

Angular Gotchas

Best of Web...but watch out
Responsive application build with HTML5 and Javascript

Mike R. Emo
Founder iEmoSoft consulting
Featuring Nick Thurow
<http://www.tinyurl.com/mikeemo>
memo@iEmosoft.com

Shared vs. Isolate

\$Scope

```
<body ng-controller='myCTRL'>  
  <full-name/>  
</body>
```

```
<body ng-controller='myCTRL'>
  <full-name/>
</body>
```

```
app.directive("fullName", function(){
  return {
    scope: false,
    template: ..."
  };
};
```

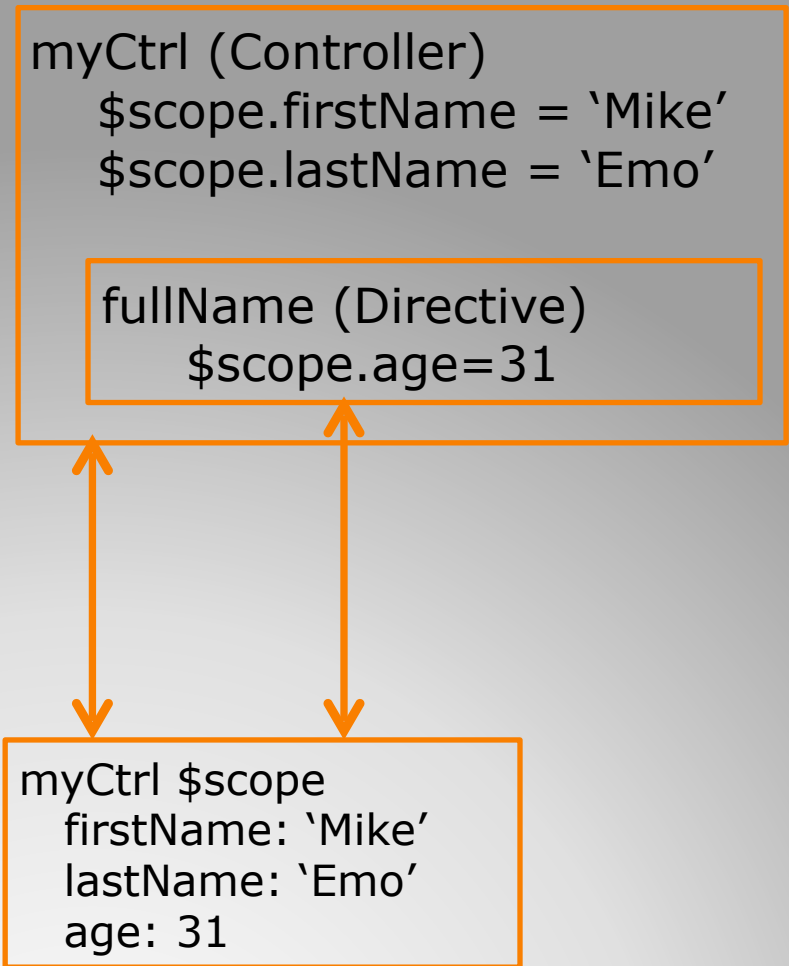
```
<body ng-controller='myCTRL'>
  <full-name/>
</body>
```

```
app.directive("fullName", function(){
  return {
    scope: false,
    template: ..."
  };
};
```

```
myCtrl (Controller)
  $scope.firstName = 'Mike'
  $scope.lastName = 'Emo'
```

```
fullName (Directive)
  $scope.age=31
```

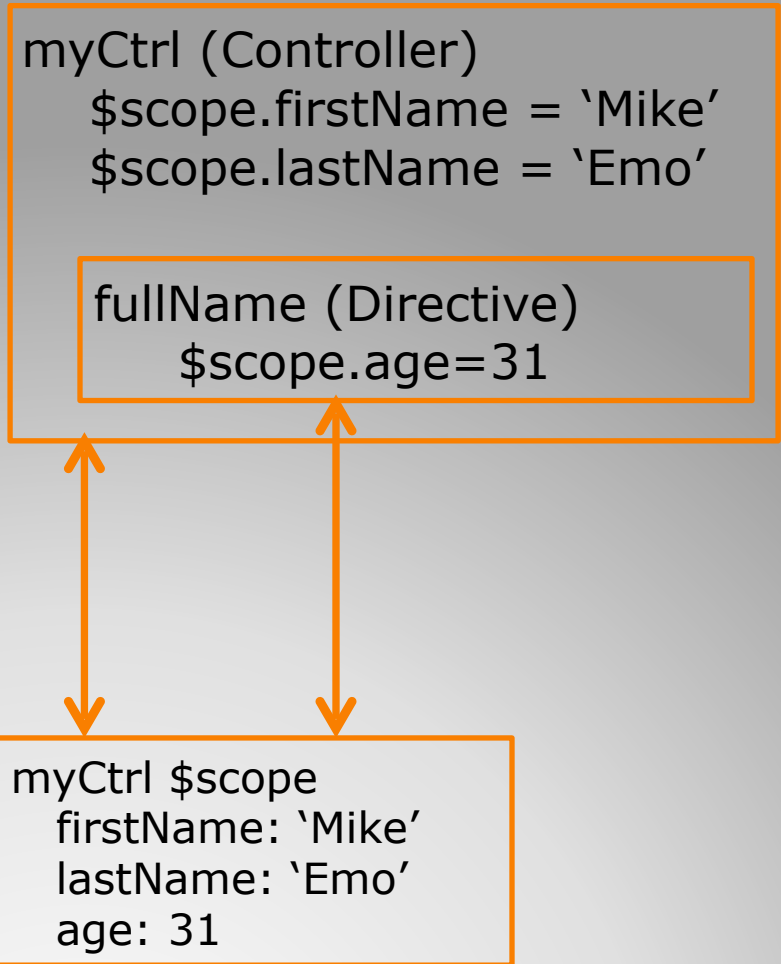
```
myCtrl $scope
  firstName: 'Mike'
  lastName: 'Emo'
  age: 31
```



```
<body ng-controller='myCTRL'>
  <full-name/>
</body>
```

