



Possibilités





AngularJS 1.5

Février 2016

Regular JS 😊



Angular 2.0

Septembre 2016

TypeScript

ECMAScript (ES6)

Babel.js

SystemJS



facebook / react

Watch 3,846 Star 54,455 Fork 9,702

Code Issues 501 Pull requests 106 Projects 0 Wiki Pulse Graphs

angular / angular.js

Watch 4,452 Star 53,479 Fork 26,466

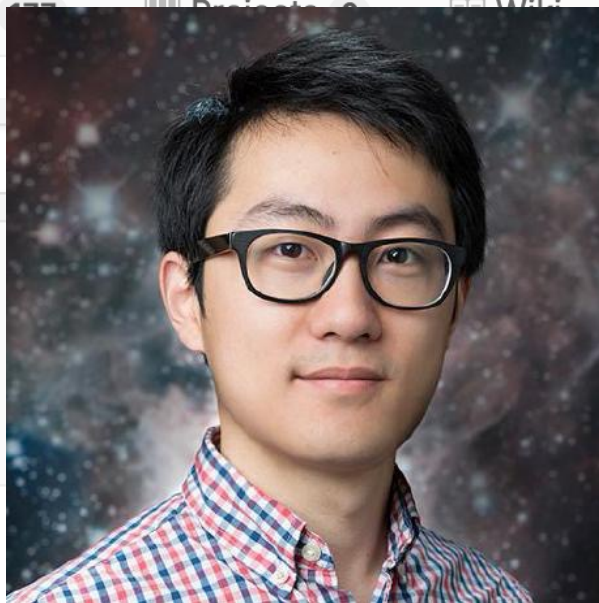
Code Issues 676 Pull requests Projects Wiki Pulse Graphs

vuejs / vue

Watch 1,965 Unstar 34,327 Fork 4,021

Code Issues 41 Pull requests 2

Simple yet powerful library for building modern



1,344 commits 12 branches 178 releases 50 contributors MIT

Branch: dev New pull request

Create new file Upload files Find file Clone or download

yyx990803 support destructuring in v-for alias Latest commit a0d8603 2 hours ago

.github	Add guide for PR title (#3936)	a month ago
benchmarks	adjust build	a day ago
build	adjust build	a day ago
dist	improve build file readme	a day ago

