



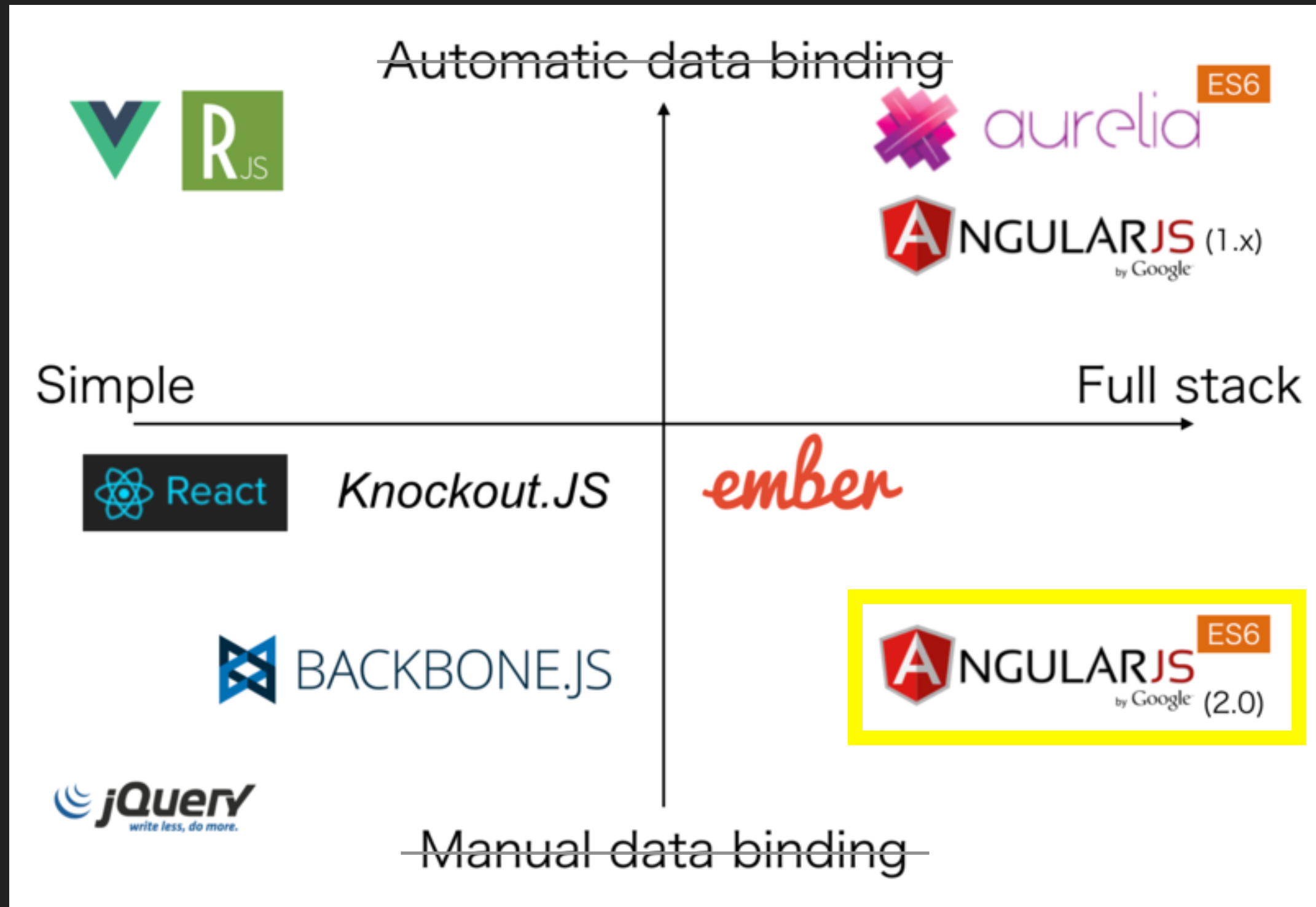
BETABEERS - 3 NOV 2016 - @MGLPIA

---

# ANGULAR 2



[Miguel García López](#) – Licensed under a [Creative Commons Attribution 4.0 Intl. License](#)



# (ANTES QUE NADA)



<https://www.youtube.com/watch?v=hQVTIJBZook>

README.md

## You Don't Know JS (book series)

This is a series of books diving deep into the core mechanisms of the JavaScript language. The first edition of the series is now complete.