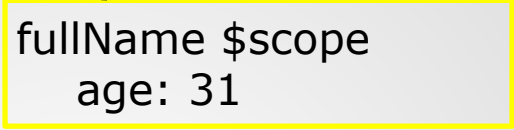
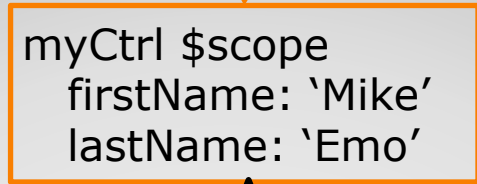
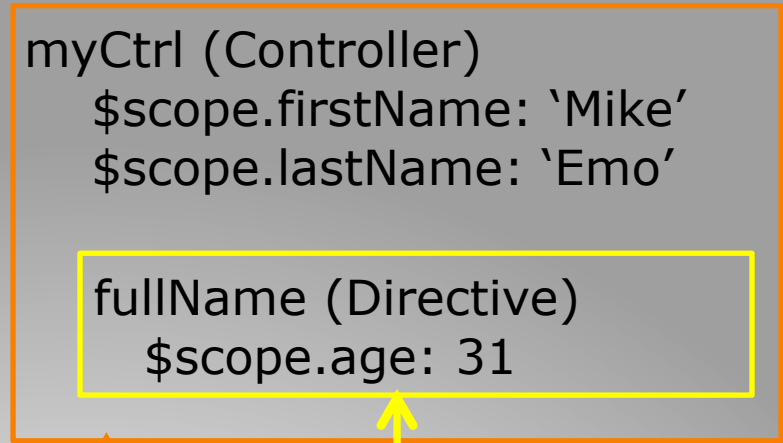
```
app.directive("fullName", function(){
  return {
    scope: false,
    template: ..."
  };
};
```

They share the SAME scope



```
<body ng-controller='myCTRL'>
  <full-name/>
</body>
```

```
app.directive("fullName", function(){
  return {
    scope: true,
    template: ..."
  };
};
```



fullName scope INHERITS from controller scope



```
<body ng-controller='myCTRL'>
  <full-name/>
</body>
```

```
app.directive("fullName", function(){
  return {
```

```
    scope: {
```

```
    },
```

```
    template: ..."
```

```
  };
```

Isolate Scope

```
myCtrl (Controller)
  $scope.firstName = 'Mike'
  $scope.lastName = 'Emo'
```

```
fullName (Directive)
```

```
myCtrl $scope
  firstName: 'Mike'
  lastName: 'Emo'
```

