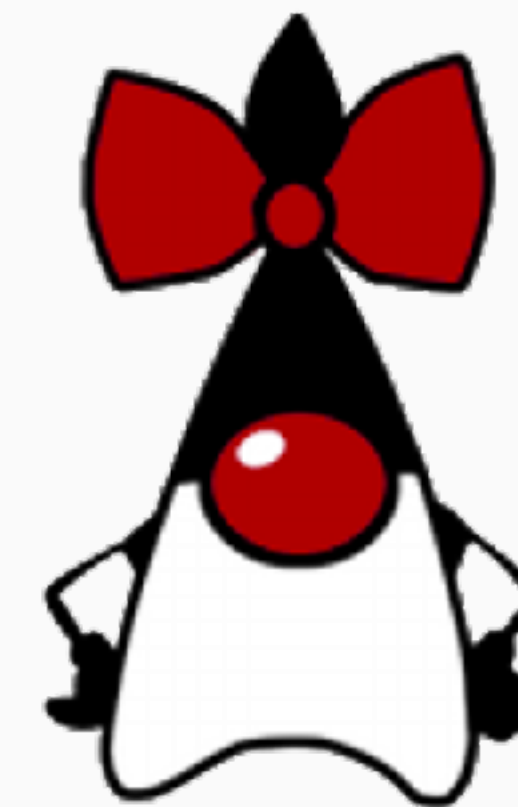






Mercedes Wyss
@itrjwyss





Angularjs

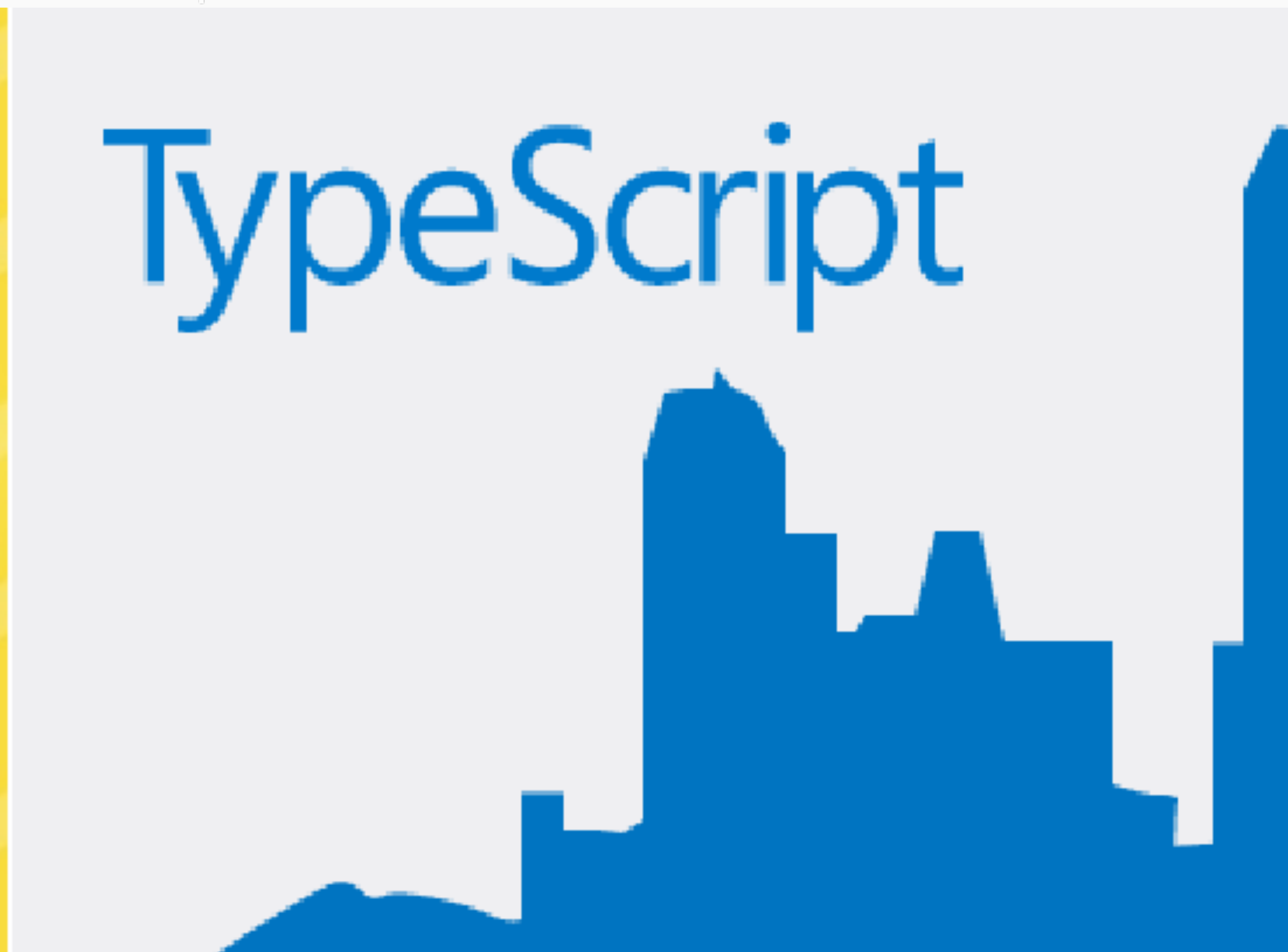
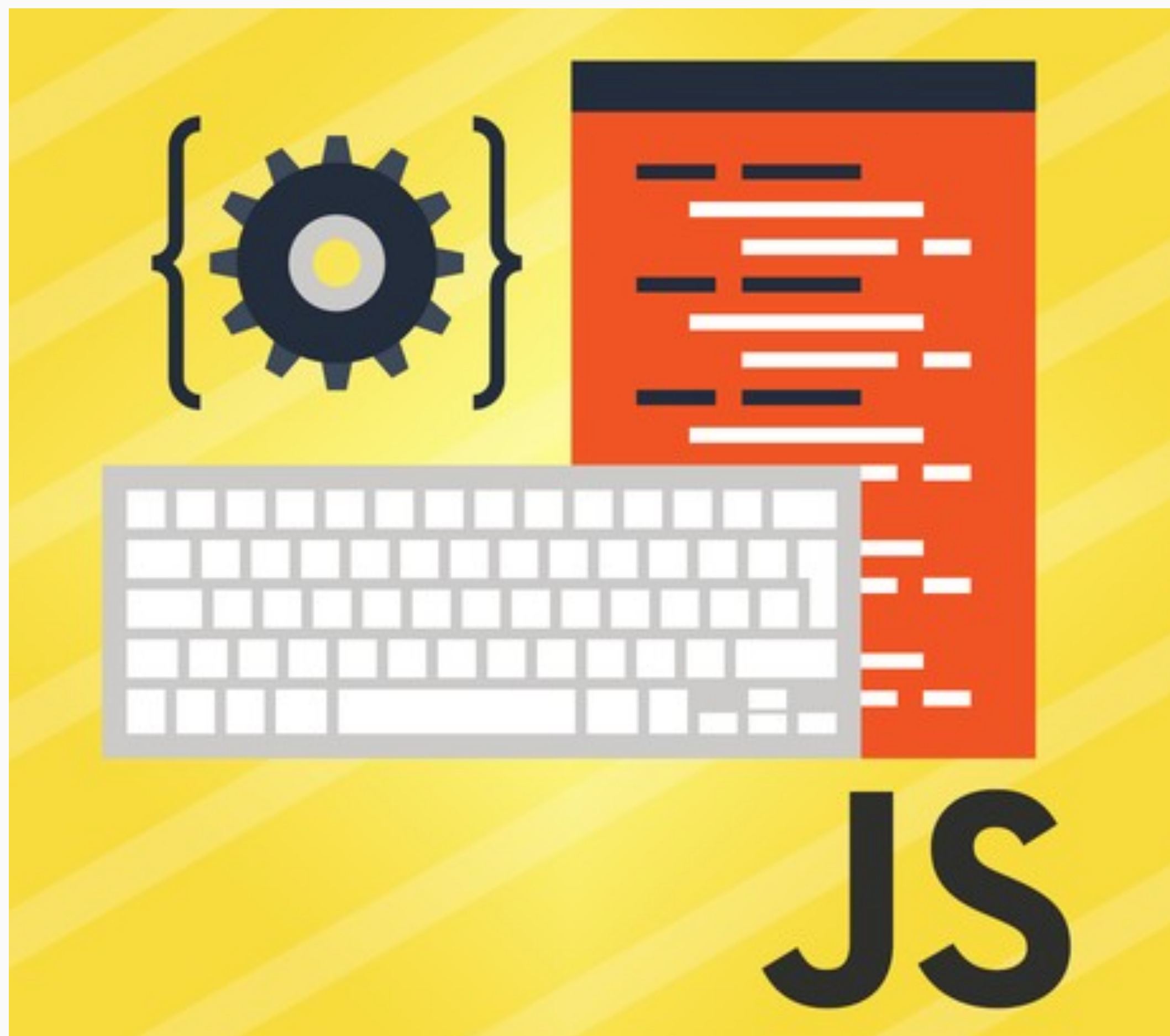
vs



