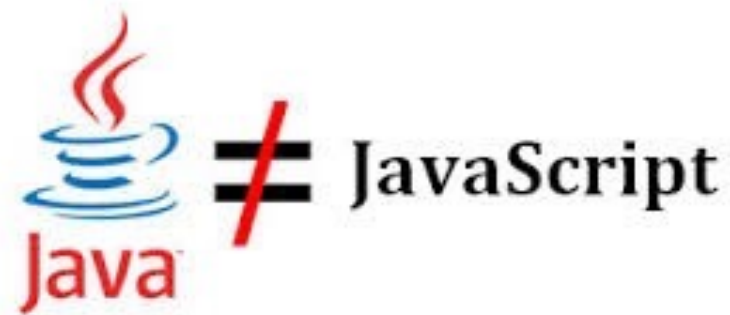


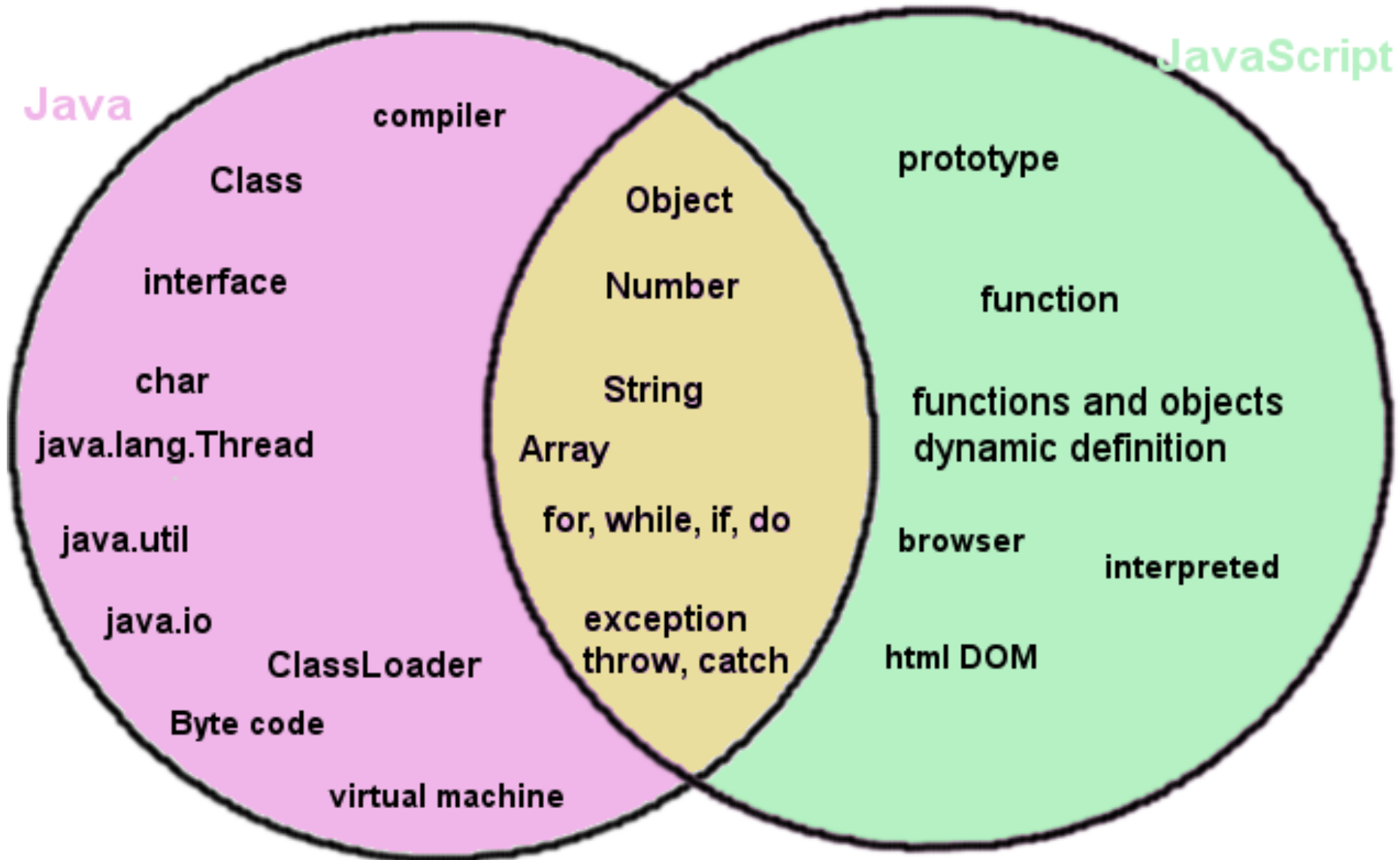
Introducción a JavaScript con Node.JS, TypeScript, AngularJS2

Alejandro Romero - @alejo8591

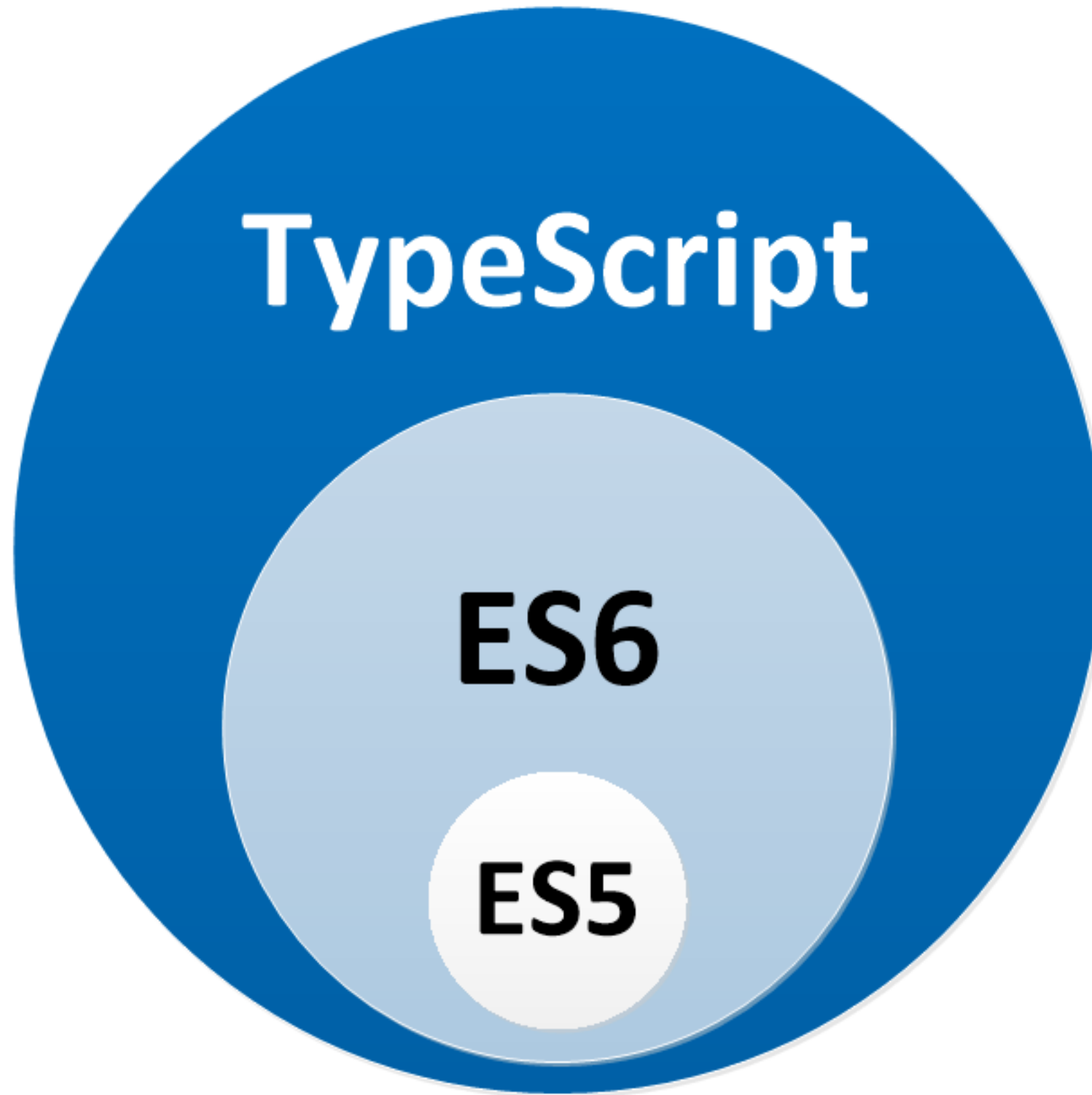
JS



Java is NOT JavaScript









STOP!