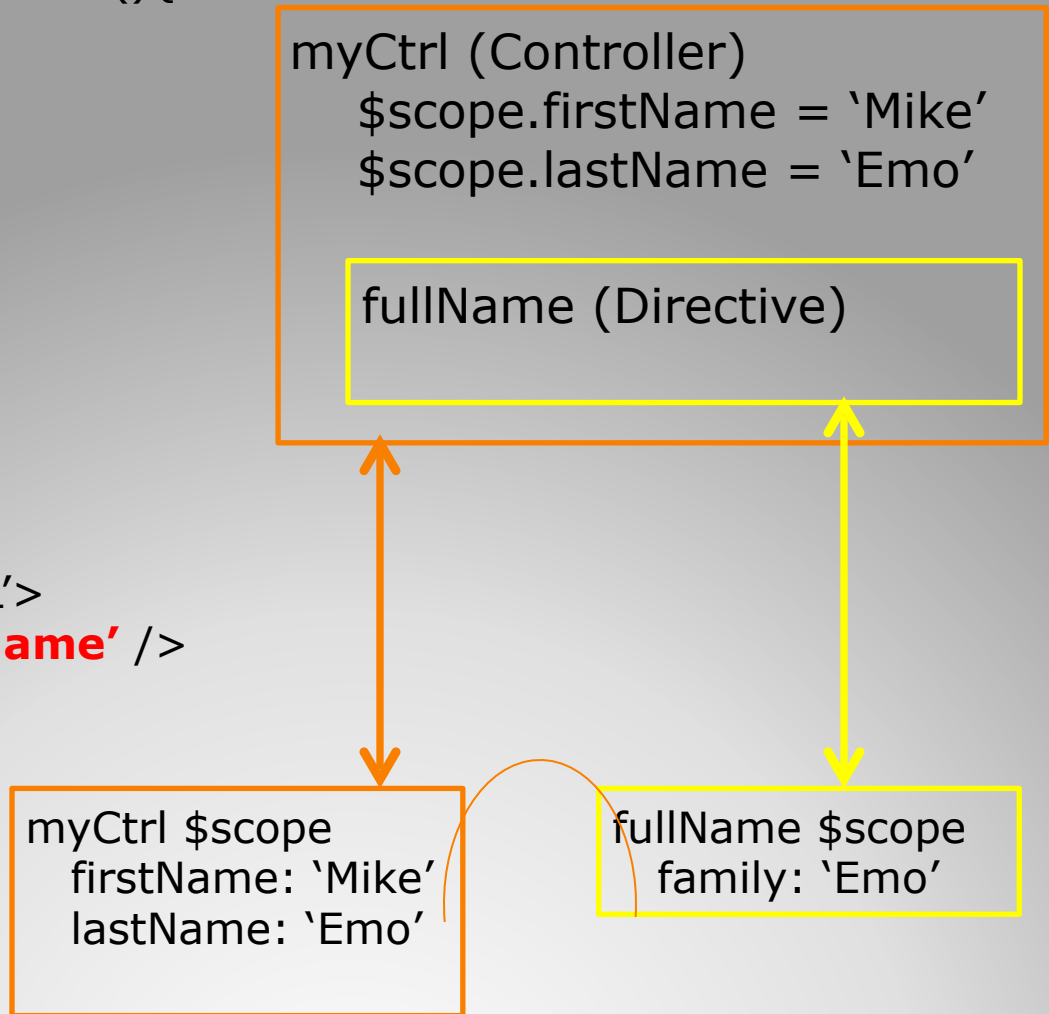
```
fullName $scope
```

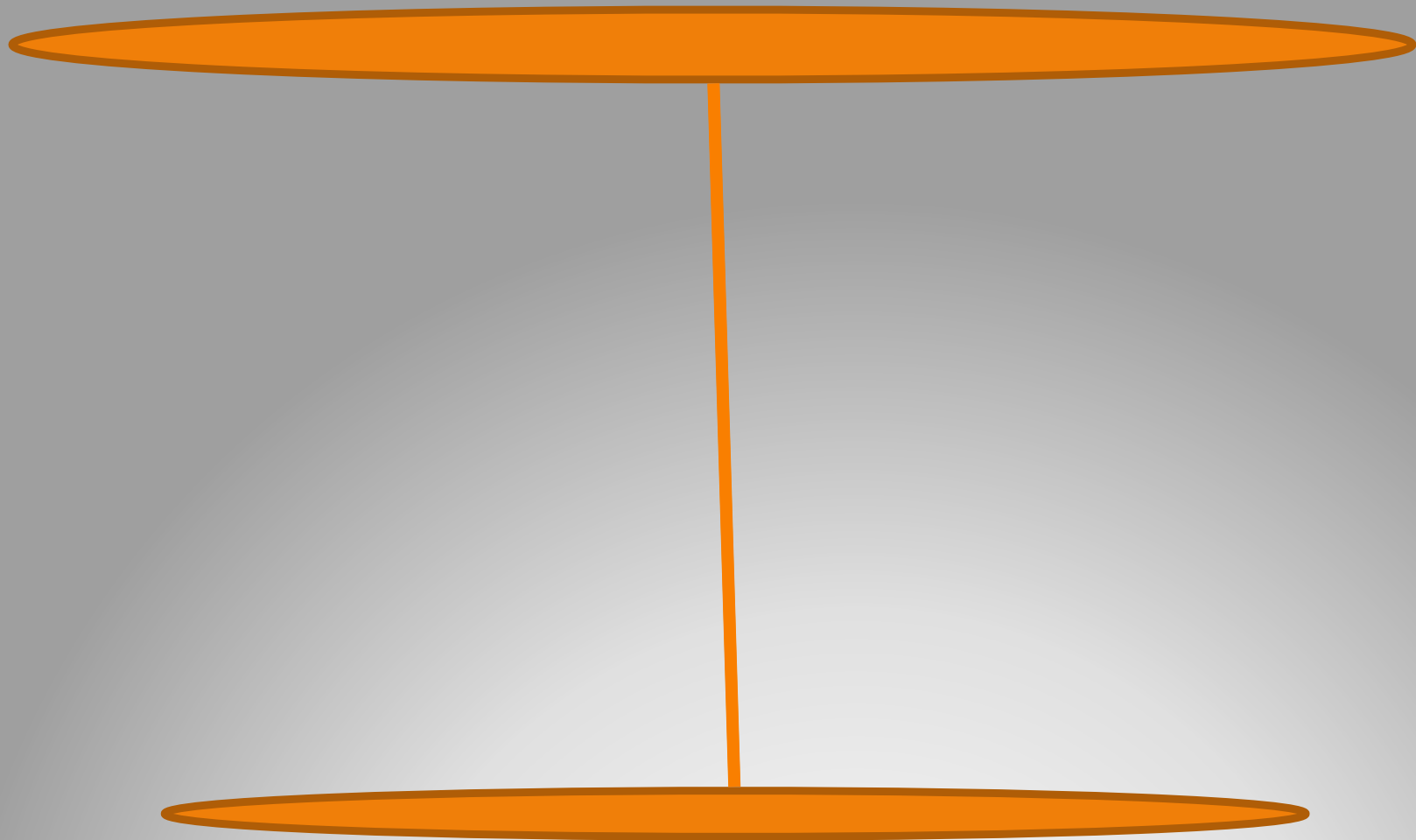


```
app.directive("fullName", function(){
  return {
    scope: {
      family:'='
    },
    template: ..."
  };
```

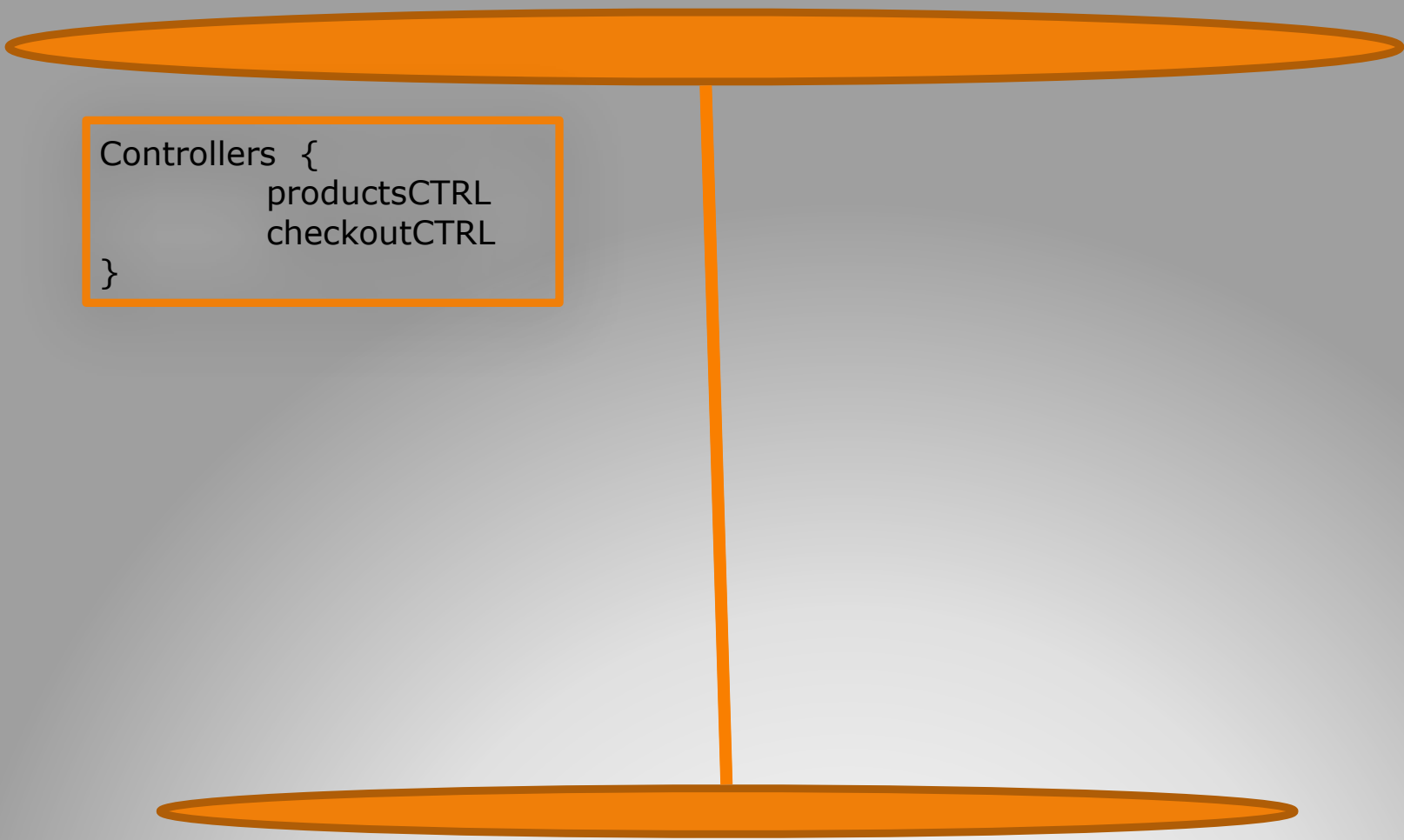
```
<body ng-controller='myCTRL'>
  <full-name family='lastName' />
</body>
```

