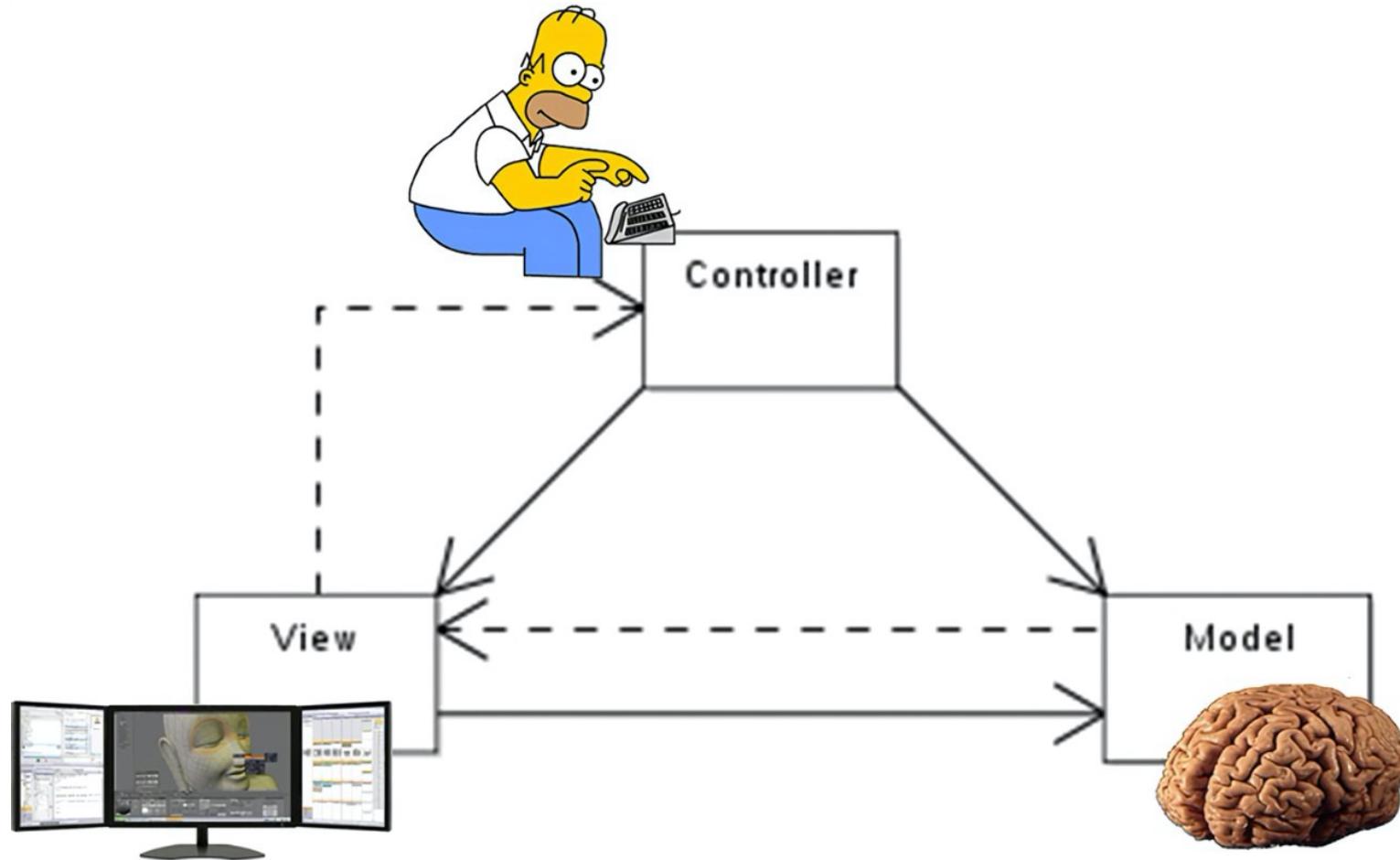


# **Technologies for E- Commerce**

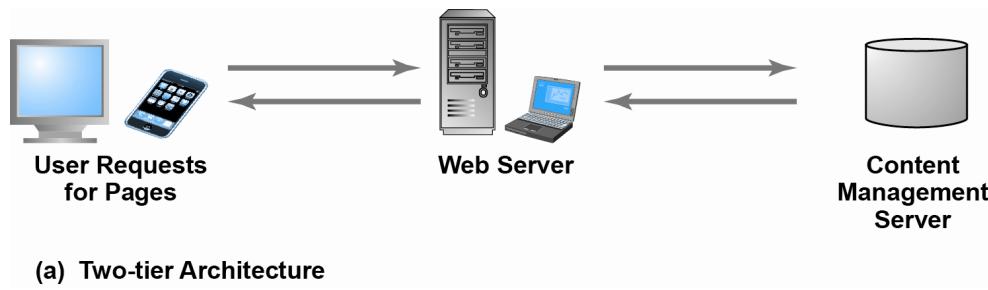
**CAN302**

**Department of Communications and Networking  
Xi'an Jiaotong-Liverpool University (XJTLU)**

# Week 5 – MVC & Front/Back Split

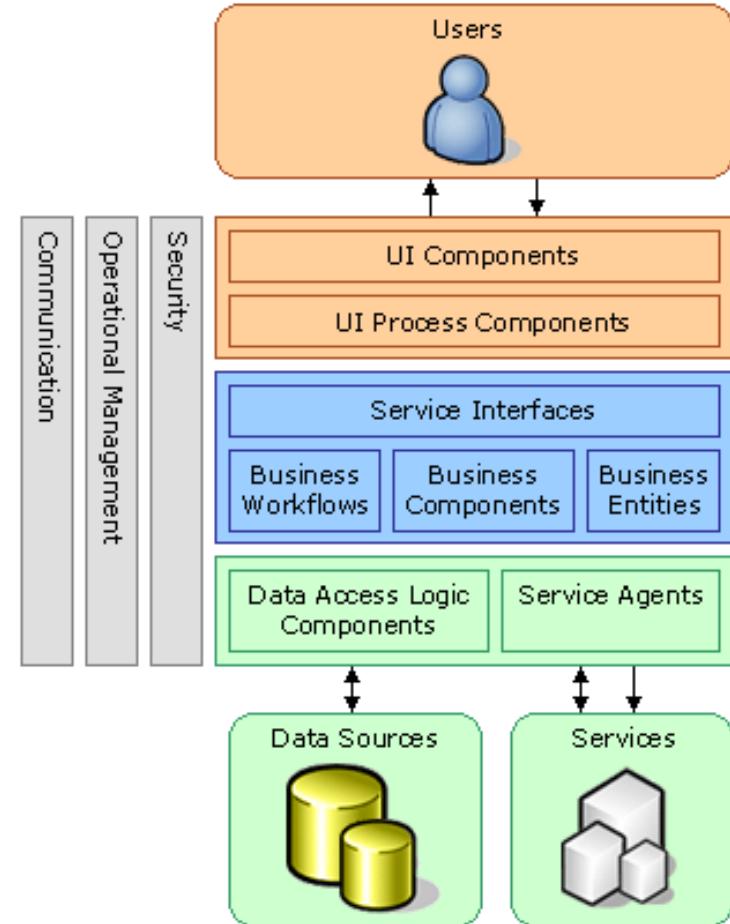


# Two and three (multi) layers



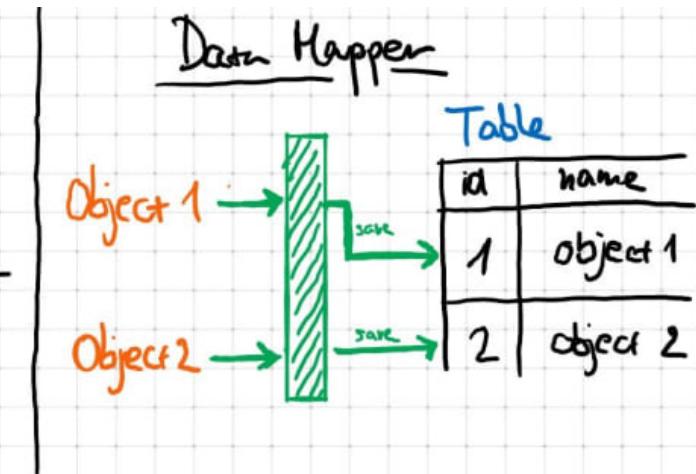
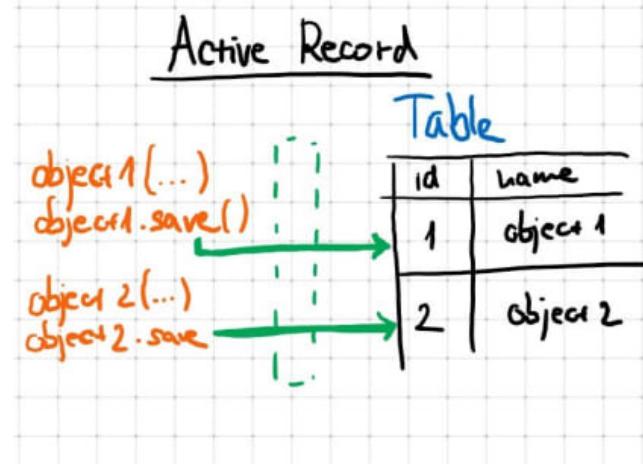
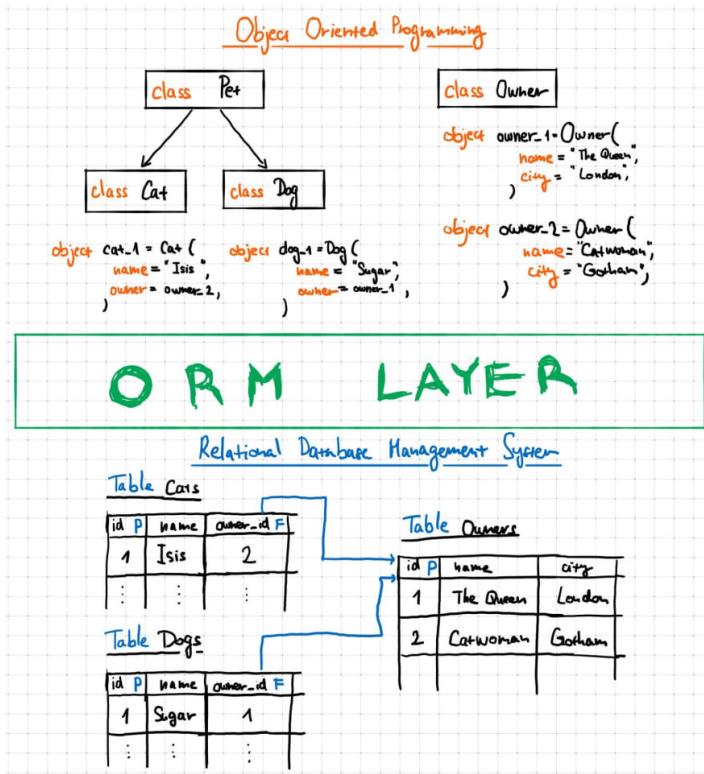
(a) Two-tier Architecture

- Why split?
  - Separate the codes to components
- What is the guideline for splitting?
  - Microsoft had an architecture as: UI, BL and DA



[https://docs.microsoft.com/en-us/previous-versions/msp-n-p/ff647339\(v=pandp.10\)](https://docs.microsoft.com/en-us/previous-versions/msp-n-p/ff647339(v=pandp.10))

