SPA development using Angular

But I heard React is good?

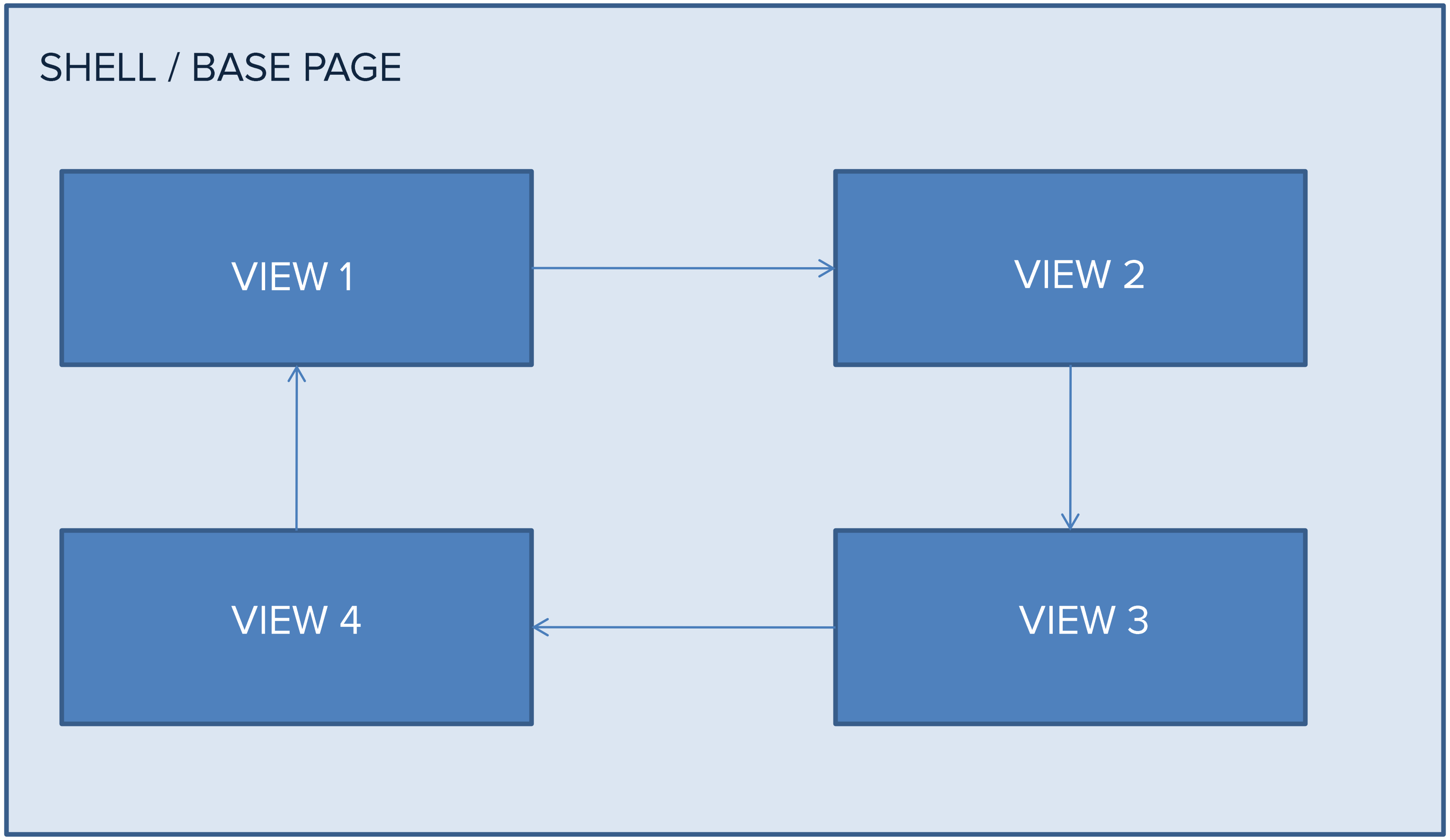
Let’s get back to 1990’s

- The WWW went live on Aug 6, 1991
- JS was created on May 1995
- Dot-com collapse during 1997 - 2001
- Google was incorporated on Sep 4, 1998
- Angular was introduced in 2012

What is AngularJS?

- SPA (Single Page Application Framework)
- MVC – MVVM ?
- Two Way Data Binding
- Works well with Custom attributes

SPA? What?