Isolate Scope
TWO WAY Binding





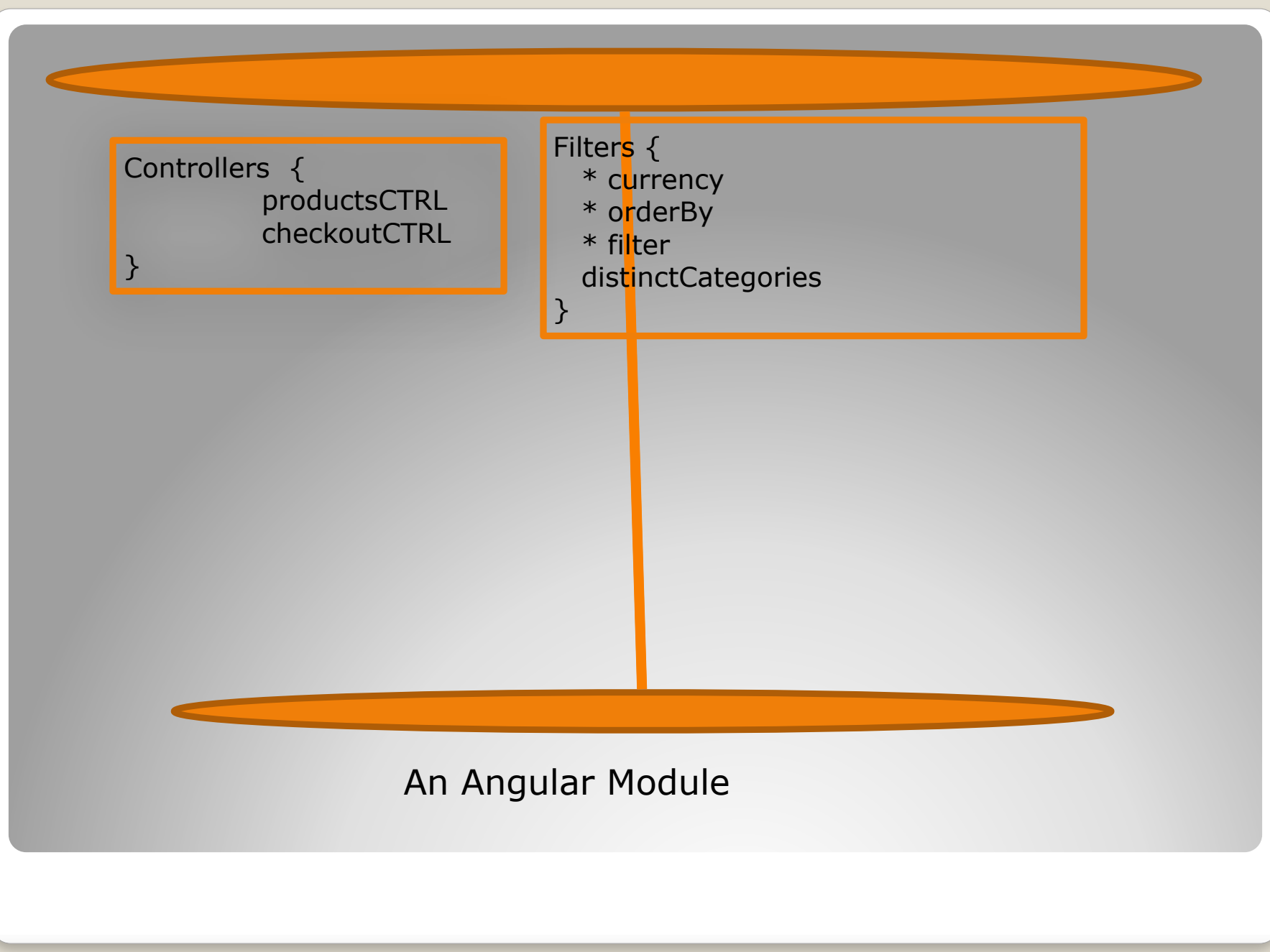
An Angular Module



The diagram illustrates the structure of an Angular Module. It features a large orange oval at the top, a vertical orange line in the center, and another large orange oval at the bottom. A rectangular box on the left side, outlined in orange, contains the following code snippet:

```
Controllers {  
  productsCTRL  
  checkoutCTRL  
}
```