Angular 2



Auth0



Diferencias Principales

AngularJS	Angular
JavaScript	TypeScript
JS Library	Framework
Concepto de programación modular	Programación basada en componentes
Utiliza controlador y \$Scope	Utiliza components y directivas
Uso fácil, solo requiere agregar la librería	Necesita ser configurado, depende de otras librerías y debe configurarse un ambiente de desarrollo

SEMMVER

semantic versioning



Semantic Versioning

MAJOR . MINOR . PATCH



Semantic Versioning



5.2.1 (2018-01-17)

Bug Fixes

- **animations:** fix increment/decrement aliases example ([#18323](#)) ([48c1898](#))
- **benchpress:** should still support selenium_webdriver < 3.6.0 ([#21477](#)) ([3c6a506](#))
- **common:** set correct timezone for ISO8601 dates in Safari ([#21506](#)) ([8e9cd57](#)), closes [#21491](#)
- **compiler:** cache external reference resolution ([#21359](#)) ([c32e833](#))
- **compiler:** make `.ngsummary.json` files idempotent ([#21448](#)) ([a931a41](#))
- **core:** fix chained http call ([#20924](#)) ([54e7576](#)), closes [#20921](#)
- **language-service:** Clear caches when program changes ([#21337](#)) ([cc9419d](#)), closes [#19405](#)
- **service-worker:** properly handle invalid hashes in all scenarios ([#21288](#)) ([51eb3d4](#))

Features

- **core:** add binding name to content changed error ([#20352](#)) ([4556532](#))
- **forms:** handle string with and without line boundary on pattern validator ([#19256](#)) ([75f8522](#))

6.0.0-beta.0 (2018-01-17)

Bug Fixes

- **animations:** fix increment/decrement aliases example (#18323) (d2aa8ac)
- **benchpress:** should still support selenium_webdriver < 3.6.0 (#21477) (9b84a32)
- **common:** set correct timezone for ISO8601 dates in Safari (#21506) (05208b8), closes #21491
- **compiler:** cache external reference resolution (#21359) (e3e2fc0)
- **compiler:** make `.ngsummary.json` files idempotent (#21448) (e64b1e9)
- **core:** fix chained http call (#20924) (7e3f9a4), closes #20921
- **ivy:** Add workaround for AJD in google3 (#21488) (6af3672)
- **language-service:** Clear caches when program changes (#21337) (43e1520), closes #19405
- **service-worker:** properly handle invalid hashes in all scenarios (#21288) (3951098)

Features

- **bazel:** allow `ng_module` rules to control whether type checking is enabled (#21460) (cffa0fe)
- **core:** add binding name to content changed error (#20352) (d3bf54b)
- **forms:** handle string with and without line boundary on pattern validator (#19256) (54bf179)

Performance Improvements

- **ivy:** add missing dom element in render3_function tree benchmark (#21476) (9b5a485)