+



Node Package Modules (NPM)

JS



Imagen: <http://ow.ly/M4ZMZ>



JS

package.json



TypeScript

JS

Variables y Comentarios en TypeScript

TypeScript

JS

Arrays en TypeScript

TypeScript

JS

Objetos y Funciones en TypeScript

Loops y Saltos en TypeScript

TypeScript

JS

OOP en TypeScript

TypeScript

JS

Accesibles en	public	private	protected
Instancias de clases	Sí	No	No
clases	Sí	Sí	Sí
Clases hijas	Sí	No	Sí

Eventos: Document Object Model (DOM)

TypeScript

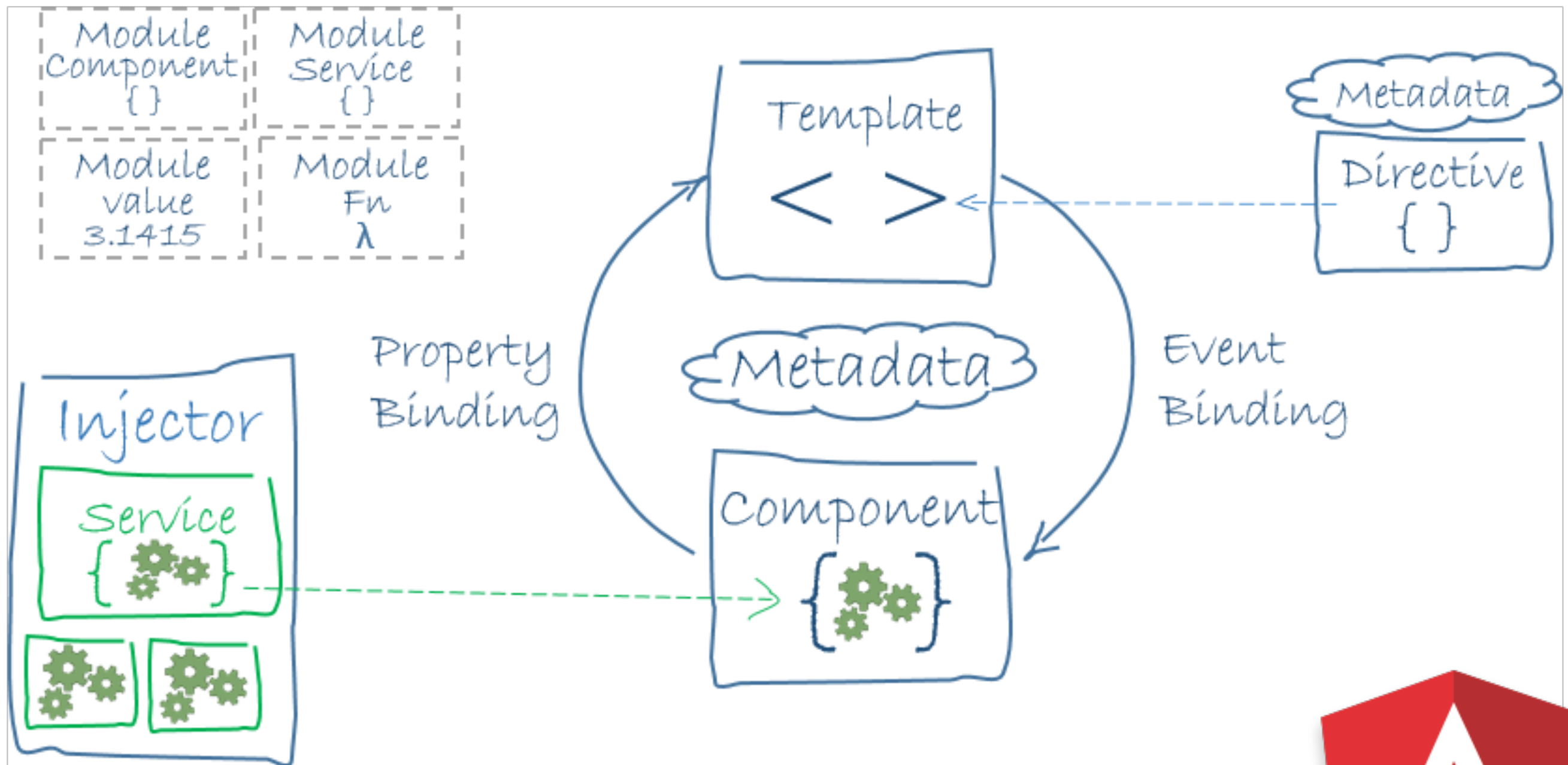


STOP!



Imagen: <https://goo.gl/83ygwT>







JS

Module
Component
{ }



JS

app/app.module.ts

```
import { NgModule }      from '@angular/core';
import { BrowserModule }  from '@angular/platform-browser';

@NgModule({
  imports:      [ BrowserModule ],
  providers:    [ Logger ],
  declarations: [ AppComponent ],
  exports:      [ AppComponent ],
  bootstrap:    [ AppComponent ]
})
export class AppModule { }
```



JS

app/main.ts

```
import { platformBrowserDynamic } from '@angular/platform-browser-dynamic';  
import { AppModule } from './app.module';  
  
platformBrowserDynamic().bootstrapModule(AppModule);
```




JS

Library Module

Component
{ }

Directive
{ }

Service
{ }

Value
3.1415

Fn
 λ



JS

app/app.module.ts (minimal)

```
import { NgModule } from '@angular/core';
import { BrowserModule } from '@angular/platform-browser';

import
    { AppComponent } from '../app.component';

@NgModule({
  imports: [ BrowserModule ],
  declarations: [ AppComponent ],
  bootstrap: [ AppComponent ]
})
export class AppModule { }
```



JS

app/app.component.ts (minimal)

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

```
@Component({  
  selector: 'my-app',  
  template: '<h1>{{title}}</h1>',  
})  
export class AppComponent {  
  title = 'Minimal NgModule';  
}
```