An Angular Module

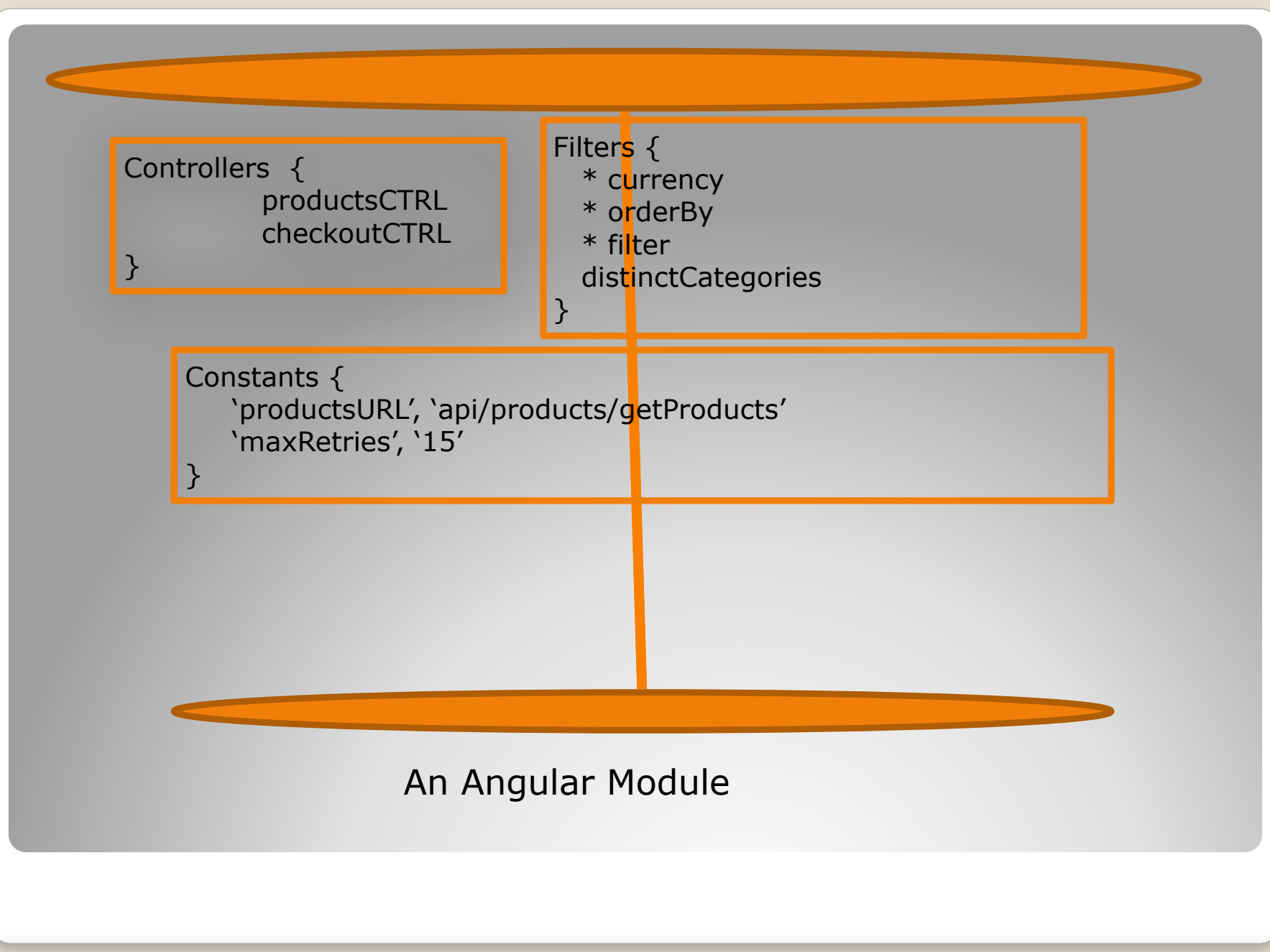


The diagram illustrates the structure of an Angular Module. At the top, a horizontal orange oval is connected by a vertical orange line to a second, identical horizontal orange oval at the bottom. The text 'An Angular Module' is centered between these two ovals. Two rectangular boxes with orange borders are positioned above the bottom oval. The left box contains the code for 'Controllers', listing 'productsCTRL' and 'checkoutCTRL'. The right box contains the code for 'Filters', listing '* currency', '* orderBy', '* filter', and 'distinctCategories'.

```
Controllers {  
  productsCTRL  
  checkoutCTRL  
}
```

```
Filters {  
  * currency  
  * orderBy  
  * filter  
  distinctCategories  
}
```

An Angular Module



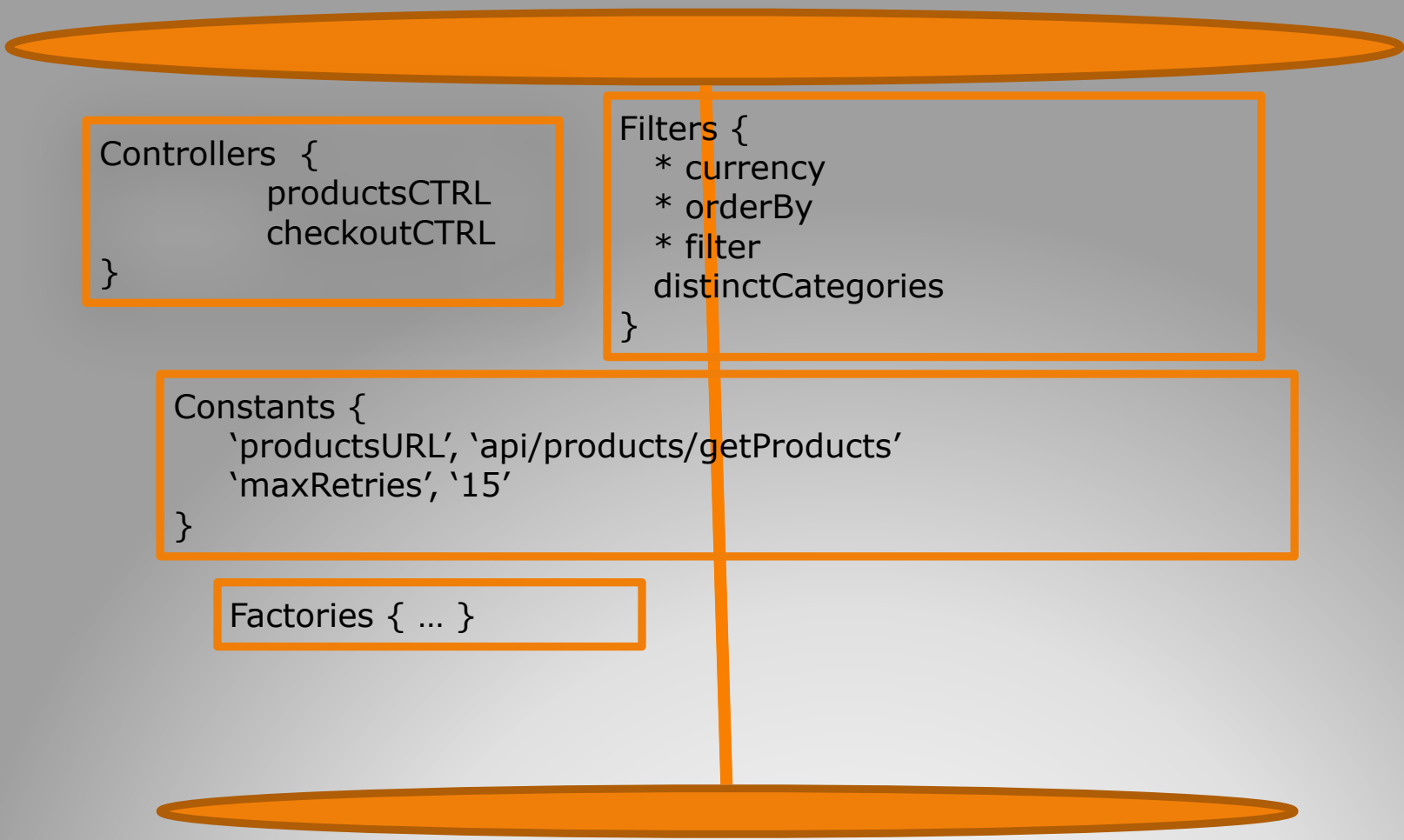
The diagram illustrates the structure of an Angular Module. It features a central vertical orange line that connects two orange ovals at the top and bottom. Three orange-bordered boxes are arranged around this central line. The top-left box contains the 'Controllers' object, the top-right box contains the 'Filters' object, and the middle box contains the 'Constants' object. The bottom oval is labeled 'An Angular Module'.

```
Controllers {  
  productsCTRL  
  checkoutCTRL  
}
```

```
Filters {  
  * currency  
  * orderBy  
  * filter  
  distinctCategories  
}
```

```
Constants {  
  `productsURL`, `api/products/getProducts`  
  `maxRetries`, `15`  
}
```

An Angular Module



The diagram illustrates the structure of an Angular Module. It features a central vertical line with a horizontal bar across it. Four orange-outlined boxes are arranged around this central structure: one on the left, one on the right, one below the horizontal bar on the left, and one below the horizontal bar on the right. Two orange ovals are positioned at the top and bottom of the diagram, with the central vertical line extending through them. The text within the boxes represents the configuration of the module.