Web Components



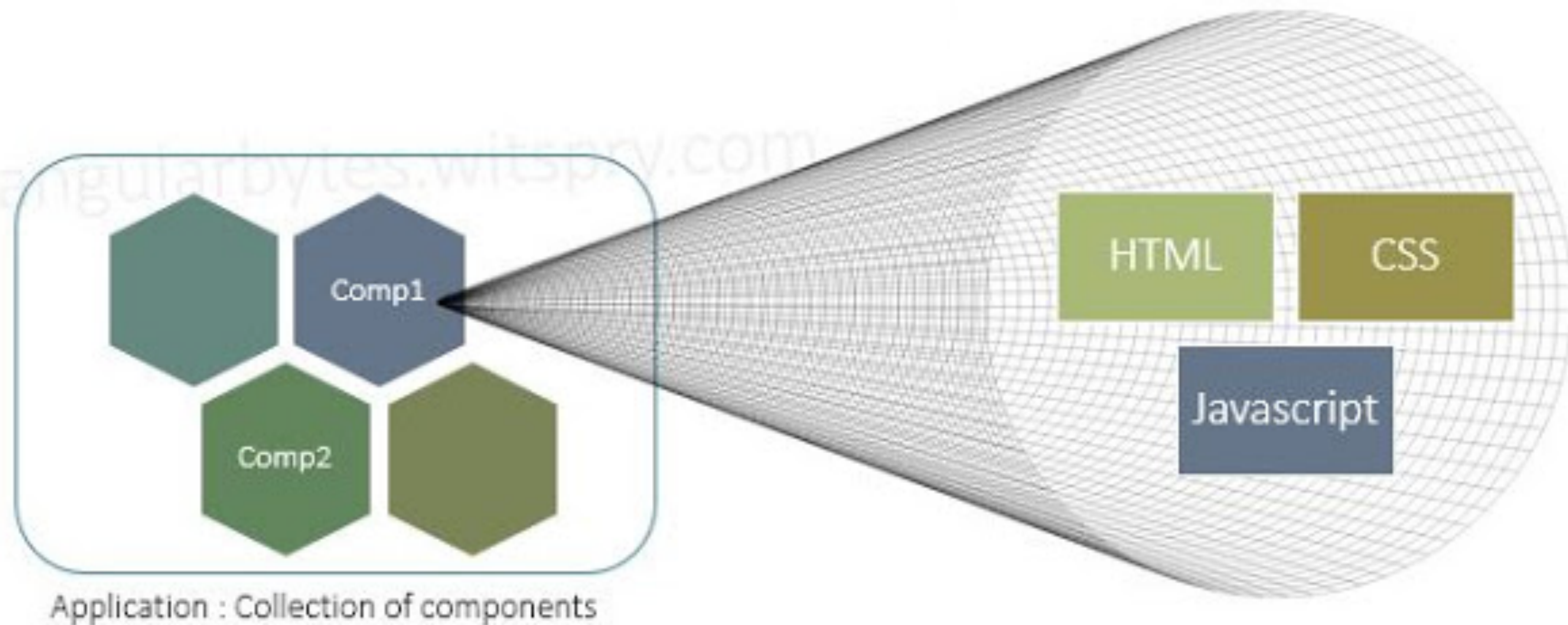
Especificaciones Web Components

- Custom Elements
- Shadow DOM
- HTML imports
- HTML Template

Desarrollo Modular Orientado a Objetos



Desarrollo Web Components



Web Components

What is this even

This is a Polymer element that you can drop on any web page and make it rain emoji. Because, seriously, why wouldn't you?

Make it rain! ☁️ with Twitter emoji

MADE WITH ❤️ BY MONICA. FIND THIS ON GITHUB

```
<link rel="import" href="../emoji-rain/emoji-rain.html">
```

```
...
```

```
<emoji-rain active></emoji-rain>
```



Web Components

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

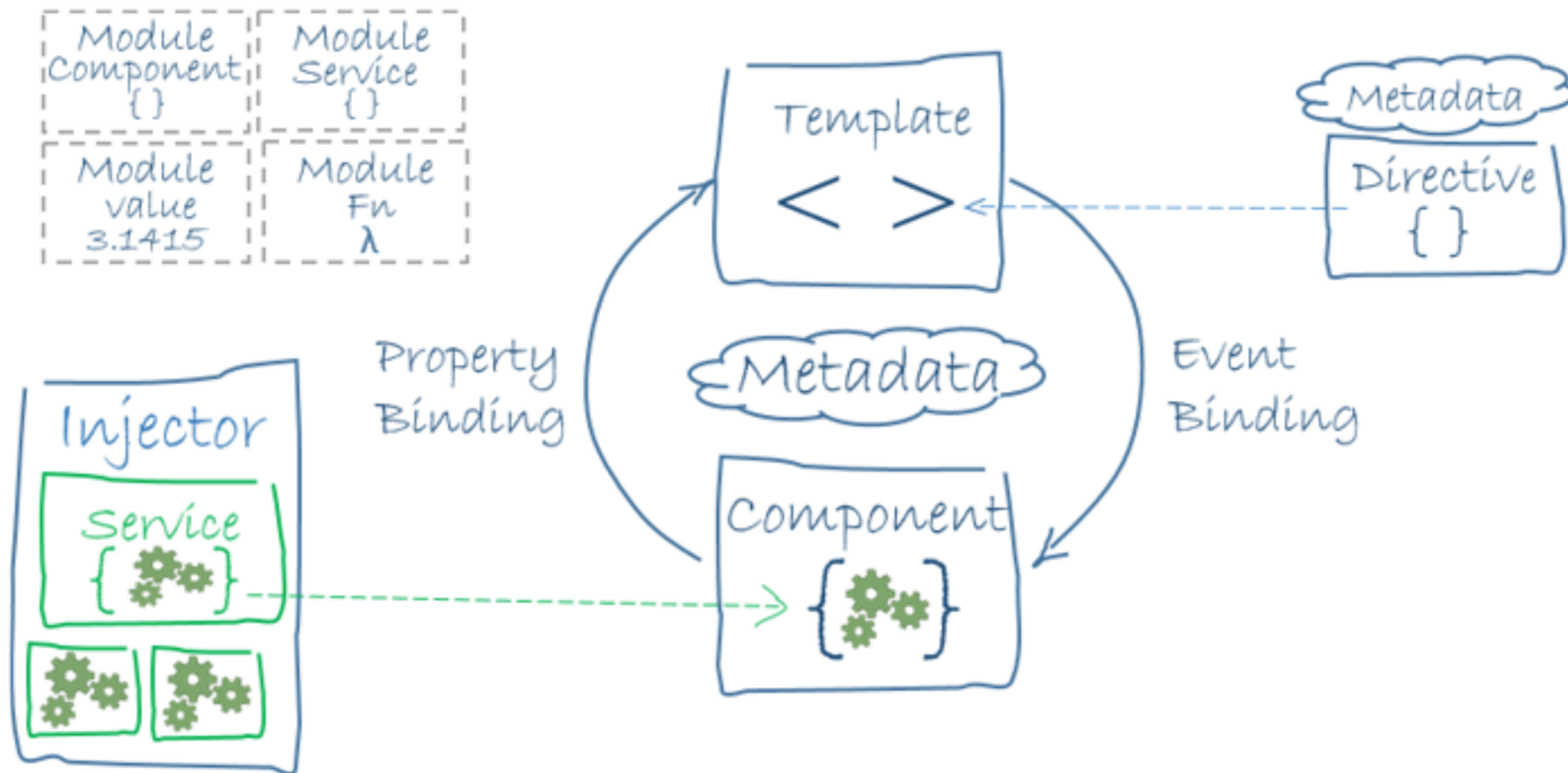
export class Player {
  id: number;
  name: string;
}