UP & GOING	SCOPE & CLOSURES	this & OBJECT PROTOTYPES	TYPES & GRAMMAR	ASYNC & PERFORMANCE	ES6 & BEYOND
JS	JS	JS	JS	JS	JS

Please feel free to contribute to the quality of this content by submitting PR's for improvements to code snippets, explanations, etc. While typo fixes are welcomed, they will likely be caught through normal editing processes, and are thus not necessarily as important for this repository.

To read more about the motivations and perspective behind this book series, check out the [Preface](#).



<https://github.com/getify/You-Dont-Know-JS>

## AGENDA

- ▶ Generalidades Angular 2
- ▶ TypeScript
- ▶ Qué trae Angular 2
- ▶ Live Demo
- ▶ Q&A? (Interrupciones bienvenidas)

## ANGULAR 2

- ▶ Framework (vs. Library) para desarrollo aplicaciones JS
- ▶ Inicialmente: Web Apps (SPA)
  - ▶ Browser
  - ▶ Electron, NodeWebKit
- ▶ También: Server-Side rendering (WIP)
- ▶ También: Mobile con NativeScript

# HISTORIA

- ▶ Desarrollado por Google, sucesor Angular 1
- ▶ Poco o nada que ver con Angular 1 - Marketing
- ▶ Beta, CR releases problemáticas
  - ▶ Cambios API frecuentes
  - ▶ Introducción conceptos last-minute - Modules
  - ▶ Router "con cuña" - todo cambiado
  - ▶ Ahora: asentado - semantic versioning

# LENGUAJES

- ▶ Para Web: HTML + CSS + (JS)
- ▶ -> TypeScript <- – "Angular 2: Why TypeScript?"
- ▶ JavaScript
- ▶ Dart





# TECNOLOGÍAS

- ▶ NPM - `package.json`
- ▶ TypeScript compiler
- ▶ Polyfills
- ▶ Web Package Manager
  - ▶ Originalmente, SystemJS
  - ▶ Ahora: -> Webpack <-
  - ▶ Cualquiera, en teoría
- ▶ => *angular-cli*





