

Framework JS

joel.cavat@hesge.ch

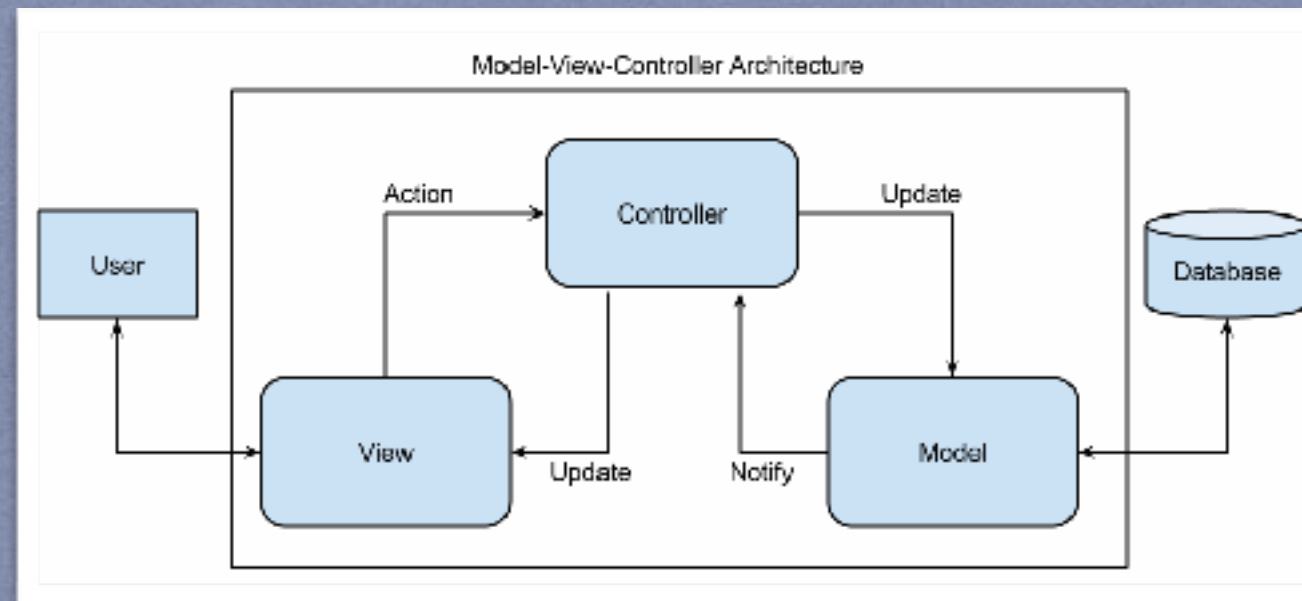
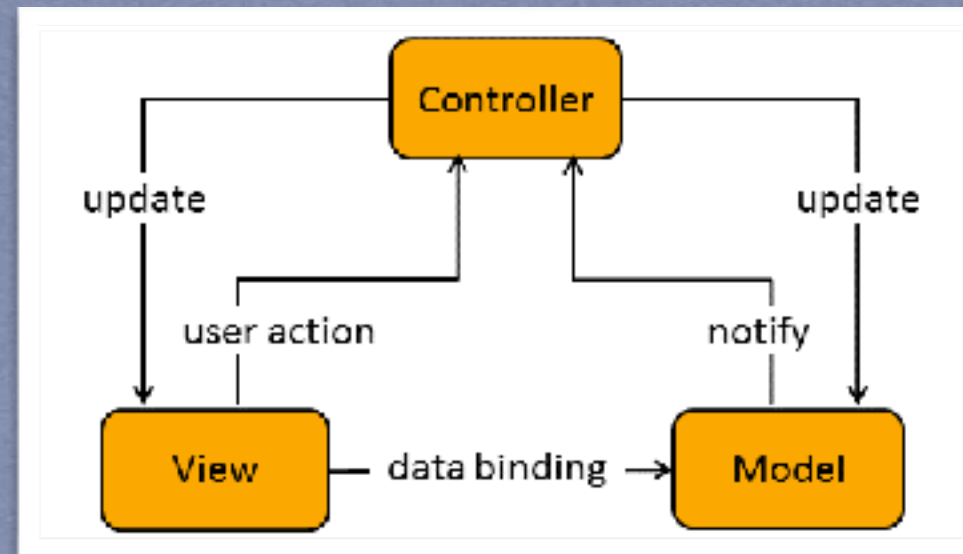
JS frameworks overview



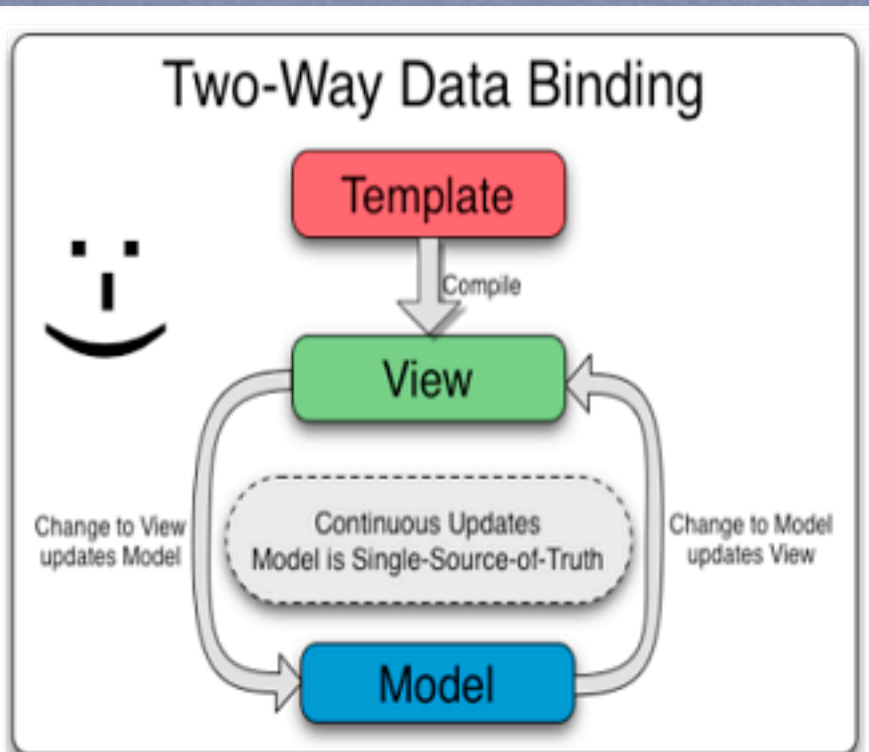
JS frameworks overview

- JS + HTML + CSS
- MV* or MVW(hatever)
- "Frameworks"
- Mobile (Ionic/Cordova) & Desktop (Electron)