@Component({
  selector: 'my-app',
  template: `
    <h1>{{title}}</h1>
    <h2>{{player.name}} details!</h2>
    <div><label>id: </label>{{player.id}}</div>
    <div>
      <label>name: </label>
      <input [(ngModel)]="player.name" placeholder="name">
    </div>
  `,
})
export class AppComponent {
  title = 'Chess Tournament';
  player: Player = {
    id: 1,
    name: 'Jeff'
  };
}
```

Arquitectura de Angular



Arquitectura Angular



Modulos

Module
Component
{ }

Modulos

- **NgModule**
- Declaraciones
- Exports
- Imports
- Proveedores
- bootstrap

Modulos

- Metadata
- Provee servicios
- Importar otros módulos
- Definir componentes públicos o privados

Module Root

src/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 { }
```



Auth0

Importando Modulo Root

src/main.ts

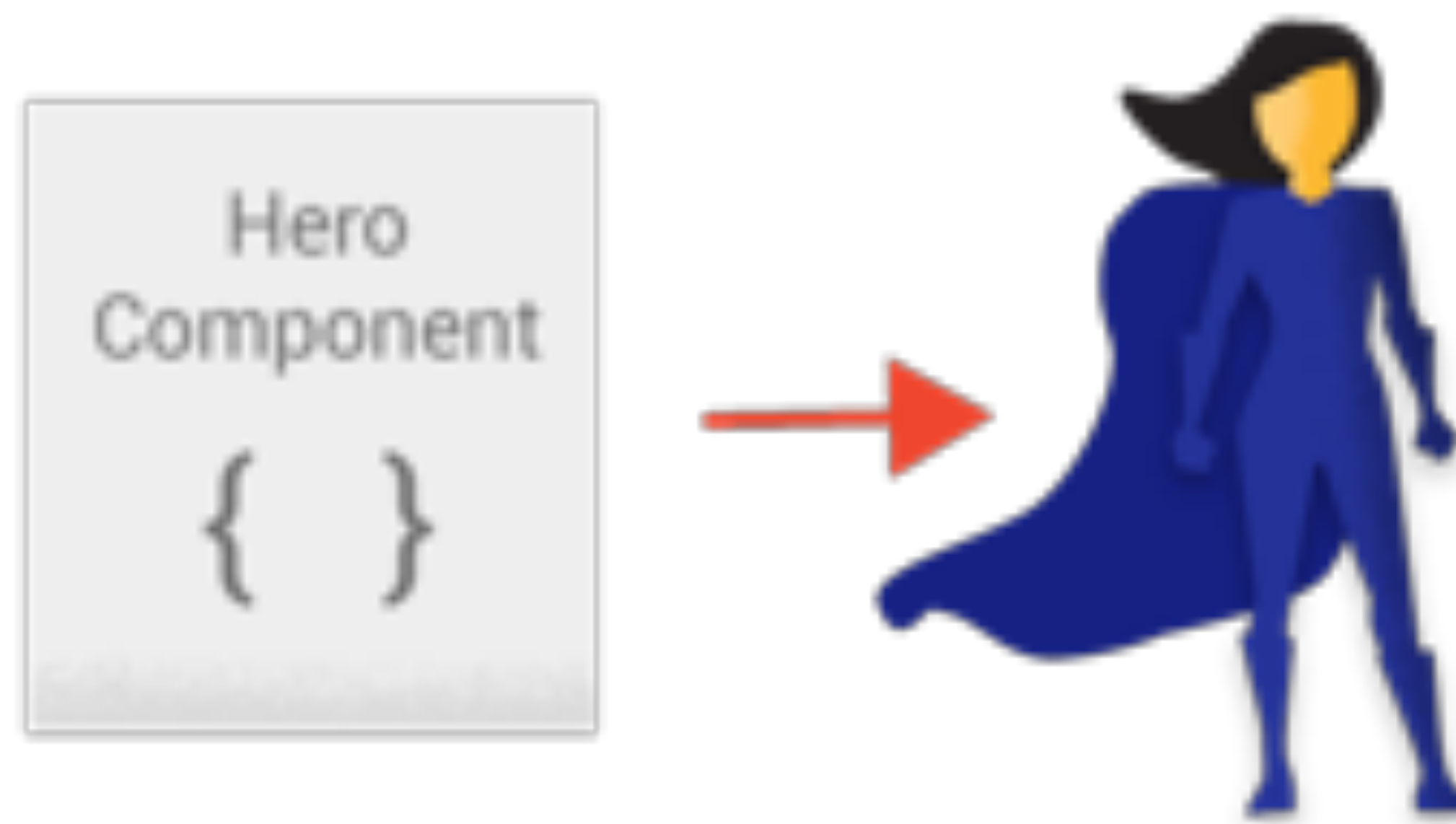
```
import { enableProdMode } from '@angular/core';
import { platformBrowserDynamic } from '@angular/platform-browser-dynamic';

import { AppModule } from './app/app.module';
import { environment } from './environments/environment';

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

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


Componentes



Componentes

Controlar un pedazo de nuestras vistas

- **Templates**

Componentes

src/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; }  
}
```