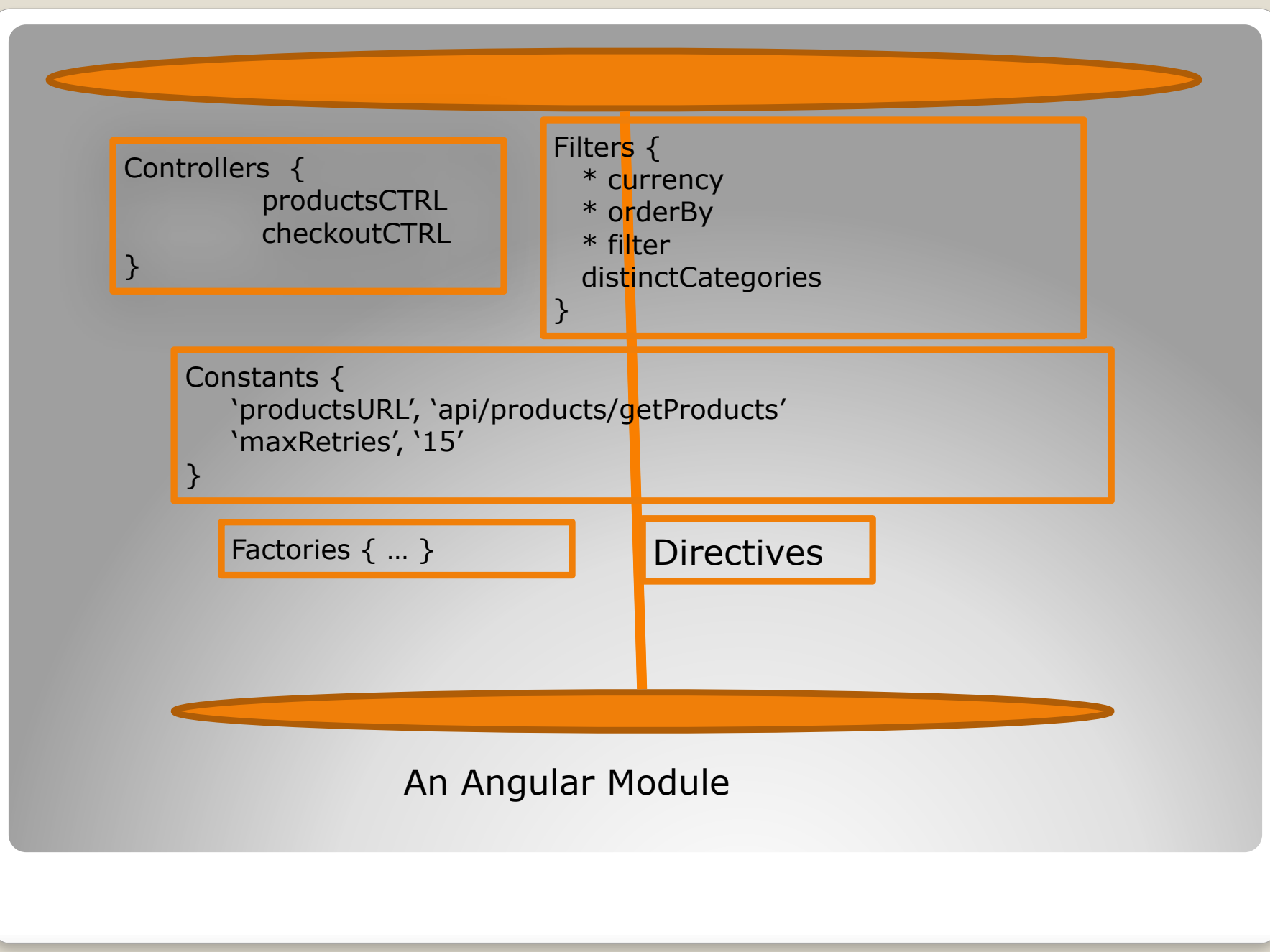
```
Controllers {  
  productsCTRL  
  checkoutCTRL  
}
```

```
Filters {  
  * currency  
  * orderBy  
  * filter  
  distinctCategories  
}
```

```
Constants {  
  'productsURL', 'api/products/getProducts'  
  'maxRetries', '15'  
}
```

```
Factories { ... }
```

An Angular Module



The diagram illustrates the structure of an Angular Module. It features a central vertical line with a horizontal bar across it. At the top and bottom of the vertical line are orange ovals. Five boxes are connected to the central line: 'Controllers' and 'Filters' are at the top; 'Constants' is in the middle; 'Factories' and 'Directives' are at the bottom. Each box contains a list of module components.

```
Controllers {  
  productsCTRL  
  checkoutCTRL  
}
```

```
Filters {  
  * currency  
  * orderBy  
  * filter  
  distinctCategories  
}
```

```
Constants {  
  `productsURL`, `api/products/getProducts`  
  `maxRetries`, `15`  
}
```

```
Factories { ... }
```

```
Directives
```

An Angular Module

```
Controllers {  
  productsCTRL  
  checkoutCTRL  
}
```

```
Filters {  
  * currency  
  * orderBy  
  * filter  
  distinctCategories  
}
```