Pros of SPA over Traditional Webapp

- Traditional app loads the entire page
- Not very bandwidth efficient, especially on mobile
- On Contrary, SPA loads multiple views on fly and renders them on the client

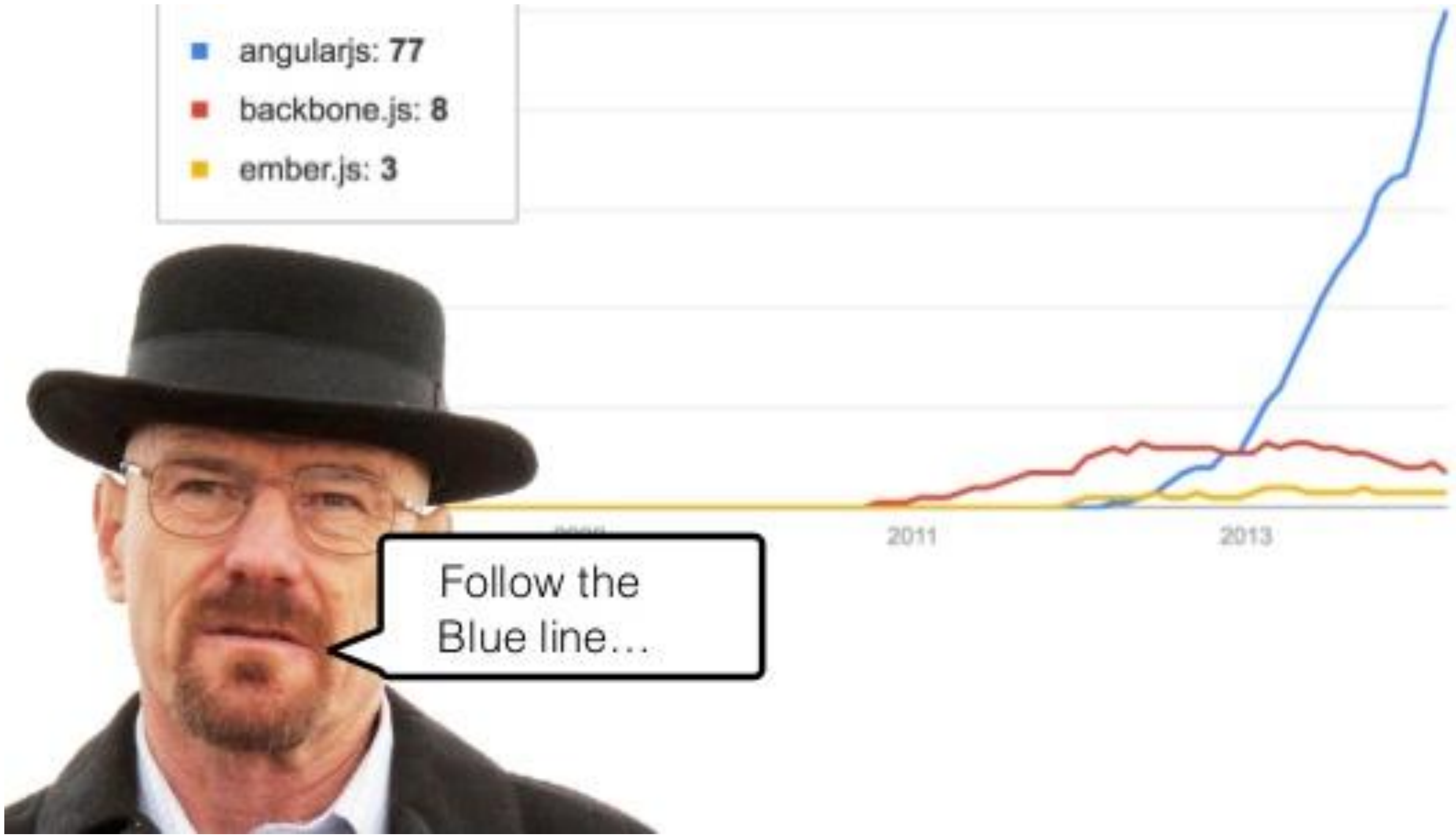
But who uses SPA?

Google

facebook

NETFLIX

Should we use AngularJS?