Auth0

Templates



Templates

src/app/hero-list.component.html

```
<h2>Hero List</h2>

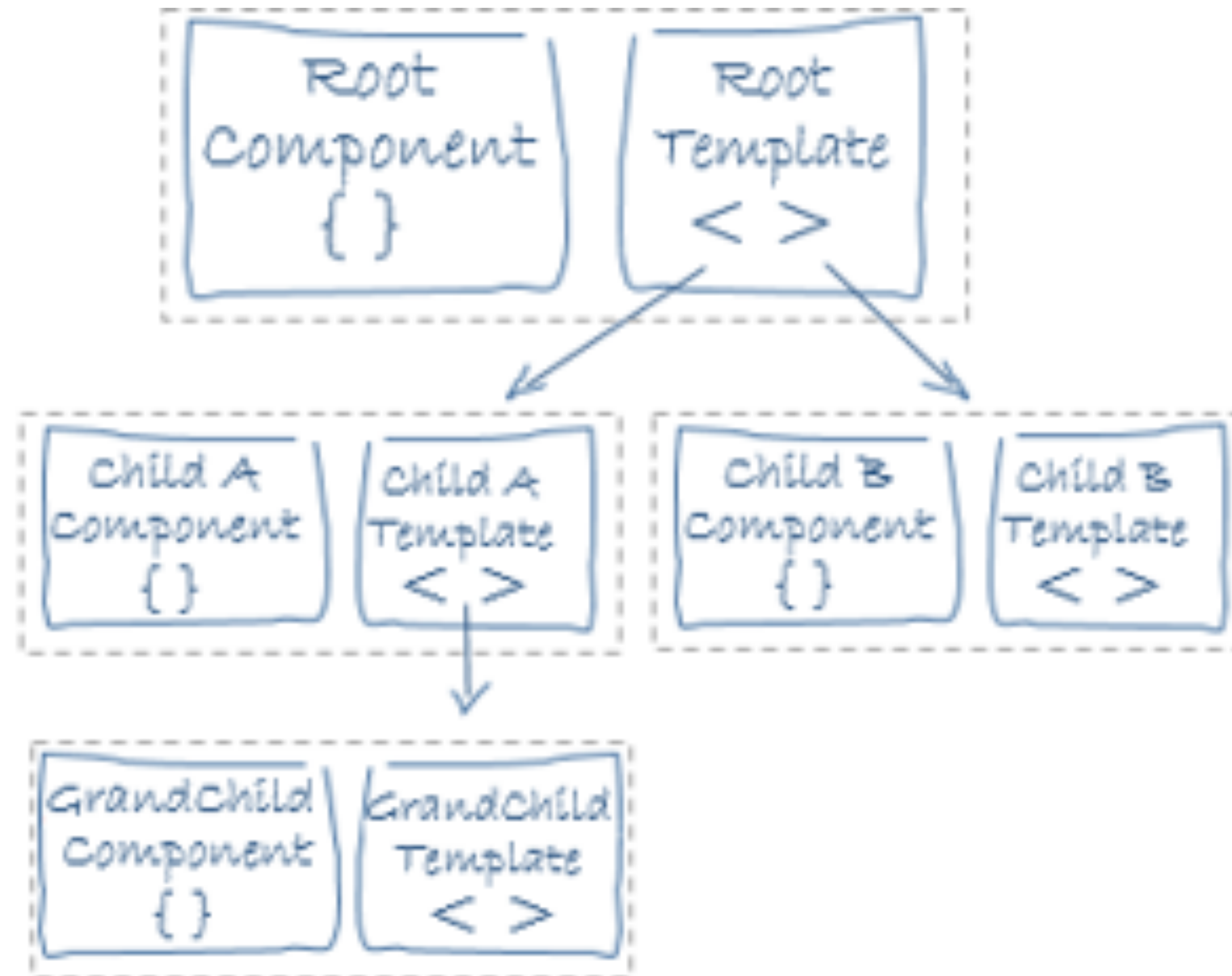
<p><i>Pick a hero from the list</i></p>
<ul>
  <li *ngFor="let hero of heroes" (click)="selectHero(hero)">
    {{hero.name}}
  </li>
</ul>

<app-hero-detail *ngIf="selectedHero" [hero]="selectedHero"></app-hero-detail>
```