JS

Hero
Component

{ }





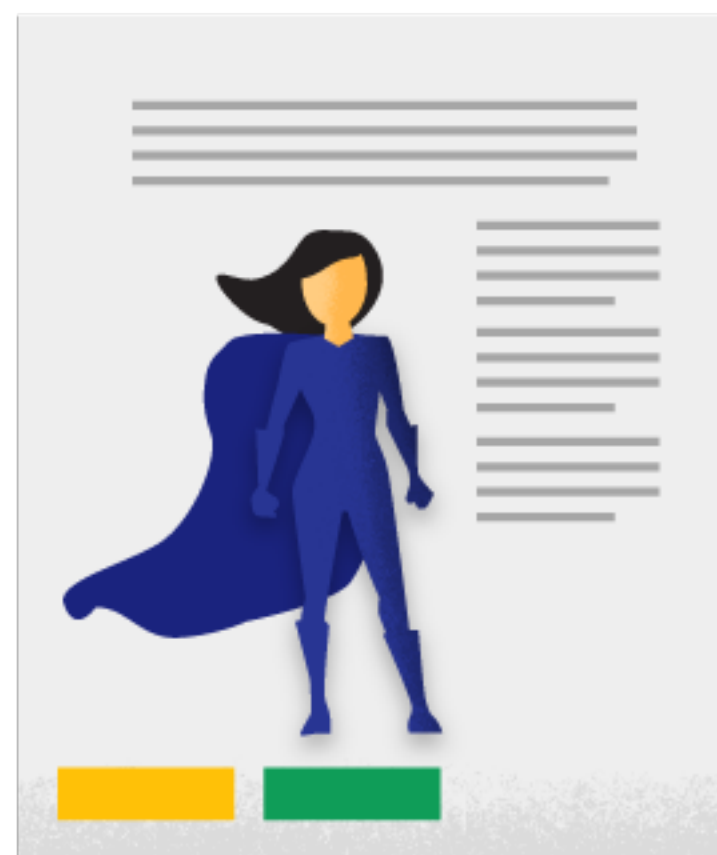
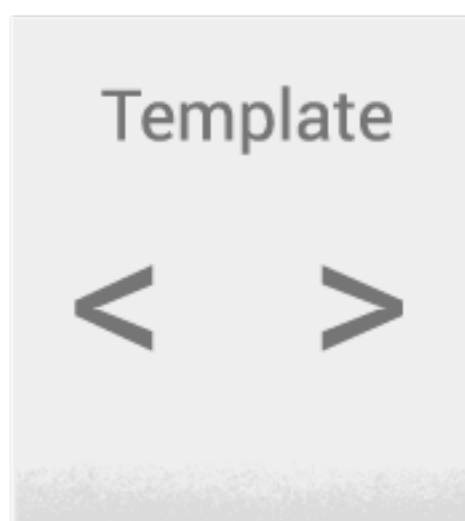
JS

app/hero-list.component.ts (class)

```
export class HeroListComponent implements OnInit {  
  heroes: Hero[];  
  selectedHero: Hero;  
  
  constructor(private service: HeroService) { }  
  
  ngOnInit() {  
    this.heroes = this.service.getHeroes();  
  }  
  
  selectHero(hero: Hero) { this.selectedHero = hero; }  
}
```




JS





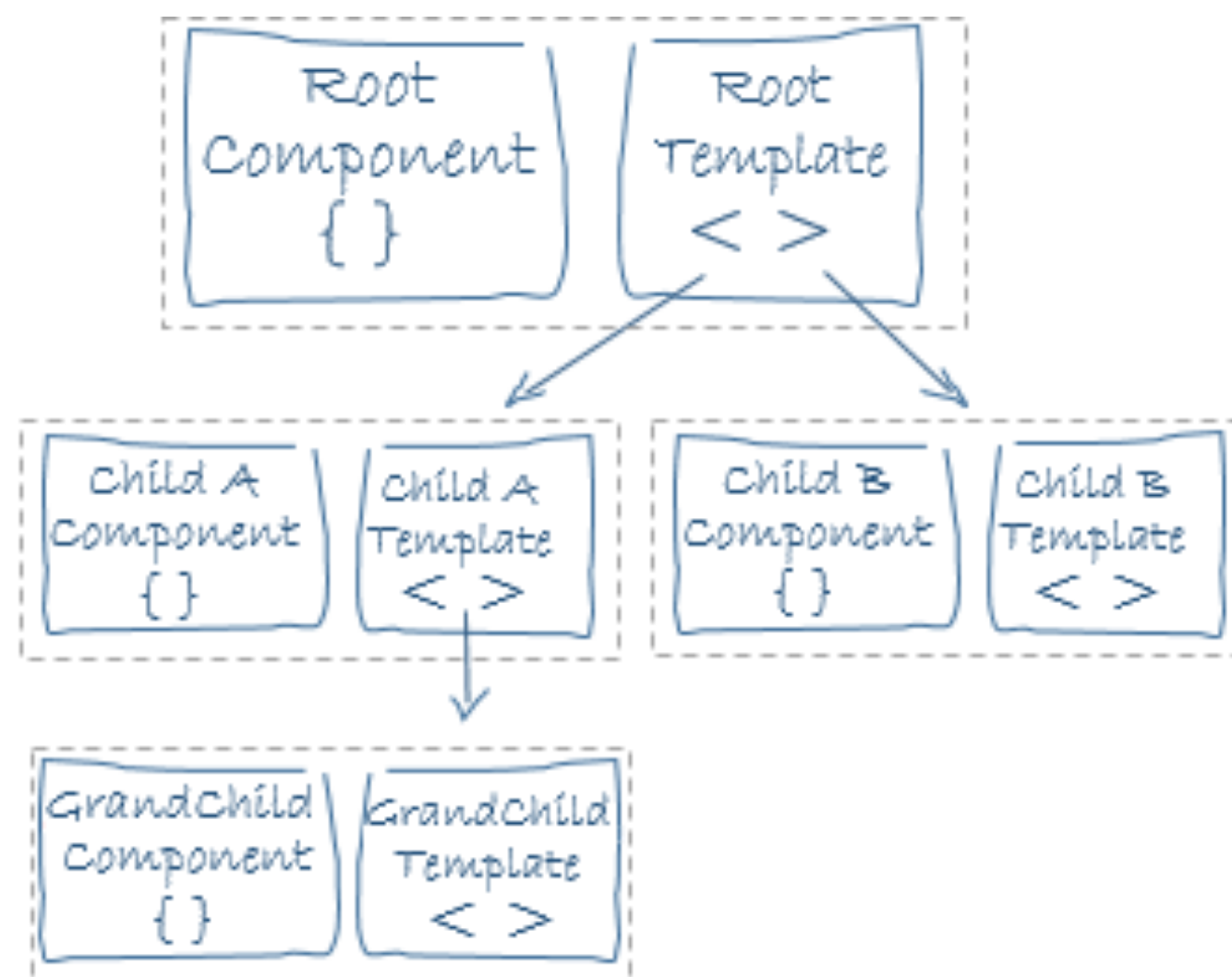
JS

app/hero-list.component.html

```
1. <h2>Hero List</h2>
2.
3. <p><i>Pick a hero from the list</i></p>
4. <ul>
5.   <li *ngFor="let hero of heroes" (click)="selectHero(hero)">
6.     {{hero.name}}
7.   </li>
8. </ul>
9.
10. <hero-detail *ngIf="selectedHero" [hero]="selectedHero"></hero-detail>
```