# Object Relational Mapping



<https://dev.to/tinazhouhui/introduction-to-object-relational-mapping-the-what-why-when-and-how-of-orm-nb2>  
<https://stackoverflow.com/questions/1279613/what-is-an-orm-how-does-it-work-and-how-should-i-use-one>

# A sample of ORM

```
class User(db.Model):
    username = db.Column(db.String(80), unique=True)
    pw_hash = db.Column(db.String(80))
```

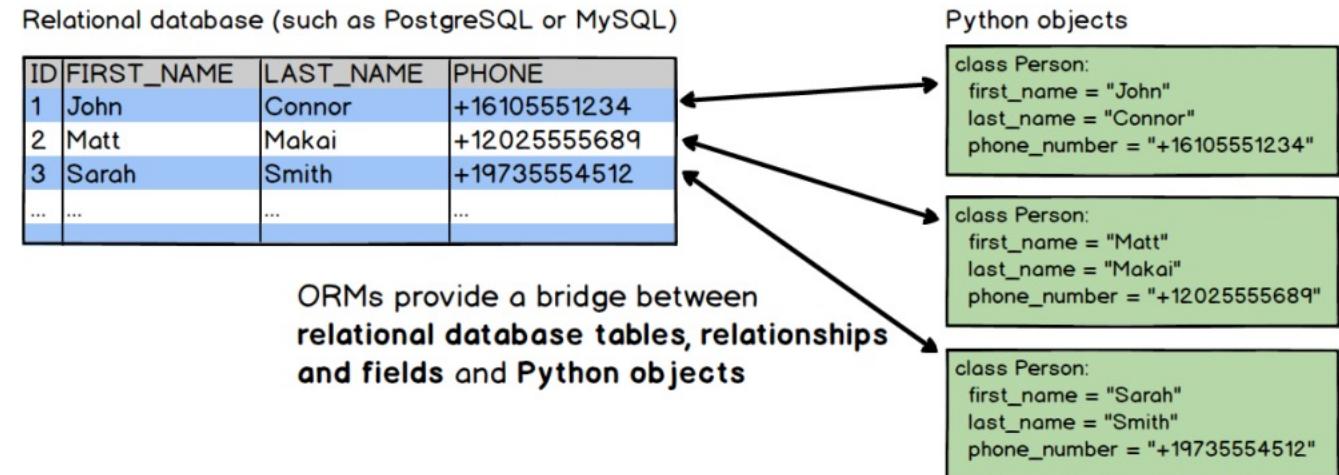
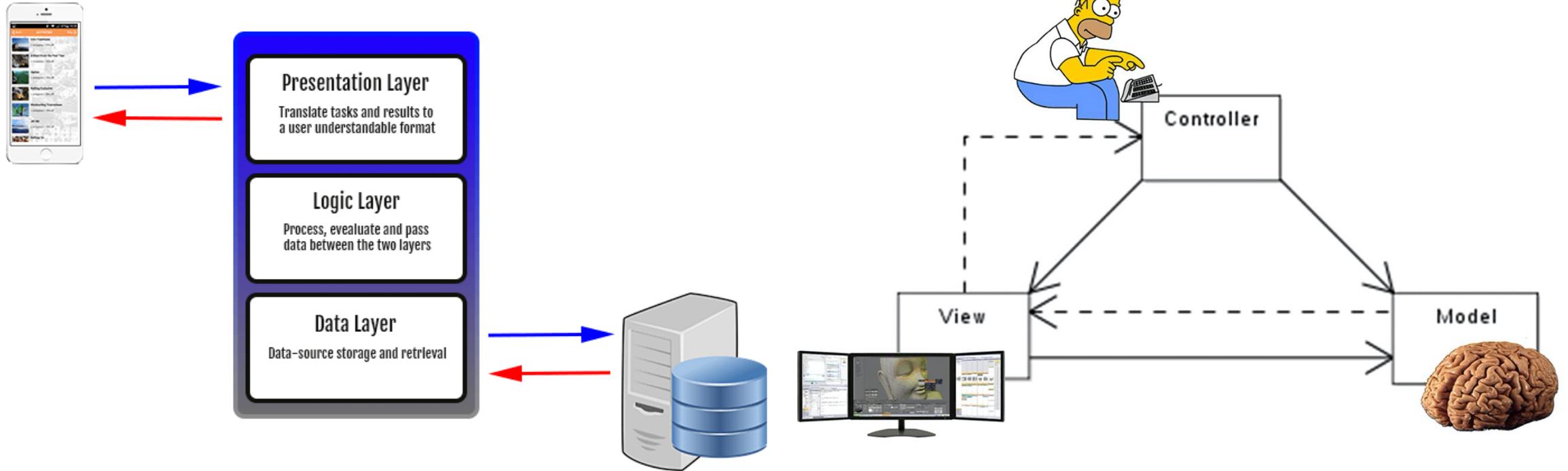


Figure 0: Object Relational Mapping

[https://cs.ulb.ac.be/public/\\_media/teaching/entityframework\\_2017.pdf](https://cs.ulb.ac.be/public/_media/teaching/entityframework_2017.pdf)

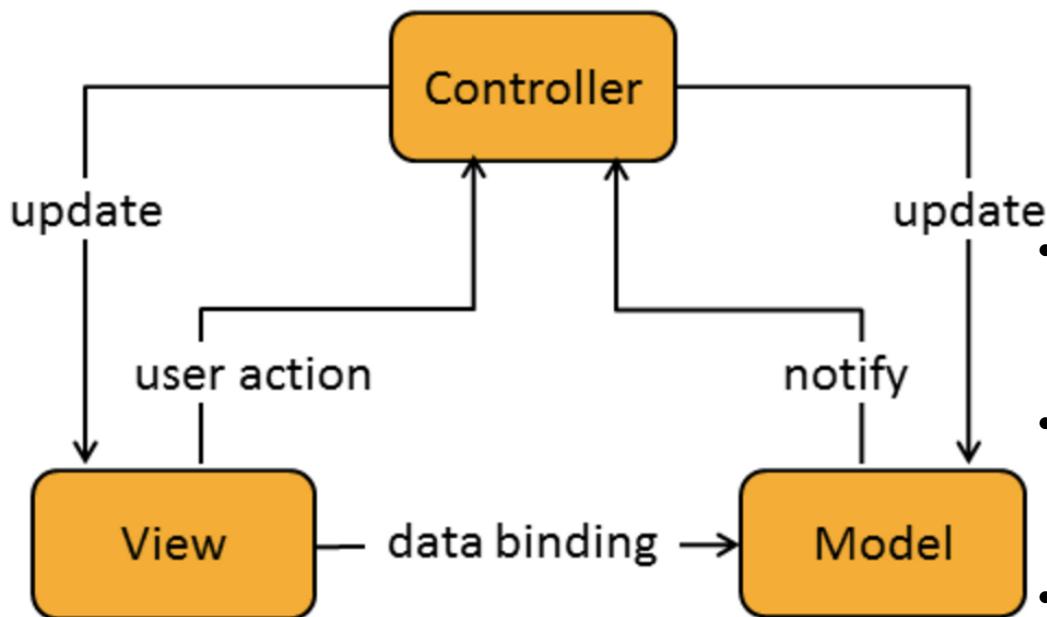
<http://www.pythondoc.com/flask-sqlalchemy/api.html#flask.ext.sqlalchemy.SQLAlchemy>

# Three layers and MVC



**Discussion I : Does three layers corresponding to MVC?**

# MVC



- MVC is an architecture helps to **split** applications into logical units. MVC is much old than web technology. Many time, you cannot cut the application so clear. Microsoft three layers is a good split but they are **NOT** corresponding to MVC one by one.
- In simple terms, this paradigm separates **business** logic from **interface** logic.
- The **model** represents a unique entity - it could be a single object or more likely a structure. In this way, the processing of data takes place only in the model, which ensures internal data consistency.
- The **view** is used to present the graphical visualization of the user interface. **To see the status of the application objects**, the view queries the model through the controller.
- The **controller** provides the link between the user interface (view) and the application processing logic (model). In a sense the controller enables a user to make changes and see results.

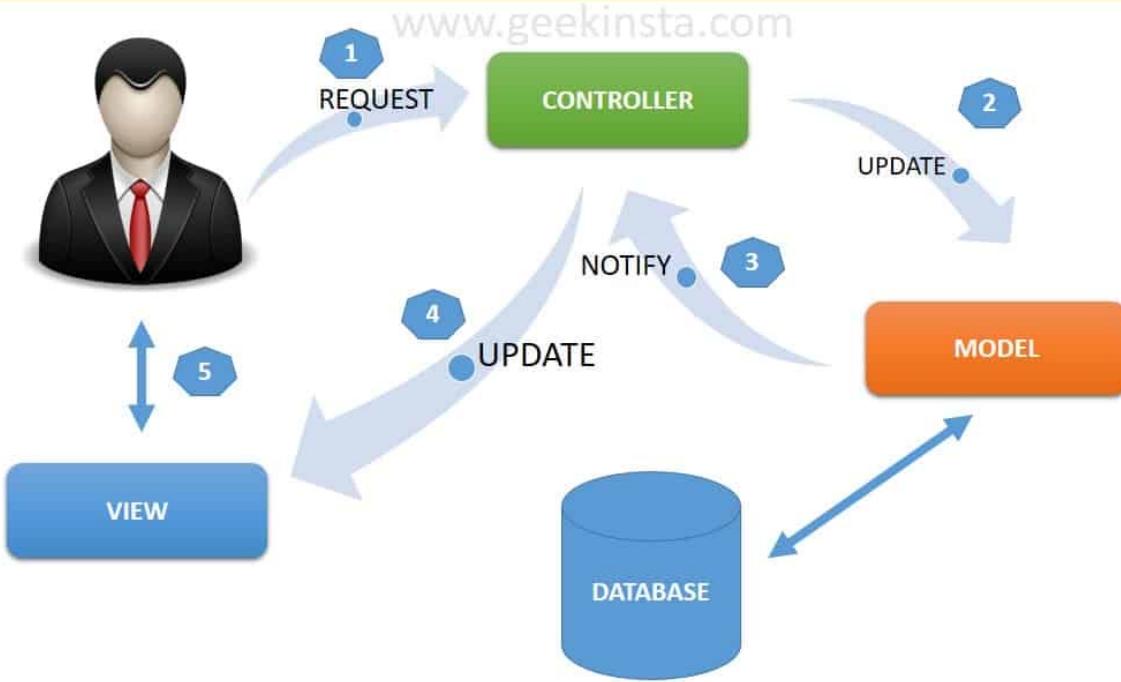
<https://blogs.sap.com/2016/01/02/an-insight-into-model-view-controller-mvc-in-the-context-of-sap-ui5/>

<https://www.techopedia.com/definition/3842/model-view-controller-mvc>

<https://www.w3schools.in/mvc-architecture/>

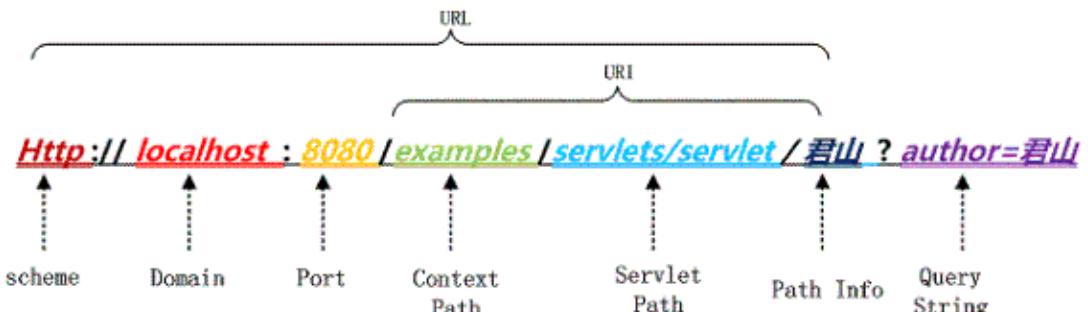
# MVC

## MVC DESIGN PATTERN



- The classic MVC pattern works by managing the state of an application.
- When a user performs an action or makes a request, an action in the Controller is called.
- The Controller then either tells Model to make changes and update the View or returns a View based on a Model.
- Hence we can say that the View is controlled by the Controller and Model.