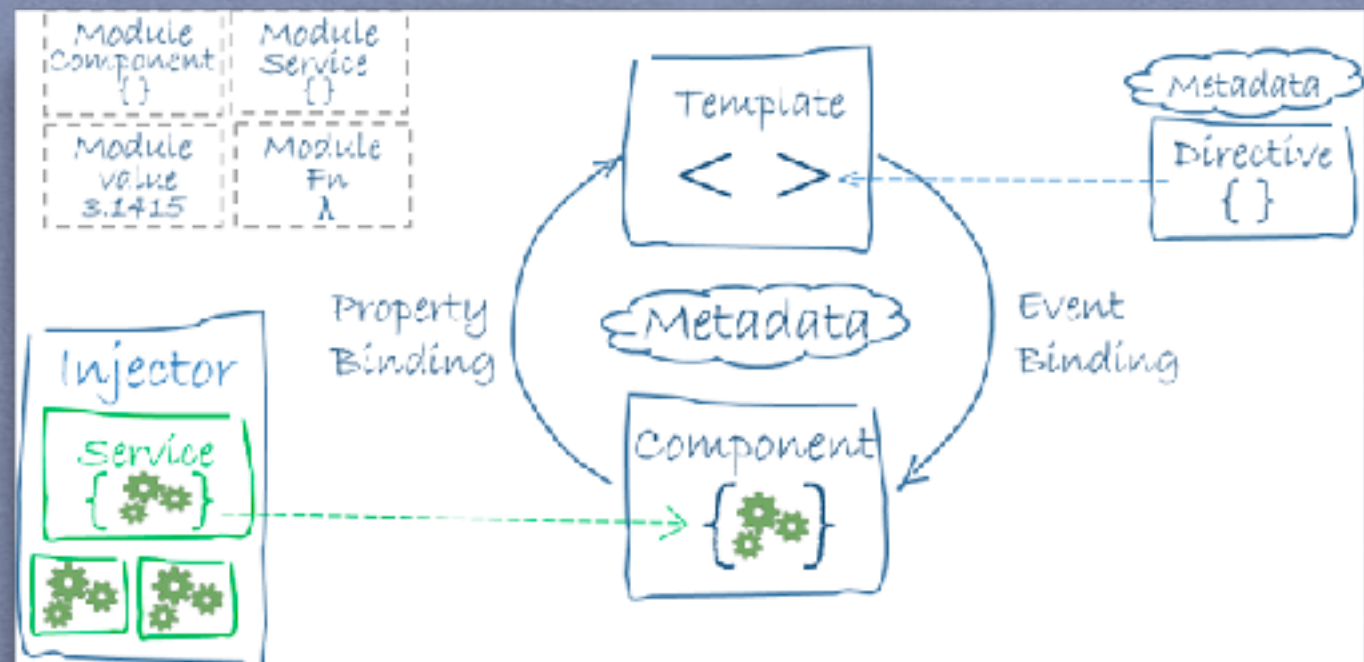
Frameworks MV*



Angular Data Binding

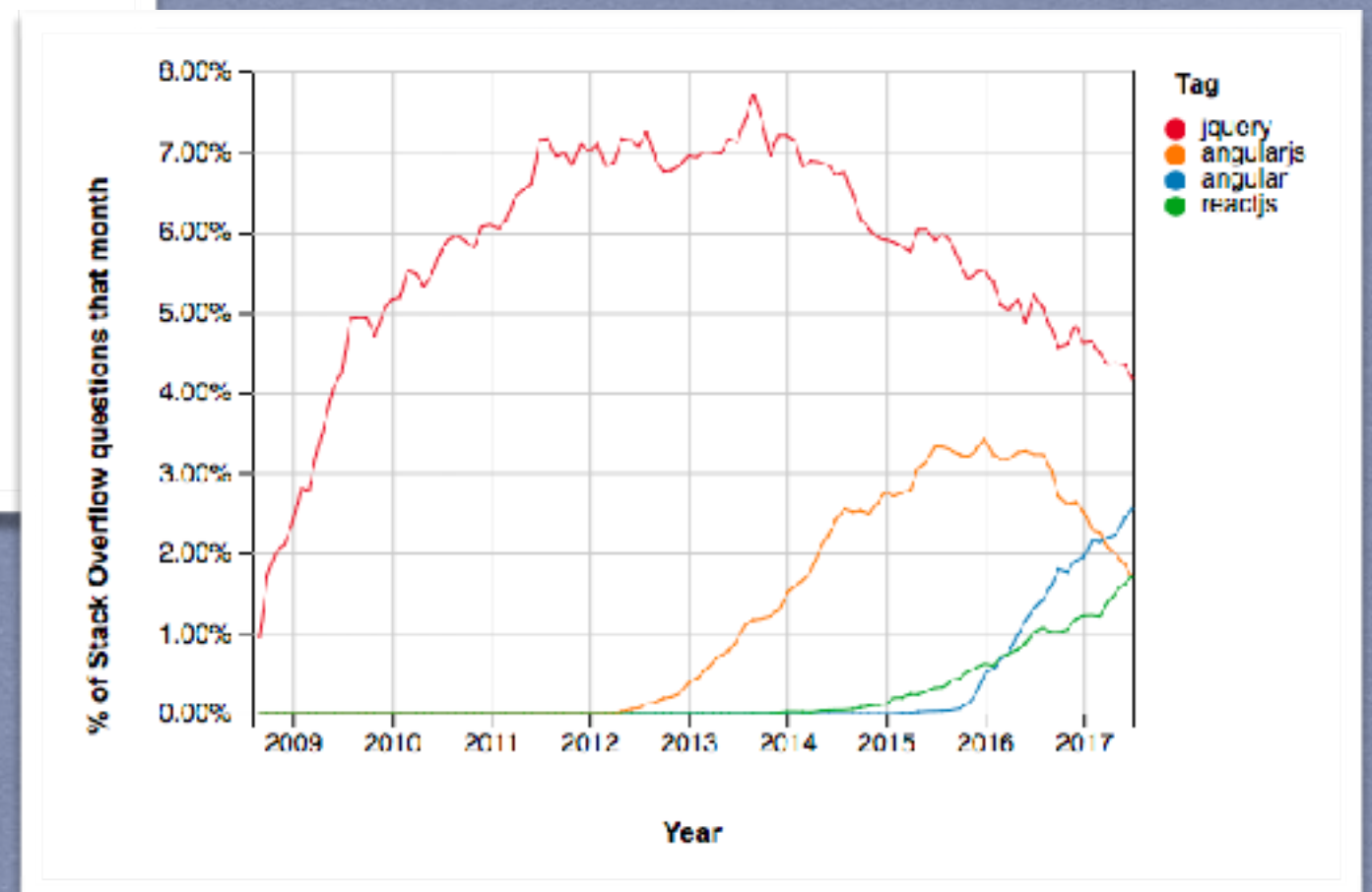
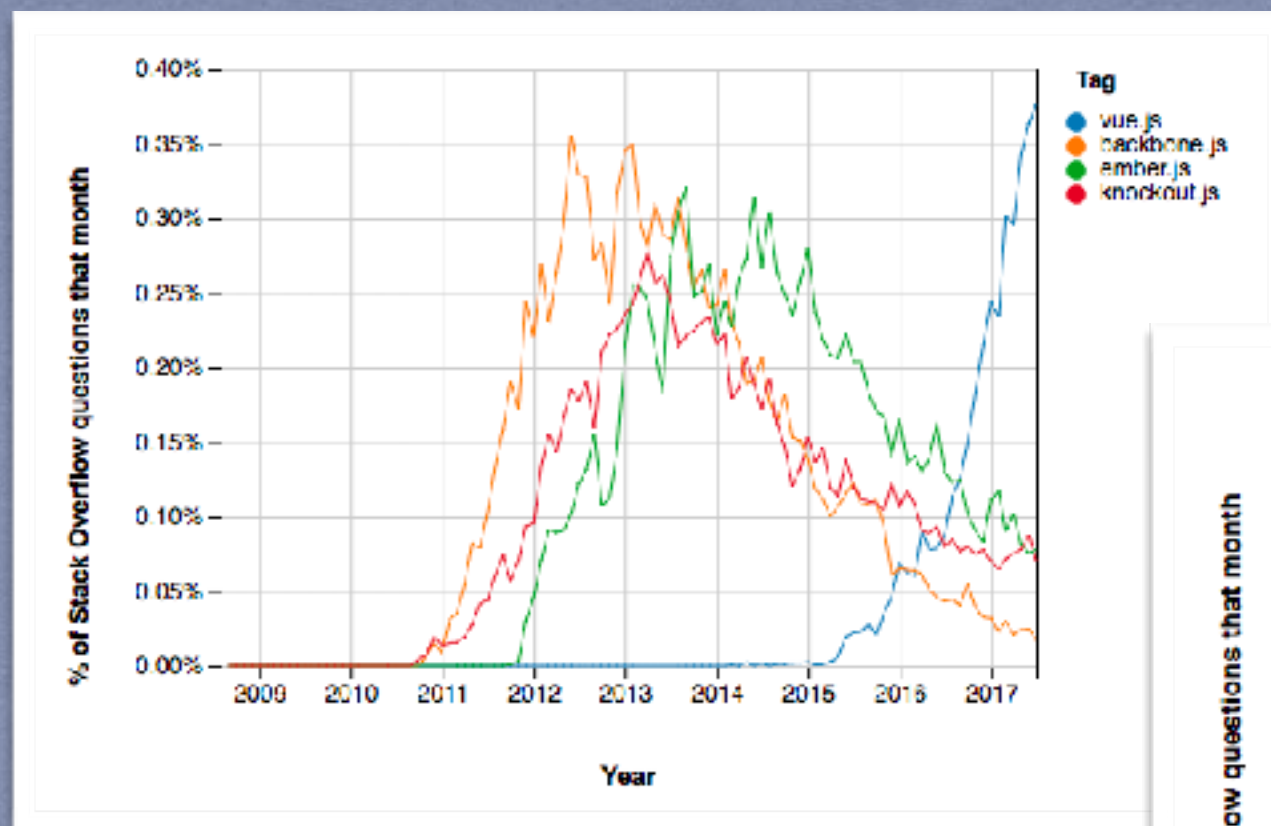