JS





JS

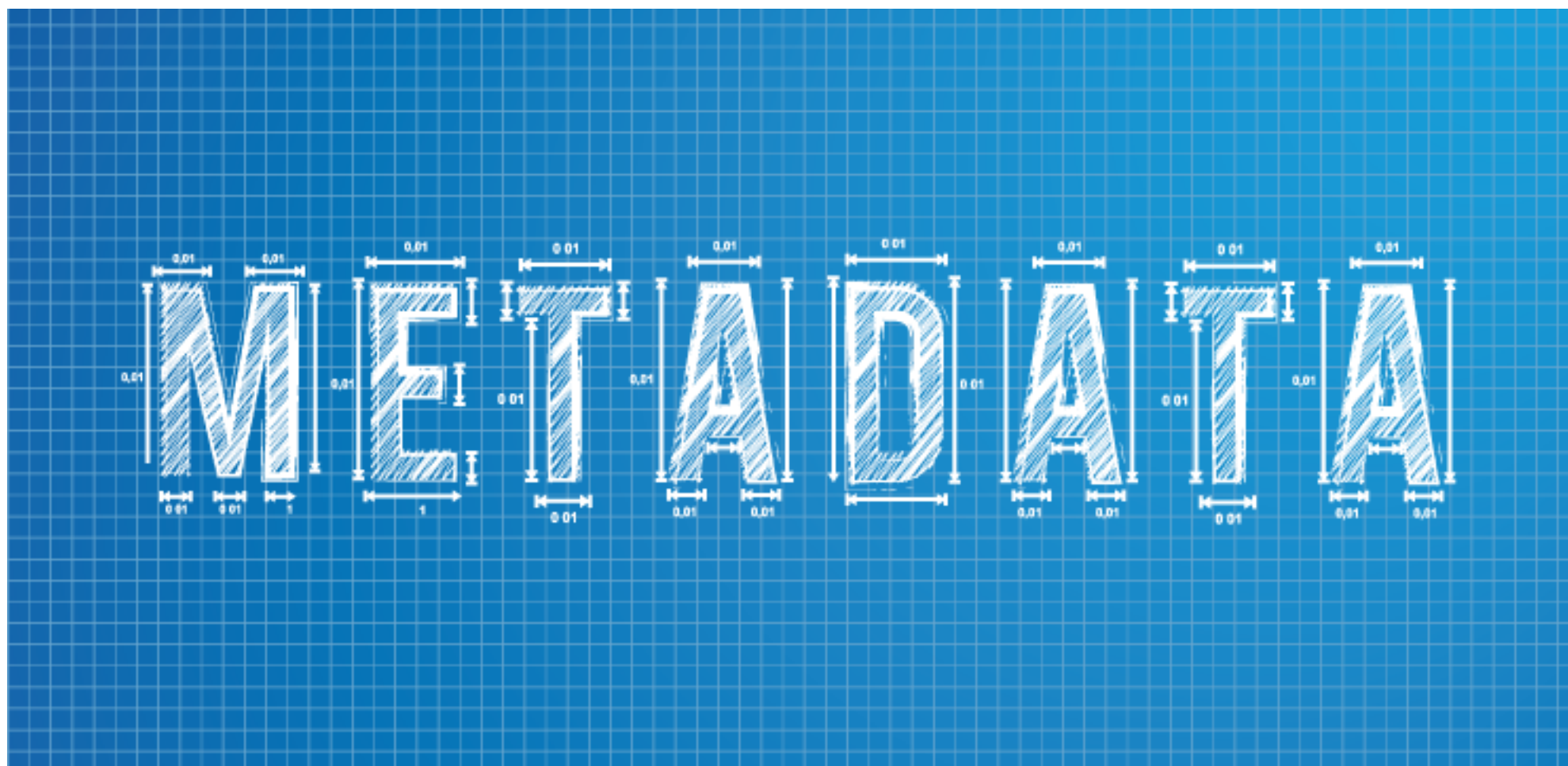


Imagen: <https://goo.gl/tGk6oX>





JS

app/hero-list.component.ts (metadata)

```
@Component({  
  moduleId: module.id,  
  selector: 'hero-list',  
  templateUrl: 'hero-list.component.html',  
  providers: [ HeroService ]  
})  
export class HeroListComponent implements OnInit {  
  /* . . . */  
}
```



JS

Template

< >

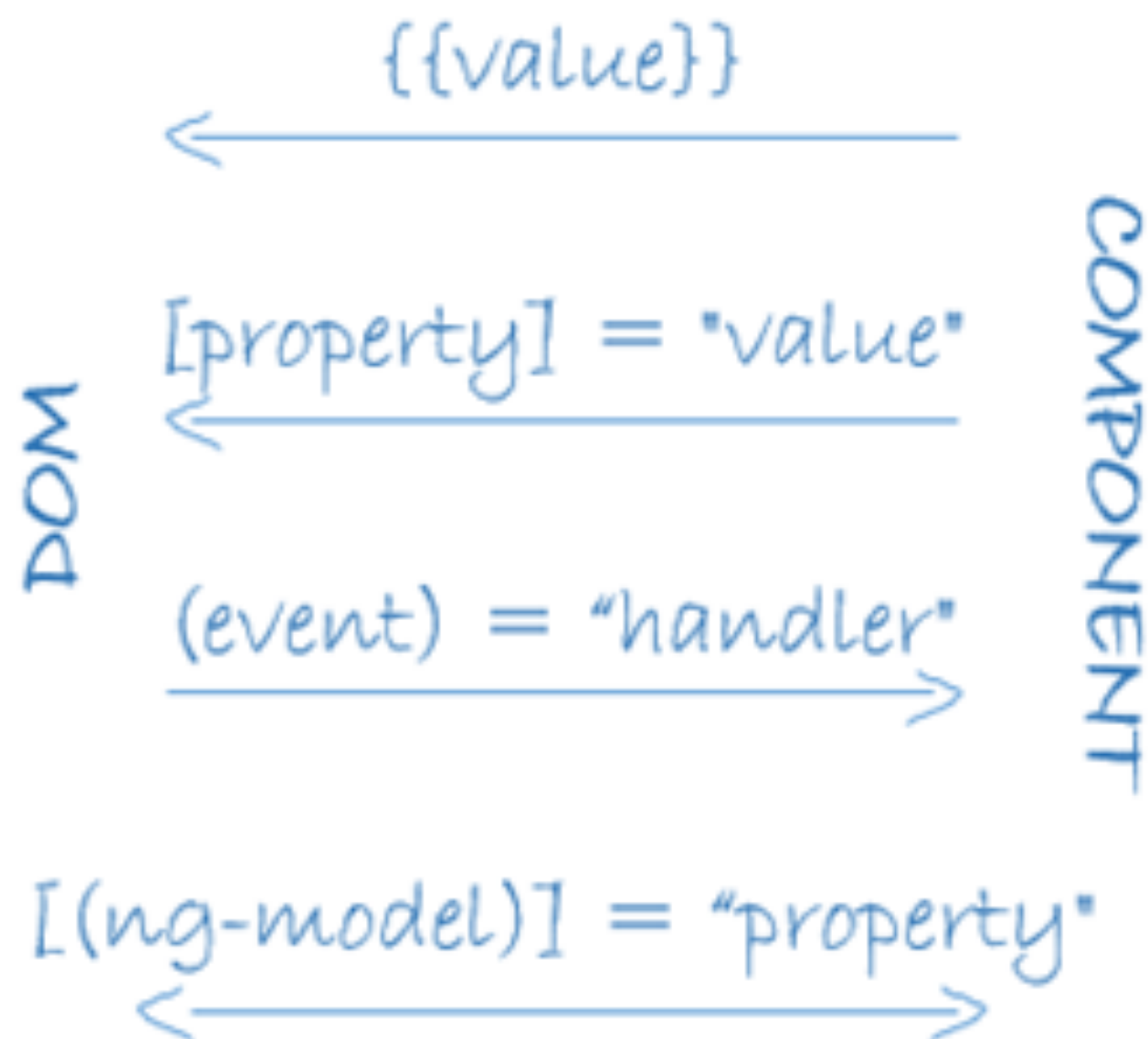
Metadata

Component

{ }



JS





JS

app/hero-list.component.html (binding)

```
<li>{{hero.name}}</li>  
<hero-detail [hero]="selectedHero"></hero-detail>  
<li (click)="selectHero(hero)"></li>
```



JS

Metadata

Directive
{ }



JS

app/hero-list.component.html (structural)

```
<li *ngFor="let hero of heroes"></li>  
<hero-detail *ngIf="selectedHero"></hero-detail>
```