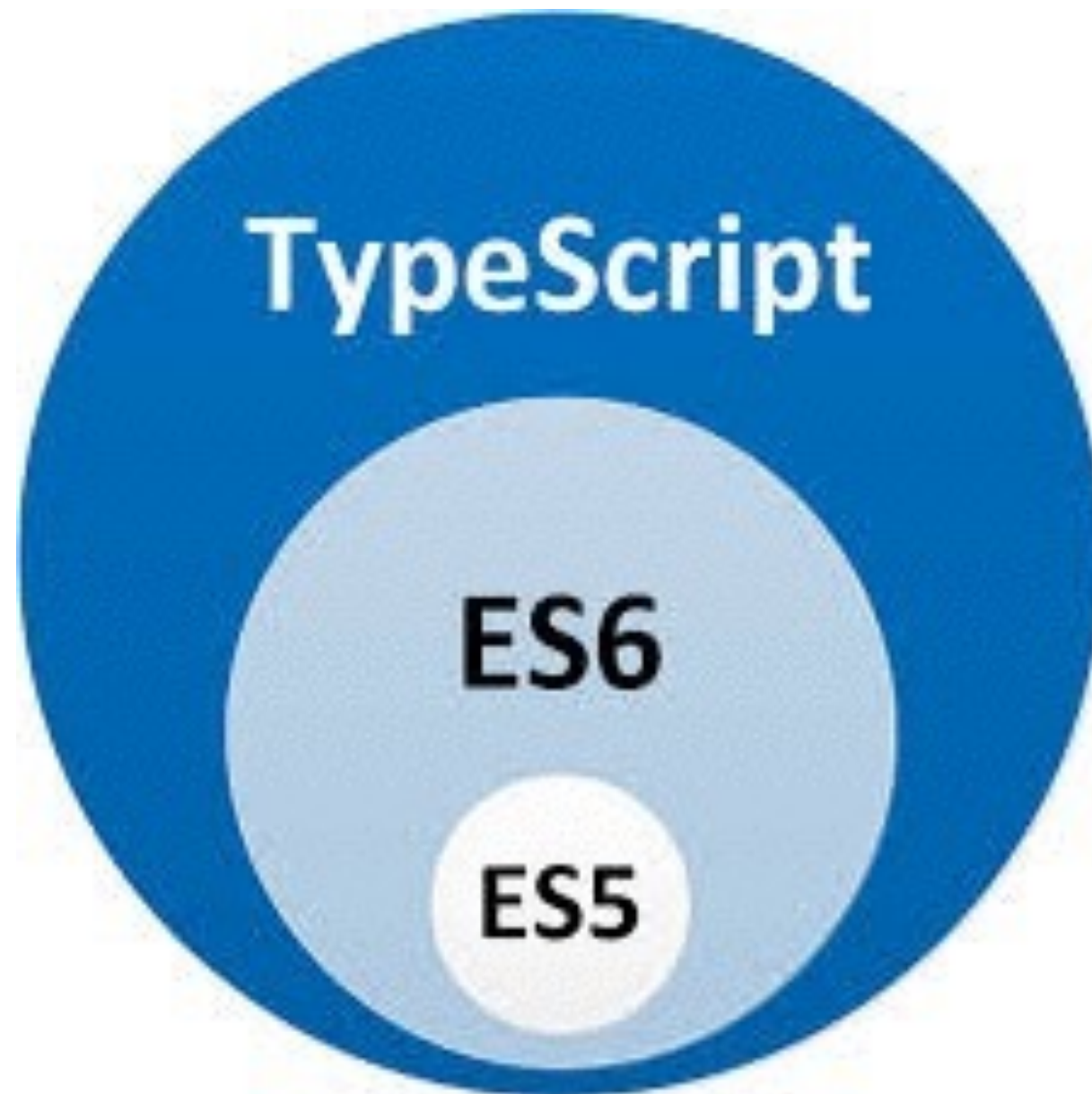
## ADVANCED ANGULAR 2 DEMO SEED

- ▶ <https://github.com/NathanWalker/angular-seed-advanced>

*"An advanced Angular2 seed project with support for ngrx/store, ngrx/effects, ng2-translate, angularartics2, lodash, NativeScript (\*native\* mobile), Electron (Mac, Windows and Linux desktop) and more."*

- ▶ En esta presentación: Web, exclusivamente

# LENGUAJE CON TIPADO QUE COMPILA A JAVASCRIPT



# TIPADO ESTRUCTURAL — INFERENCIA DE TIPOS

```
interface User {
  email: string;
  fullName?: string;
}
```

OPCIONAL

```
function registerUser(user: User) {
  // ...
  return user;
}
```

INFERENCIA DEL TIPO  
RETORNO

NO COMPILA

TIPADO ESTRUCTURAL: NO "IMPLEMENTS"

```
const doesNotCompile = registerUser({});
const registeredUser1 = registerUser({ email: 'u@e.com' });
const registeredUser2 = registerUser({ email: 'u@e.com',
                                       fullName: 'Miguel G' });
```

INFERENCIA DEL TIPO DE LA  
VARIABLE

# FLEXIBLE — PRODUCTIVO

```
class UserDisplayComponent implements OnInit {
```

AUTO ATTRIBUTE

DEFAULT VALUE

```
// no need for: private readonly http: Http;
```

```
  constructor(private http: Http, public username: string = 'nousername') {
    // ... (no need for: this.http = http)
  }
```

CASCA SI NO (POR EL "IMPLEMENTS")

```
  ngOnInit() {
```

```
// ...
```

```
    const userInfo = loadUserInformation();
```

```
    if (userInfo instanceof Promise) {
```

```
      userInfo.then(...);
```

```
    } else if (userInfo instanceof Observable) {
```

```
      userInfo.subscribe(...);
```

```
    }
```

```
  }
```