# URI – controller

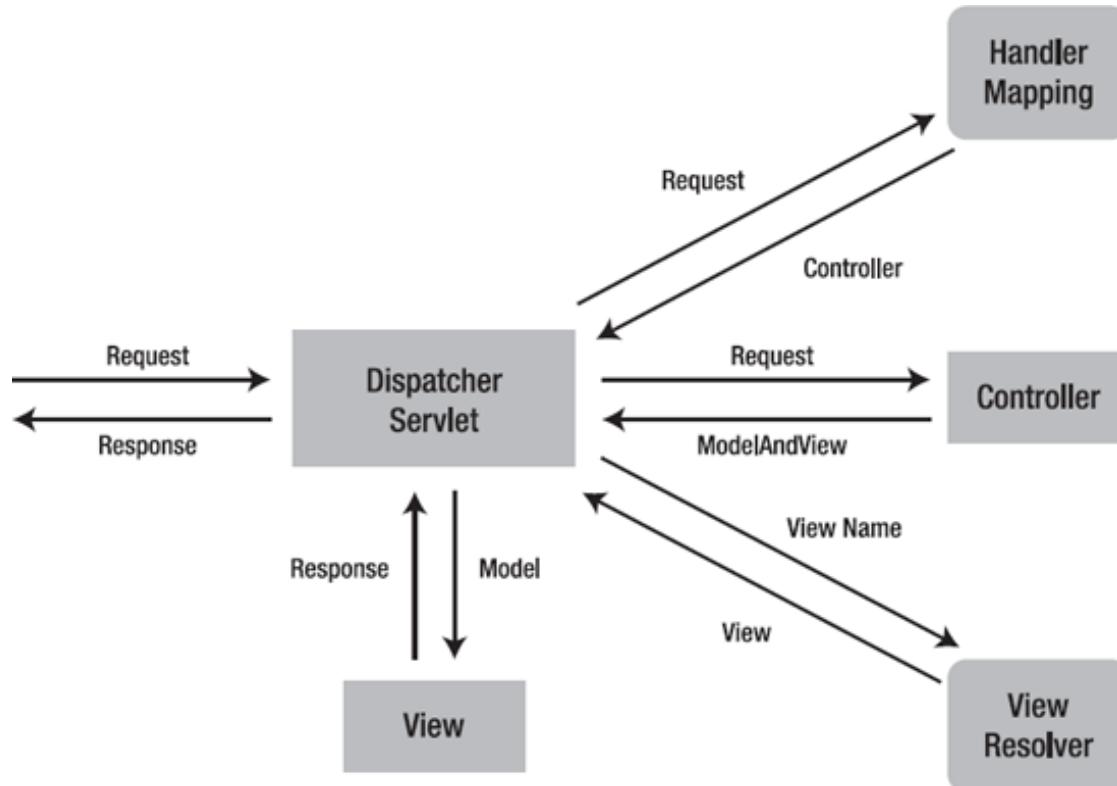
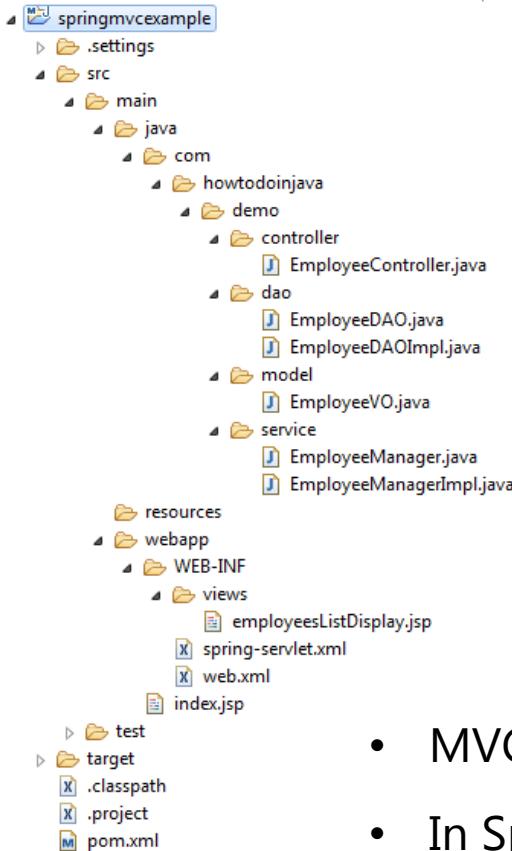


```
1 | from flask import Flask
2 | app = Flask(__name__)
3 |
4 | @app.route('/')
5 | def hello_world():
6 |     return 'Hello World!'
7 |
8 | if __name__ == '__main__':
9 |     app.run()
```

```
1 | @Controller
2 | @RequestMapping(value = "/", method = RequestMethod.GET)
3 | public class ContactController {
4 |
5 |     @RequestMapping(value = "/", method = RequestMethod.GET)
6 |     public String redirectToContactPage() {
7 |         return "redirect:contact";
8 |     }
9 |
10 |     @RequestMapping(value = "/admin", method = RequestMethod.GET)
11 |     public String toAdminPage() {
12 |         return "admin";
13 |     }
14 |
15 |     @RequestMapping(value = "/contact", method = RequestMethod.GET)
16 |     public String toContactForOthersPage() {
17 |         return "contact";
18 |     }
19 |
20 | }
```

<https://www.jianshu.com/p/27b01fb4d688>  
<https://www.jianshu.com/p/59b6f19fcfc>

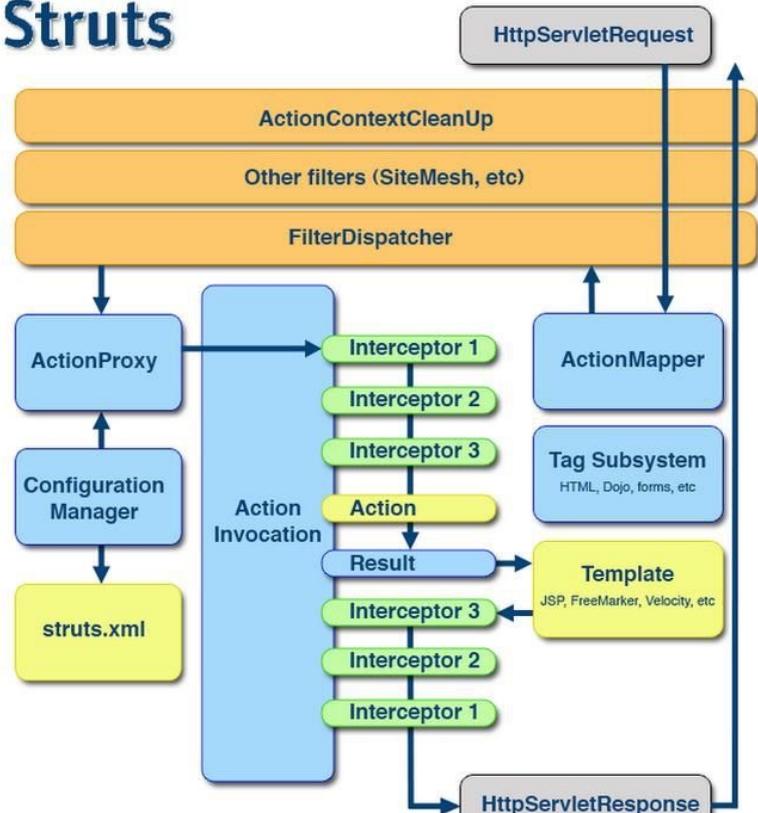
# Spring MVC – “Hello world”



- MVC is only a guideline to split components.
- In Spring MVC, there is a central controller, every request must go to controller and find its corresponding route.
- Understand the router and URI

# Struts and Spring MVC

## Struts



Key:

Servlet Filters

Struts Core

Interceptors

User created

知乎 @夏微凉秋微暖

SSH: Spring, Struts2 and Hibernate

JSP is used for view layer.



知乎 @夏微凉秋微暖

SSM: Spring, SpringMVC and Mybatis

<https://zhuanlan.zhihu.com/p/44149566>

<https://www.zhihu.com/question/22014461>

# Hibernate and Mybatis



**HIBERNATE**

**VS**



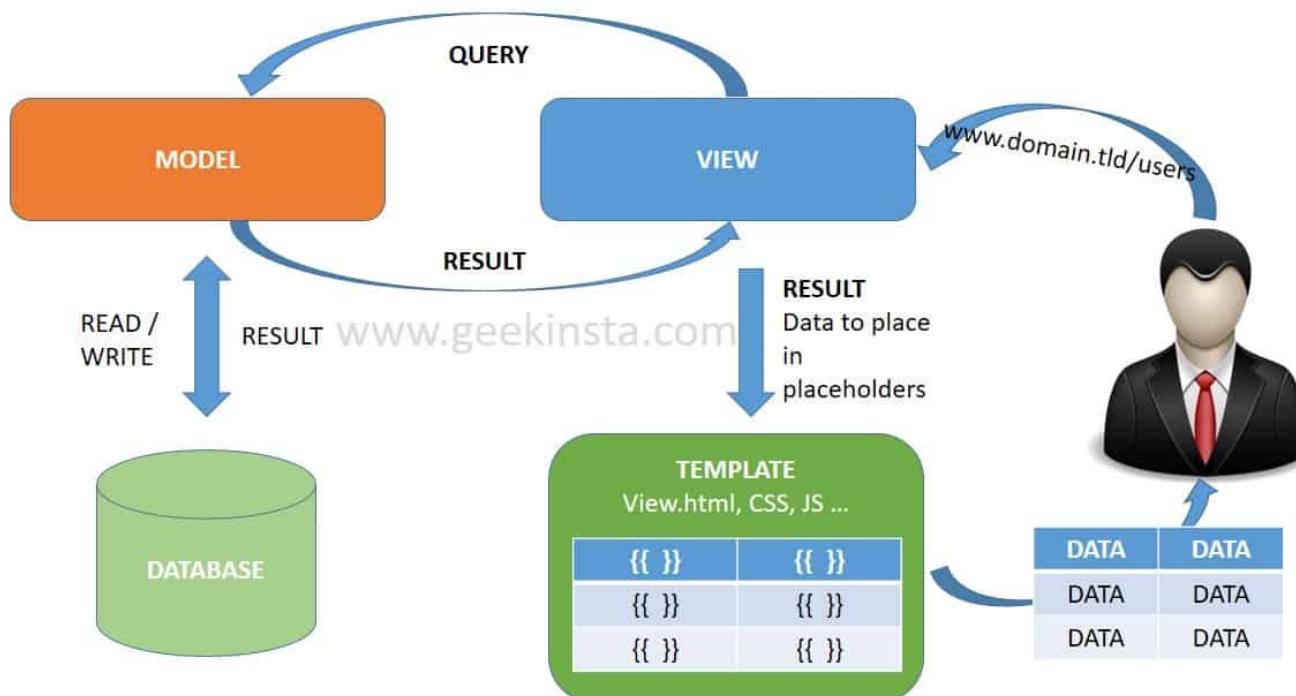
**MYBATIS**

- General CRUD(CREATE, READ, UPDATE, and DELETE ) functionality.
- The environment is driven by object models and needs to generate SQL automatically.
- Session management.
- Analytic fetch queries.
- Stored procedures and dynamic SQL.
- Support complicated search queries, where search criteria are dynamic, and paging of results.