JS





JS

app/hero.service.ts (class)

```
export class HeroService {  
  private heroes: Hero[] = [];  
  
  constructor(  
    private backend: BackendService,  
    private logger: Logger) { }  
  
  getHeroes() {  
    this.backend.getAll(Hero).then( (heroes: Hero[]) => {  
      this.logger.log(`Fetched ${heroes.length} heroes.`);  
      this.heroes.push(...heroes); // fill cache  
    });  
    return this.heroes;  
  }  
}
```

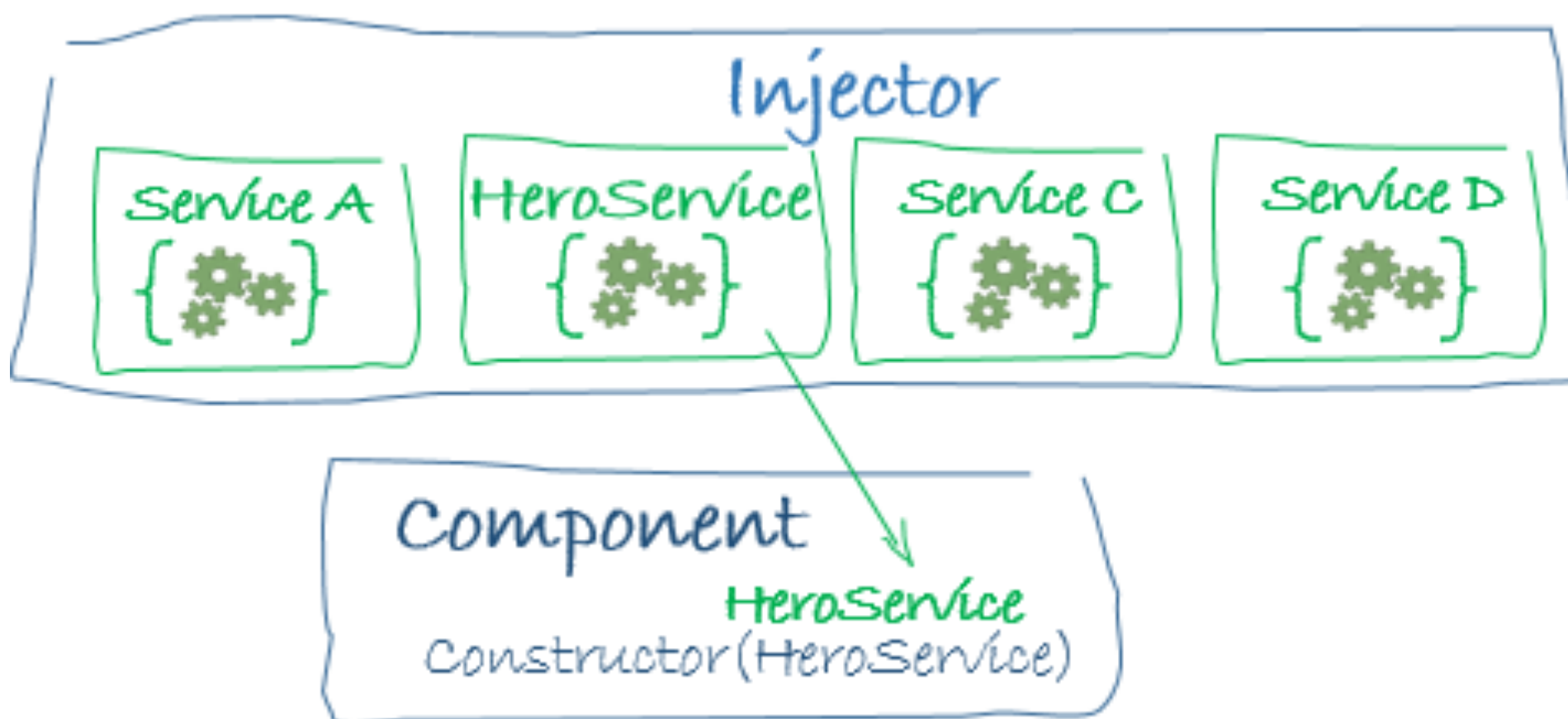


Dependency Injection





JS





JS

app/app.module.ts (module providers)

```
providers: [  
    BackendService,  
    HeroService,  
    Logger  
],
```



JS

app/hero-list.component.ts (component providers)

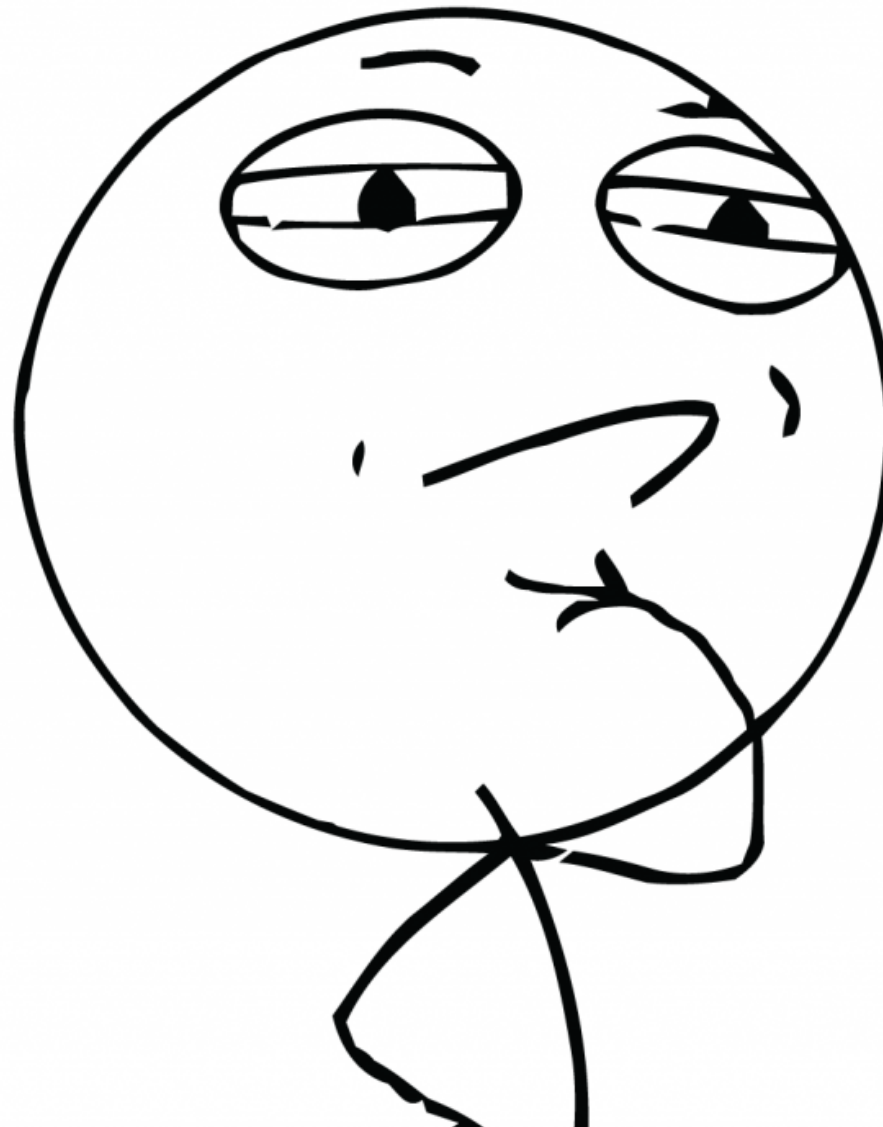
```
@Component({  
  moduleId: module.id,  
  selector: 'hero-list',  
  templateUrl: 'hero-list.component.html',  
  providers: [ HeroService ]  
})
```



First Application

JS

CHALLENGE CONSIDERED





JS

package.json



JS

tsconfig.json



JS

typings.json



JS

system.config.json



JS

Aplicando Clase de un modelo



JS

Aplicando Clase de un
modelo y separando
templates



JS

Aplicando Clase de un modelo e iterando



JS

Multipl Componen



JS

¿Qué es un Mock?

