzo di risorse autodescrittive

Identificazione delle risorse

Per risorsa si intende un qualsiasi elemento oggetto di elaborazione

Il modo di identificare una risorsa in maniera univoca sul Web è utilizzare un URI come ad esempio:

`/clienti`

`/prodotti`

`/ordini`

Metodi HTTP

Le azioni da compiere sulle risorse, sono identificate attraverso i verbi (o metodi) HTTP

Verbo o Metodo	Azione	Descrizione
POST	Create	Crea una risorsa
GET	Read	Ottiene una risorsa
PUT	Update	Aggiorna una risorsa
DELETE	Delete	Cancella una risorsa

Stateless ed autodescrittiva

Basandosi sul protocollo HTTP, il modello ne eredita le limitazioni.

Ogni richiesta deve essere autodescrittiva, deve cioè contenere tutte le informazioni necessarie per essere portata a termine senza fare riferimento a chiamate precedenti.

Es. se voglio ottenere le informazioni su una entità devo fornire l'identificatore di quella determinata entità:

```
GET /prodotti/33
```

REST

In maniera più sintetica l'utilizzo di REST sostituisce le query string, con una sintassi più compatta e più leggibile.

L'URL con la query string:

`/get_prodotto.php?id=33`

diventa:

`GET /prodotti/33`

URL RESTful

POST `www.dominio.com/articoli`

GET `www.dominio.com/articoli`

GET `www.dominio.com/articoli/33`

PUT `www.dominio.com/articoli/33`

DELETE `www.dominio.com/articoli/33`

Web framework

Un Web framework è una “boilerplate” per un ambiente adatto ad facilitare lo sviluppo di un progetto Web.

Quello che fa solitamente un framework Web è fornire:

- una struttura di directory ben definite
- il pattern MVC
- la gestione di url REST
- tante utility



Noi useremo il framework Laravel, il cui sito di riferimento è: <https://laravel.com/>

Molte funzioni del framework si utilizzano attraverso la riga di comando (CLI) usando il comando:

```
$ php artisan <comando> [opzioni]
```

Ho preparato un piccolo manuale per aiutarvi con i comandi di base: bit.ly/360ka46



È giunto alla versione 9.x è quindi un framework maturo, stabile, ben consolidato

Usa il linguaggio PHP che è semplice da comprendere e molto performante (nella sua versione 7.x)

È estendibile attraverso i package

Implementa molte funzioni utili senza dover aggiungere package esterni (come l'autenticazione)

È molto ben documentato: <https://laravel.com/docs/>



Laravel utilizza diverse convenzioni COC
(Conventions Over Configuration)

Per i nomi dei Modelli, dei Controller, delle View, ma anche convenzioni stilistiche per la scrittura del codice.

Laravel naming conventions:

<https://xqsit94.github.io/laravel-coding-guidelines/naming-conventions.html>

Code style PSR-12:

<https://www.php-fig.org/psr/psr-12/>

Demo

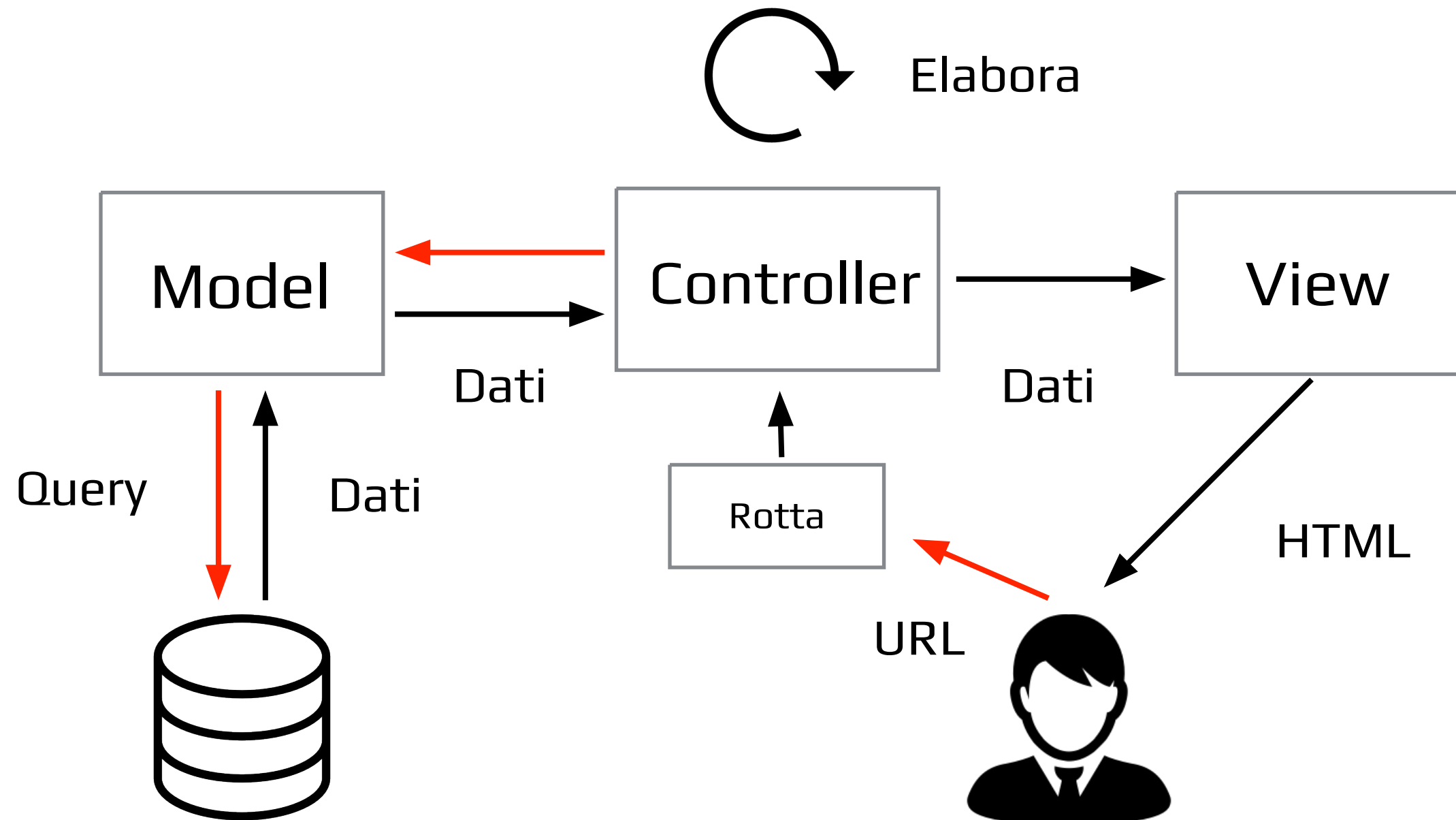
Istanziamento di un progetto Web su framework Laravel