FORCES



- Meilleure organisation du code JavaScript



- Sites dynamiques et rapides



- Peu de code à écrire



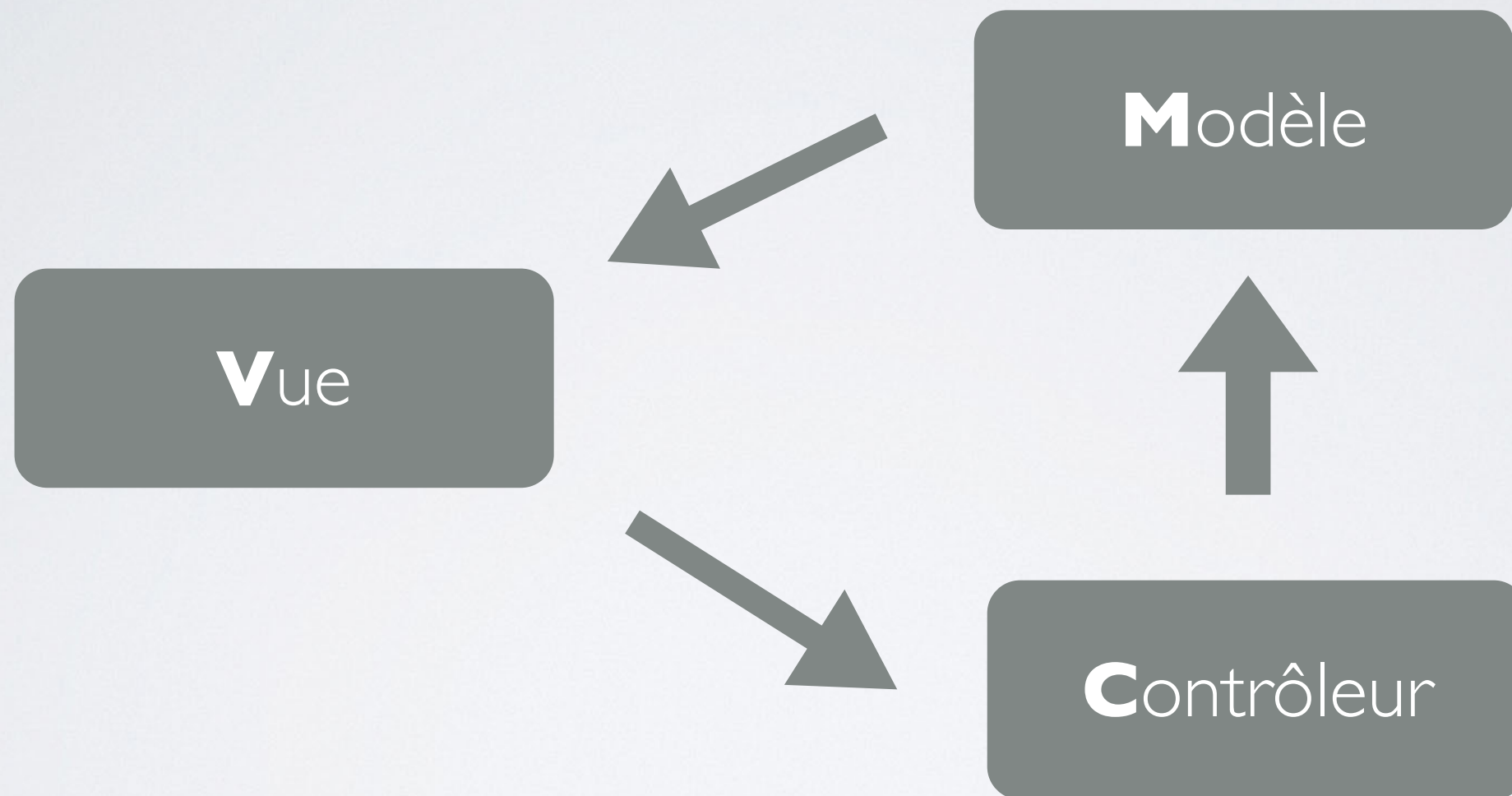
- Facile à tester



- 2 way data-binding

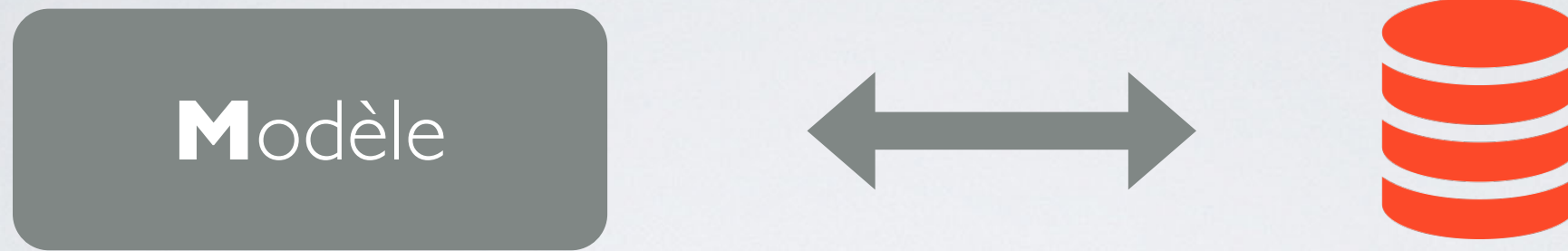
LE MODÈLE MVC

ARCHITECTURE MVC



ARCHITECTURE MVC

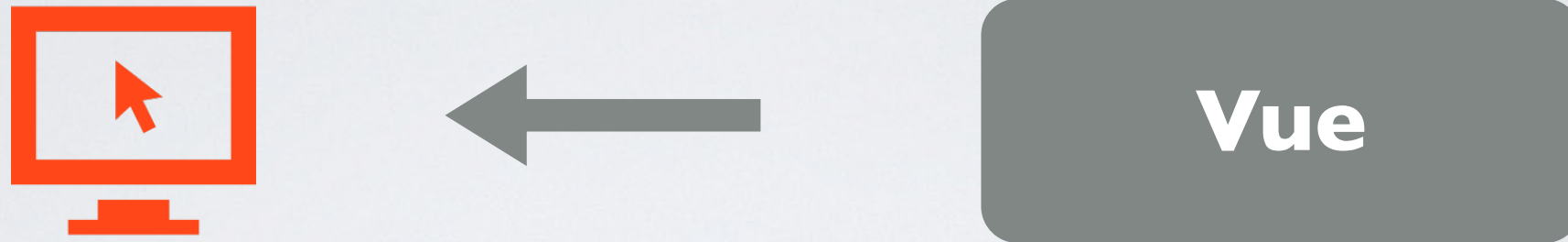
MODÈLE



- Il représente les données de l'application
- Interaction avec la base de données
- Traitement des données

ARCHITECTURE MVC

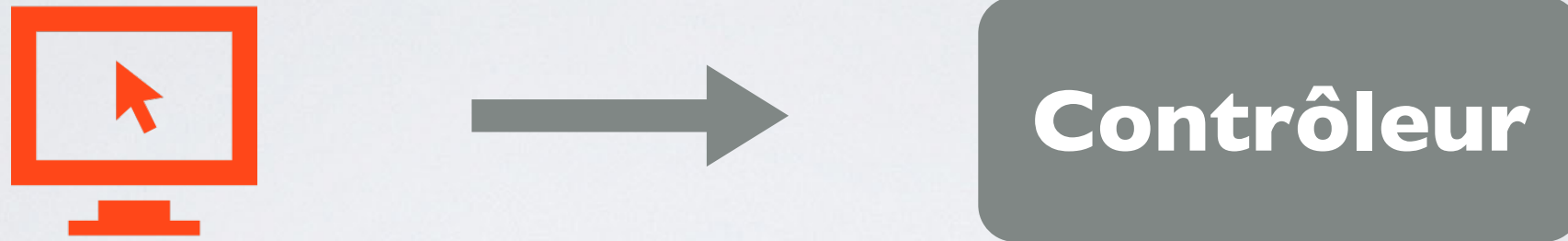
VUE



- Interface utilisateur
- Aucun traitement
- Affiche les données que fournit le modèle

