



ANGULAR... !

LA CUVÉE DE L'ANNÉE ~~2016~~ 2017

MOI ?

KILLIAN CHARPENTIER

- Front-End / mobile developer
- OPEN, Lannion

@KillianCharpent

Github: ShamanisTe

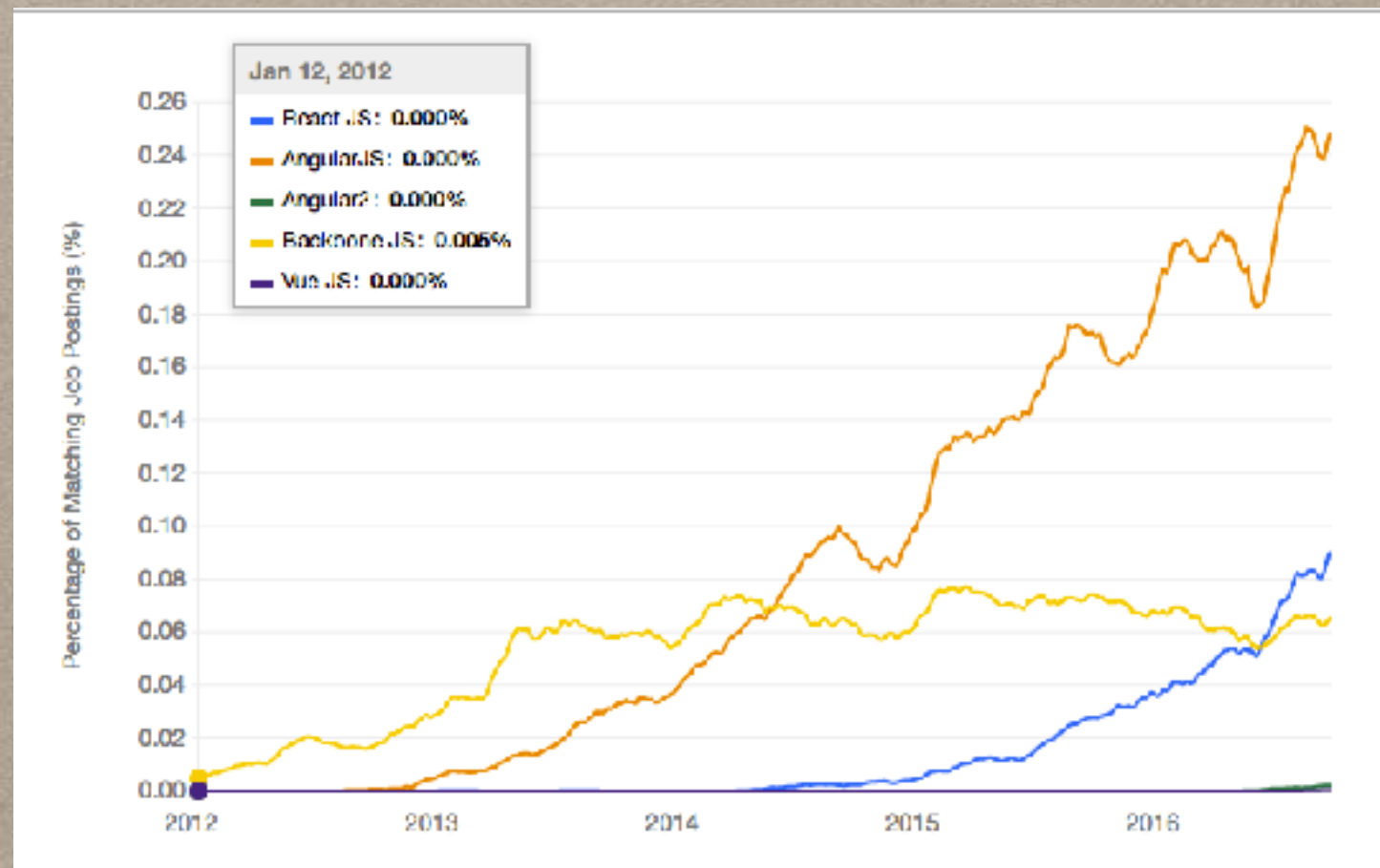


SOMMAIRE

1. Rappel AngularJS
2. Angular, c'est quoi ?
3. AngularJS versus Angular
4. Oui... mais ?
5. Conclusion

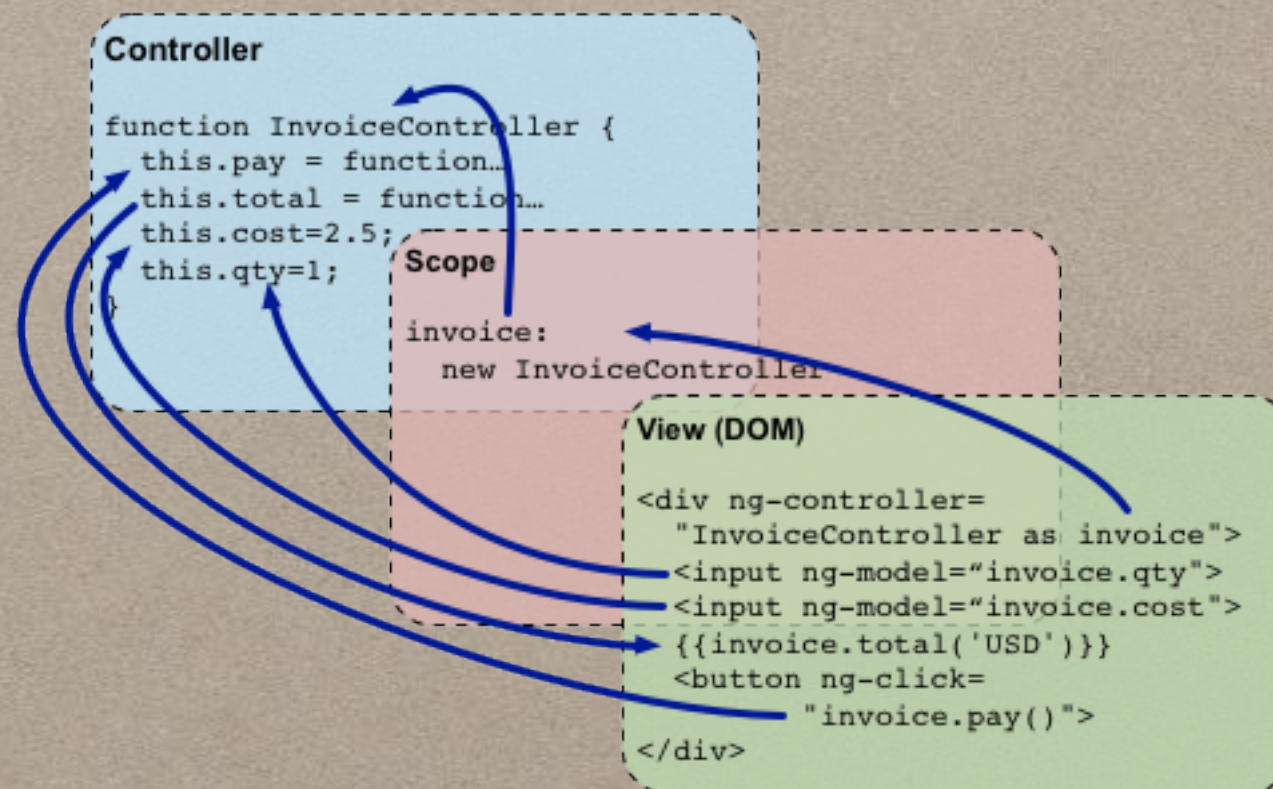
RAPPEL ANGULARJS

- 2009, propulsé par Google
- Très populaire



RAPPEL ANGULARJS

- 2009, propulsé par Google
- Très populaire
- MV*
- Un framework client complet





RAPPEL ANGULARJS

- Dirty checking
- Third library
- Scope
- Performance render

SOMMAIRE

1. ~~Rappel AngularJS~~
2. Angular, c'est quoi ?
3. AngularJS versus Angular
4. Oui... mais ?
5. Conclusion

ANGULAR, C'EST QUOI ?