Two-Way Data Binding (AngularJS 1.*)



JS Frameworks' lifecycle

frameworks as a product





Angular

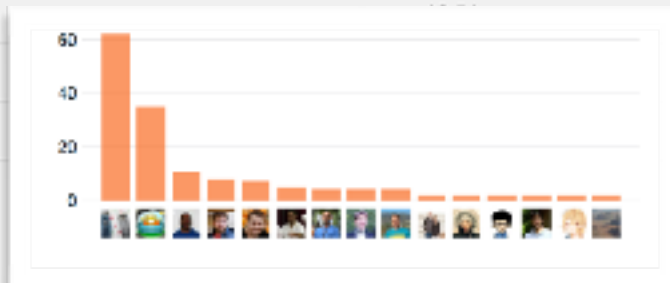
Watch 3,008 Star 35,249 Fork 8,597

Repository Details

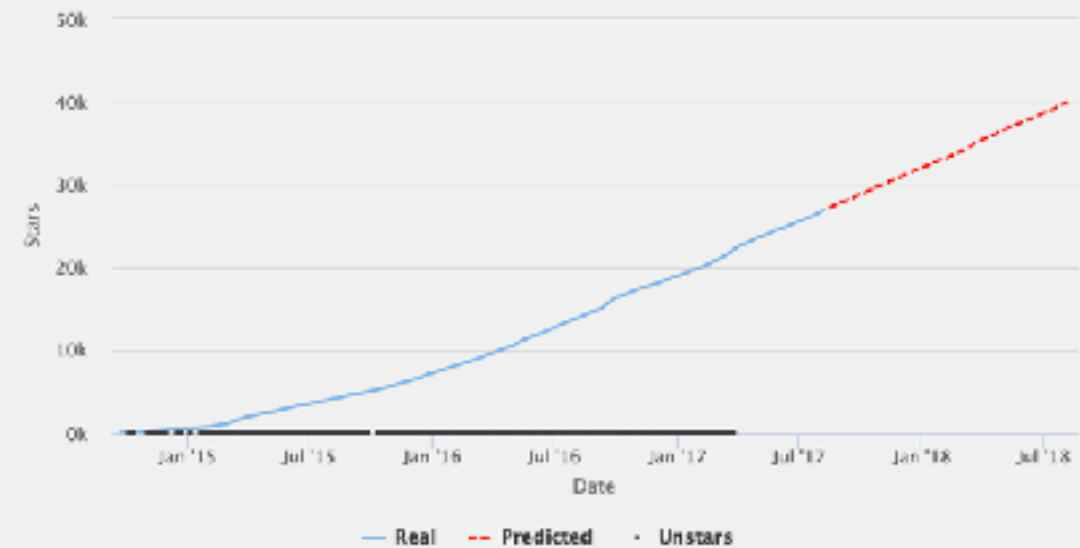
Name	angular
Owner	angular
Language	TypeScript
Domain	Web libraries and frameworks
Growth	Fast
Stars	26,765
Unstars	974
Forks	6,744
Created at	September 18, 2014
Last update	August 10, 2017

Truck Factor = 6 (branch master)

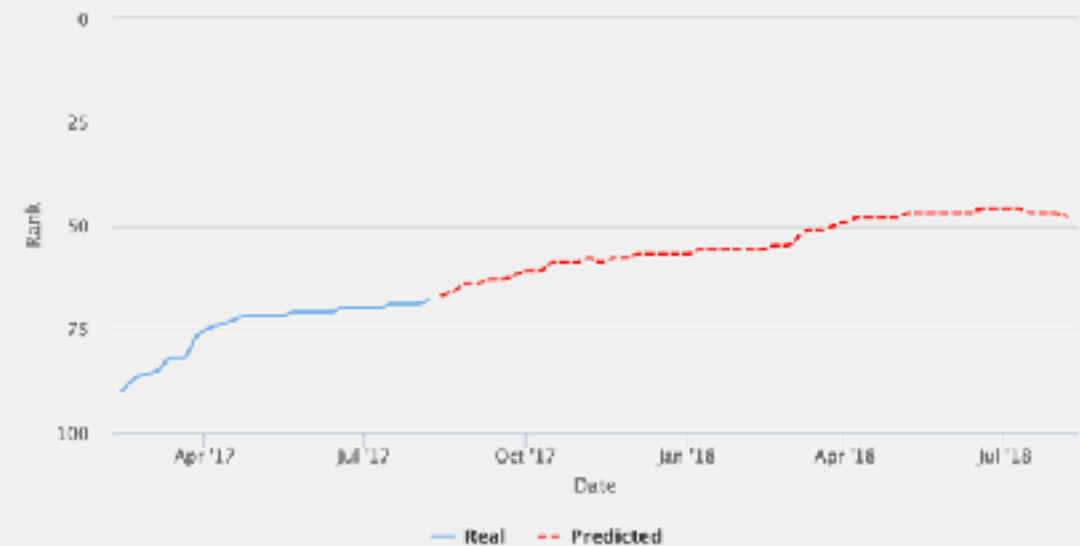
Developer	% Files
Tobias Bosch	26.91
vsavkin	15.88
Victor Berchet	11.11
Misko Hevery	10.11
Igor Minar	9.11
Matias Niemela	8.11