URL

Gli URL hanno il compito di determinare quale controller e quale metodo invocare all'interno del nostro codice:

Rotta: Mappa l'URL + Verbo HTTP in una chiamata ad un determinato metodo di un determinato controller.

MVC



Rotte

Le rotte rendono possibile l'implementazione del modello REST:

`www.mioblog.com/articoli/33`

Verbo + Path => Action

GET `articoli/{id}` => 'ArticoloController@nomeMetodo'

Demo

Rotte, View e Controller

Laravel Views

- Creare una nuova View
- Sintassi Blade
- Inserire file CSS e JS (assets)

Laravel Controllers

- Creare un nuovo controller
- Legare una rotta ad un controller
- Passare i dati ad una View

View e Controller

- Passaggio di parametri da url
- Gestione di un Form
- Layouts

ORM

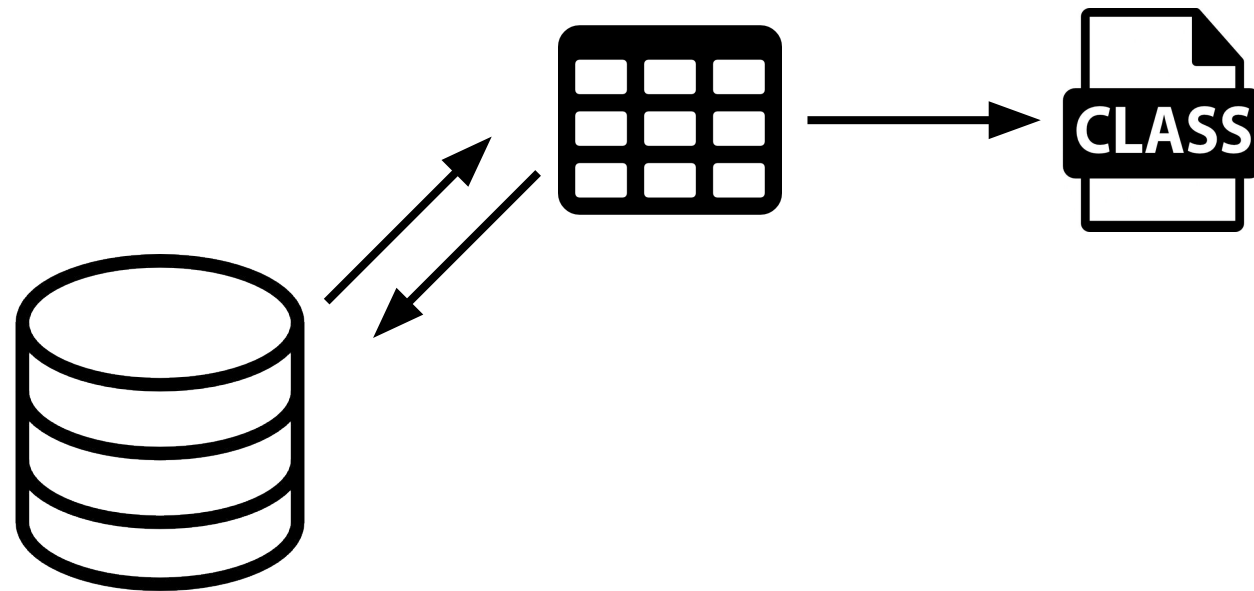
Object Relational Mapping

Object Relational Mapping

Un ORM fornisce, mediante un'interfaccia orientata agli oggetti, tutti i servizi utili alla persistenza dei dati fornendo una astrazione dalle caratteristiche implementative del RDBMS utilizzato.