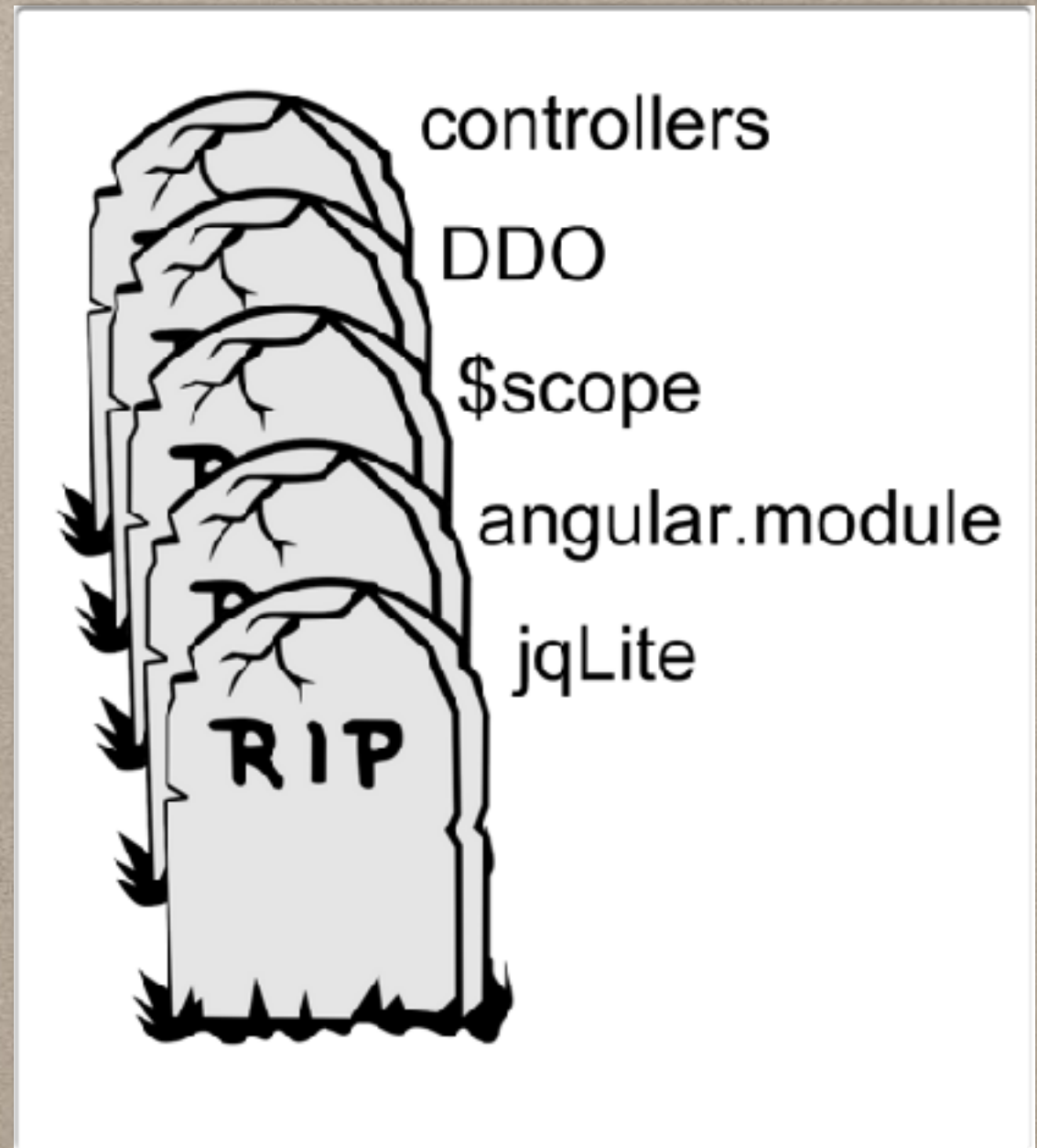
SOMMAIRE

- Présentation
- Nouveaux concepts
- Environnements

ANGULAR, C'EST QUOI ?

PRÉSENTATION

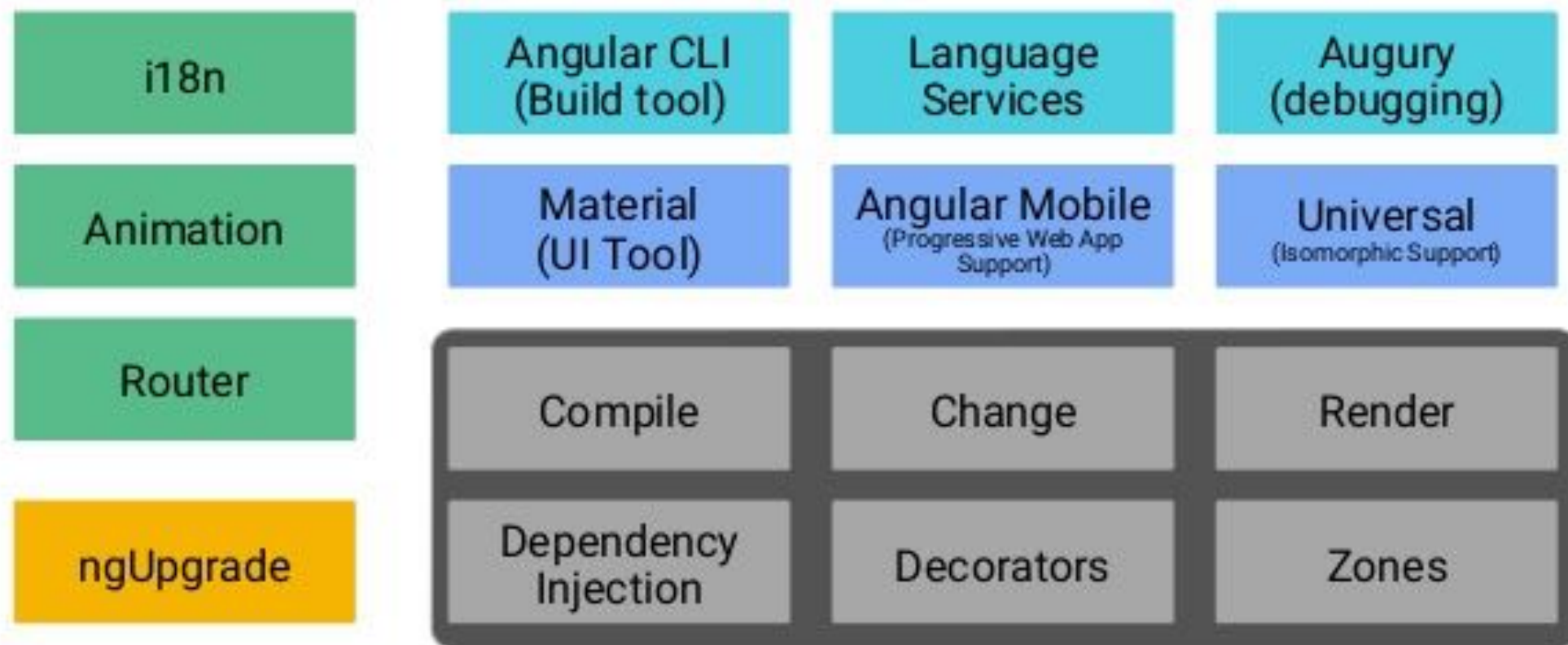
- Version 2.4.8
- Angular !== AngularJS
- YAF
- Basé sur les web standards



ANGULAR, C'EST QUOI ?

PRÉSENTATION

Framework to Platform



ANGULAR, C'EST QUOI ?

NOUVEAUX CONCEPTS

- Component Architecture (pattern)

ANGULAR, C'EST QUOI ?