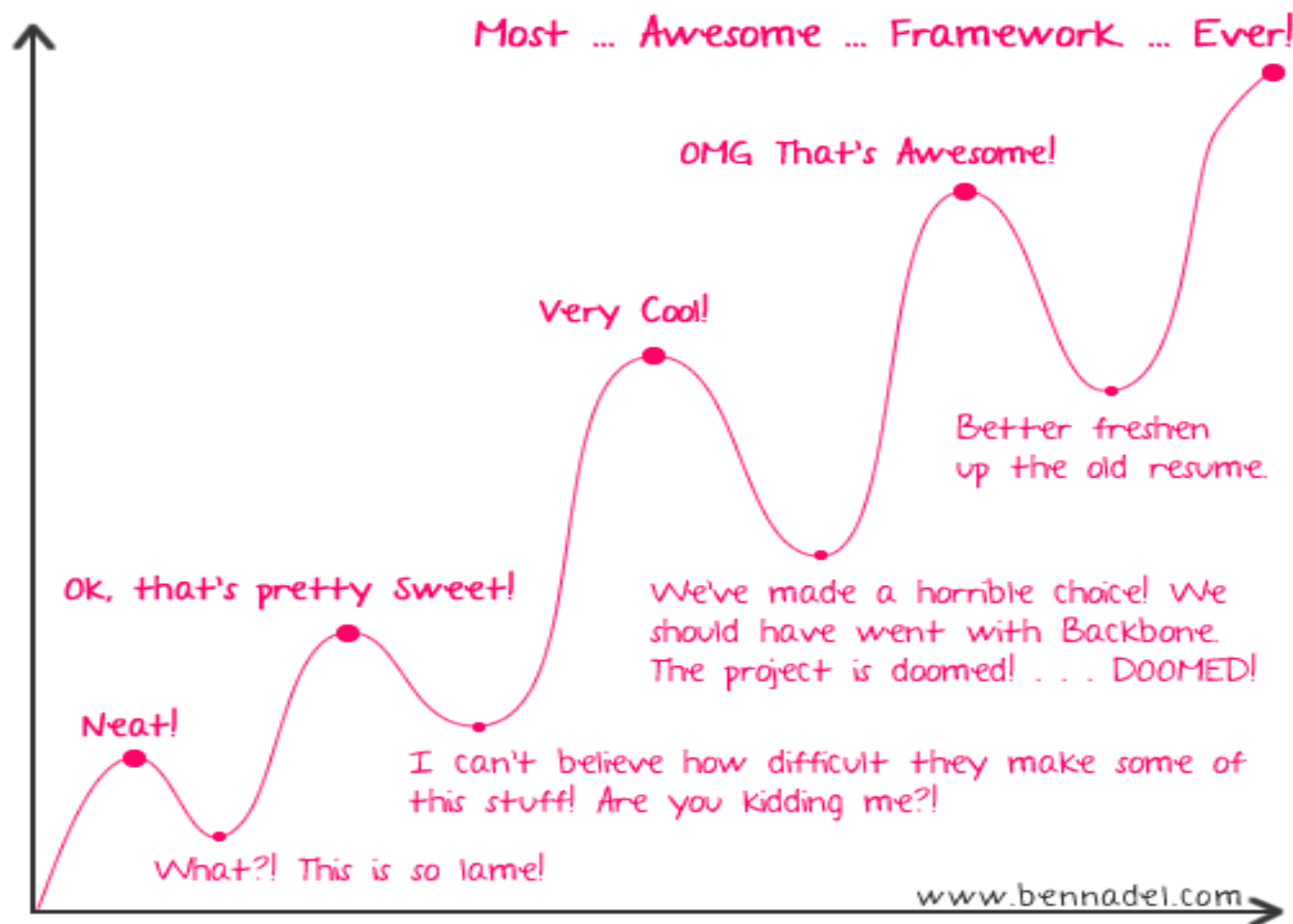
```
Constants {  
  'productsURL', 'api/products/getProducts'  
  'maxRetries', '15'  
}
```

```
Factories { ... }
```

```
Directives
```

```
Module(s)  
"Dependencies"
```

An Angular Module



My Feelings About AngularJS Over Time

SportzApp Module "Bucket"

listOfProducts.html

SportzApp Module "Bucket"

listOfProducts.html



mainCtrl

\$scope

SportzApp Module "Bucket"