ARCHITECTURE MVC

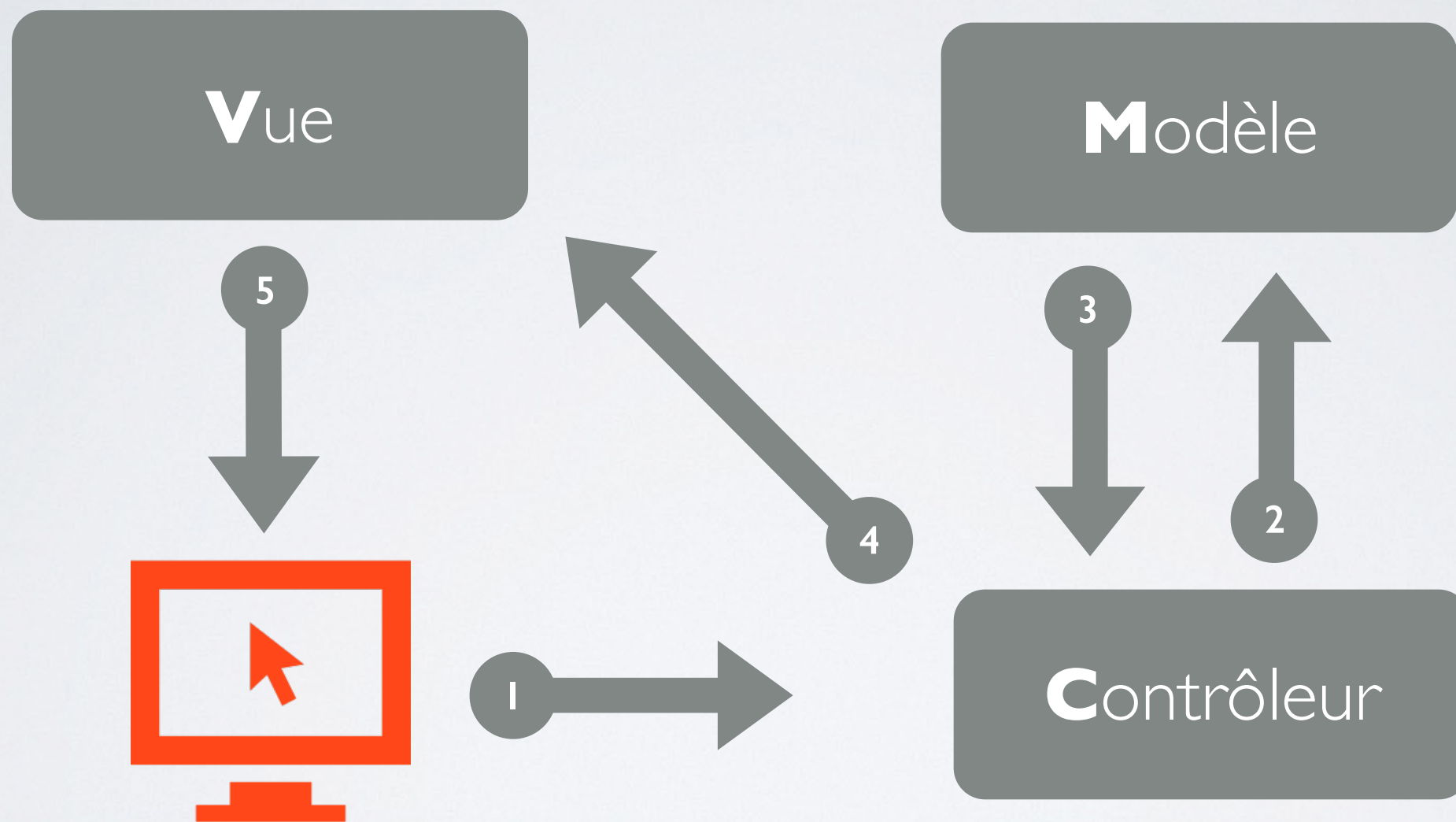
CONTRÔLEUR



- Interface entre le modèle et le client
- Interprète la requête client et met à jour le modèle