Number of stars



Ranking



Statistiques : <http://gittrends.io/>



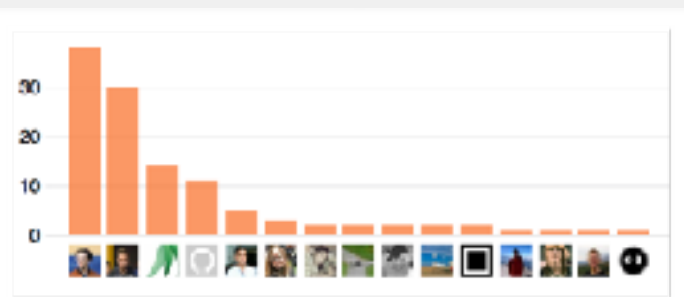
Watch 5,759 Star 93,894 Fork 17,694

Repository Details

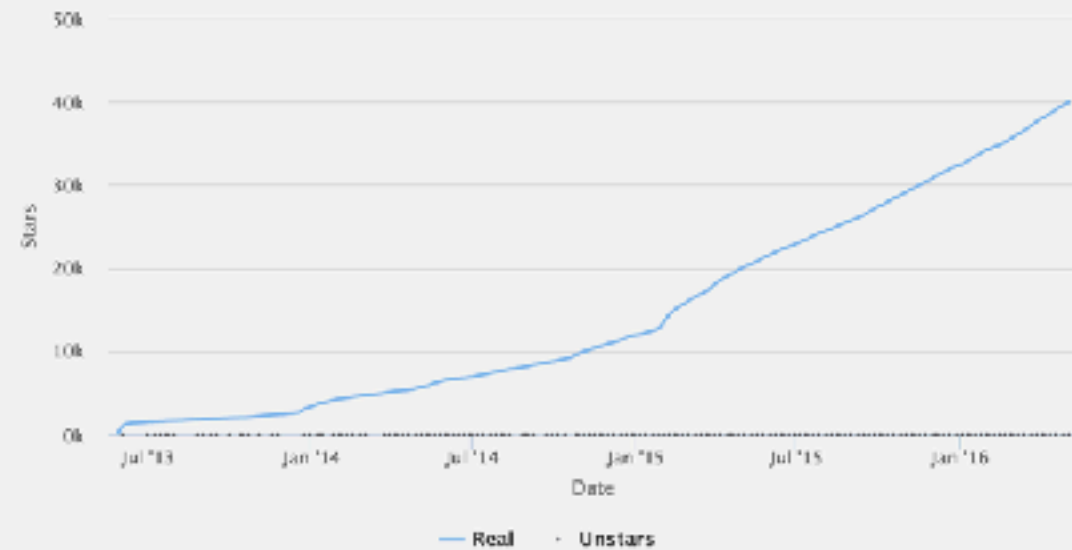
Name	react
Owner	facebook
Language	JavaScript
Domain	Web libraries and frameworks
Stars	73,151
Unstars	946
Forks	13,760
Created at	May 24, 2013
Last update	August 10, 2017

Truck Factor = 3 (branch master)

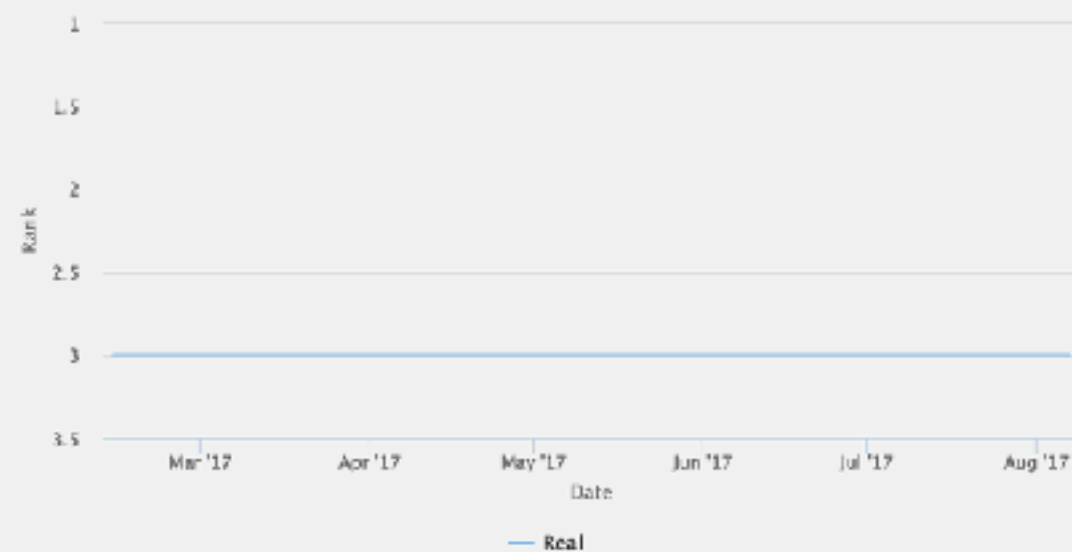
Developer	% Files
Ben Alpert	27.57
Sebastian Markbage	26.64
Paul O'Shannessy	22.9