NOUVEAUX CONCEPTS

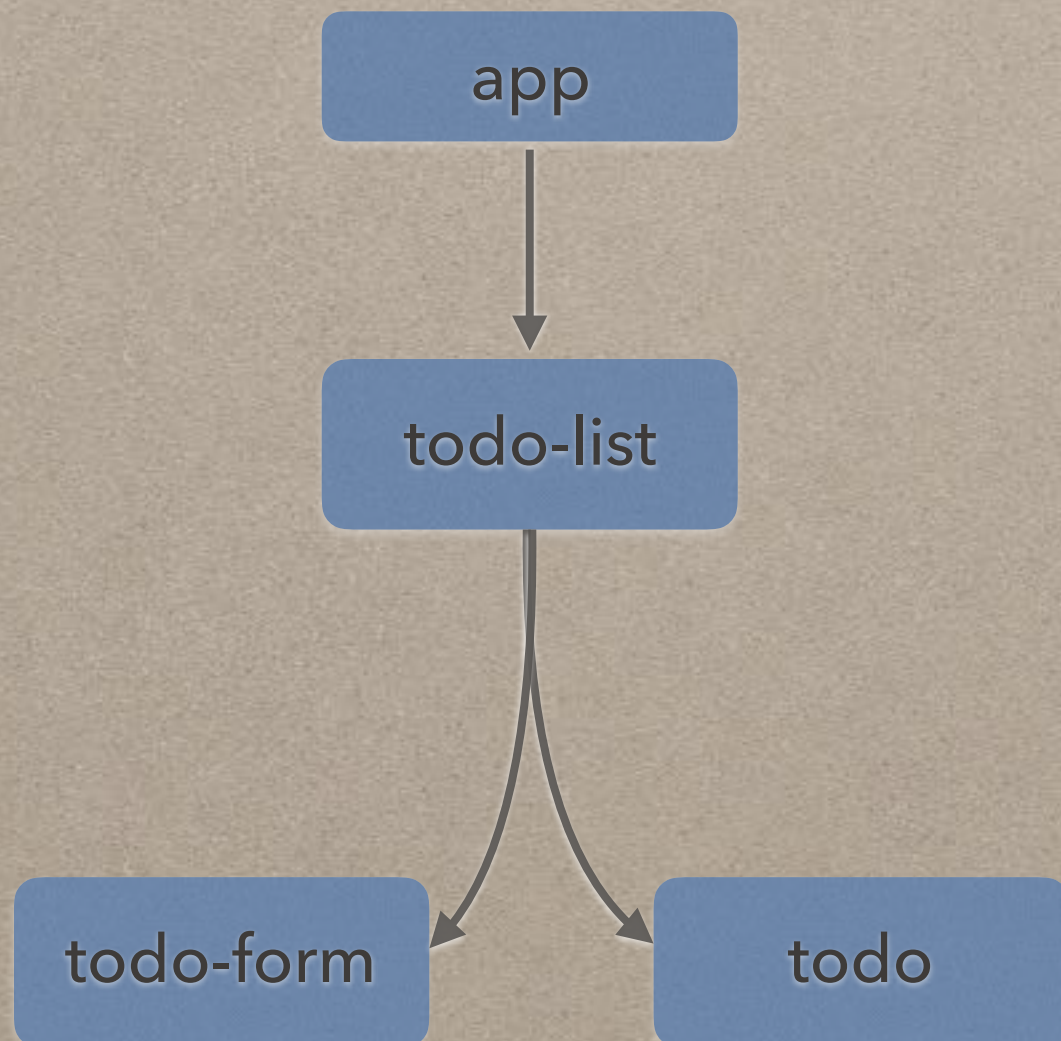
- Component Architecture (pattern)



ANGULAR, C'EST QUOI ?