<https://www.perfomatix.com/hibernate-vs-mybatis-java-development-company/>  
<https://www.zhihu.com/question/309662829>

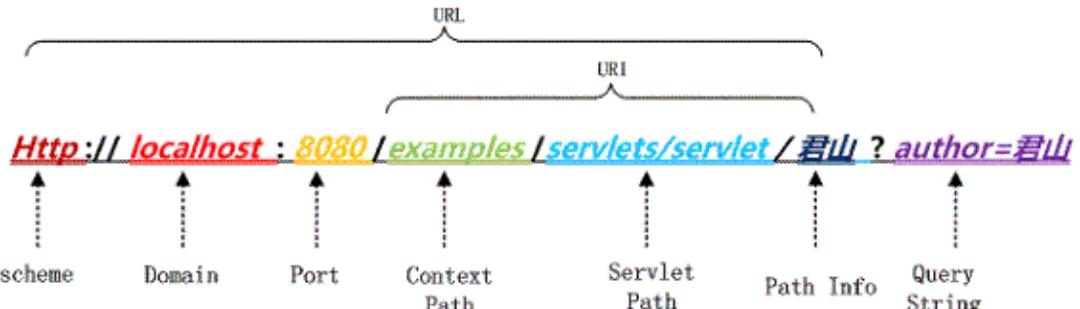
# MVT

## MVT DESIGN PATTERN



- MVT takes a slightly different approach and introduced by Python Django.
- When a user makes an HTTP request, the corresponding view performs a query on the Model and collects the result set from the Model.
- The View then fills the result in a template and sends it to the user.

# Does native PHP has controller?



```
1 | from flask import Flask
2 | app = Flask(__name__)
3 |
4 | @app.route('/')
5 | def hello_world():
6 |     return 'Hello World!'
7 |
8 | if __name__ == '__main__':
9 |     app.run()
```

```
1 | @Controller
2 | @RequestMapping(value = "/", method = RequestMethod.GET)
3 | public class ContactController {
4 |
5 |     @RequestMapping(value = "/", method = RequestMethod.GET)
6 |     public String redirectToContactPage() {
7 |         return "redirect:contact";
8 |     }
9 |
10 |     @RequestMapping(value = "/admin", method = RequestMethod.GET)
11 |     public String toAdminPage() {
12 |         return "admin";
13 |     }
14 |
15 |     @RequestMapping(value = "/contact", method = RequestMethod.GET)
16 |     public String toContactForOthersPage() {
17 |         return "contact";
18 |     }
19 |
20 | }
```

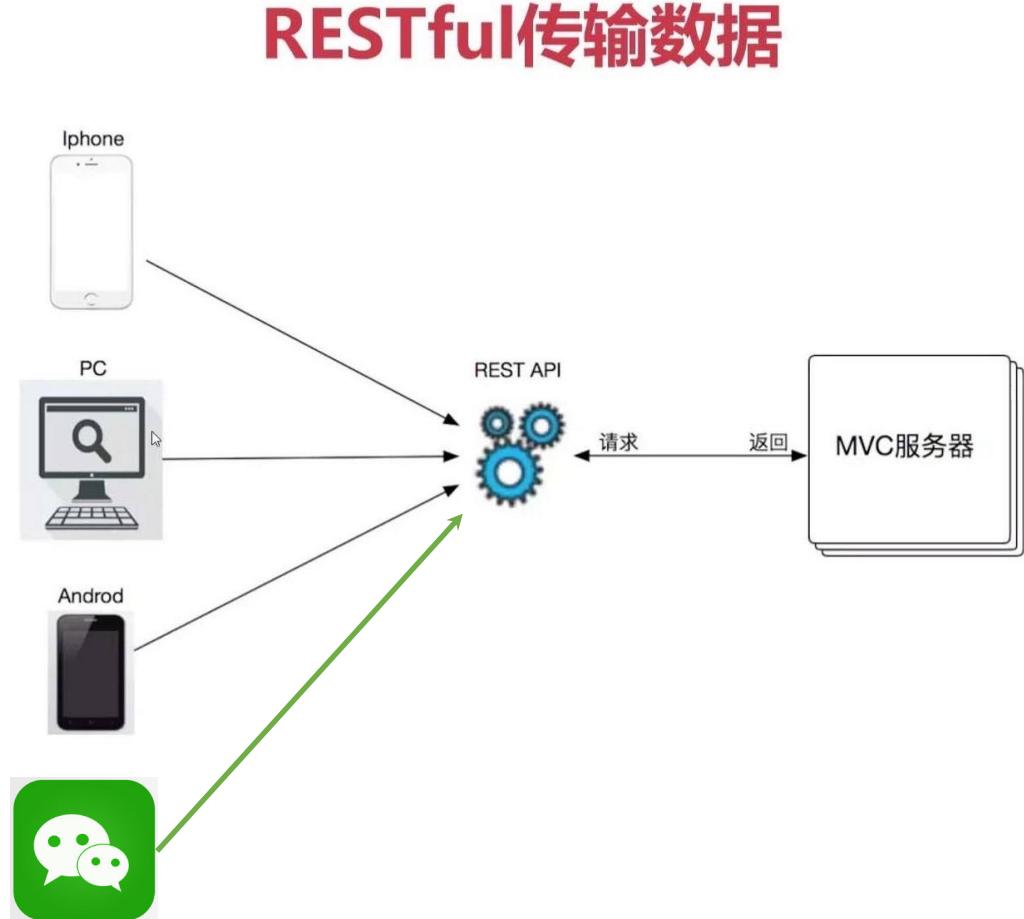
# Some PHP frameworks



**ThinkPHP®**



# Multi Front-APP age

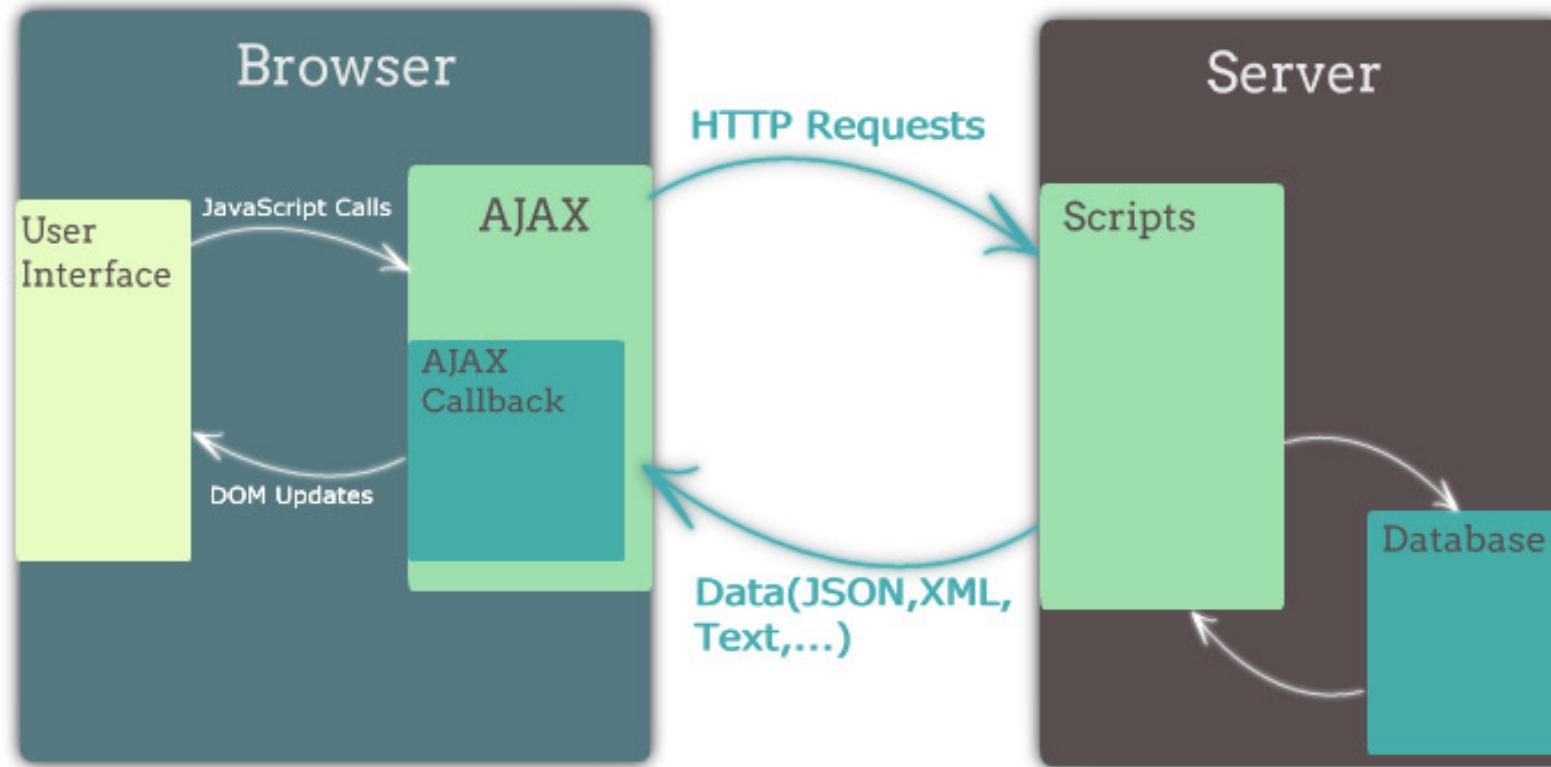


- Many new front-ends in mobile age, the main types are: PC web, mobile web, iOS/android/HarmonyOS native APP and **WeChat** application.
- Their appearance are very different from each other while back-end logic are most same.
- Let the present layer define their layouts and back-end process the data is a way to split.
- Restful APIs can support different front-ends.

# F/B split:



# Web - AJAX (Asynchronous JavaScript And XML)



- Ajax enable browser (front-end) to interact with web server (back-end) within the same webpage.
- Ajax can be applied to make dynamic front-end developing.

<https://developer.mozilla.org/en-US/docs/Web/Guide/AJAX>

# XML

```
<?xml version="1.0" encoding="UTF-8"?>
<teacheres>
  <teacher position="CEO">
    <number>1</number>
    <name>刘老师</name>
    <sex>男</sex>
    <phone>808</phone>
  </teacher>
  <teacher>
```

根节点  
开始标签  
根节点的子节点  
number的父节点

XML信息：  
版本号和编码格式  
节点的属性

- Within a webpage, using JS can get XML format data and replace the original webpage

<https://www.cnblogs.com/web-learn/p/15228702.html>

# XML & json

```
<employees>
  <employee>
    <firstName>Bill</firstName>
    <lastName>Gates</lastName>
  </employee>
  <employee>
    <firstName>Steve</firstName>
    <lastName>Jobs</lastName>
  </employee>
  <employee>
    <firstName>Elon</firstName>
    <lastName>Musk</lastName>
  </employee>
</employees>
```

```
{"employees": [
  { "firstName":"Bill", "lastName":"Gates" },
  { "firstName":"Steve", "lastName":"Jobs" },
  { "firstName":"Elon", "lastName":"Musk" }
]}
```

- JSON - JavaScript Object Notation, a better format for reading.
- JSON is more popular.