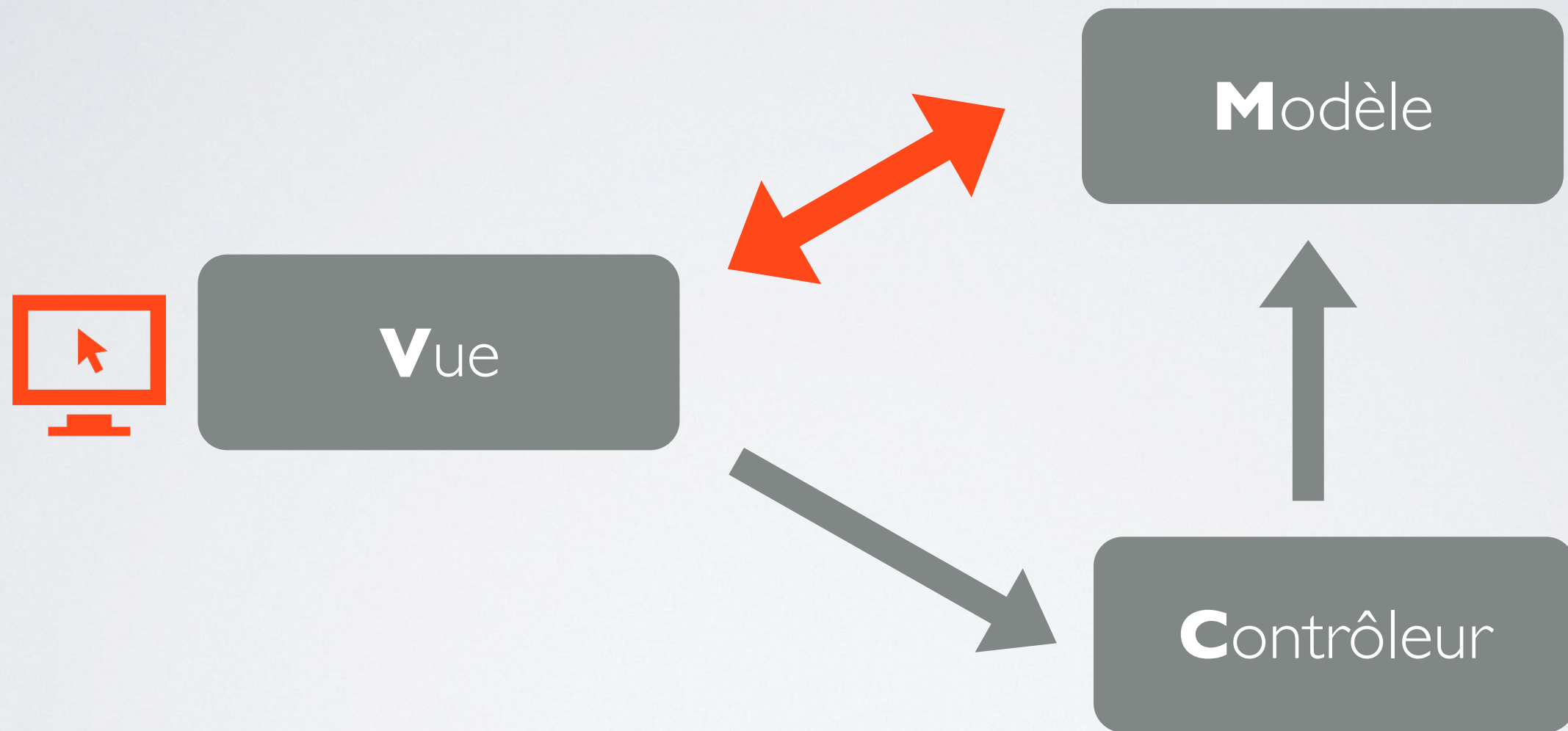
MVC SYMFONY2

DÉTAILS



MVC ANGULAR

DÉTAILS

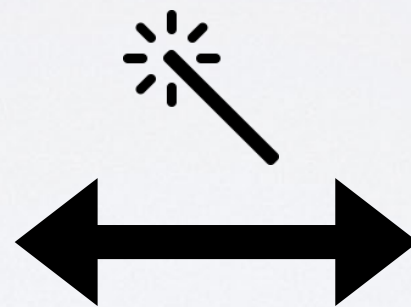


TWO-WAY DATA BINDING

- La Vue et le Modèle sont liés



View



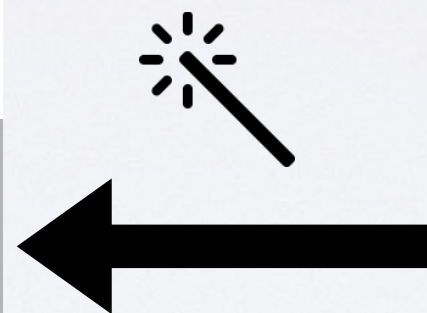
Model

TWO-WAY DATA BINDING

- Lorsque le modèle change, la vue change automatiquement



View



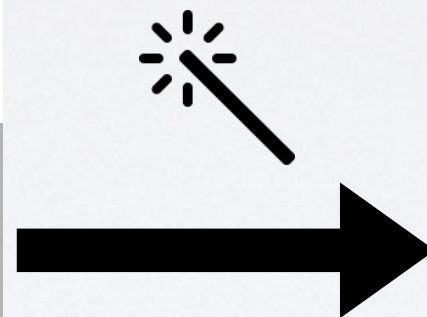
Model

TWO-WAY DATA BINDING

- Lorsque la vue change, le modèle change automatiquement



View



Model

⚡ PERFORMANCES

Site web traditionnel



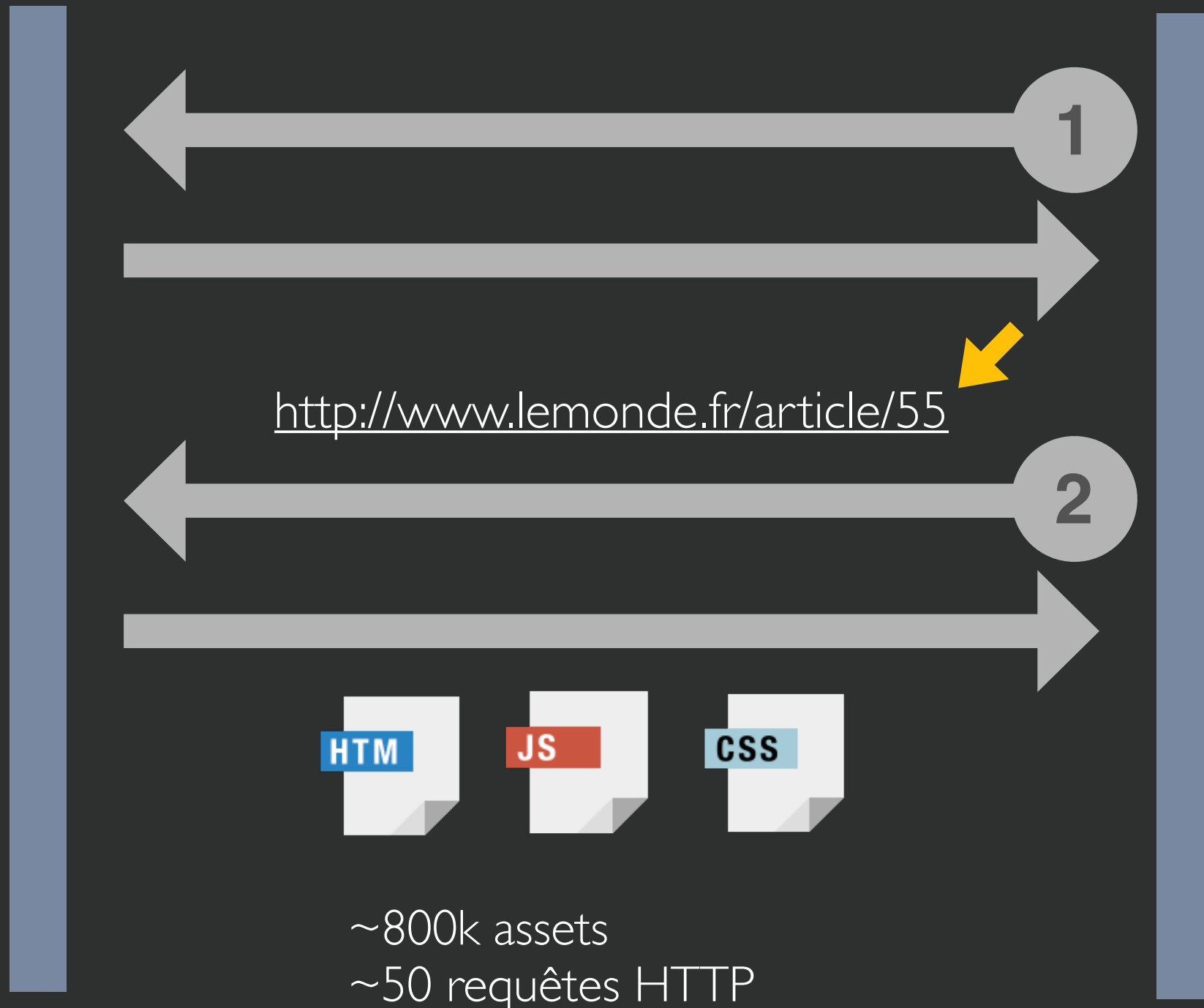
<http://www.lemonde.fr/article/10>



