Shifting to a Component-based Design – Angular (2+) Thinking

Miguel A. Castro
PRINCIPAL CONSULTANT

@miguelcastro67 www.melvicorp.com

Overview

What are components?

How they differ from directives

Component dissection

Why a component-based design

Design comparisons

com·po·nent

/kəmˈpōnənt/

A part or element of a larger whole, especially a part of a machine or vehicle

What Are Components?

Component

- Appearance
- Behavior
- Interface

```
<main-navigator></main-navigator>

<course course-id=“10156”
              show-details=“true”></course>
```

The Angular way !

and AngularJS

Difference from Directives

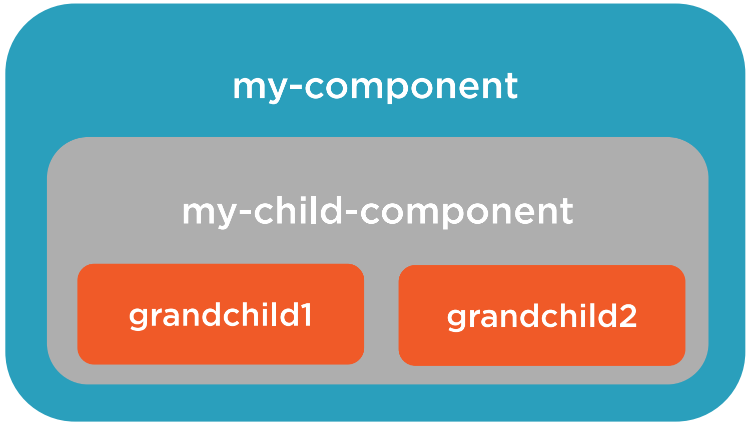
Directive

```
<my-directive
    ng-model="boundValue" />

<input type="text"
    ng-model="boundValue"
    ng-blur="doSomething()" />
```

Directive instructing the HTML input tag to fire a function when focus is lost.

Component

```
<my-component
    item-id="321" />

<my-child-component
    item-id="outer-item-id" />
```



Dissecting a Component

Template

```
var template = `
  <div class="row“
     ng-repeat="module in
         vm.course.Modules">
    <div class="col-md-4">
        {{ $index }} -
        {{ module.Title }}
    </div>
    <div class="col-md-1"></div>
    <div class="col-md-2“>
      {{ vm.timeFormat(modu
    </div>
  </div>
`
```

Controller

```
var controller =
 function (courseService) {
  var vm = this;
  vm.timeFormat = function (
                      module) {
    var hours = 0;
    var minutes =
       Number(module.Minutes);
    var seconds =
       Number(module.Seconds);
```

Bindings

```
var bindings = {
    course: '<‘
}
```

Component

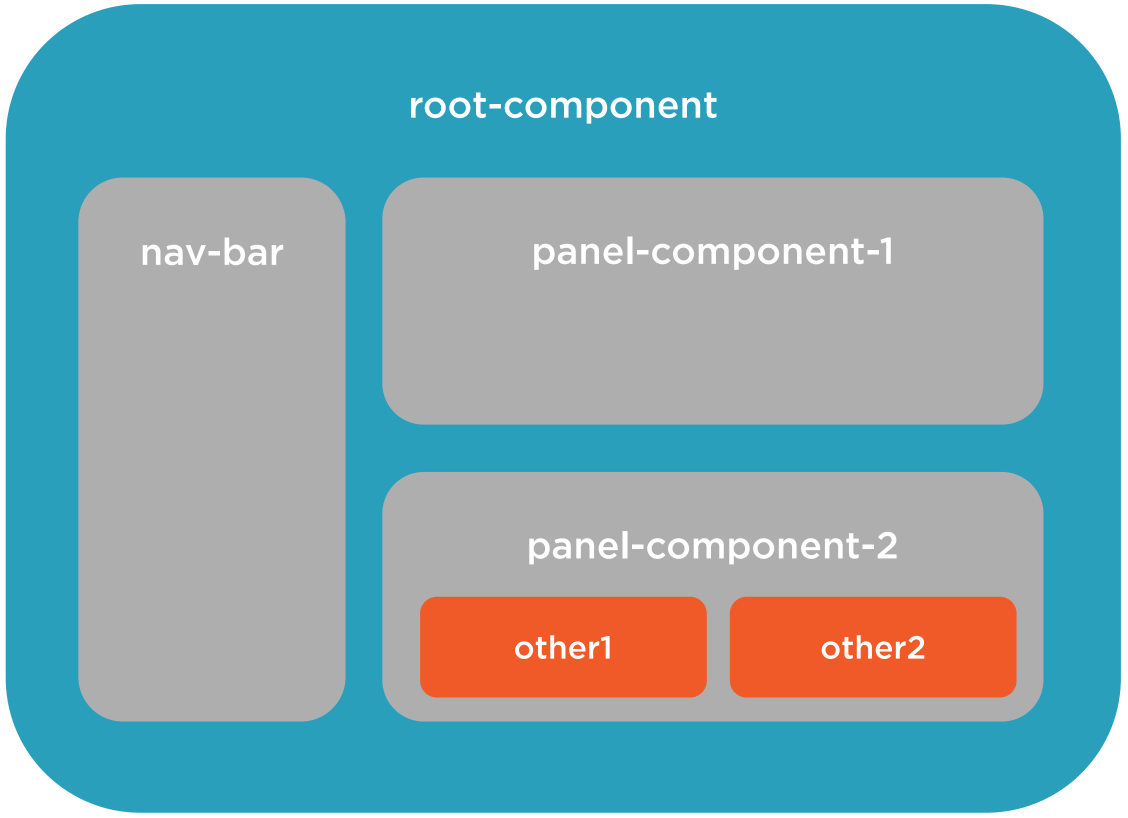
```
angular.module('courseViewer').component('courseModules',
 {
    bindings: bindings,
    controller: controller,
    template: template
});
```