Number of stars



Ranking



Statistiques : <http://gittrends.io/>



Polymer

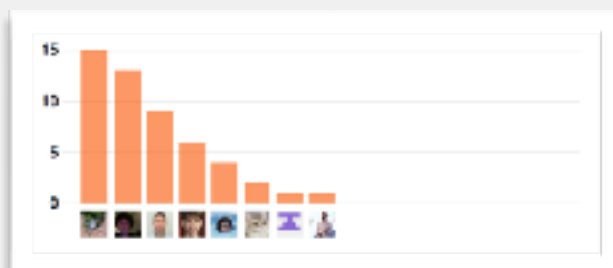
Watch 1,093 Star 19,417 Fork 1,528

Repository Details

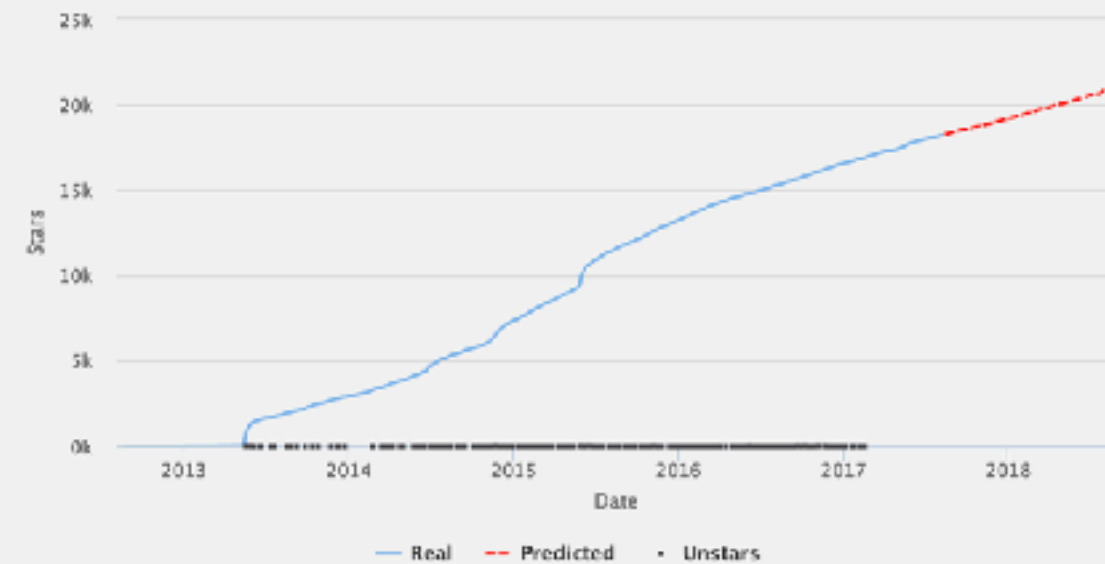
Name	polymer
Owner	Polymer
Language	HTML
Domain	Web libraries and frameworks
Growth	Moderate
Stars	18,079
Unstars	425
Forks	1,801
Created at	August 23, 2012
Last update	August 10, 2017

Track Factor = 2 (branch master)

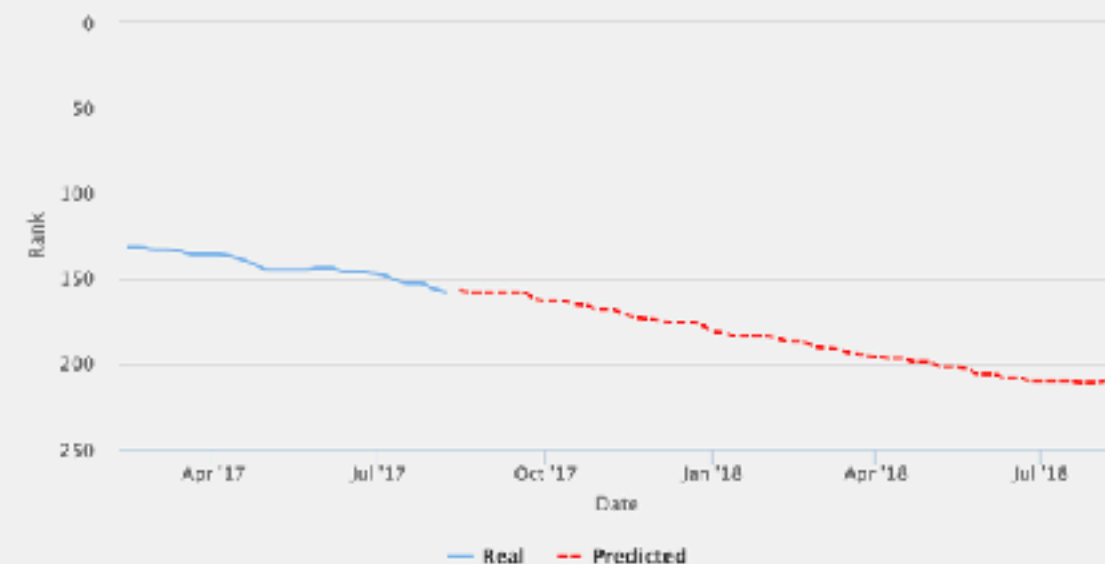
Developer	% Files
Steve Orvell	53.59
Kevin Schaaf	23.21



Number of stars



Ranking



Statistiques : <http://gittrends.io/>



Vue

Watch 4,619

Star 81,389

Fork 13,625

Repository Details

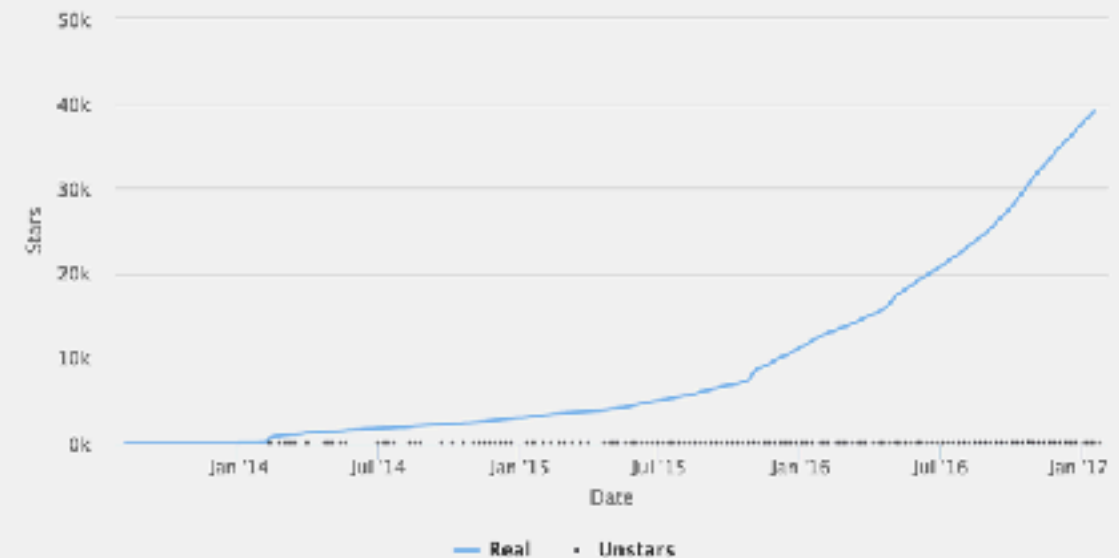
Name	vue
Owner	vuejs
Language	JavaScript
Domain	Web libraries and frameworks
Growth	Fast
Stars	62,971
Unstars	1,478
Forks	8,958
Created at	July 28, 2013
Last update	August 10, 2017

Truck Factor = 1 (branch master)