[https://www.w3school.com.cn/js/js\\_json\\_xml.asp](https://www.w3school.com.cn/js/js_json_xml.asp)

<https://www.toptal.com/web/json-vs-xml-part-1>

<https://twobithistory.org/2017/09/21/the-rise-and-rise-of-json.html>

# A sample page with jQuery

```
<!doctype html>
<html lang="en">
<head>
  <meta charset="utf-8">
  <meta http-equiv="X-UA-Compatible" content="IE=edge">
  <title>jQuery AJAX example</title>
</head>
<body>  droplist of id="states"
<select id="states">
  <option>AK</option>
  <option>HI</option>
  <option>NY</option>
  <option>TX</option>
</select>
<p>It's capital is <span id="capital"></span></p>

<script src="http://codeorigin.jquery.com/jquery-2.0.3.min.js">
</script>
<script>
$(document).ready(function() {
  });
</script>      jQuery stuff
</body>
</html>
```

```
<script>
$(document).ready(function() {
  $('#states').change(function() {
    selectedState = $("#states option:selected").text();
    $.get('computecapitals.php?state=' + selectedState,
    function( data ) {
      $('#capital').html( data );
    });
  });
});
</script>
```

The selected state in dropdown: AK, HI, TX, etc

AJAX call

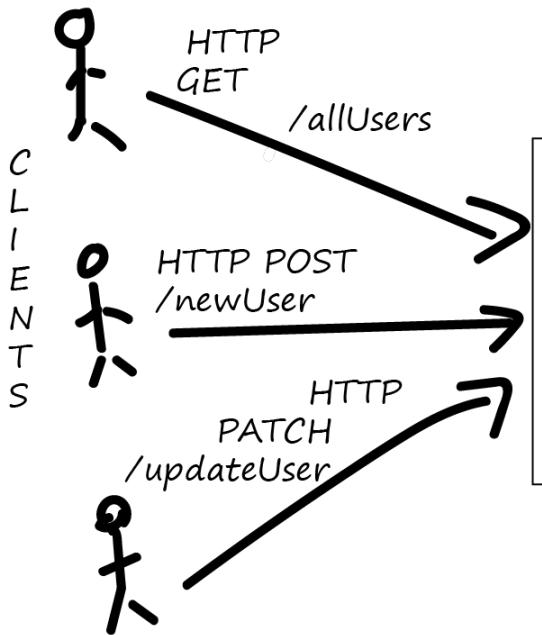
pass the URL

pass the callback function which does this with the retrieved data

- jQuery is a JS project.
- It likes an SDK which has more functions than Ajax.
- Bootstrap includes jQuery

# Rest API

## Rest API Basics

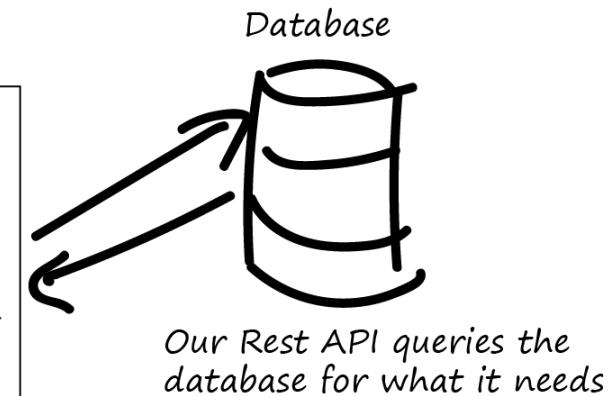


Our Clients, send HTTP Requests and wait for responses

### Rest API

Receives HTTP requests from Clients and does whatever request needs. i.e create users

Typical HTTP Verbs:  
GET -> Read from Database  
PUT -> Update/Replace row in Database  
PATCH -> Update/Modify row in Database  
POST -> Create a new record in the database  
DELETE -> Delete from the database



Our Rest API queries the database for what it needs

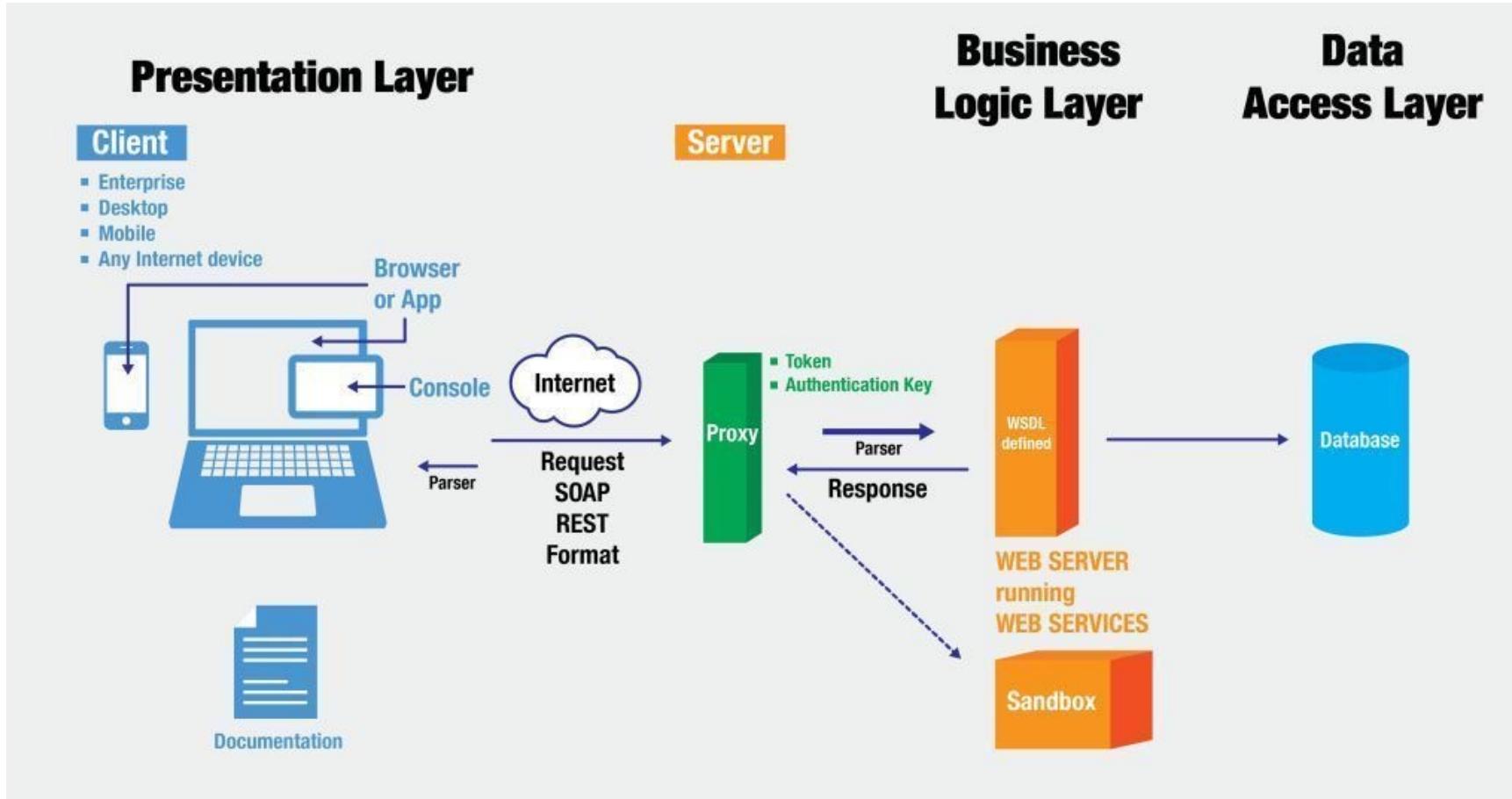
Response: When the Rest API has what it needs, it sends back a response to the clients. This would typically be in JSON or XML format.

**REST** - Representational State Transfer.

**API** - Application Programming Interface

<https://tutorialedge.net/software-eng/what-is-a-rest-api/>

# A new three layers



- There is still view in back-end, what is it?

# View may **NOT** be the webpage

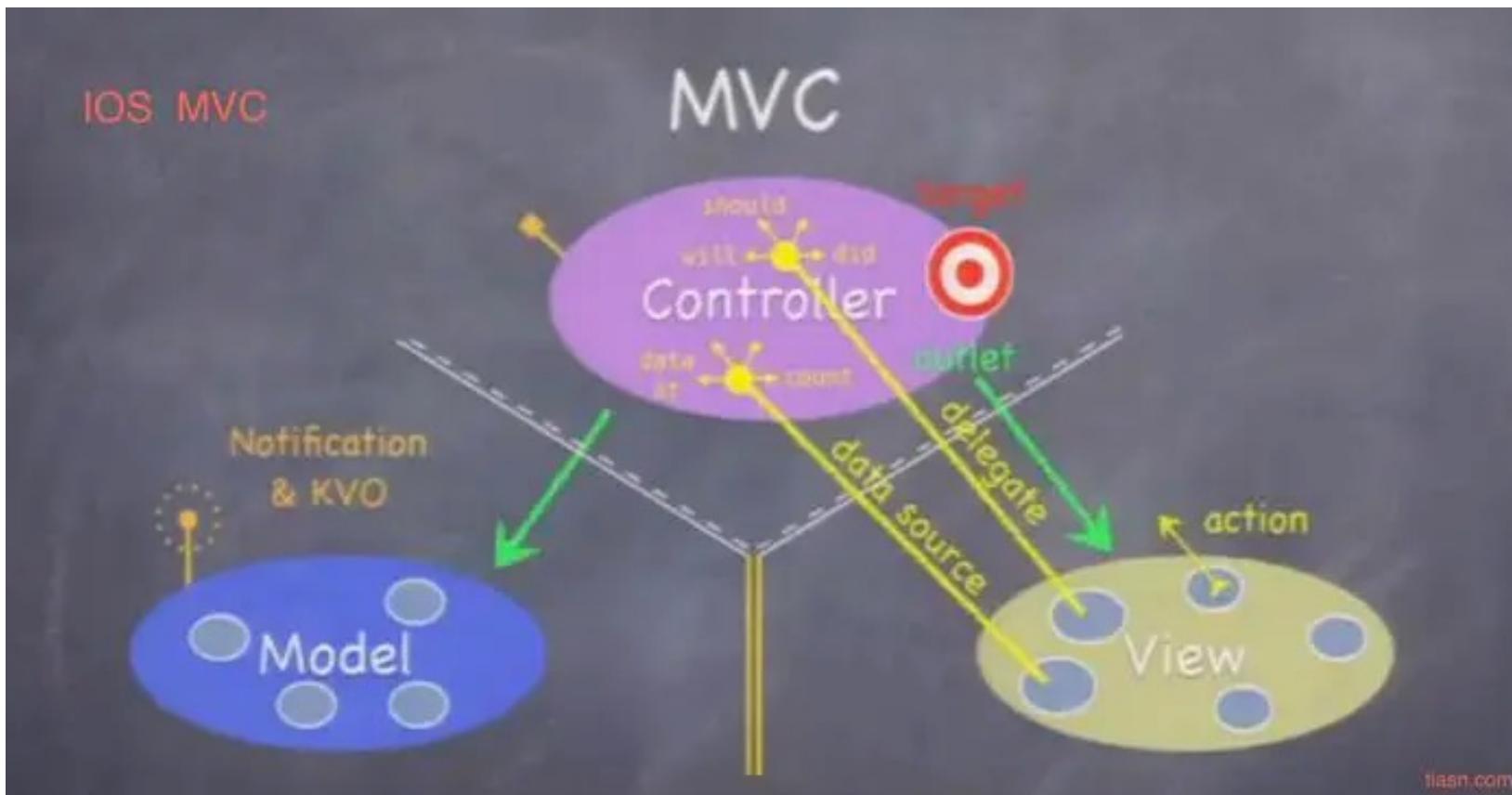


- JSON view .

```
{  
    "Type": "sap.ui.core.mvc.JSONView",  
  
    "controllerName": "sap.hcm.Address",  
  
    "content": [  
  
        {"Type": "sap.ui.commons.Image",  
  
            "id": "MyImage",  
  
            "src": "http://www.sap.com/global/ui/images/global/sap-logo.png"  
  
        },  
  
        {  
  
            "Type": "sap.ui.commons.Button",  
  
            "id": "MyButton",  
  
            "text": "Press Me"  
  
        }  
    ]  
}
```

- The **view** is used to present the graphical visualization of the user interface.
- **To see the status of the application objects**, the view queries the model through the controller.
- How about your target audiences are machines but not human beings?