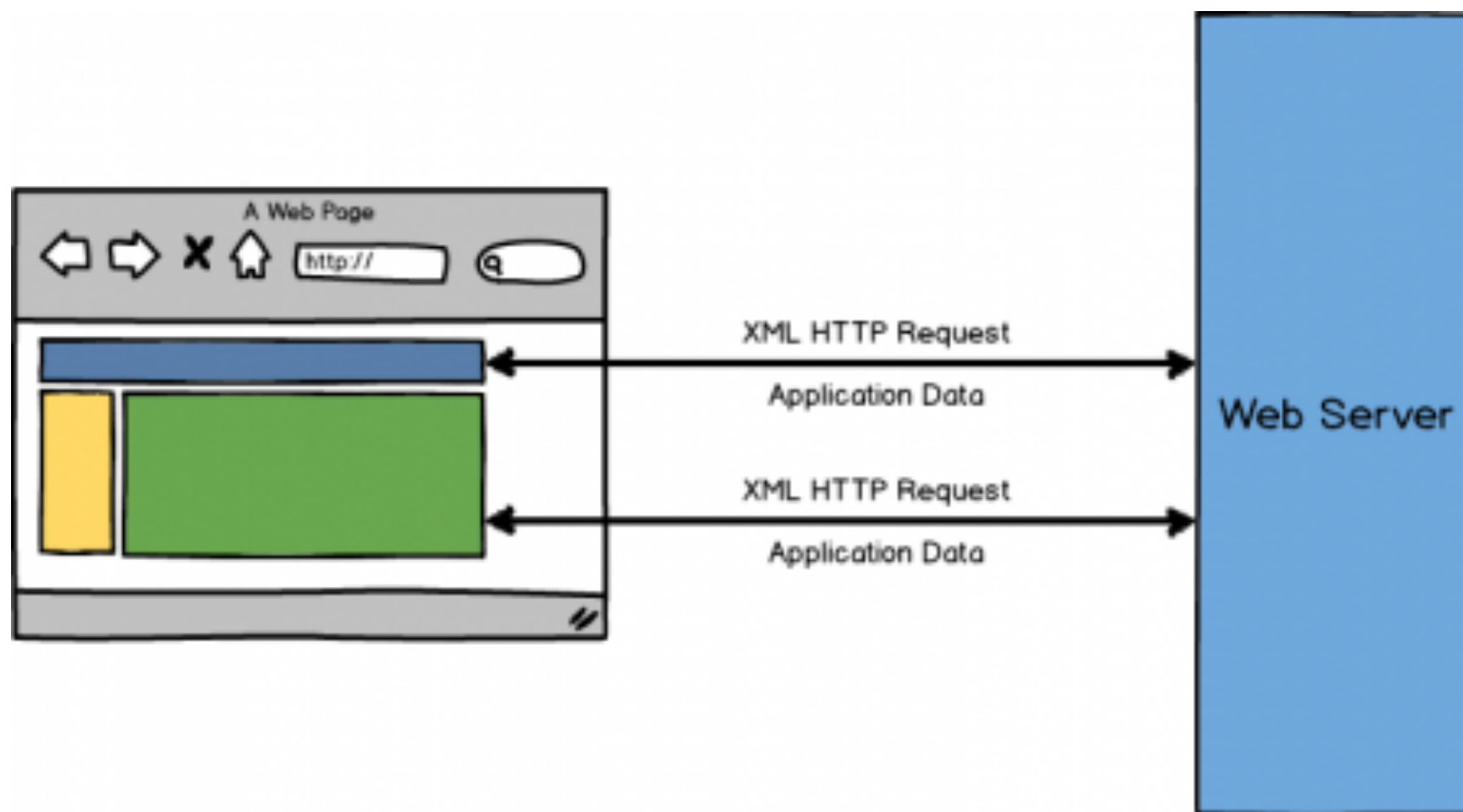


JS

SPI?