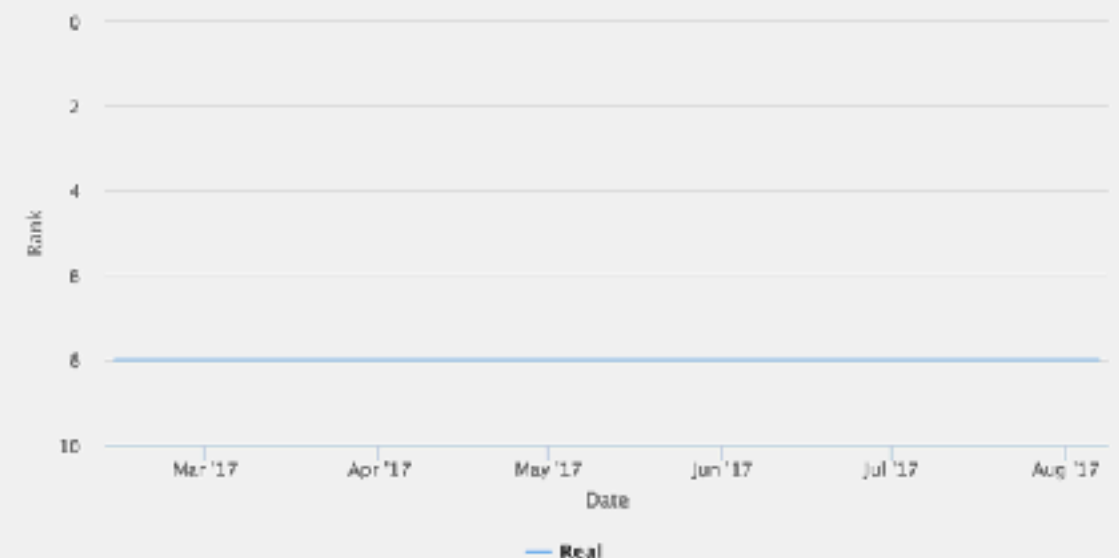
Developer	% Files
Even You	96.82



Number of stars



Ranking



Statistiques : <http://gittrends.io/>

METEOR

Watch 1,802
 Star 38,625
 Fork 4,985

Repository Details

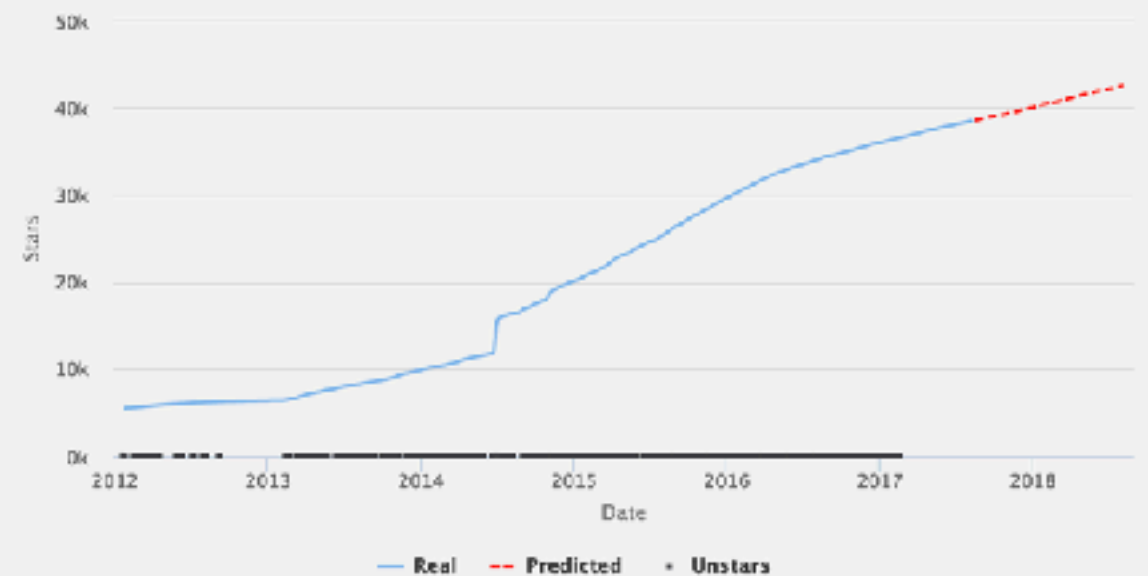
Name	meteor
Owner	meteor
Language	JavaScript
Domain	Web libraries and frameworks
Growth	Moderate
Stars	38,009
Unstars	1,080
Forks	4,922
Created at	January 19, 2012
Last update	August 10, 2017

Track Factor = 4 (branch master)

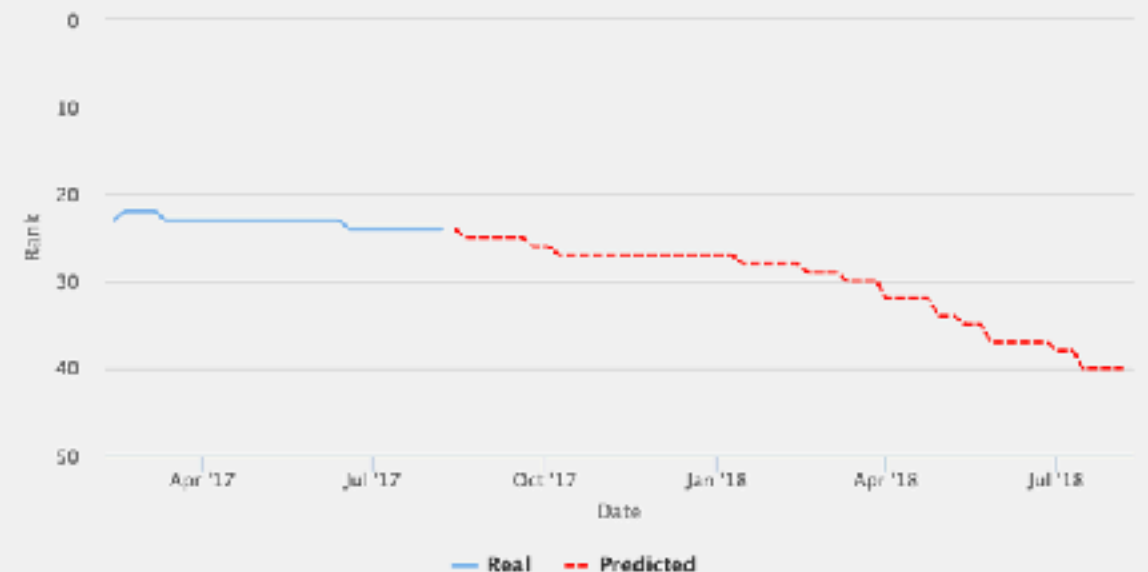
Developer	% Files
Ben Newman	25.85
David Glasser	18.73
Steve Kim	18.3
Avital Oliver	