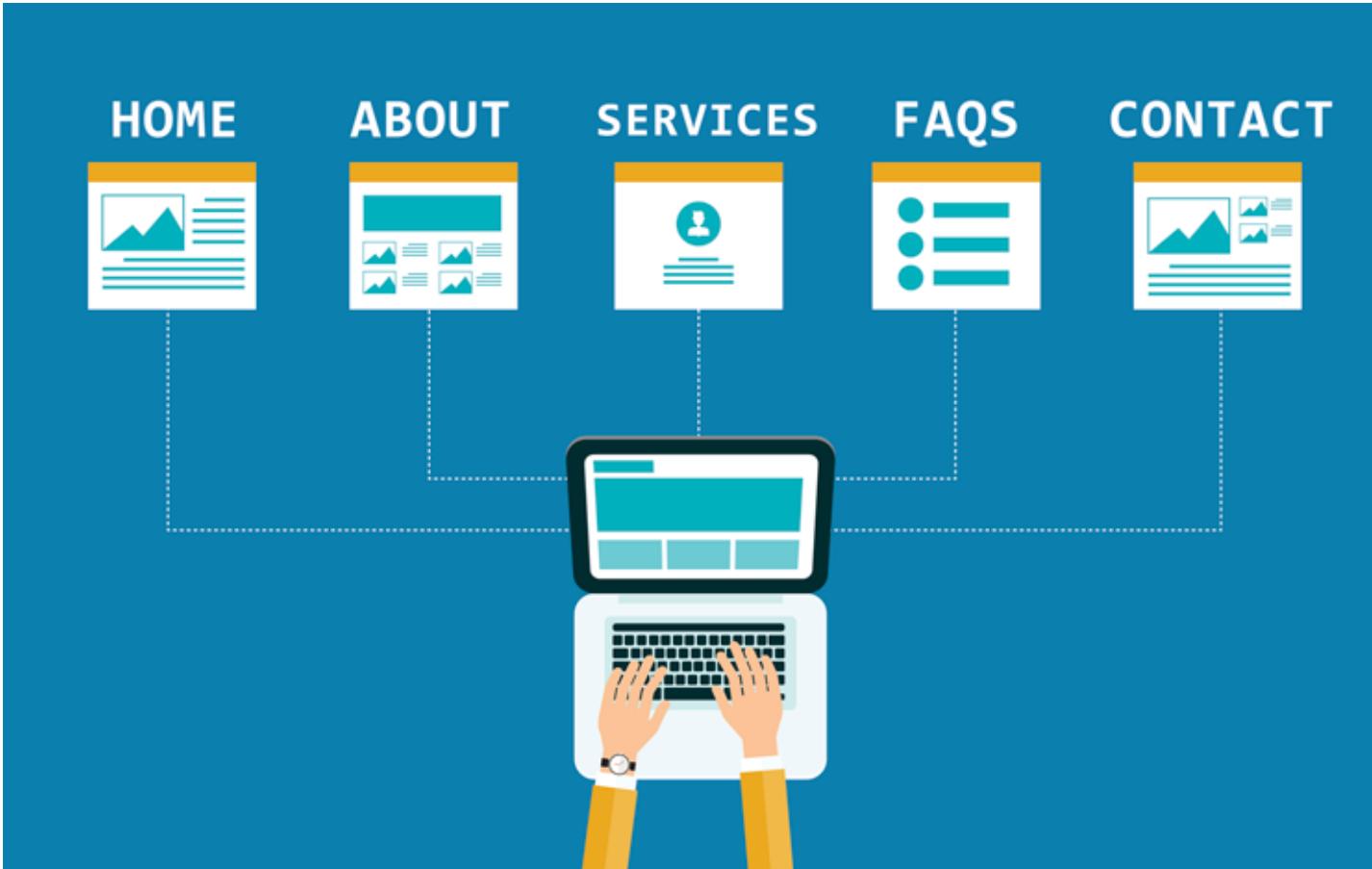
# MVC of front-end



- After split, each front-end is an individual system, it may have their own model.

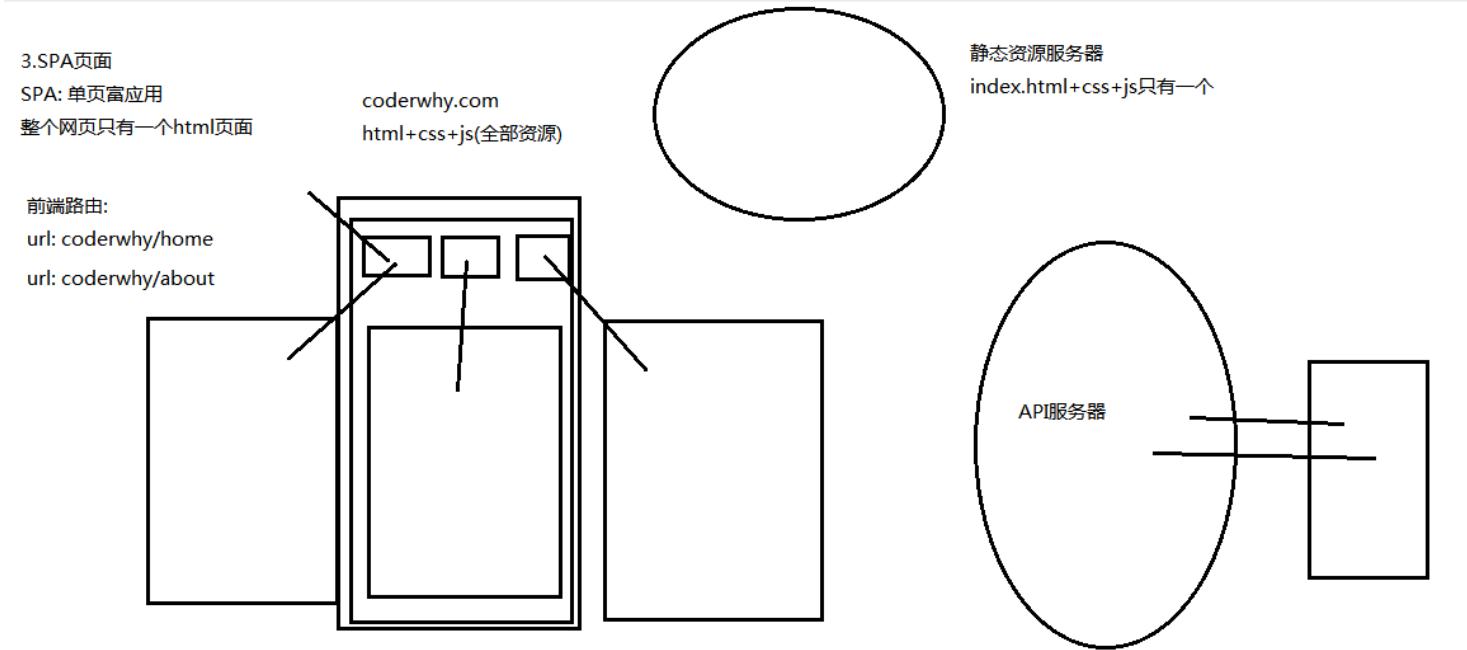
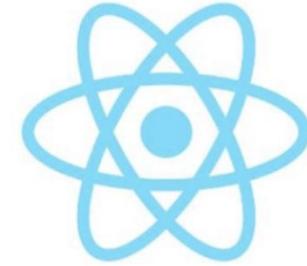
<https://www.jianshu.com/p/309f0477aac1>

# How about many web pages?



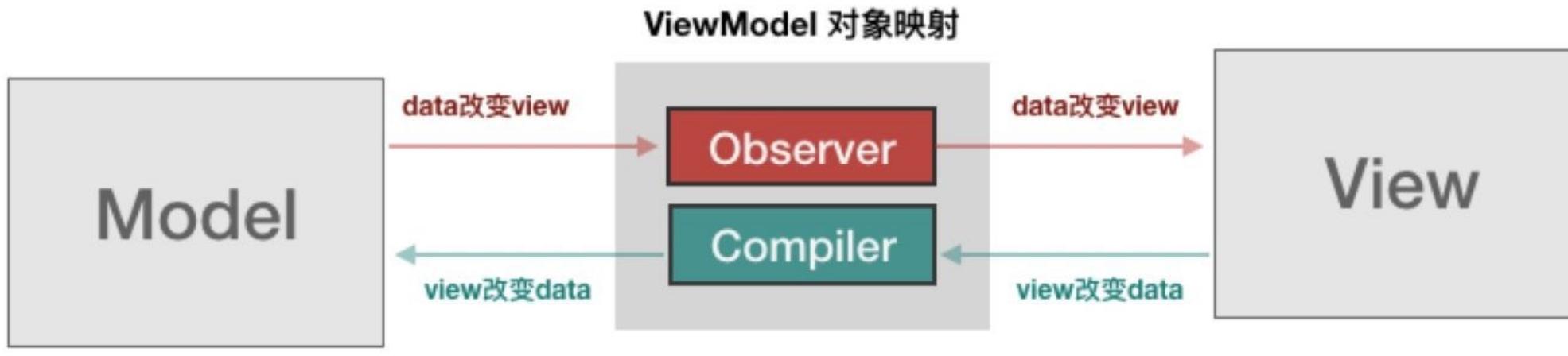
- Each page can be dynamic one with jQuery.
- It still need many web pages in a project.

# Front end framework



- SPA means single page application.
- The whole project has only one index.html file.
- The URI change is caught by **front-end router**.

# Vue – Model-View-ViewModel (MVVM)



不管你数据变化是从外部来的  
还是内部的，一视同仁

知乎 @LIN.JY666

<https://blog.csdn.net/youlingxx/article/details/79361979>  
<https://zhuanlan.zhihu.com/p/143876237>

# Vue – two-way binding



Hello Wor

Hello Wor

Hello World!

Hello World!

A screenshot of a browser's developer tools console. The 'Elements' tab is active, showing the DOM structure with the 'body' element selected. A tooltip is displayed over the 'body' element, listing its properties: \_\_defineGetter\_\_, \_\_defineSetter\_\_, \_\_lookupGetter\_\_, \_\_lookupSetter\_\_, \_\_ob\_\_, constructor, hasOwnProperty, isPrototypeOf, message, propertyIsEnumerable, toLocaleString, and valueOf. The 'Properties' section of the console shows the '\$scope' property. The bottom right corner of the screenshot contains the text 'Vue.js development experience: vue.js:10060'.