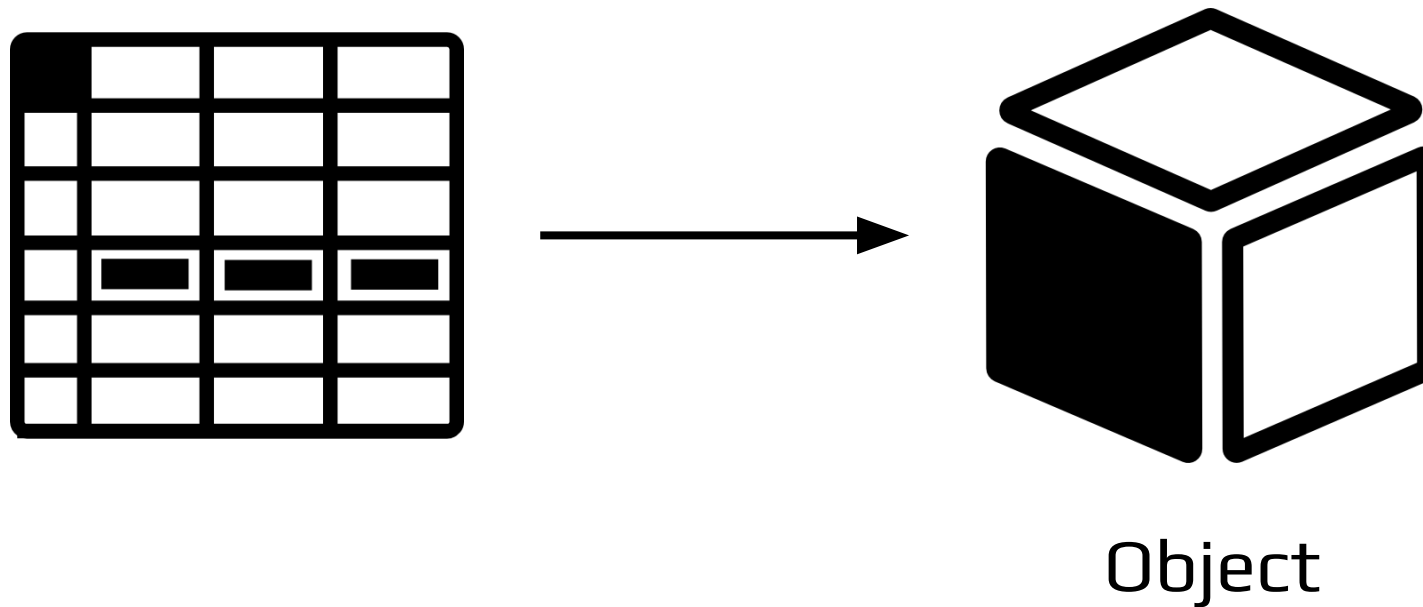
Object Relational Mapping

In altre parole, un ORM mappa le entità di un database in una classe.



Object Relational Mapping

E le tuple di una entità in un oggetto della classe.



Object Relational Mapping

Facciamo un esempio classico dalla lezione 5

id	first_name	last_name	salary
1	Mario	Rossi	20000
2	Luigi	Albertini	22000
3	Piero	Solerti	21500
4	Luisa	Bianchi	21500

```
class Employee extends ORM {  
  
    private $id;  
    private $first_name;  
    private $last_name;  
    private $salary;  
  
}
```

Object Relational Mapping

L'ORM mette a disposizione dello sviluppatore una serie di funzioni per gestire i dati, come ad esempio per estrarli dal database...

id	first_name	last_name	salary
1	Mario	Rossi	20000
2	Luigi	Albertini	22000
3	Piero	Solerti	21500
4	Luisa	Bianchi	21500

```
class Employee extends ORM ...
```

```
$emp = Employee::find(2);
```

Object Relational Mapping

... o salvarli nel database

id	first_name	last_name	salary
1	Mario	Rossi	20000
2	Luigi	Albertini	23000
3	Piero	Solerti	21500
4	Luisa	Bianchi	21500

```
class Employee extends ORM ...
```

```
$emp->salary = 23000;  
$emp->save();
```

Object Relational Mapping

Posso estrarre un elemento specifico, tutti oppure costruire query più complesse...

```
$emp = Employee::find(2) ;
```

```
$emps = Employee::all() ;
```

```
$emps = Employee::where('salary', '>', 21000) ;
```

Eloquent

Laravel ha un ORM il cui nome è Eloquent e tutti i modelli per poter utilizzare l'ORM devono estendere la classe Model

Demo

Model

Dove trovare aiuto

Per approfondire:

<https://laracasts.com/>

Per chi non frequenta o per non si accontenta:

<https://laracasts.com/series/laravel-8-from-scratch>

Domande ?