NOUVEAUX CONCEPTS

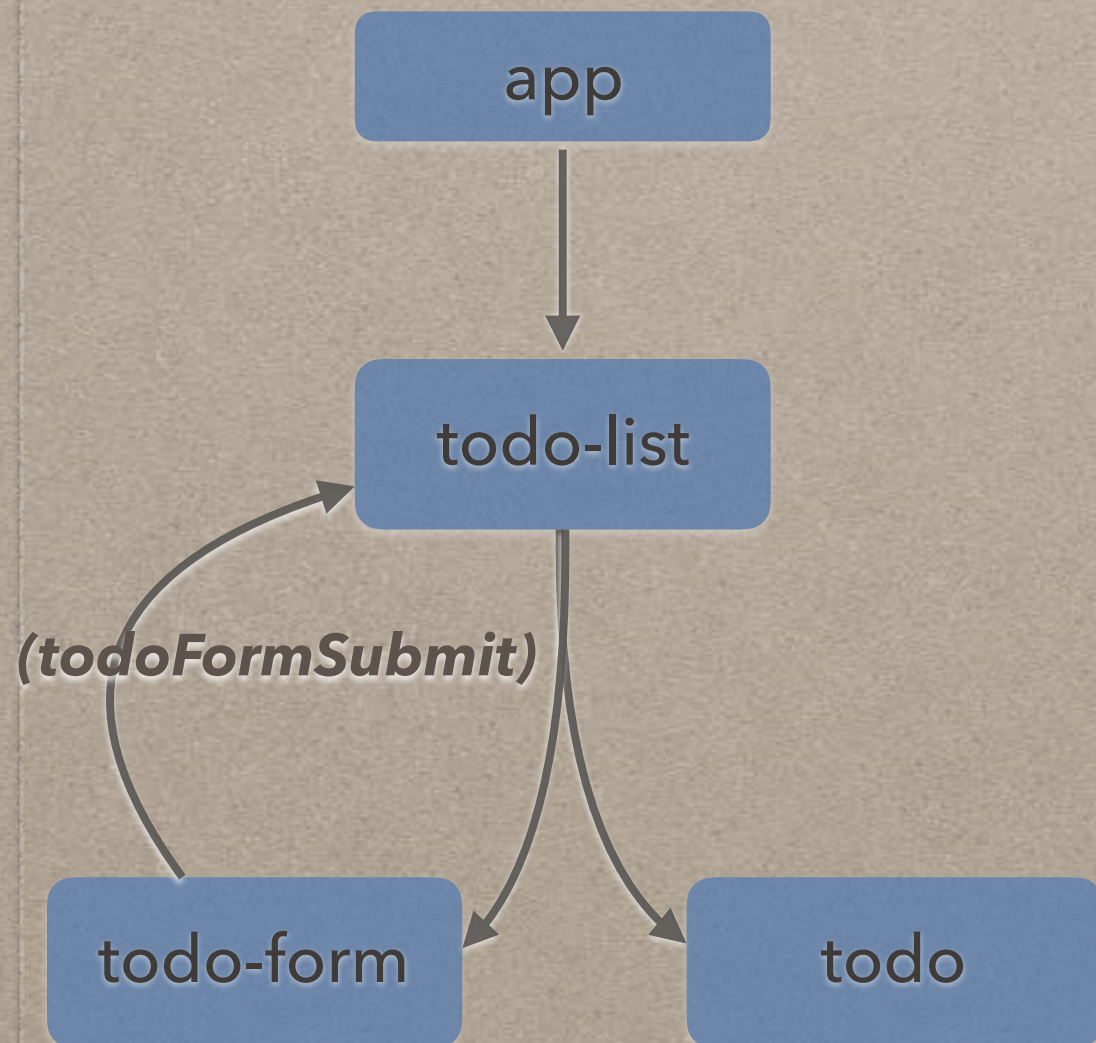
- Component Architecture (pattern)



ANGULAR, C'EST QUOI ?

NOUVEAUX CONCEPTS

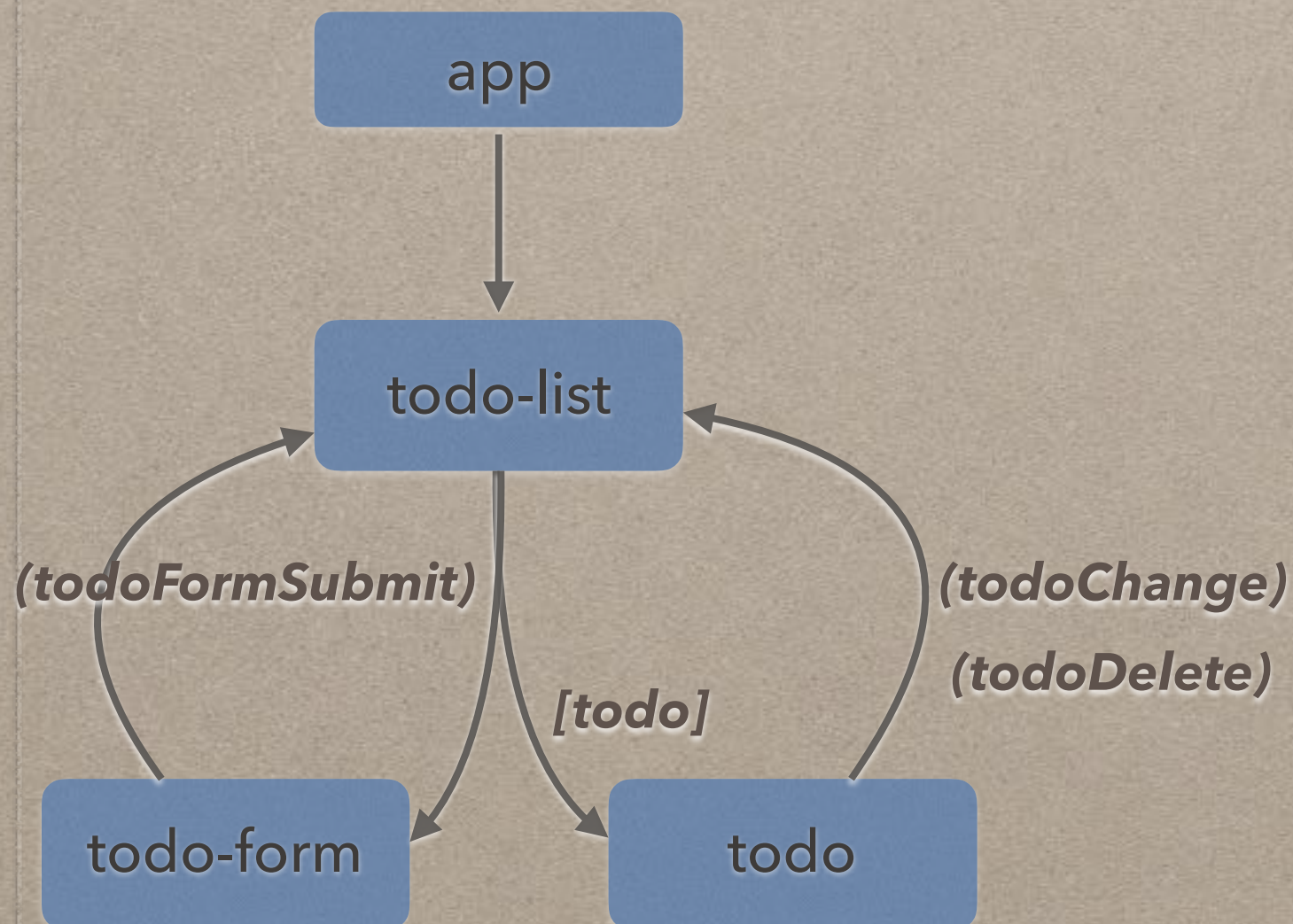
- Component Architecture (pattern)