Lets Start!

```
<html ng-app>
    <head><title>Hello World</title></head>
    <body>
         <div>
               <input type="text" ng-model=“userText” />
               <p> Hello {{userText}} !</p>
         </div>
         <script src=“angular.min.js”></script>
   </body>
</html>
```

Break Down

Directive that initialize Angular Application

```
<html ng-app>
    <head><title>Hello World</title></head>
    <body>
         <div>
               <input type="text" ng-model=“userText“ />
               <p>
                   Hello {{userText}} !
               </p>
         </div>
         <script src=“angular.min.js”></script>
   </body>
</html>
```

Directive that defines our model

Expressions
(Binds our model with HTML)

Directives to the Rescue

- Directives help HTML to play new tricks
- Some of the commonly used directives
  - ng-repeat
  - ng-show
  - ng-hide
  - ng-if

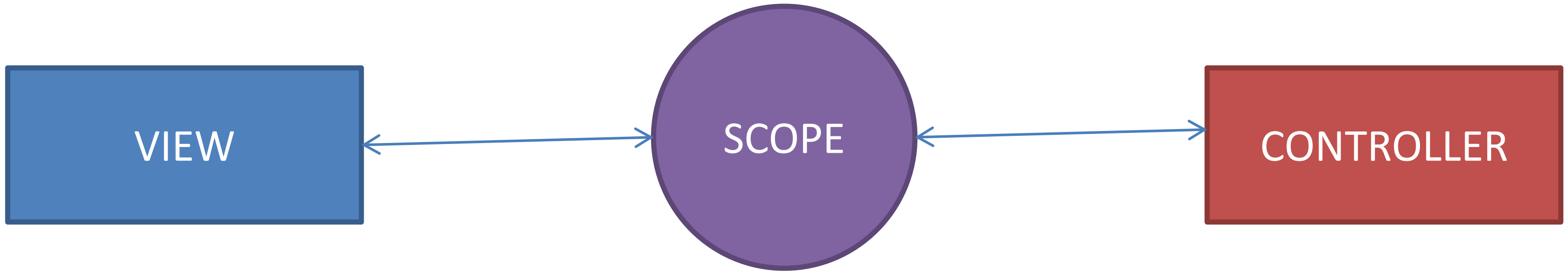
Play with ng-repeat

```
<html ng-app>
    <head><title>Hello World</title></head>
    <body>
         <div ng-init=“names=[‘raghav’,’vijay’,’fazil’]”>
                <ul>
                       <li ng-repeat=“name in names”>
                              {{name}}
                       </li>
                </ul>
         </div>
         <script src=“angular.min.js”></script>
   </body>
</html>
```

Controllers – the fun part

- Controller is simply a JS which controls what data gets to which view and performs other data operations and services.

- View is glued to the controller through Scope.



$$\begin{tikzcd} \text{VIEW} \arrow[r, leftrightarrow] & \text{SCOPE} \arrow[r, leftrightarrow] & \text{CONTROLLER} \end{tikzcd}$$

Controllers – Demo

```
<div ng-controller=“helloWorldCtrl”>
       <h1>{{userText}}</h1>
</div>

function helloWorldCtrl ( $scope )  {
   $scope.userText = “World”;
}
```

Use ***ng-controller*** directive to assign a controller to the attribute scope.

HTTP Requests

- Use *$http* service to make AJAX requests.

- Inject *$http* as dependency to controller that makes requests

```
function getMyData( $scope, $http )  {

     $http.get(“/mydata.json”)

     .success(function(data){
             $scope.myData = data;
     })
     .error(function(err) {
             console.log(err);
     });
}
```

Binding Events

- Angular provides several directives to bind events
    - ng-click
    - ng-dbl-click
    - ng-keypress
    - ng-change

```
<button ng-click=“saveData( )”>Save</button>
```

- Make sure the function that’s triggered is within scope

Using Factory & Services

- Use factory & services to implement reusable components and to enable data sharing between controllers.

```
myApp.service(‘myService’,function(){
     this.getAggregateScore = function() {
          return processedData;
     }
})

myApp.factory(‘myService’,function(){
     return {
          getAggregateScore: function() {}
     }
})
```

Design Pattern

- Modularize your application

```
var myApp = angular.module( “myApp”, [ ] );
```