# Vue – components library



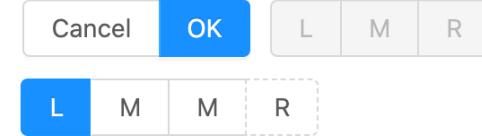
## Component Registration

```
import { Button } from 'ant-design-vue';
Vue.use(Button);
```

## Examples

This screenshot shows a section of the Ant Design Vue documentation for the Button component. It displays several examples of different button types: Primary (blue), Default (light gray), Dashed (dashed border), Danger (red), and Link (light blue). Below these examples, there is a section titled "Type" with a note stating: "There are primary button, default button, dashed button, danger button and link button in antd."

## Basic



## With Icon



- UI libraries, like Bootstrap for traditional CSS/JS

# Vue: sample code

```
1  <!DOCTYPE html>
2  <html lang="">
3    <head>
4      <meta charset="utf-8">
5      <meta http-equiv="X-UA-Compatible" content="IE=edge">
6      <meta name="viewport" content="width=device-width,initial-scale=1.0">
7      <link rel="icon" href="<%= BASE_URL %>favicon.ico">
8      <title><%= htmlWebpackPlugin.options.title %></title>
9    </head>
10   <body>
11     <noscript>
12       <strong>We're sorry but <%= htmlWebpackPlugin.options.title %> doesn't work properly without J
13     </noscript>
14     <div id="app"></div>
15     <!-- built files will be auto injected -->
16   </body>
17 </html>
18
```

- Only one “index.html”

# Vue: sample code

```
1  <template>
2    <div class="hello">
3      <h1>{{ msg }}</h1>
4      <h3>
5        | It's my Vue now!
6        |</h3>
7      </div>
8    </template>
9
10   <script>
11     export default {name: 'HelloWorld'...}
12   </script>
13
14   <!-- Add "scoped" attribute to limit CSS to this component only -->
15   <style scoped>
16     h3 {...}
17     ul {
18       list-style-type: none;
19       padding: 0;
20     }
21   </style>
```

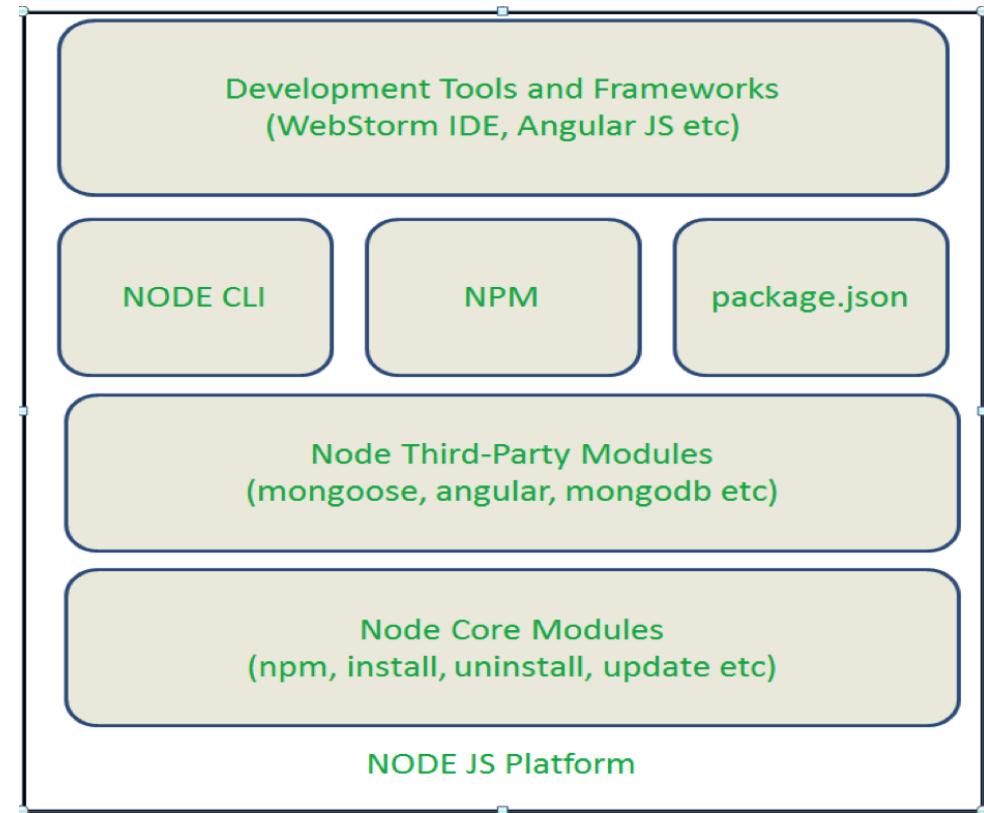
- Are you familiar with it?

# Vue: front-end router

```
51  ],
52  {
53   path: '/mine',
54   name: 'mine',           ←
55   component: () => import('../views/Mine.vue'),
56   children: [
57     {
58       path: '',
59       component: Login //默认登录界面
60     },
61     {
62       path: 'login',      ←
63       name: 'login',
64       component: Login
65     },
66     {
67       path: 'reg',        ←
68       name: 'reg',
69       component: Reg
70     }
71   ],
72 }
```

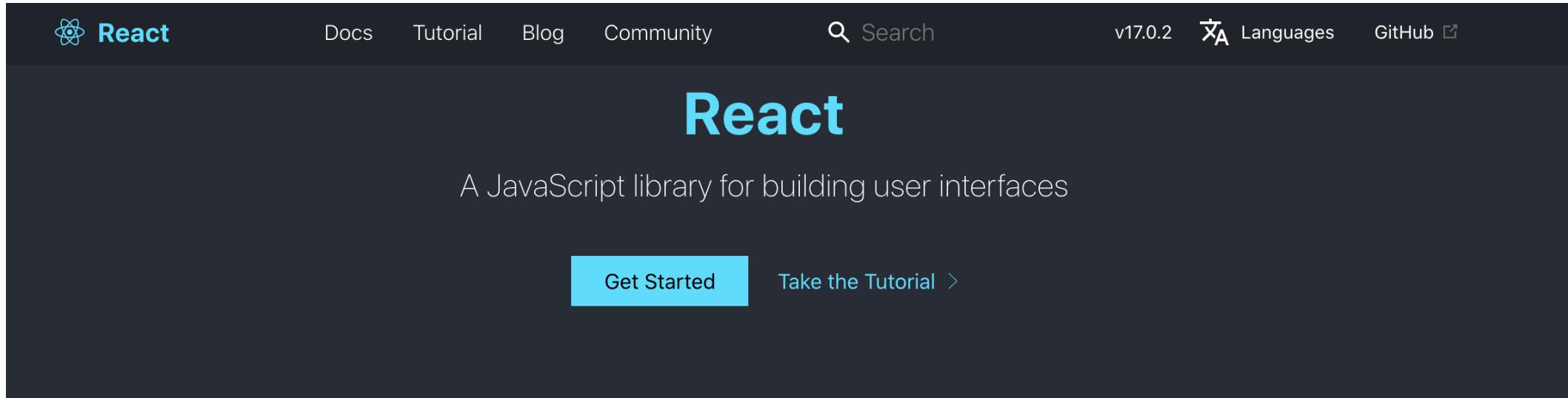
- Capture the path by js and change the corresponding component

# Vue and Node.Js



- NodeJs is a webserver powered by JavaScript
- Vue need NodeJs to do the develop
- Vue can be packaged to static files (mainly Js type) to run on any web server.

# React



The image shows the official React website homepage. At the top is a dark navigation bar with the React logo, "Docs", "Tutorial", "Blog", "Community", a search bar, "v17.0.2", "Languages", and "GitHub". Below the bar is a large white section with the word "React" in large blue letters, followed by "A JavaScript library for building user interfaces". There are two buttons: "Get Started" (in white) and "Take the Tutorial >".

# React

A JavaScript library for building user interfaces

[Get Started](#) [Take the Tutorial >](#)

## Declarative

React makes it painless to create interactive UIs. Design simple views for

## Component-Based

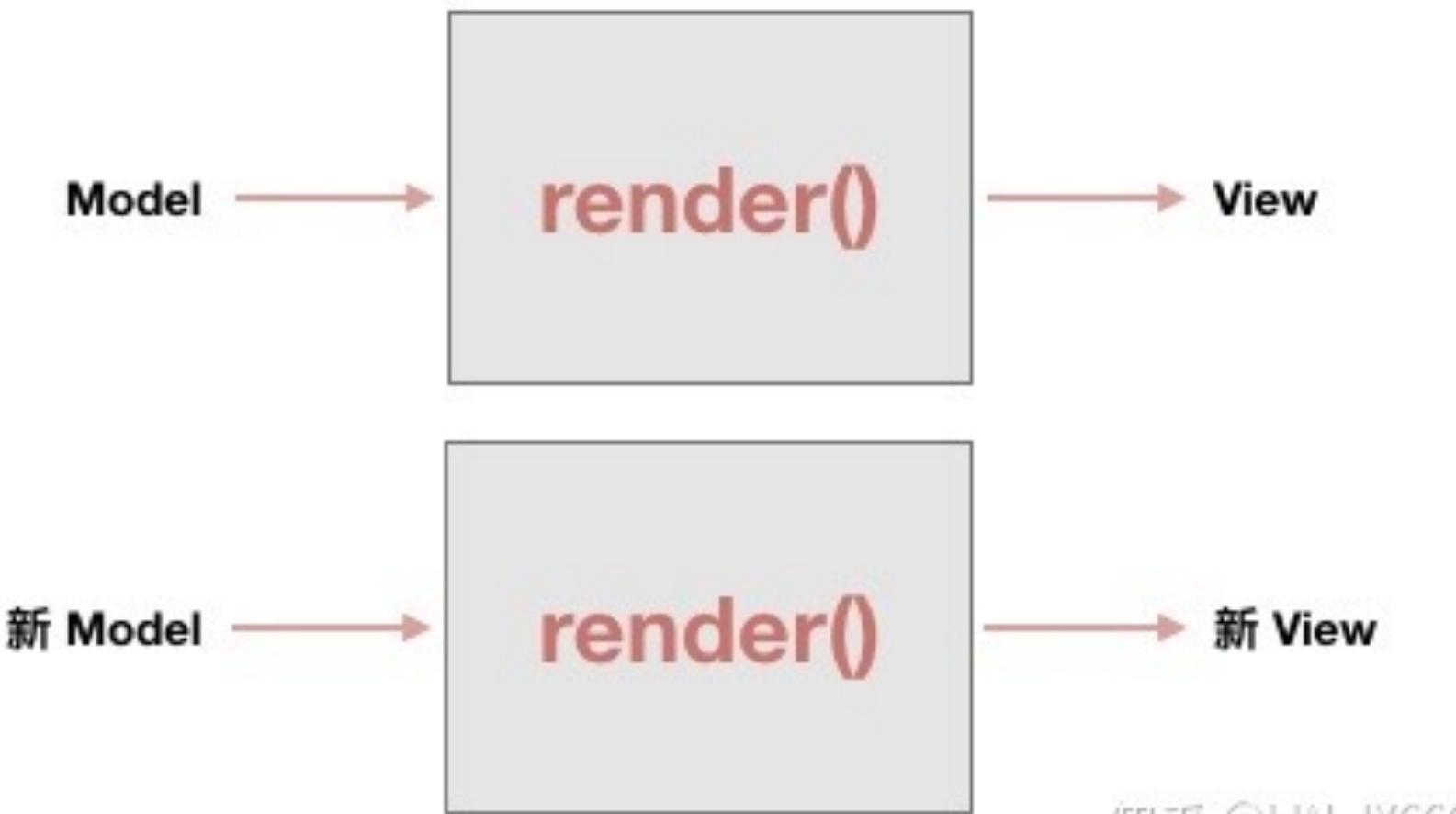
Build encapsulated components that manage their own state, then compose