ANGULAR, C'EST QUOI ?

NOUVEAUX CONCEPTS

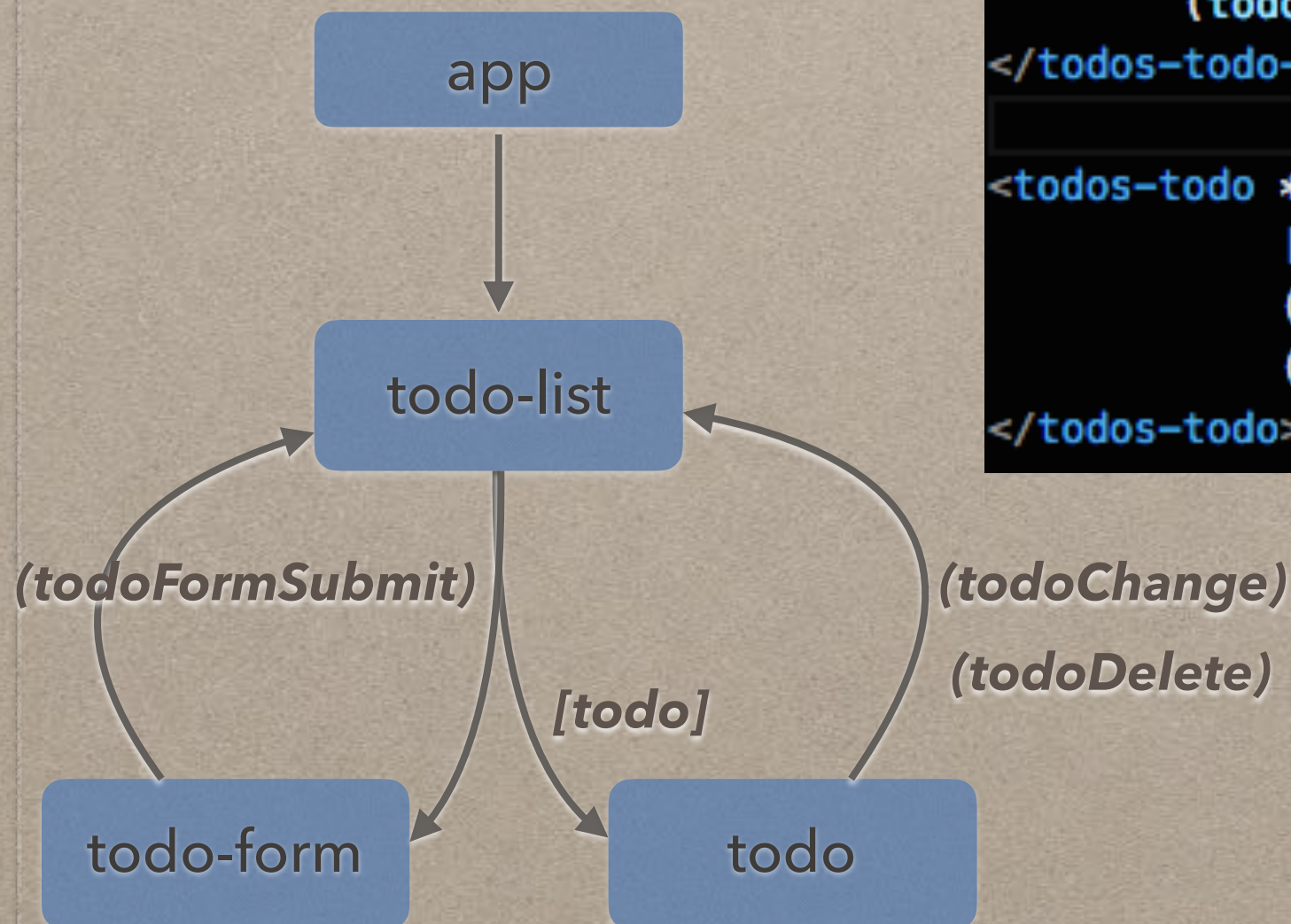
- Component Architecture (pattern)



ANGULAR, C'EST QUOI ?

NOUVEAUX CONCEPTS

- Component Architecture (pattern)



```
<todos-todo-form  
  (todoFormSubmit)="onTodoFormSubmit($event)">  
</todos-todo-form>  
  
<todos-todo *ngFor="let t of todos"  
  [todo]="t"  
  (todoChange)="onTodoChange($event)"  
  (todoDelete)="onTodoDelete($event)">  
</todos-todo>
```


ANGULAR, C'EST QUOI ?