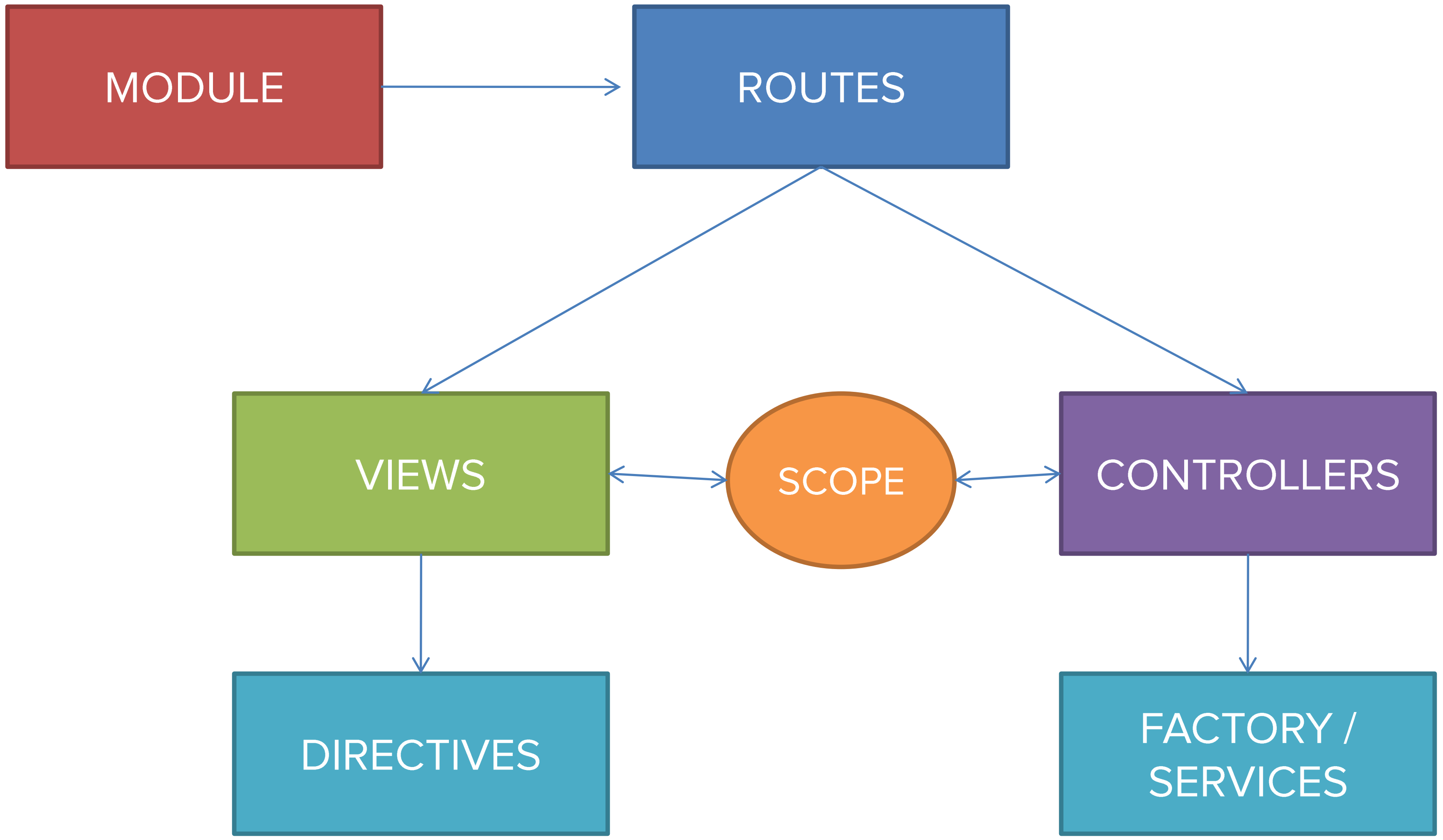
- Inject dependency only when needed.

- Check for memory leaks and profile the footprint.