SIN CASTING

UNION TYPES

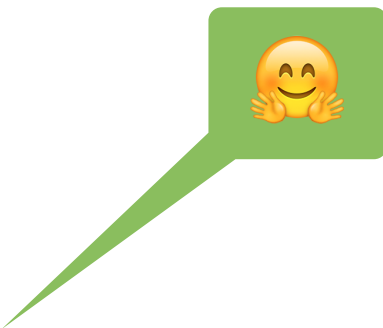

```
  private loadUserInformation(): UserInformation
    | Promise<UserInformation>
    | Observable<UserInformation> {
```

```
// ... this.http(...); return ...
```

```
  }
```

```
}
```

# OJO! — ESTO ES JAVASCRIPT POR DEBAJO

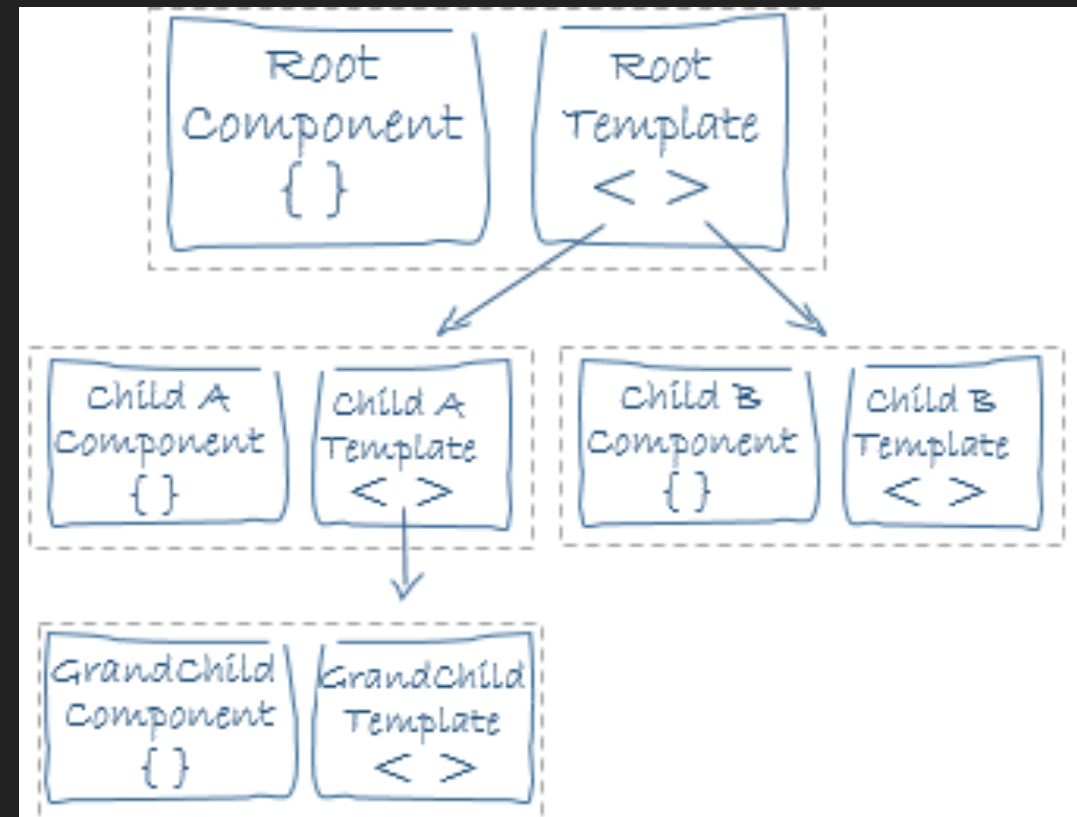
- ▶ Ni "private" ni leches
- ▶ "any" Y a tomar vientos
- ▶ `return response.json() as User[];`


- ▶ Esto no es Java
- ▶ Compila a ES5 o ES6 – common-js, amd, ...