Auth0

Componentes y Templates



Metadata

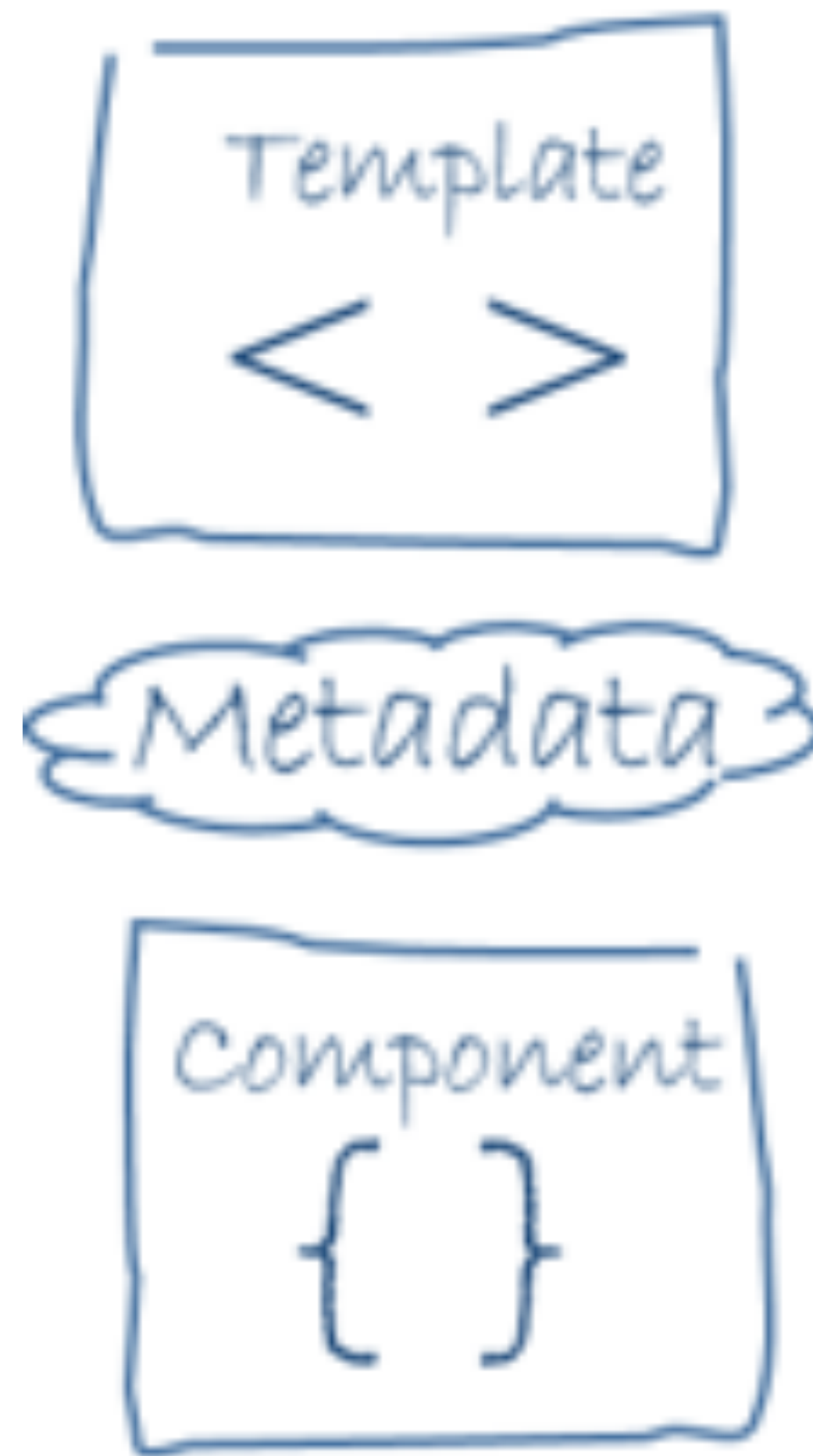
Metadata

Metadata

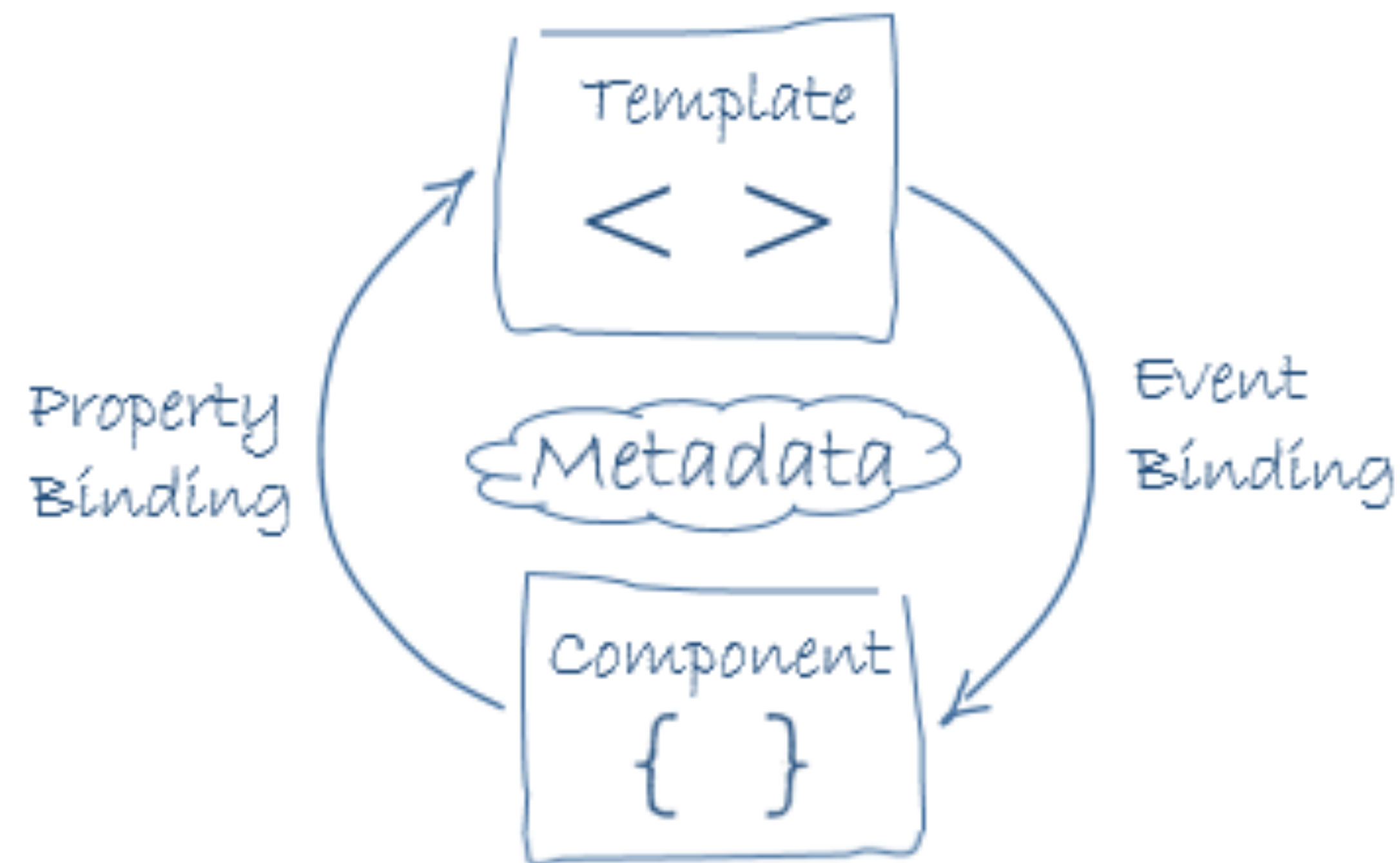
src/app/hero-list.component.ts (metadata)