JS



Single Page Application Elements within a Page



JS

Multiples
Componentes + mock
+ routing



JS

Multiples Componentes más mock y routing

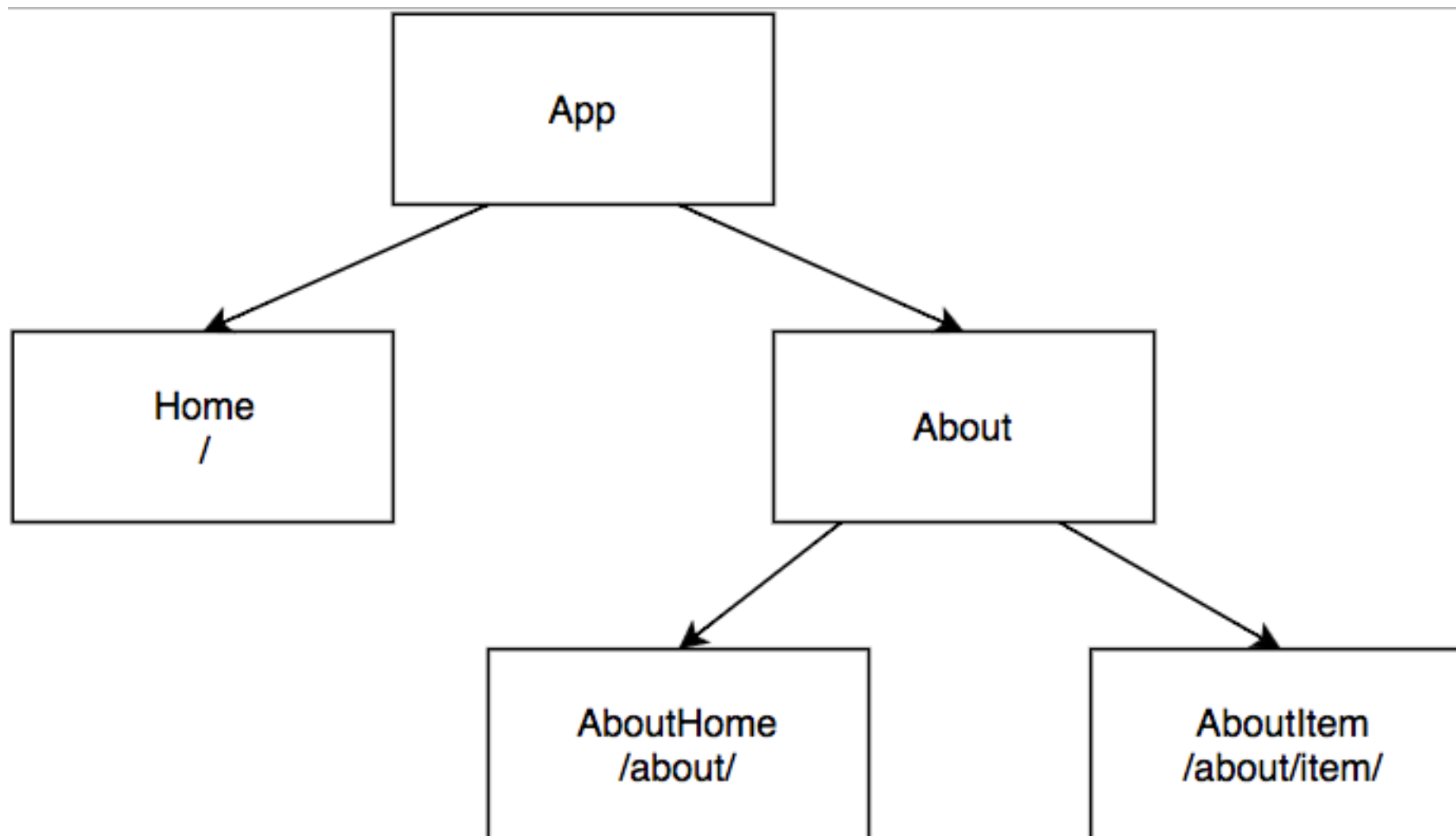


JS

Routing

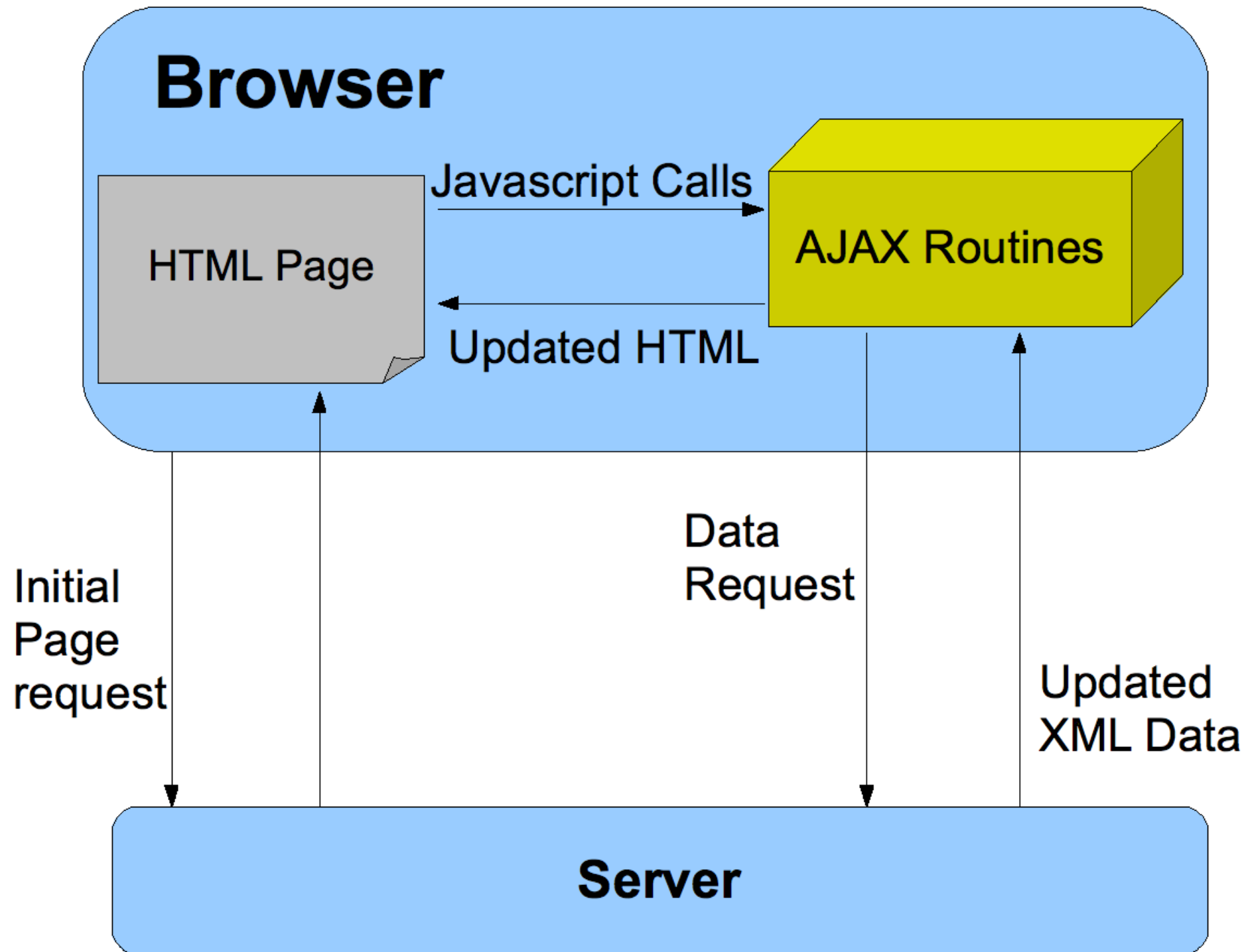


JS



JS



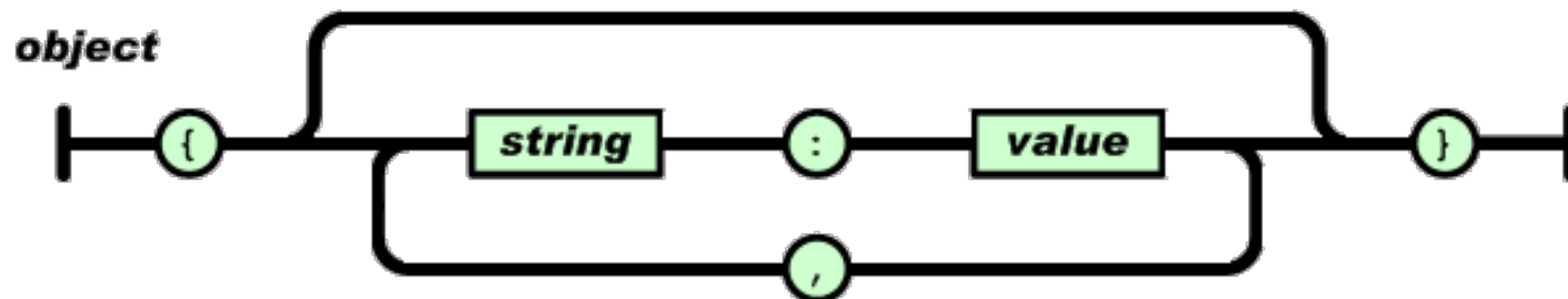


JS

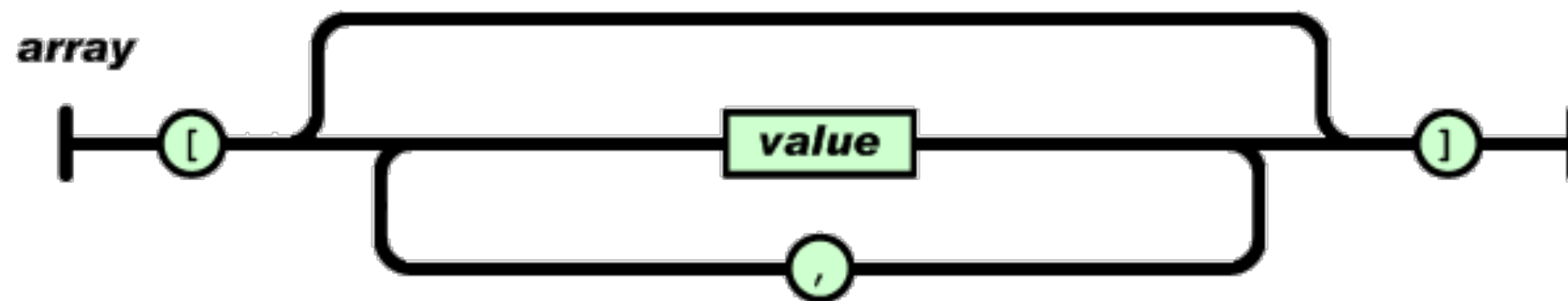
{JSON}



{JSON}

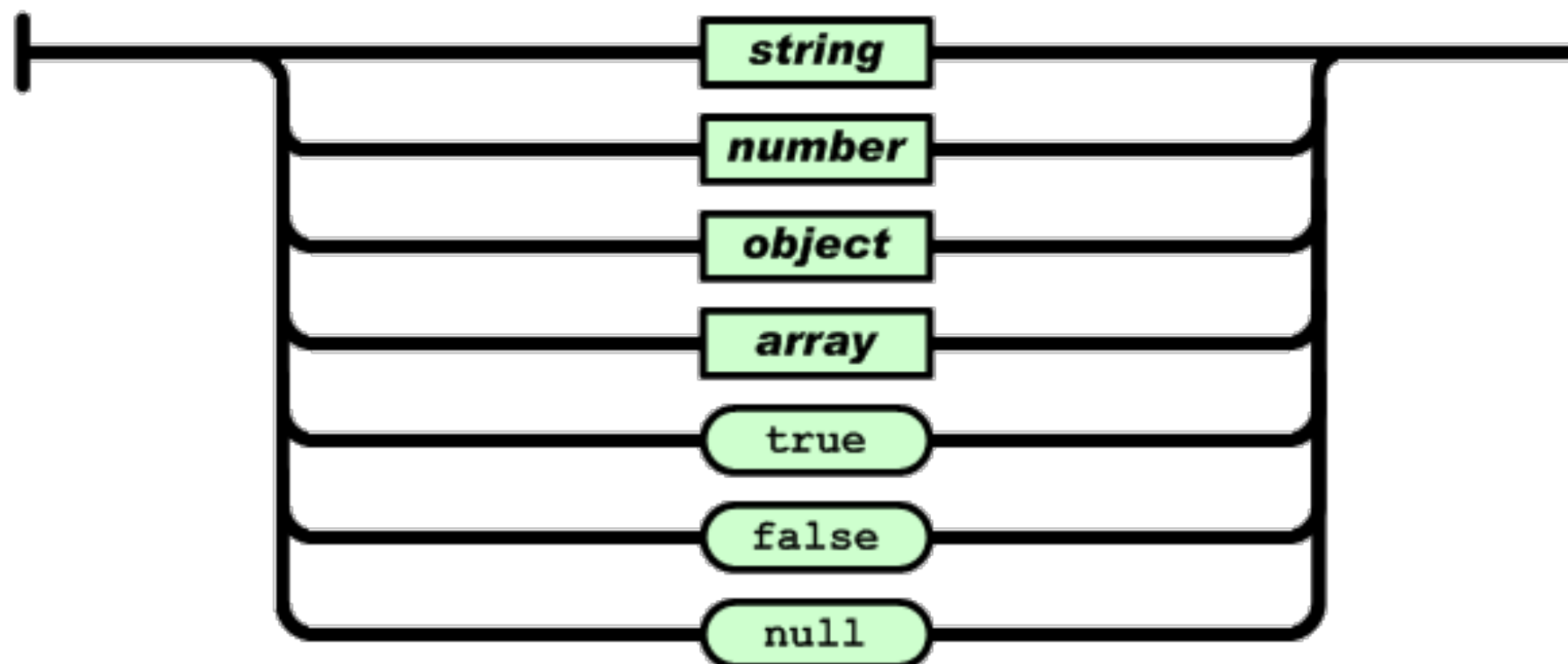


{JSON}