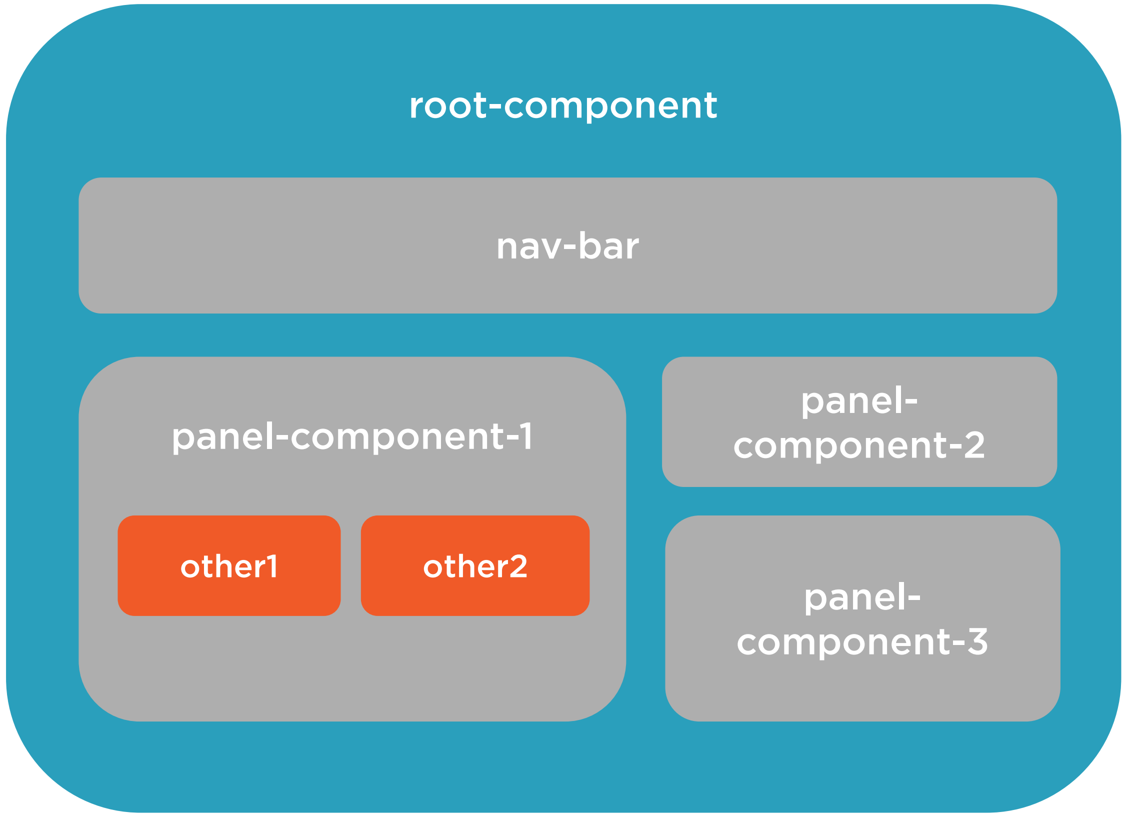
Dissecting a Component

Component

```
angular.module('courseViewer').component('courseModules', {
    bindings:  {
        course: '<'
    },
    controller:  function (courseService) {
        var vm = this;
        vm.timeFormat = function (module) {
            var hours = 0;
            var minutes = Number(module.Minutes);
            var seconds = Number(module.Seconds);
            var moduleLength = courseService.timeFormat(hours, minutes, seconds);
            return moduleLength;
        }
    },
    template:  `
              <div class="row" ng-repeat="module in vm.course.Modules">
                  <div class="col-md-4">{{ $index }} - {{ module.Title }}</div>
                 <div class="col-md-1"></div>
                 <div class="col-md-2">{{ vm.timeFormat(module) }}</div>
              </div>
              `
});
```