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

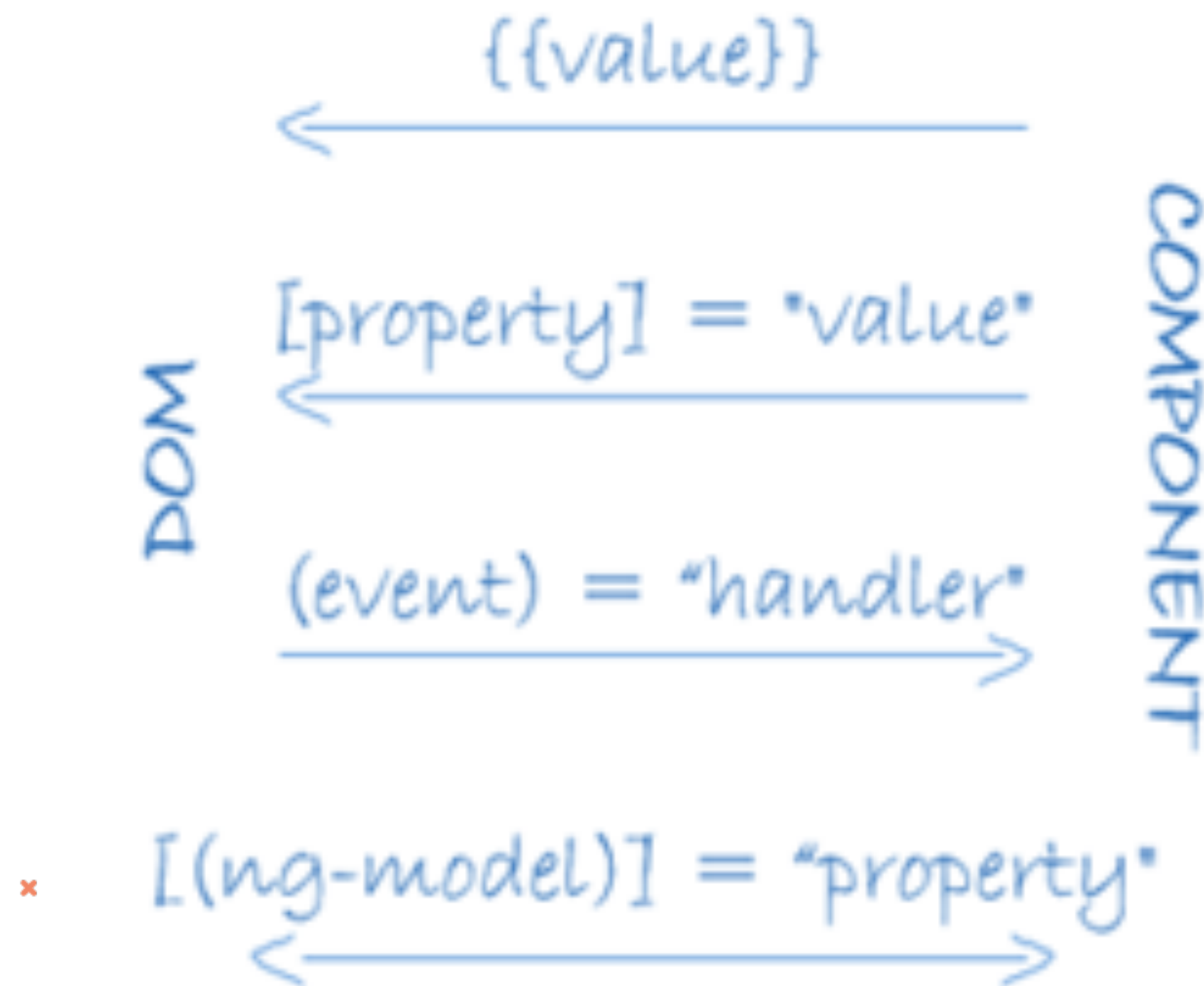
Metadata



Intercambio de Información



Intercambio de Información

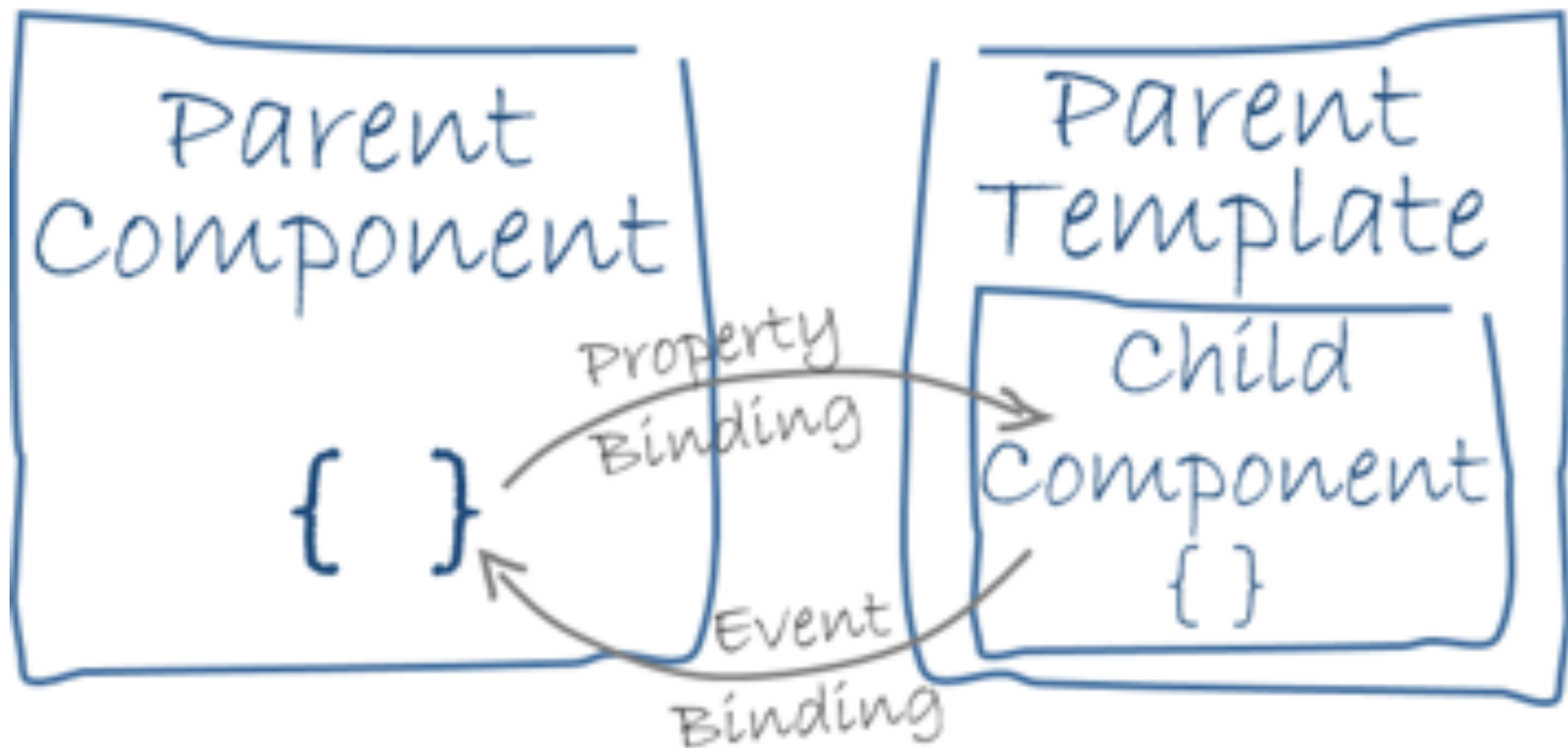


Intercambio de Información

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

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

Intercambio de Información



Auth0

Servicios



Servicios

- Un servicio es típicamente una clase con un propósito estrecho y bien definido.
- Los componentes son grandes consumidores de servicios.

Servicios

src/app/logger.service.ts (class)

```
export class Logger {  
  log(msg: any) { console.log(msg); }  