{JSON}

value



JS

In Memory

Web API



JS

http://

Imagen: <https://goo.gl/VmrtBv>



JS



Imagen: <https://goo.gl/YPu3Qy>



Observables	Promise
Observables handle multiple values over time	Promises are only called once and will return a single value
Observables are cancellable	Promises are not cancellable



JS

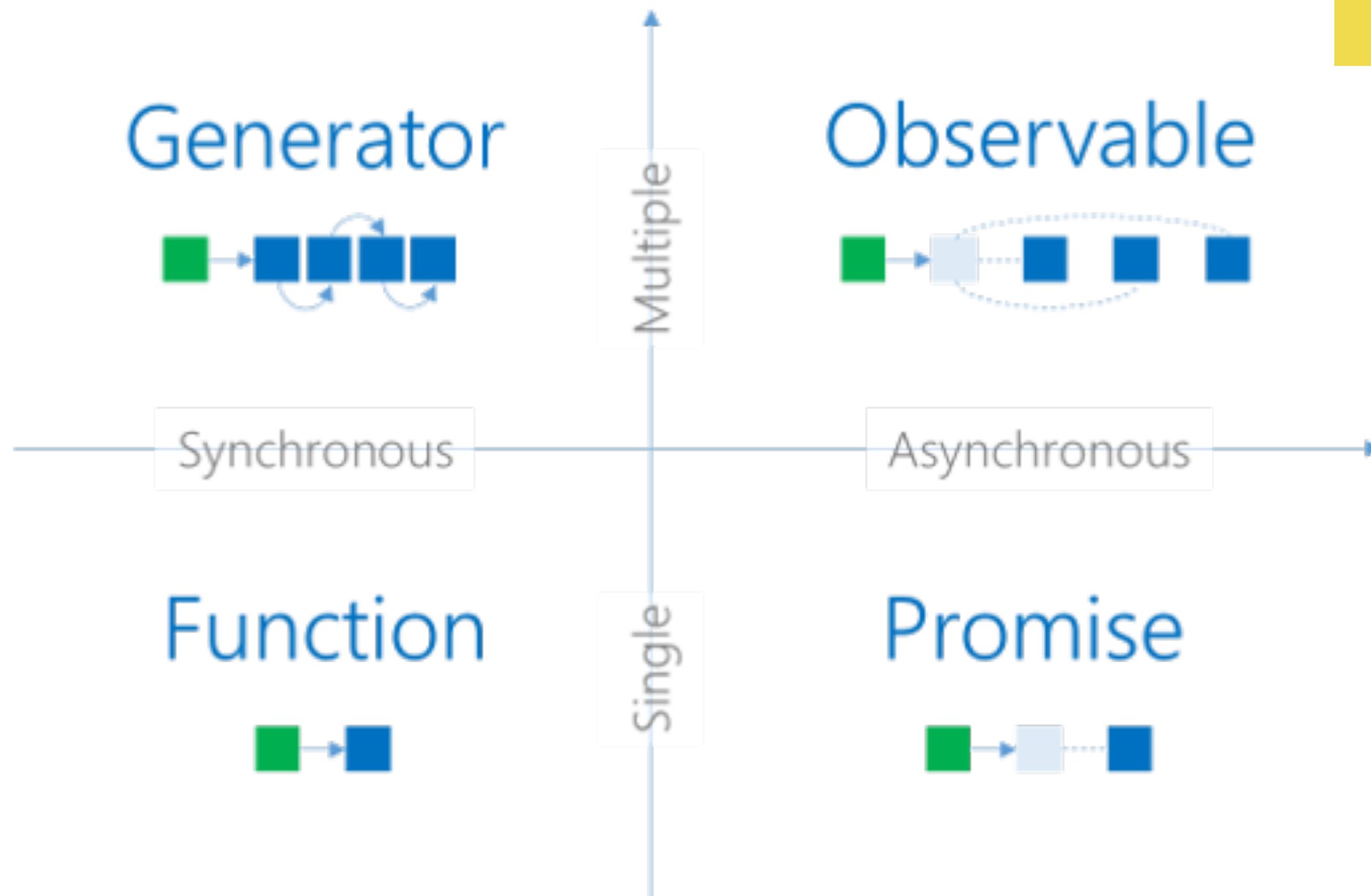


Imagen: <https://goo.gl/4xeo9L>



¡Gracias!