NOUVEAUX CONCEPTS

- Component Architecture (pattern)
- Plate forme

ANGULAR, C'EST QUOI ?

NOUVEAUX CONCEPTS

- Component Architecture (pattern)
- Plate forme
 - Browser
 - Angular mobile toolkit
 - Angular Universal
 - Ionic2, NativeScript, ReactNative

ANGULAR, C'EST QUOI ?

NOUVEAUX CONCEPTS



ANGULAR, C'EST QUOI ?

NOUVEAUX CONCEPTS

- Component Architecture (pattern)
- Plate forme
- i18n

ANGULAR, C'EST QUOI ?

NOUVEAUX CONCEPTS

- Component Architecture (pattern)
- Plate forme
- i18n
 - ng-xi18n
 - format XLF

```
<trans-unit id="de18b431a067310d007a8c4fc1d28e8422f7421d" datatype="html">  
  <source>I'm hungry</source>  
  <target>Je suis affamé</target>  
  <note priority="1" from="description">this is a state of my body</note>  
  <note priority="1" from="meaning">i want to eat a big pony</note>  
</trans-unit>
```


ANGULAR, C'EST QUOI ?

NOUVEAUX CONCEPTS

- Component Architecture (pattern)
- Plate forme
- i18n
- ZoneJS

ANGULAR, C'EST QUOI ?

NOUVEAUX CONCEPTS

- Component Architecture (pattern)
- Plate forme
- i18n
- ZoneJS

```
> debutPresentation();  
> obtenirNomParticipants();  
> setTimeout(function(){  
    verifierSiNomCourt();  
}, 0);  
> donnerMangerNomCourt();  
> finPresentation();
```


ANGULAR, C'EST QUOI ?

NOUVEAUX CONCEPTS

- Component Architecture (pattern)
- Plate forme
- i18n
- ZoneJS
 - Génération de contexte d'exécution
 - Meilleur suivi des tâches asynchrones

ANGULAR, C'EST QUOI ?

NOUVEAUX CONCEPTS

- Component Architecture (pattern)
- Plate forme
- i18n
- ZoneJS
- Reactive programming

ANGULAR, C'EST QUOI ?

ENVIRONNEMENTS

"A command line interface for Angular"

Angular CLI

- initialisation du projet

```
$ ng new my-project
```

- création de fichiers

```
$ ng generate my-component
```

- build

```
$ ng build --target=production --env=production
```

- serve

```
$ ng serve --env=production
```


ANGULAR, C'EST QUOI ?

ENVIRONNEMENTS

"Server-side Rendering for Angular apps"

Angular Universal

ANGULAR, C'EST QUOI ?

ENVIRONNEMENTS

"Server-side Rendering for Angular apps"

Angular Universal

```
<html>
<head>
  <meta charset="UTF-8">
  <title>Universal Test</title>
</head>
<body>
  <demo-app></demo-app>
</body>
</html>
```


ANGULAR, C'EST QUOI ?

ENVIRONNEMENTS

"Server-side Rendering for Angular apps"

Angular Universal

```
<html>
<head>
  <meta charset="UTF-8">
  <title>Universal Test</title>
</head>
<body>
  <demo-app></demo-app>
</body>
</html>
```

```
<html><head>
  <meta charset="UTF-8">
  <title>Universal Test</title>
</head>
<body>
  <demo-app ng-version="4.0.0-rc.1-213e210">
    <h1>Universal Demo</h1>
    <a routerLink="/" href="/">Home</a>
    <a routerLink="/lazy" href="/lazy">Lazy</a>
    <router-outlet></router-outlet>
    <home-view ng-version="4.0.0-rc.1-213e210">
      <h3>Home View</h3>
    </home-view>
  </demo-app>
</body>
</html>
```


ANGULAR, C'EST QUOI ?

ENVIRONNEMENTS

"Server-side Rendering for Angular apps"

Angular Universal

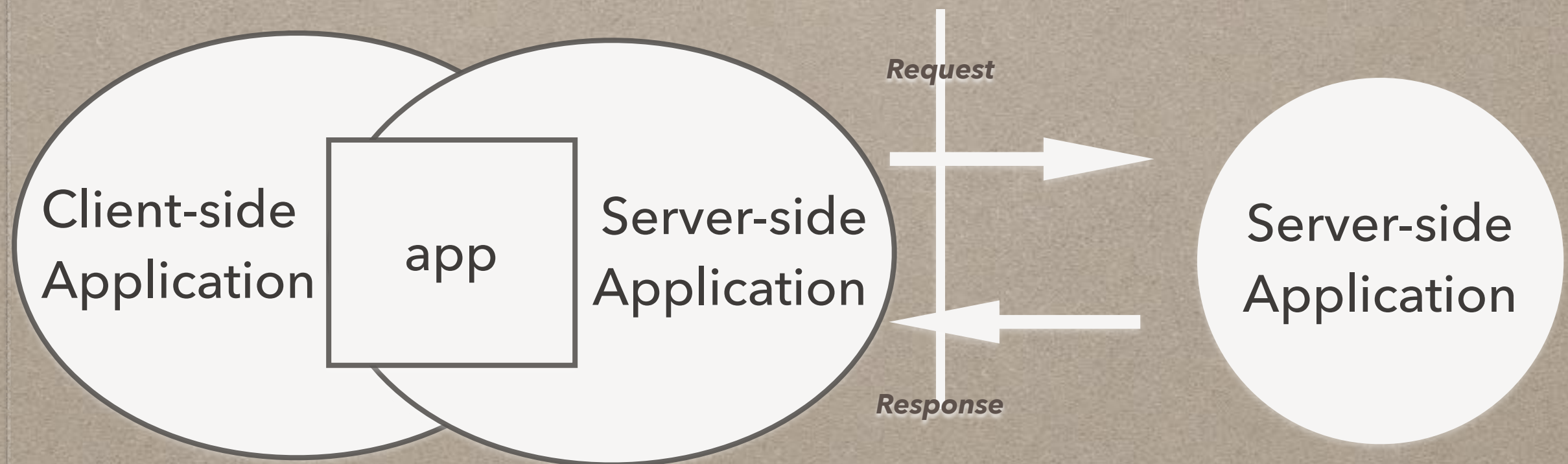
- Back End
- server-rendering
- SEO
- proxy
- cache