Number of stars



Ranking



Statistiques : <http://gittrends.io/>



Watch

3,545

★

 Star

48,721

🍴

 Fork

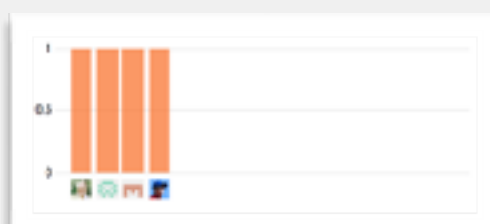
15,398

Repository Details

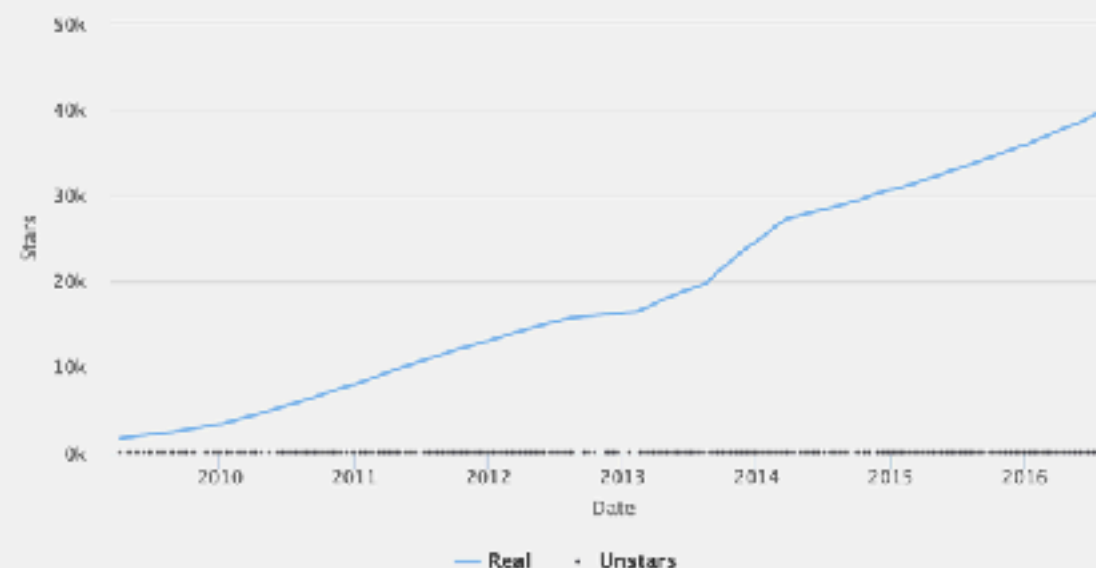
Name	jquery
Owner	jquery
Language	JavaScript
Domain	Web libraries and frameworks
Stars	45,926
Unstars	1,019
Forks	15,162
Created at	April 3, 2009
Last update	August 10, 2017

Trunk Factor = 4 (branch master)

Developer	% Files
Timmy Willison	33.19
Michał Gołębiewski	27.95
jaubourg	18.78
John Resig	16.16



Number of stars



Ranking



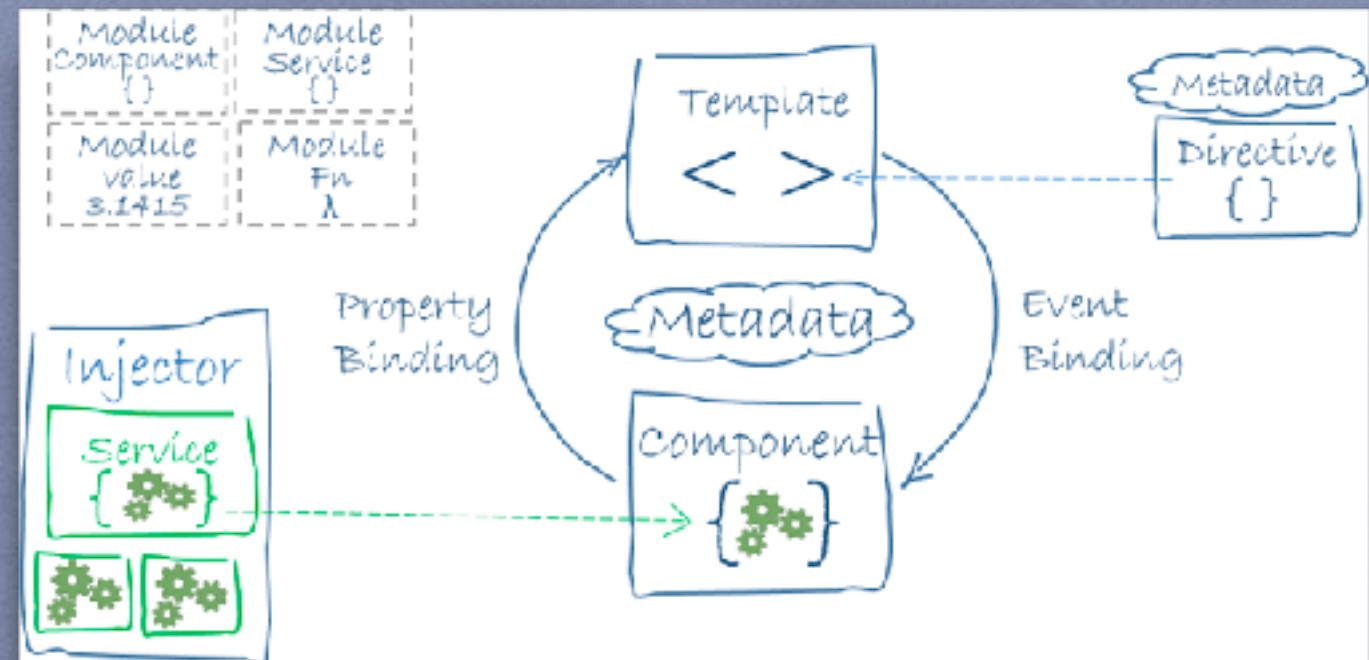
Statistiques : <http://gittrends.io/>



Angular

Architecture overview

- framework for building web application
- module as a set of components and services
- component : data, logic & html template
- dependency injection and services
- one/two ways data-binding
- routing : navigation between views
- lazy loading modules





Angular

Features

- TypeScript (JavaScript superset)
- Documentation and good practice
- CLI
- Performance ++
- Fairly Opinionated +/-
- Learning Curve --



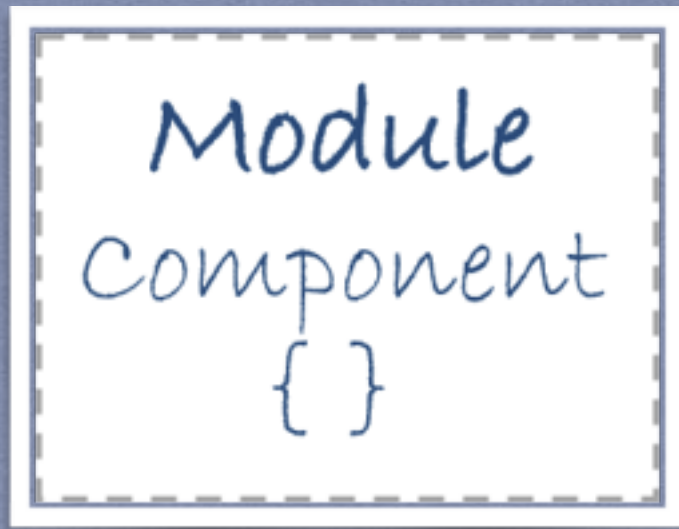
Angular by Example

Summary

- Tutorials
 - ✦ Angular CLI
 - ✦ Module
 - ✦ Component
 - ✦ Directive
 - ✦ Service
 - ✦ Pipe
 - ✦ Router



Introduction to Modules

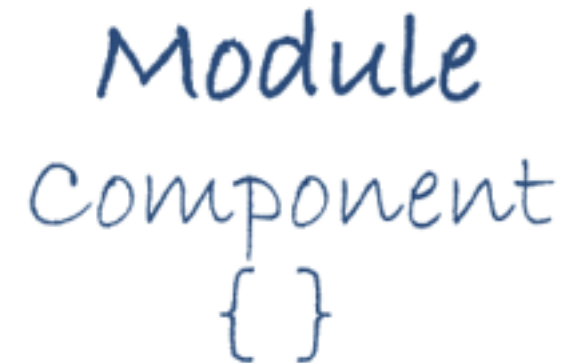


- cohesive block of code representing app domain
- contains components and services defined by a scope
- defined as a class decorated with **@NgModule**



Introduction to Modules

- **AppModule** : root module in `app/app.module.ts`
- **SharedModule** : reusable items (components, directives, pipes, ...)
- **FeatureModule** : all distinct features / domain bounded context
 - define clear boundaries
 - hide its implementation details
 - lazily routed