- Use templates to render the data

```
<div x-axis = “{{xaxis}}”   y-axis=“{{yaxis}}”> </div>
```

Design Pattern



Routing the URLs

- Inject *ngRoute* service to your module

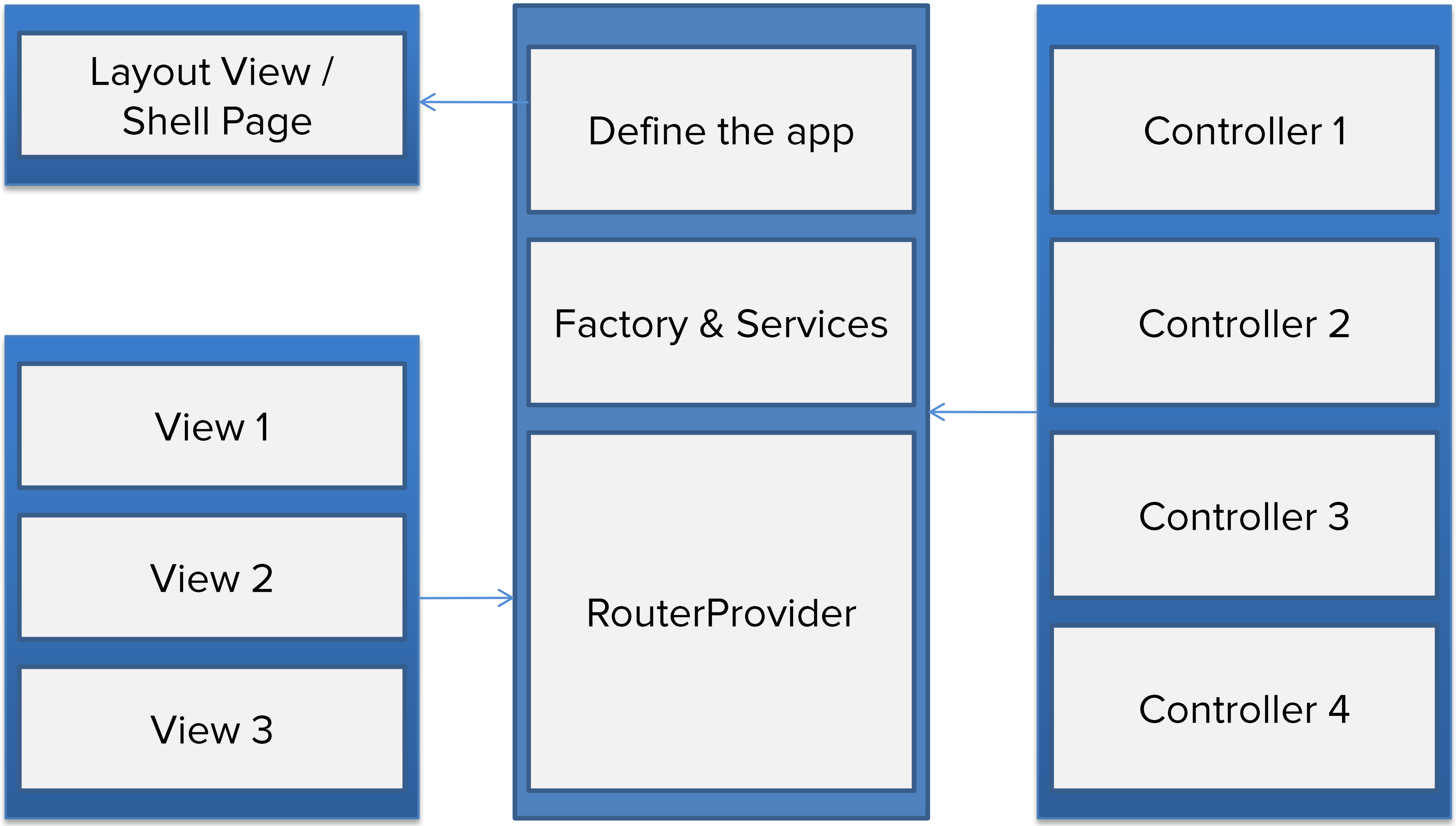
```
var myApp = angular.module(“myApp”, [“ngRoute”]);

myApp.config(function($routeProvider) {

   $routeProvider
   .when(‘/dashboard’, {
        templateUrl: ‘templates/dashboard.html’,
        controller: ‘dashboardCtrl’
   })
   .otherwise({
        templateUrl: ‘templates/404.html’
   });
});
```

Wrapping All together

A Typical Application



Testing

“Well, I’ll just launch the app and see if everything works.

We’ve never had any problem doing that.”

– No one ever

KARMA + Jasmine =