ANGULAR, C'EST QUOI ?

ENVIRONNEMENTS

"Server-side Rendering for Angular apps"

Angular Universal



ANGULAR, C'EST QUOI ?

ENVIRONNEMENTS

[illegible]

ANGULAR, C'EST QUOI ?

ENVIRONNEMENTS

- nodeJS
- TypeScript
- ES6 / ES7
- Webpack
- RxJS



Dart



SOMMAIRE

1. ~~Rappel AngularJS~~
2. ~~Angular, c'est quoi ?~~
3. AngularJS versus Angular
4. Oui... mais ?
5. Conclusion

ANGULARJS VS ANGULAR

SOMMAIRE

- Bootstrap une application
- Module
- \$controller / Component
- Binding

ANGULARJS VS ANGULAR

BOOTSTRAP UNE APPLICATION

AngularJS

```
// from JS
(function(){

    angular.bootstrap(document, ['AppModule']);

})();

// from HTML
<html data-ng-app="AppModule"></html>
```

Angular

```
import './polyfills.ts';

import { platformBrowserDynamic } from '@angular/platform-browser-dynamic';
import { enableProdMode } from '@angular/core';
import { environment } from './environments/environment';
import { AppModule } from './app/';

if (environment.production) {
    enableProdMode();
}

platformBrowserDynamic().bootstrapModule(AppModule);
```


ANGULARJS VS ANGULAR

MODULE

AngularJS

```
(function(){  
  
    angular.module("AppModule", [/*module dependencies*/]);  
  
})();
```

Angular

```
import { NgModule } from '@angular/core';  
  
@NgModule({  
    declarations: [],  
    imports:      [/*module dependencies*/],  
    bootstrap:    []  
})  
export class AppModule {}
```


ANGULARJS VS ANGULAR

\$CONTROLLER / COMPONENT

```
(function() {
  'use strict';
  function AppController() {
    var app = this;
    app.title = 'Hello Open';
  }

  angular.module("AppModule")
    .controller("AppController", AppController);

  angular.module('AppModule').component('AppComponent', {
    templateUrl: 'appComponent.html',
    controller: AppController
  });

})();

// from HTML
<div data-ng-controller="AppController as app">
  {{ app.title }}
</div>

// from appComponent
<span>Name: {{$ctrl.title}}</span>
```

```
import { Component } from '@angular/core';

@Component({
  selector: 'app-root',
  templateUrl: './app.component.html',
  styleUrls: ['./app.component.css']
})
export class AppComponent {

  title = 'Hello Open';

  constructor() { }

}

//app.component.html
<div>{{ title }}</div>

// AppModule
@NgModule({
  declarations: [AppComponent],
  /** ... */
})
export class AppModule {}

// Bootstrap
@NgModule({
  bootstrap: [AppComponent],
  /** ... */
})
export class AppModule {}
```


ANGULARJS VS ANGULAR

BINDING

AngularJS

```
angular.module('AppModule').component('TodoComponent', {  
  // .... //  
  binding:{  
    'todoChange': '&',  
    'todoDelete': '&',  
    'todo': '<'  
  },  
});
```

Angular

```
// .... //  
export class TodoComponent implements OnInit {  
  
  @Input() todo: Todo;  
  @Output() todoChange: EventEmitter<TodoChangeEventArgs> = new EventEmitter<TodoChangeEventArgs>();  
  @Output() todoDelete: EventEmitter<TodoDeleteEventArgs> = new EventEmitter<TodoDeleteEventArgs>();  
  
  constructor() {  
  }  
  // .... //  
}
```


ANGULARJS VS ANGULAR

DEMO



Let's rock'n component

ANGULARJS VS ANGULAR

SUMMARY

- Simplification
- Modularité
- Beaucoup de notions communes

SOMMAIRE

1. ~~Rappel AngularJS~~
2. ~~Angular, c'est quoi ?~~
3. ~~AngularJS versus Angular~~
4. Oui... mais ?
5. Conclusion

OUI... MAIS ?

- Migration compliquée (1 -> 2)
- Black box
- Des subtilités
- Manque d'harmonie des outils
- Tout n'est pas finalisé

OUI... MAIS ?



SOMMAIRE

1. ~~Rappel AngularJS~~
2. ~~Angular, c'est quoi ?~~
3. ~~AngularJS versus Angular~~
4. ~~Oui... mais ?~~
5. Conclusion

CONCLUSION

- Très moderne
- Meilleure fiabilisation du code
- Vers un Angular plus unifié