~800k assets
~50 requêtes HTTP

⚡ PERFORMANCES

Site web traditionnel



⚡ PERFORMANCES

Site web avec AngularJS



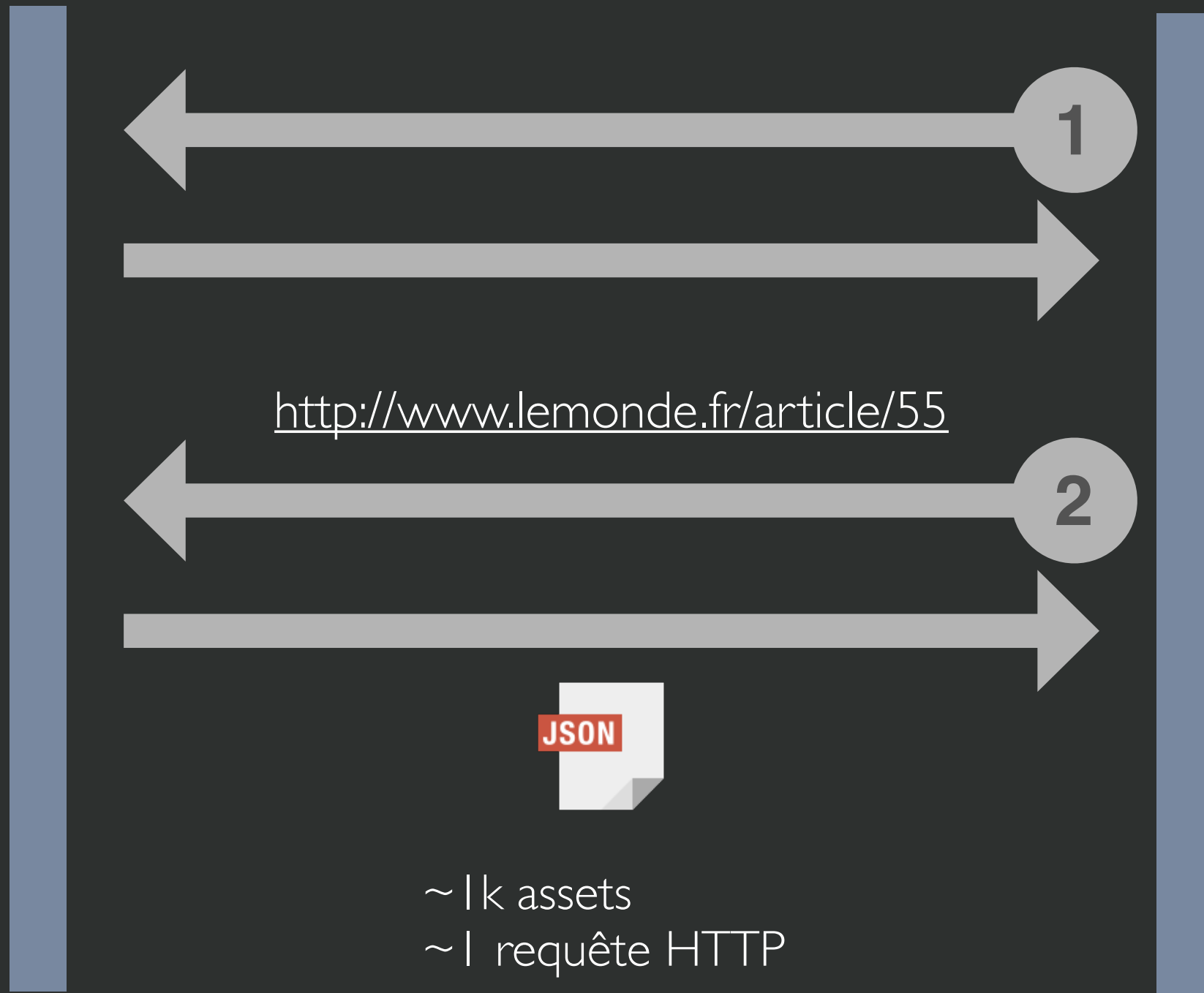
<http://www.lemonde.fr/article/10>



~800k assets
~50 requêtes HTTP

⚡ PERFORMANCES

Site web avec AngularJS



GENERATION DE LA VUE

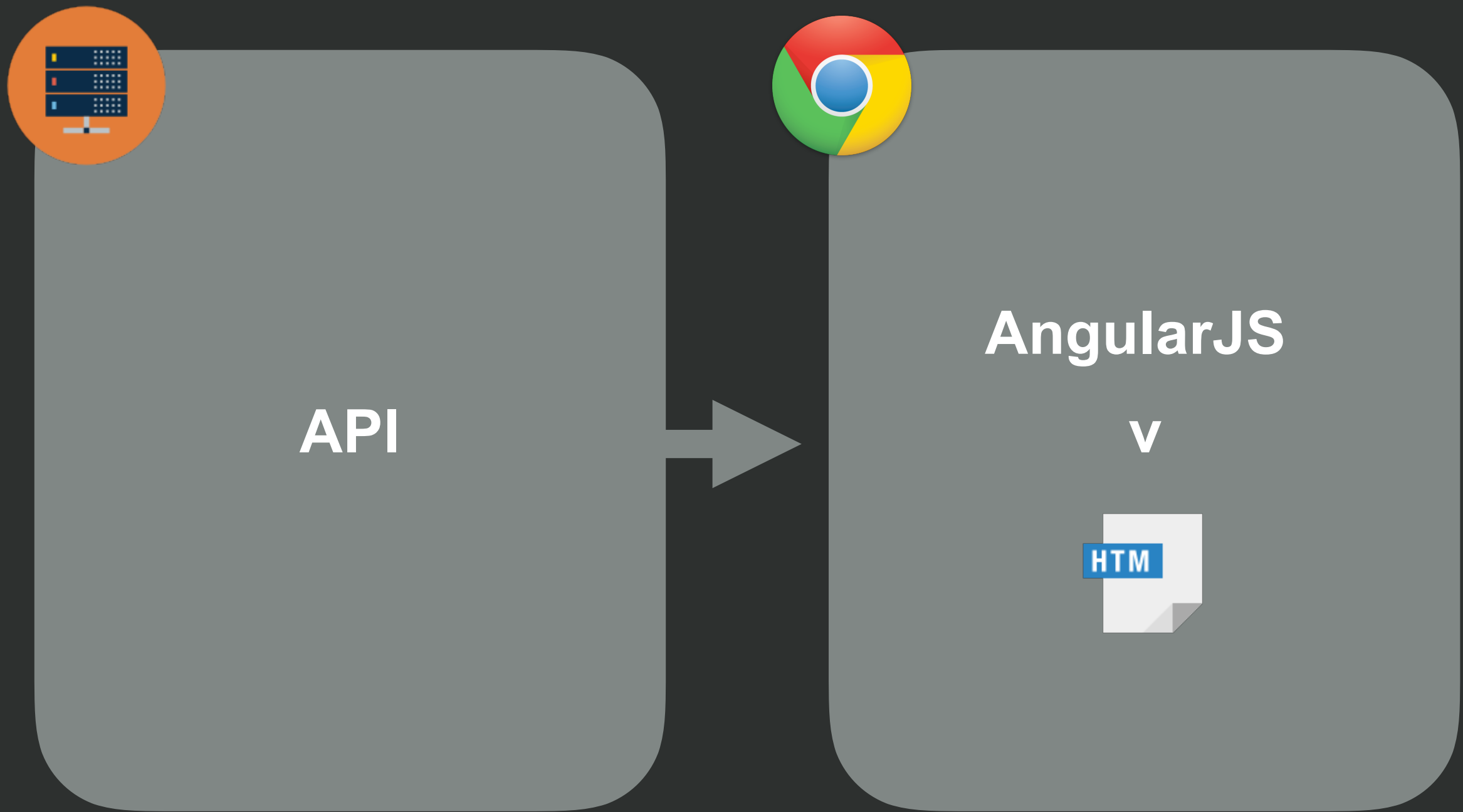


TWIG

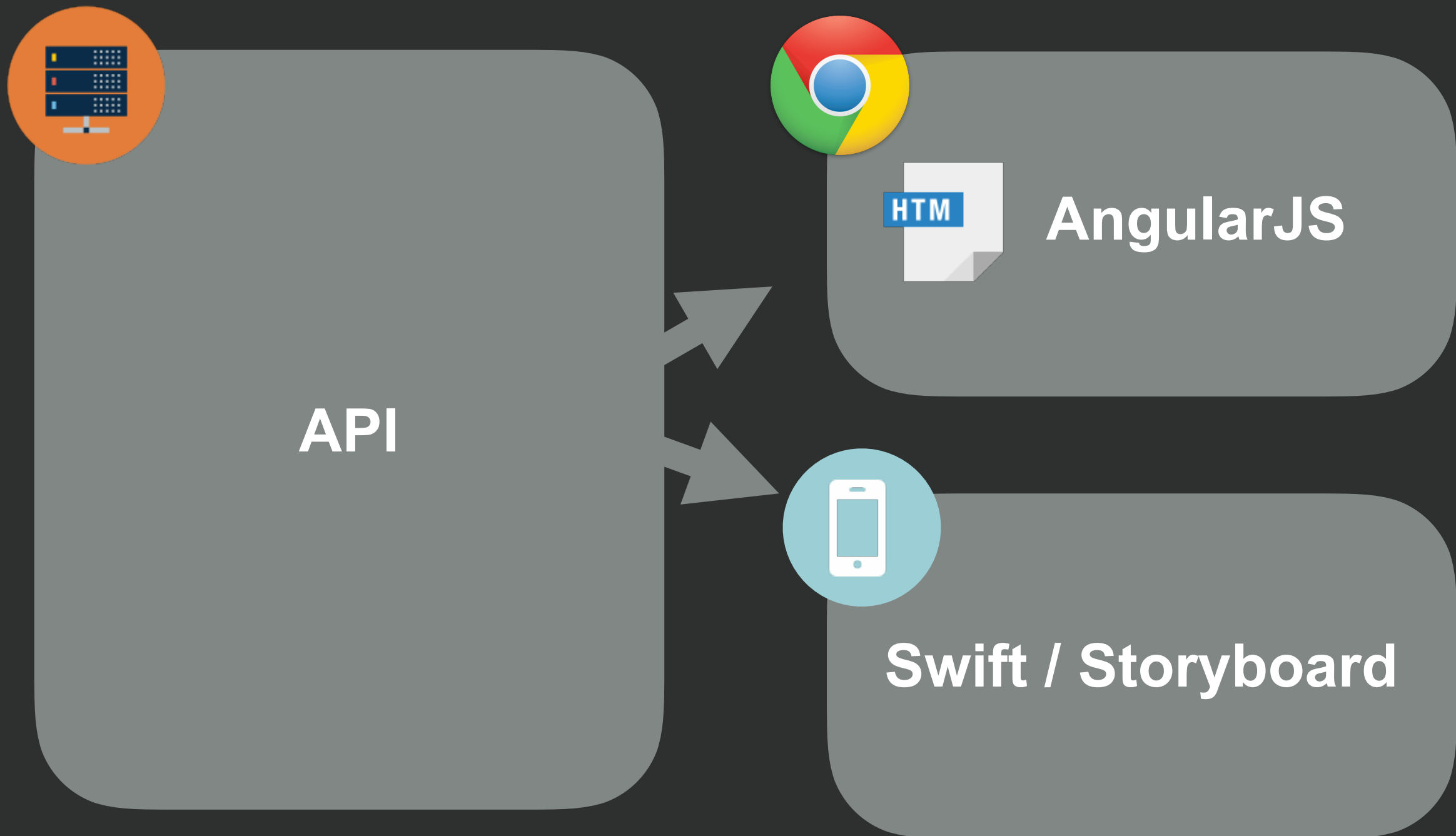
+



GENERATION DE LA VUE

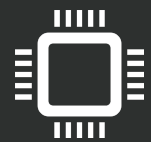


GENERATION DE LA VUE



⚡ PERFORMANCES

Site web traditionnel



~500ms backend



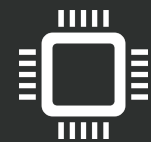
~500ms network