## Learn Once, Write Anywhere

We don't make assumptions about the rest of your technology stack, so you can

- Similar with Vue
- Using JSX language (similar to JS)

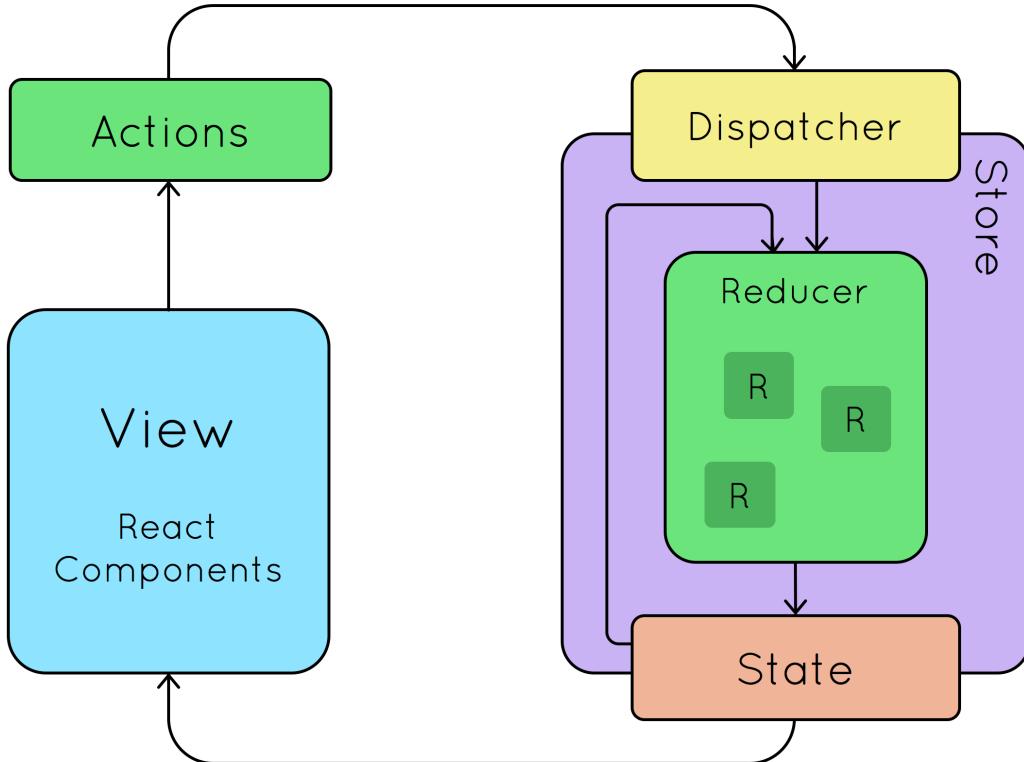
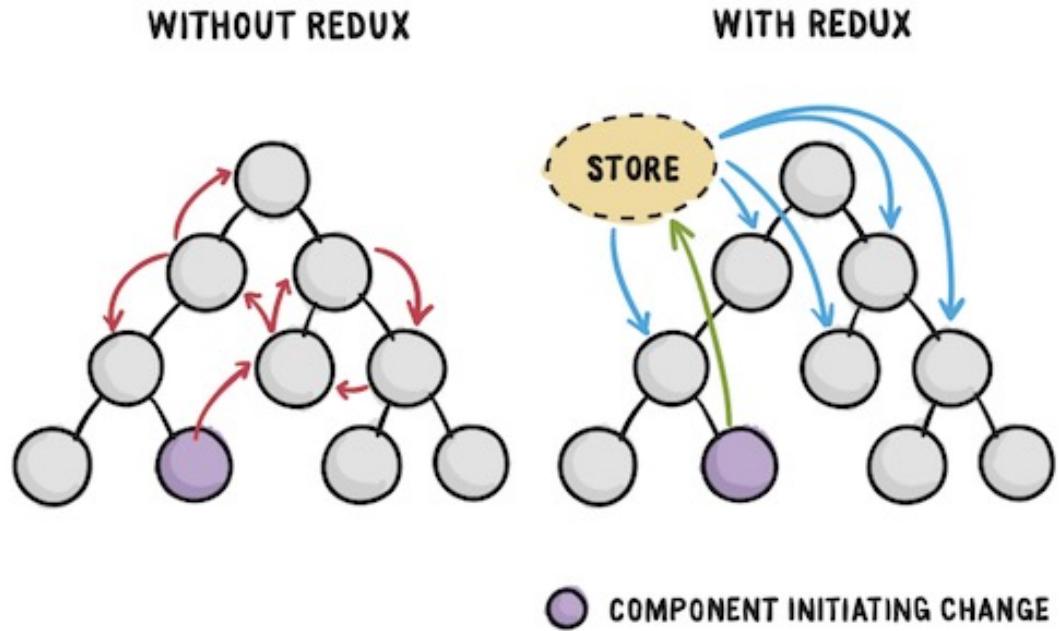
# React



知乎 @LIN.JY666

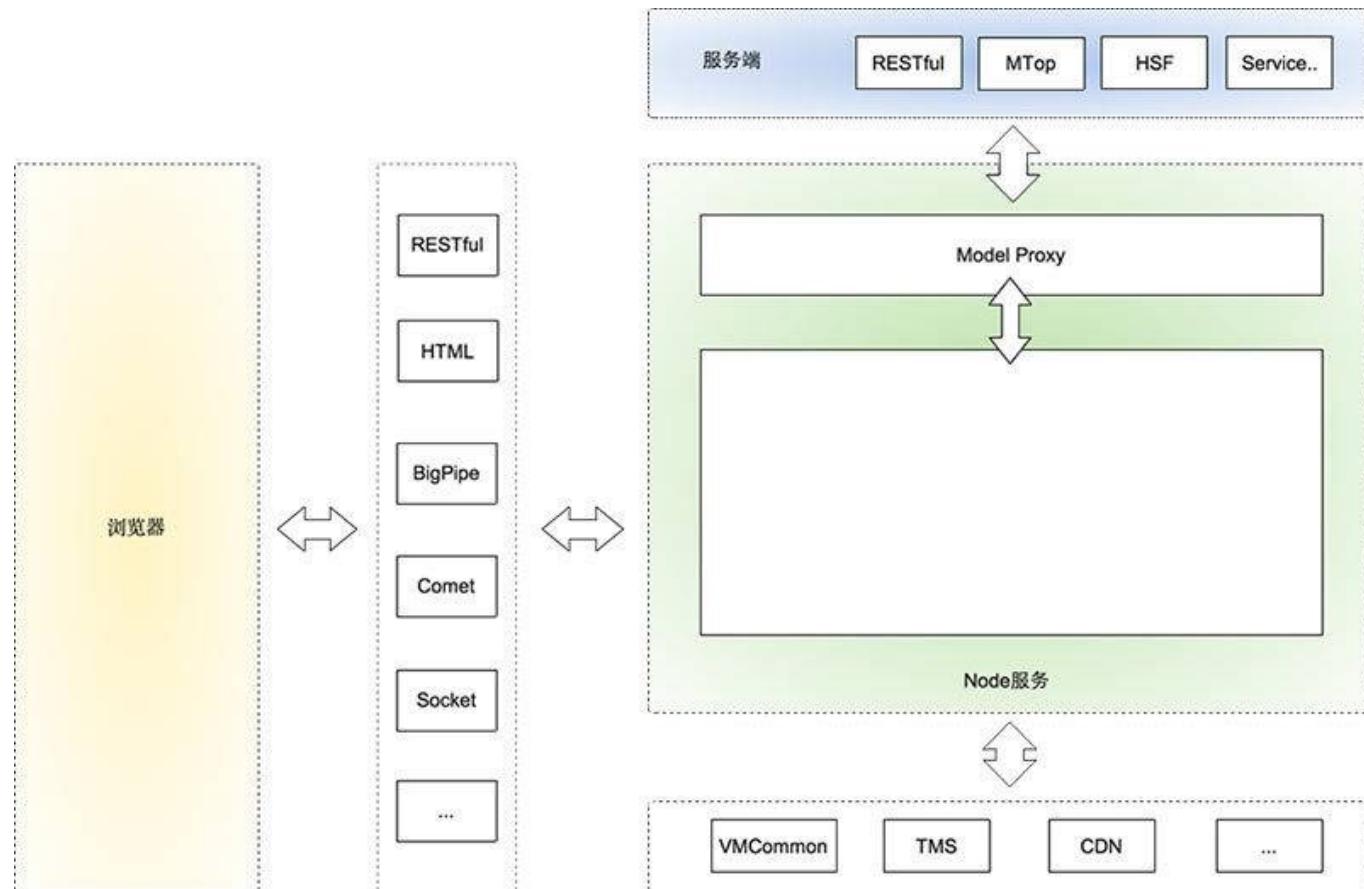
- React is just a render.

# React + redux



- **React + redux can work in MVC model.**

# Revolution has **NO** ends



- **NodeJS is a JS webserver.**
- **Insert NodeJS agent in the back-end**
- **Front-end also involve in some back-end things**

<https://www.kancloud.cn/kancloud/midway/48192>

<https://www.it610.com/article/1280970709337653248.htm>

# WeChat Mini Program

The screenshot shows the Weixin Docs website interface. At the top, there is a navigation bar with links for 'Weixin Docs', 'Mini Program', 'Development' (which is the active tab), 'Introduction', 'Design', 'Operation', 'Data', 'Community', and 'EN'. Below the navigation bar is a secondary menu with tabs for 'Guide', 'Framework' (highlighted in green), 'Components', 'API', 'Capabilities', 'Server Side', 'Weixin DevTools', and 'Cloud Base'. Underneath these tabs are two smaller buttons: 'WeChat CloudRun' and 'Update History'. The main content area on the left lists several sections: 'Mini Program configuration', 'Scene value', 'Frame interface', 'WXML Grammatical reference', and 'WXS Grammatical reference'. The right side features a large title 'Reference Documentation for Mini Program Frameworks' and a descriptive paragraph about the chapter's purpose. At the bottom right of the page is a circular button with a question mark and the Chinese characters '反馈'.

- › Mini Program configuration
- Scene value
- › Frame interface
- › WXML Grammatical reference
- › WXS Grammatical reference

## Reference Documentation for Mini Program Frameworks

This chapter provides reference documentation for Mini Program framework configurations, framework APIs, WXML, and WXS. For more information about components and APIs, refer to the following documents:

- Reference Documentation for Mini Program Components
- Reference Documentation for Mini Program APIs
- Reference Documentation for Mini Program Server APIs

The translations are provided by WeChat Translation and are for reference only. In case of any inconsistency and

<https://developers.weixin.qq.com/miniprogram/en/dev/reference/>

# WeChat Mini Program

Guide    **Framework**    Components    API    Capabilities    Server Side    Weixin DevTools    Cloud Base

WeChat CloudRun    Update History

› Mini Program configuration

Scene value

› Frame interface

› WXML Grammatical reference

Data binding

List rendering

Conditional Rendering

template

reference

› WXS Grammatical reference

## WXML

WXML (WeiXin Markup Language) is a markup language for framework design. It can be used to build the page structure when combined with [base components](#) and an [event system](#).

The following examples show what WXML can do:

### Data Binding

```
<!-- wxml -->
<view> {{message}} </view>
```

```
// page.js
Page({
  data: {
```



<https://developers.weixin.qq.com/miniprogram/en/dev/reference/>

# How to choose?



手机App

+



微信小程序

+



H5微网站

# Cannon vs. mosquito