  error(msg: any) { console.error(msg); }  
  warn(msg: any) { console.warn(msg); }  
}
```

Servicios

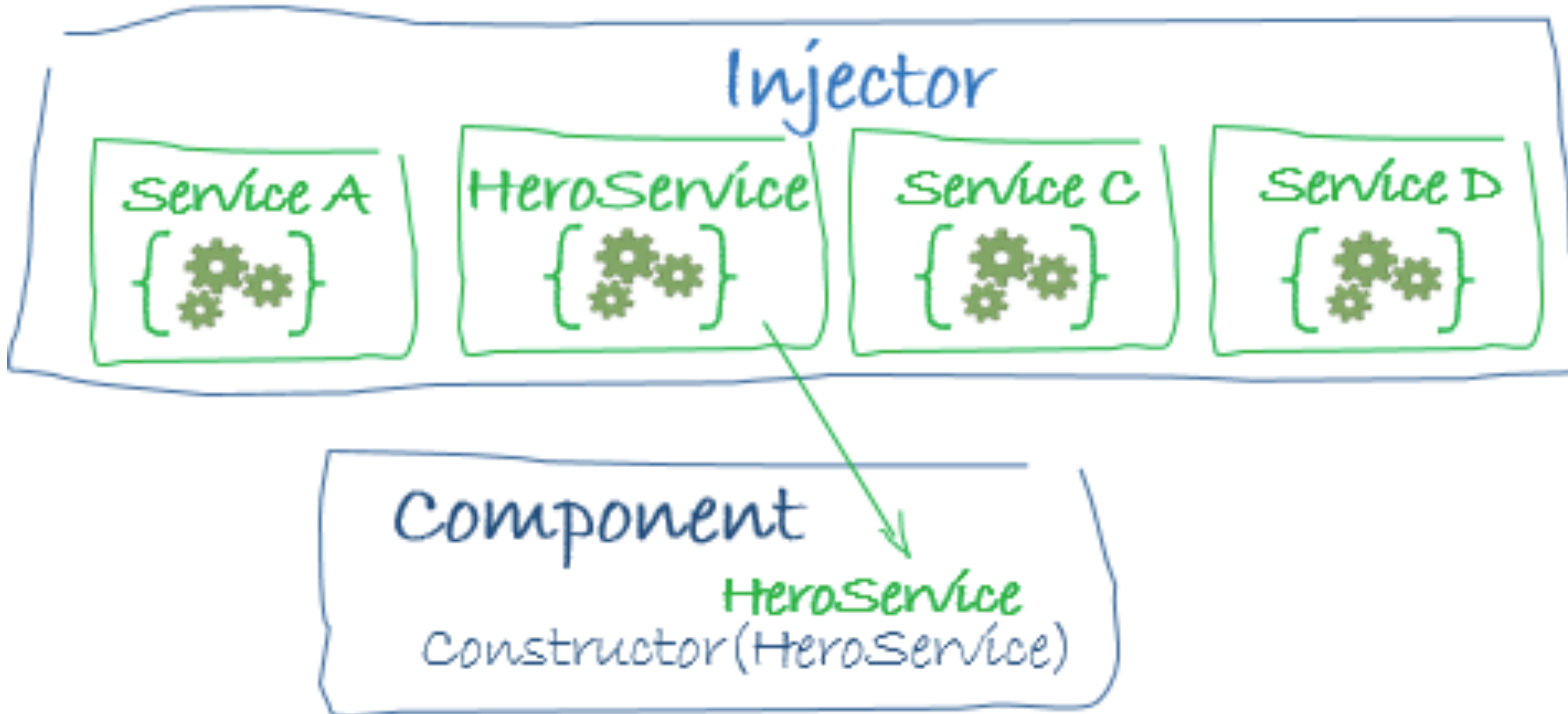
src/app/hero.service.ts (class)

```
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;  
  }  
}
```


Inyección de Dependencias

Component *Service*
{Constructor(service)}

Inyección de Dependencias



Inyección de Dependencias

src/app/app.module.ts (module providers)

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

Inyección de Dependencias

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

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


**JET
BRAINS**



THE DRIVE
TO DEVELOP