Module
Component
{ }



App Module

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

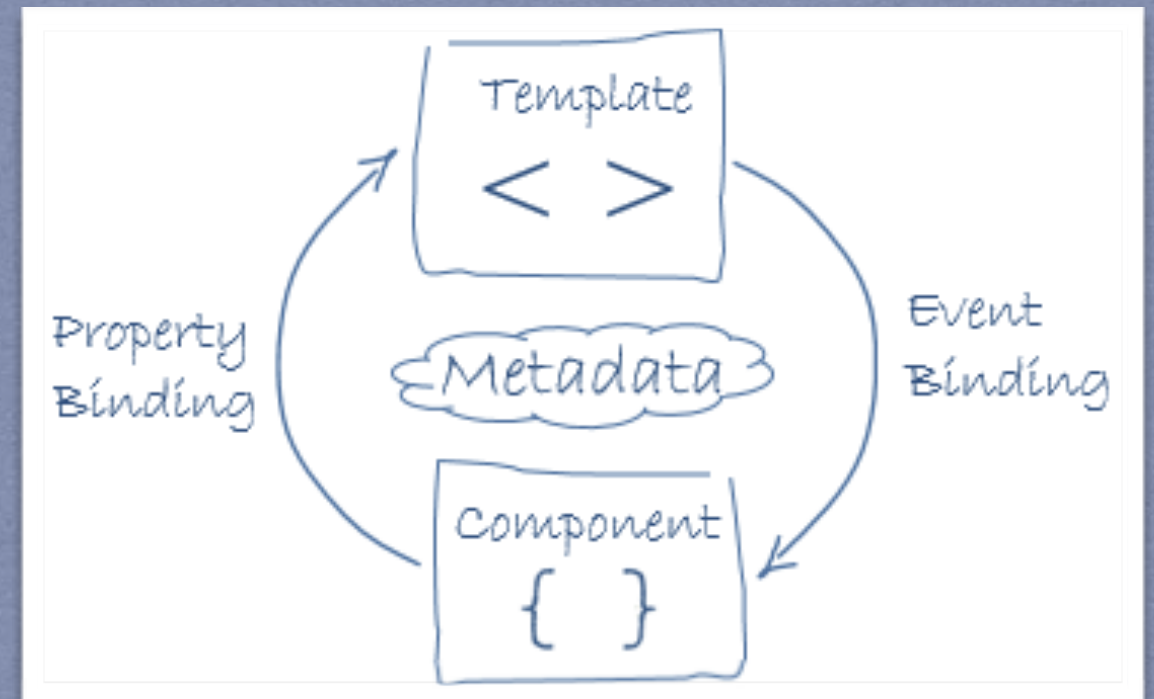
@NgModule({
  imports:      [ BrowserModule ],
  providers:    [ Logger ],
  declarations: [ AppComponent ],
  exports:      [ AppComponent ],
  bootstrap:    [ AppComponent ]
})
export class AppModule { }
```

- **declarations**—The components, *directives*, and *pipes* that belong to this NgModule.
- **exports**—The subset of declarations that should be visible and usable in the *component templates* of other NgModules.
- **imports**—Other modules whose exported classes are needed by component templates declared in *this* NgModule.
- **providers**—Creators of services that this NgModule contributes to the global collection of services; they become accessible in all parts of the app. (You can also specify providers at the component level, which is often preferred.)
- **bootstrap**—The main application view, called the *root component*, which hosts all other app views. Only the *root NgModule* should set this **bootstrap** property.



Introduction to Component

- controls a piece of screen (a view)
- component = data, logic & template
- always one root component
- data-binding defined through a selector
- defined as a class decorated with **@Component**





Introduction to Component

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

```
<app-hero-list></app-hero-list>
```




Introduction to Directive

transforms the DOM according to the given instructions

- structural directives
 - alter layout by adding, removing, and replacing elements in DOM
- attribute directives
 - alter the appearance or behavior

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

```
<input [(ngModel)]="hero.name">
```

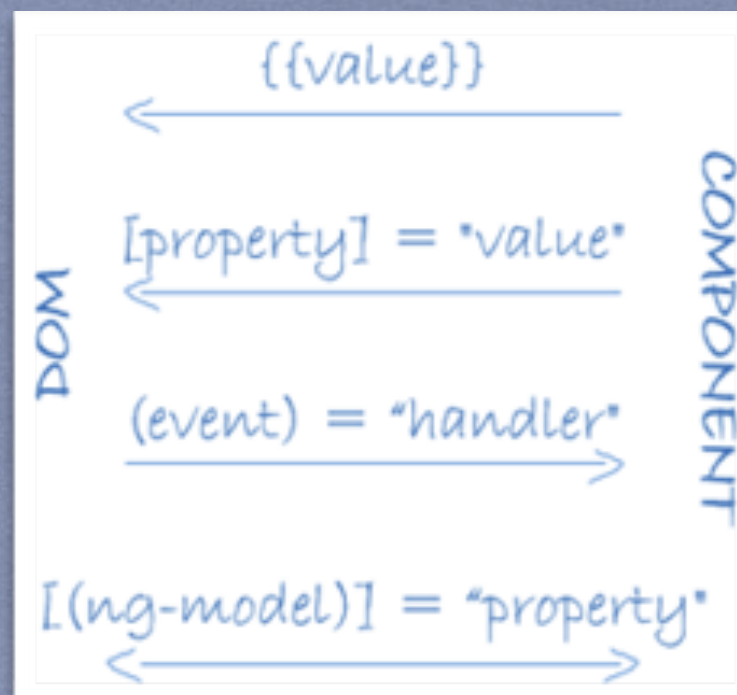



Introduction to Directive

```
<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>
```





Introduction to **Services** and dependency injection

- a service is injectable via the **@Injectable** decorator
- it will be registered and candidate for all uses
- suppliers shouldn't create a service with new
- the framework manage dependencies and lifecycle



Introduction to **Services** and dependency injection

```
class MyObject {  
    DataBase db;  
    constructor() {  
        this.db = new MySQLDataBase();  
    }  
    add(item: Item) {  
        this.db.insert(item);  
    }  
}  
  
MyObject o = new MyObject();
```



```
class MyObject {  
    DataBase db;  
    constructor(DataBase db) {  
        this.db = db;  
    }  
    add(item: Item) {  
        this.db.insert(item);  
    }  
}  
  
MyObject o1 = new MyObject(new FakeDataBase());  
MyObject o2 = new MyObject(new MySQLDataBase());  
MyObject o3 = new MyObject(new MongoDataBase());
```




Introduction to **Services** and dependency injection

```
@Injectable()
export abstract class OrderRepositoryService {
  constructor() { }
  abstract orders(): String[ ];
}

@Injectable()
export class FakeOrderRepositoryService
      extends
OrderRepositoryService {
  orders(): String[ ] {
    return ["Order 1", "Order 2", "Order 3"];
  }
}
```

```
@Component({
  ...
})
export class VendorsComponent {

  private orders: String[ ] = [ ];

  constructor(private repository:
OrderRepositoryService) {
    this.orders = repository.orders();
  }
}
```

```
@NgModule({
  ...
  providers: [
    {provide: OrderRepositoryService, useClass: FakeOrderRepositoryService}
  ],
  ...
})
export class AppModule { }
```



... but even more

- pipes
- router