listOfProducts.html

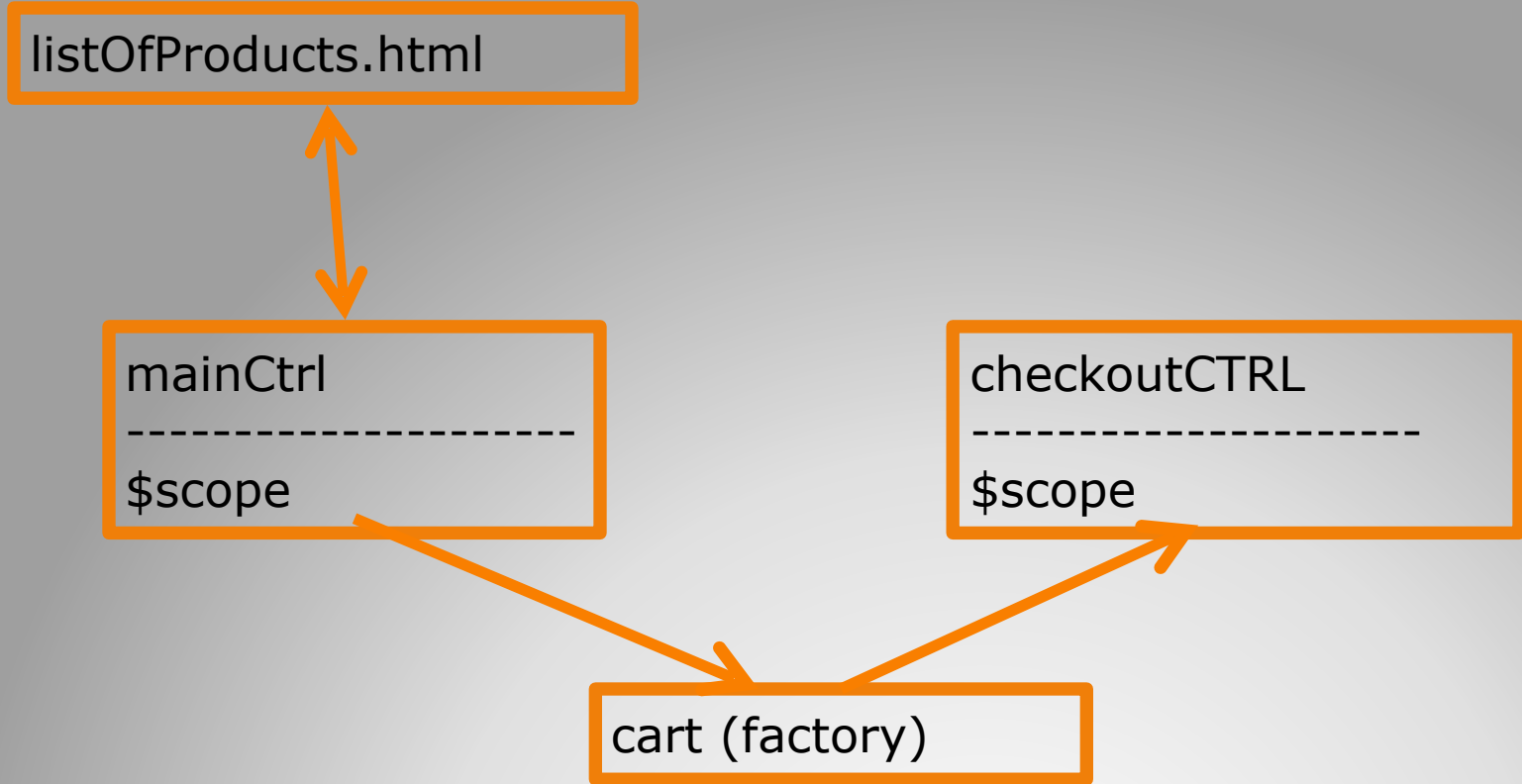


mainCtrl

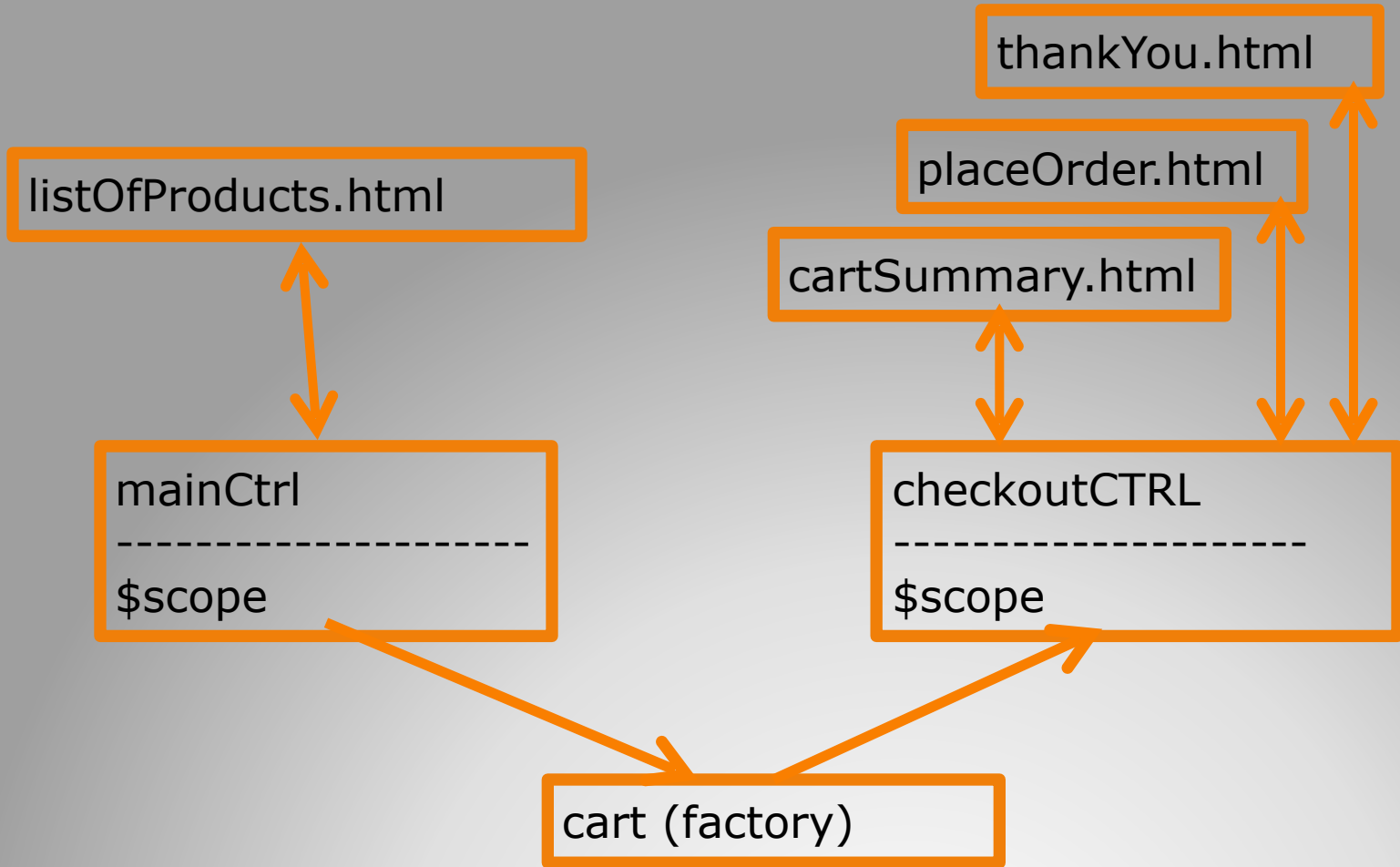
\$scope

cart (factory)

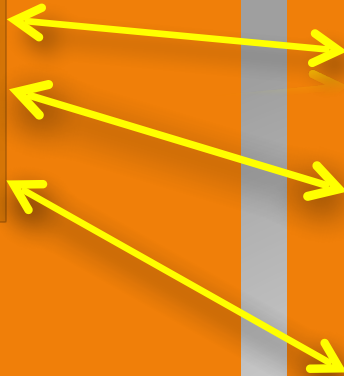
SportzApp Module "Bucket"



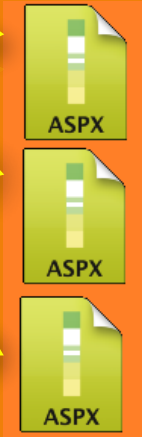
SportzApp Module "Bucket"



Client Machine



Web Server



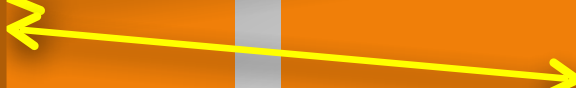
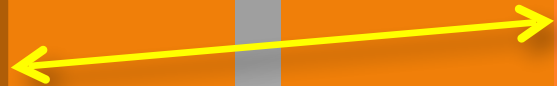
Client Machine



Angular
Controllers

Angular
Directives

Web Server



Client Machine



Angular
Controllers

Angular
Directives

Local
Server

Ajax
Cache
Factory
Biz Log

Web Server



Recap

- Don't overuse modules

Recap

- Don't overuse modules
- Don't overuse isolate scope

Recap

- Don't overuse modules
- Don't overuse isolate scope
- Consider Directives and reusability

Recap

- Don't overuse modules
- Don't overuse isolate scope
- Consider Directives and reusability
- Don't overuse watches

Recap

- Don't overuse modules
- Don't overuse isolate scope
- Consider Directives and reusability
- Don't overuse watches
- Do consider local server pattern

Recap

- Don't overuse modules
- Don't overuse isolate scope
- Consider Directives and reusability
- Don't overuse watches
- Do consider local server pattern
- Keep directives and controllers thin

Recap

- Don't overuse modules
- Don't overuse isolate scope
- Consider Directives and reusability
- Don't overuse watches
- Do consider local server pattern
- Keep directives and controllers thin
- Remove event handlers when applicable to avoid memory leaks

Angular Gotchas

Best of Web...but watch out
Responsive application build with HTML5 and Javascript

Mike R. Emo
Founder iEmoSoft consulting
Featuring Nick Thurow
<http://www.tinyurl.com/mikeemo>
memo@iEmosoft.com

Mike Emo

Email: Memo@iEmosoft.com

Blog: TheMikeEmo.com

Phone: 612-615-6937

Nick Thurow

NThurow@gmail.com

.Net Solution

<https://github.com/emomon23/IntroToAngular>

Contact Info