## ME GUSTA

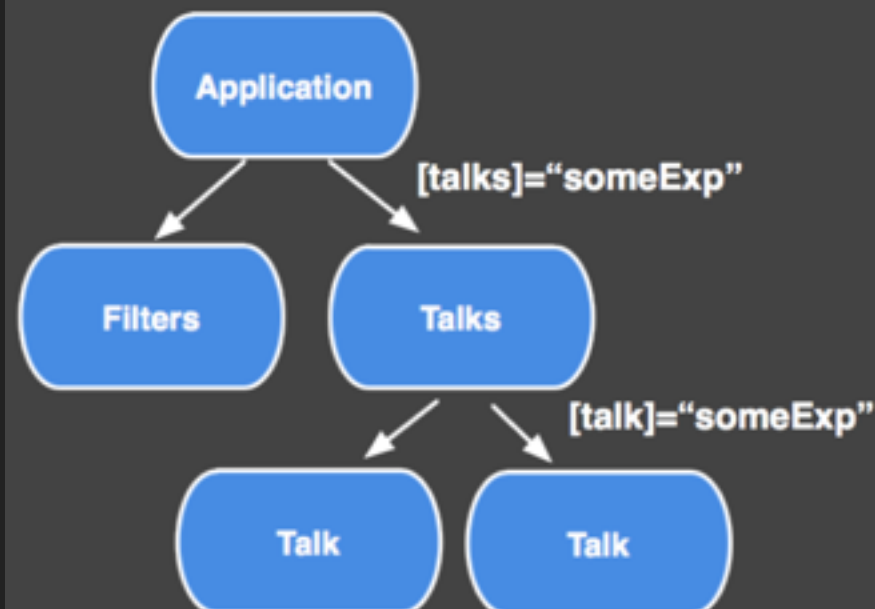
- ▶ Tipado, pero flexible
- ▶ Fácil interoperabilidad con JS – [DefinitelyTyped](#) – declare \$ : any ;
- ▶ Tipos ~ contratos
  - ▶ Expresan intencionalidad
  - ▶ Comprobables en compilación
- ▶ Tooling Support
  - ▶ Autocompletado, Autoimports
  - ▶ Refactorings – Renames semánticos varios ficheros, ...

# CONCEPTOS BASE ANGULAR 2

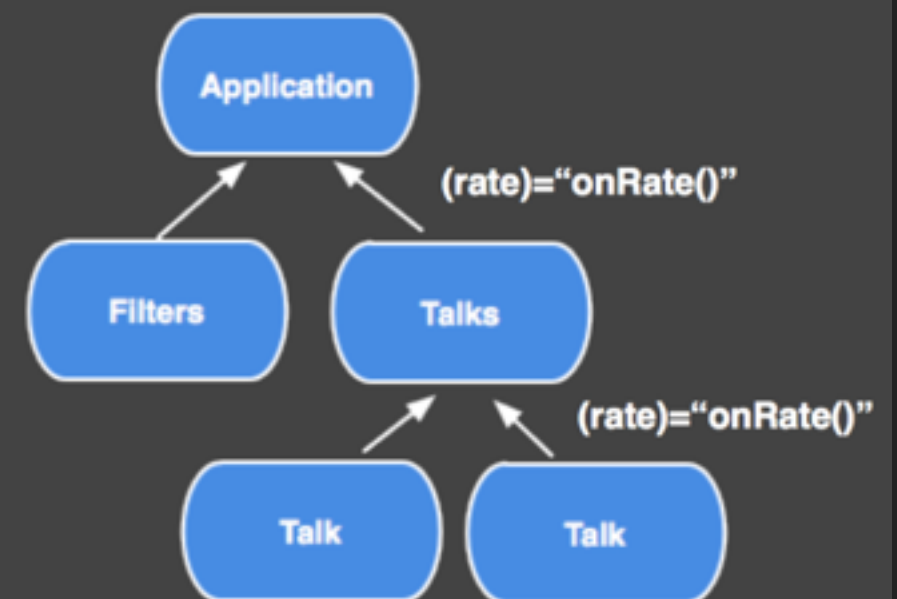
- ▶ Component, Template
- ▶ Directive
- ▶ Pipe
- ▶ Route
- ▶ Inyección Dep.
- ▶ Module



Parent -> Child



Child -> Parent





## OTROS EN ANGULAR 2

- ▶ Live vs AoT Compiling
- ▶ Http Module – Built-in JSON
- ▶ RxJs – Observables vs. Promises vs. Callbacks
- ▶ i18n – OJO AoT app por locale
- ▶ Forms + Validation
- ▶ Zone.js
- ▶ Built-in Testing Support
- ▶ No jQuery req.