~500ms render

= ~1.5 seconde

Site web avec AngularJS



~250ms backend



~50ms network



~100ms render

= ~400ms

<CODE>

INSTALLATION

index.html

```
<!DOCTYPE html>
<html>
<head>
  <title>Install</title>
</head>
<body>
  <script src="js/angular.min.js"></script> ←
  <script src="js/app.js"></script> ←
</body>
</html>
```

VOCABULAIRE ANGULAR

Modules

Controllers

Directives

Expressions

Filtres

Templates

Services

Component

MODULES

- Les modules sont l'endroit où l'on écrit notre code Angular
- Permet d'organiser notre code en briques fonctionnelles
- Permet de gérer les dépendances

app.js

```
// Crée un module  
var demo = angular.module('demo', []);
```



```
// Récupère un module par son nom  
angular.module('demo');
```


CONTROLLERS

- Les controllers permettent de définir le comportement de l'application
- Contiennent des fonctions et des valeurs

app.js

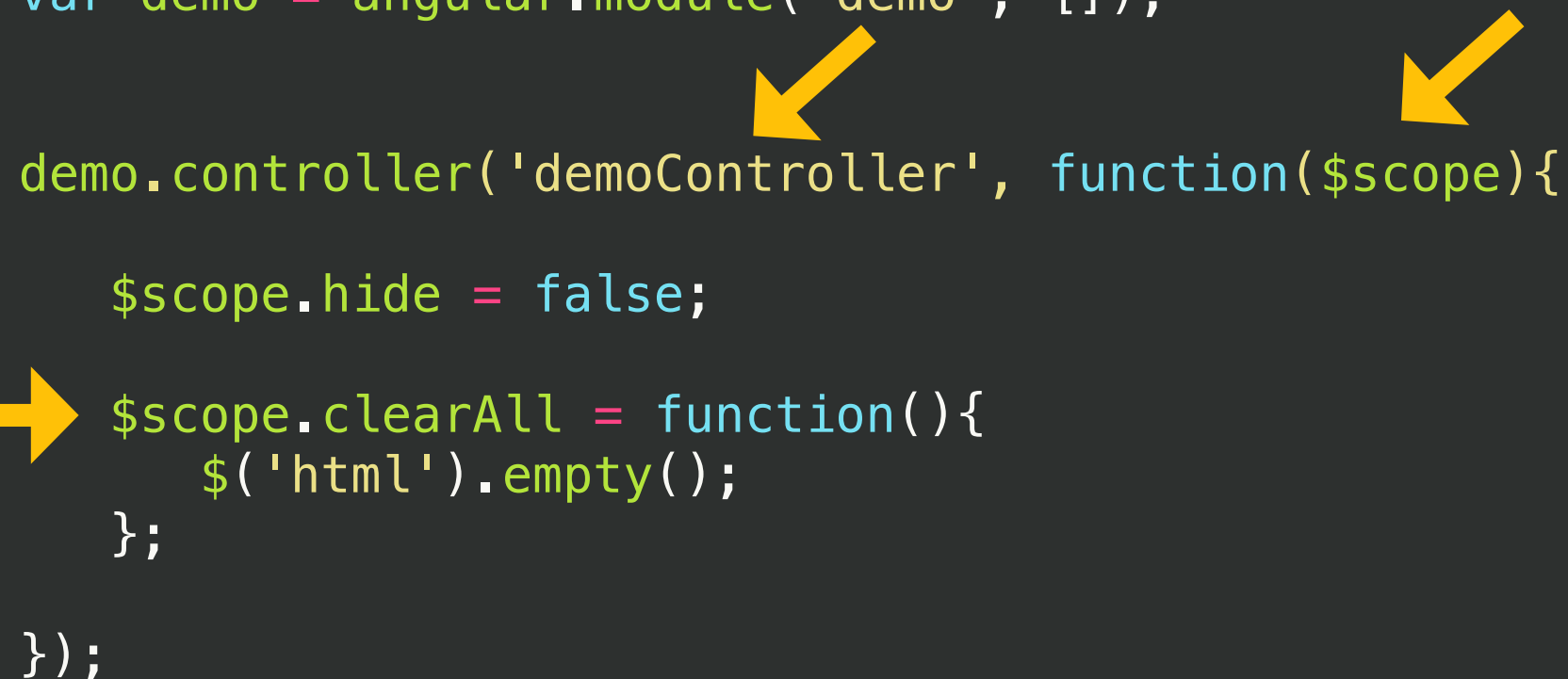
```
// Crée un module
var demo = angular.module('demo', []);

demo.controller('demoController', function($scope){

    $scope.hide = false;

    → $scope.clearAll = function(){
        $('html').empty();
    };

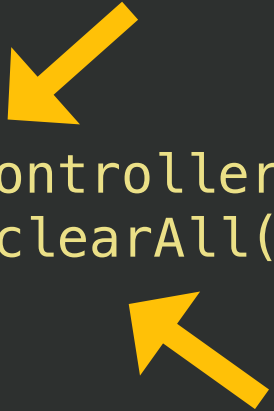
});
```



CONTROLLERS

- Les controllers permettent de définir le comportement de l'application
- Contiennent des fonctions et des valeurs

index.html




```
<div ng-controller="demoController">  
  <a href="" ng-click="clearAll()">Clear All</a>  
</div>
```

CONTROLLERS

- Les controllers permettent de définir le comportement de l'application
- Contiennent des fonctions et des valeurs

index.html



```
<div ng-controller="demoController">  
  <a href="" ng-click="clearAll()">Clear All</a>  
</div>
```

DIRECTIVES

- Une directive est un marqueur HTML qui permet à Angular de faire le liens avec du code Javascript

index.html

```
<!DOCTYPE html>
<html>
<head>
  <title>Demo 1</title>
</head>
<body ng-app="demo">
  <div ng-controller="demoController">
    Hello world
  </div>
  <script src="js/angular.min.js"></script>
  <script src="js/app.js"></script>
</body>
</html>
```



DIRECTIVES

- Une directive est un marqueur HTML qui permet à Angular de faire le liens avec du code Javascript

app.js

```
var demo = angular.module('demo', []);
```



```
demo.controller('demoController', function(){  
    alert('hello');  
});
```

DIRECTIVES

ng-app	Lie un module Angular à un élément du DOM
ng-controller	Lie un controller Angular à un élément du DOM
ng-show / ng-hide	Affiche / Masque un élément
ng-repeat	Permet de faire des boucles « foreach »
ng-src	Charge une image
ng-click	Evalue une expression au click
ng-mouseenter	Evalue une expression quand la souris entre dans la zone
ng-mouseleave	Evalue une expression quand la souris quitte dans la zone
ng-init	Permet d'initialiser des variables en HTML plutôt qu'en JS
ng-class	Permet d'assigner une classe selon une condition
ng-model	Assigne la valeur d'un champ à une variable

NG-CLICK, NG-SHOW, NG-HIDE

```
<div ng-app="demoApp">
  <div ng-controller="demoController">
    <a href="" ng-click="looged=true">Log in</a>
    <a href="" ng-click="looged=false">Log out</a>
    <div class="wrapper">
      <p ng-hide="looged">The is logged in !</p>
      <p ng-show="looged">Bye Bye !</p>
    </div>
  </div>
</div>
```

NG-MODEL & NG-CLASS

```
<h1>Choisissez un style</h1>
<input type="checkbox" ng-model="italic"> Italique
<input type="checkbox" ng-model="bold"> Gras

<!-- add the class 'text-bold' if the variable 'bold' is true -->
<p ng-class="{ 'text-italic': italic, 'text-bold': bold }">
    This is a demo text to see the result
</p>
```


EXPRESSIONS

- Permet d'insérer des valeurs dynamiques dans le HTML

index.html

```
<p>  
    Calcul : {{ 15 + 20 }}  
</p>
```

EXPRESSIONS

- Permet d'insérer des valeurs dynamiques dans le HTML

index.html

```
<p>  
  Calcul : {{ result }}  
</p>
```

app.js

```
var demo = angular.module('demo', []);  
  
demo.controller('demoController', function(){  
  $scope.result = 10 + 15;  
});
```

NG-REPEAT

```
<div ng-app="myApp" ng-controller="tableController">
  <table>
    <thead>
      <tr>
        <th>Name</th>
        <th>Capital</th>
      </tr>
    </thead>
    <tr ng-repeat="country in countries">
      <td>{{ country.name }}</td>
      <td>{{ country.capital }}</td>
    </tr>
  </table>
</div>

<script>
  var app = angular.module('myApp', []);
  app.controller('tableController', function($scope) {
    $scope.countries=
      [{"name": "India" , "capital" : "Delhi"},
      {"name": "England" , "capital" : "London"},
      {"name": "France" , "capital" : "Paris"},
      {"name": "Japan" , "capital" : "Tokyo"}];
  });
</script>
```

FILTRES

- Permettent de formater une valeur

index.html

```
<p>  
  Nom {{ name | uppercase }}  
  Prix : {{ result | currency }}  
  Publié le {{ published | date='MM/dd/yyyy' }}  
</p>
```

TP

- Construire un formulaire de commande pour une chaine de pizza avec AngularJS à la place de jQuery



Pimp my Pizza

Cooked with love

Recette

- ☐ Margherita (8€)
- ☒ Reine (10€)
- ☐ Quattro stagioni (12€)
- ☐ Sicilienne (11€)

Nombre de parts

10



Pâte

- ☒ Classique
- ☐ Extra fine (+2€)



Total
XX €

Payer