Component-based Design





Other Hierarchical Application Platforms

XAML

```
<Grid>
    <StackPanel Orienta
                Grid.R
          <TextBlock T
          <TextBlock T
    </StackPanel>
    <Button Grid.Row="
        <Button.Conter
            <StackPane
                <Image
                <TextE
            </StackPar
        </Button.Conte
    </Button>
</Grid>
```

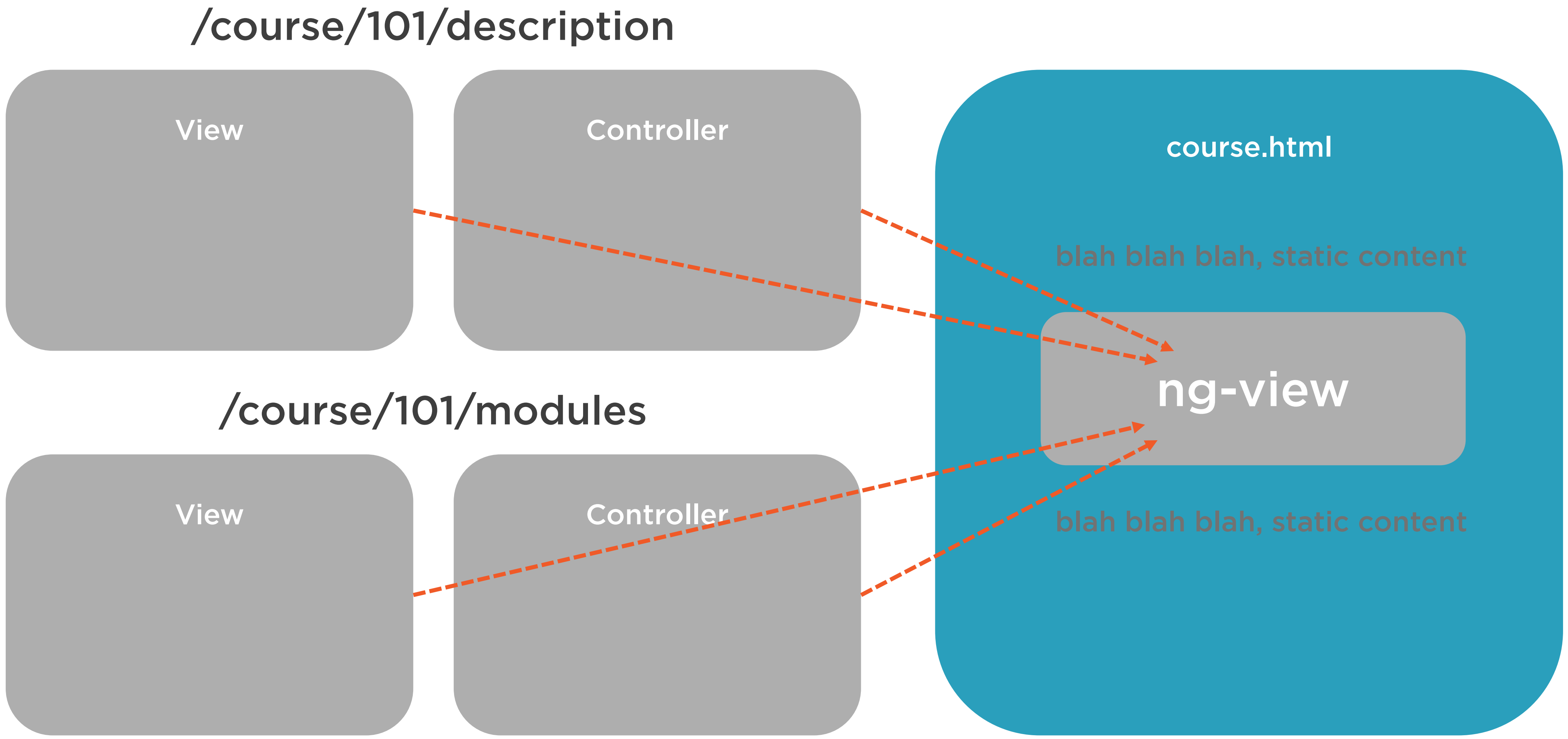
HTML

```
irst name:</label>

me.first" />


at apply:</span>
n in itemList">
eckbox"
="...." />
```

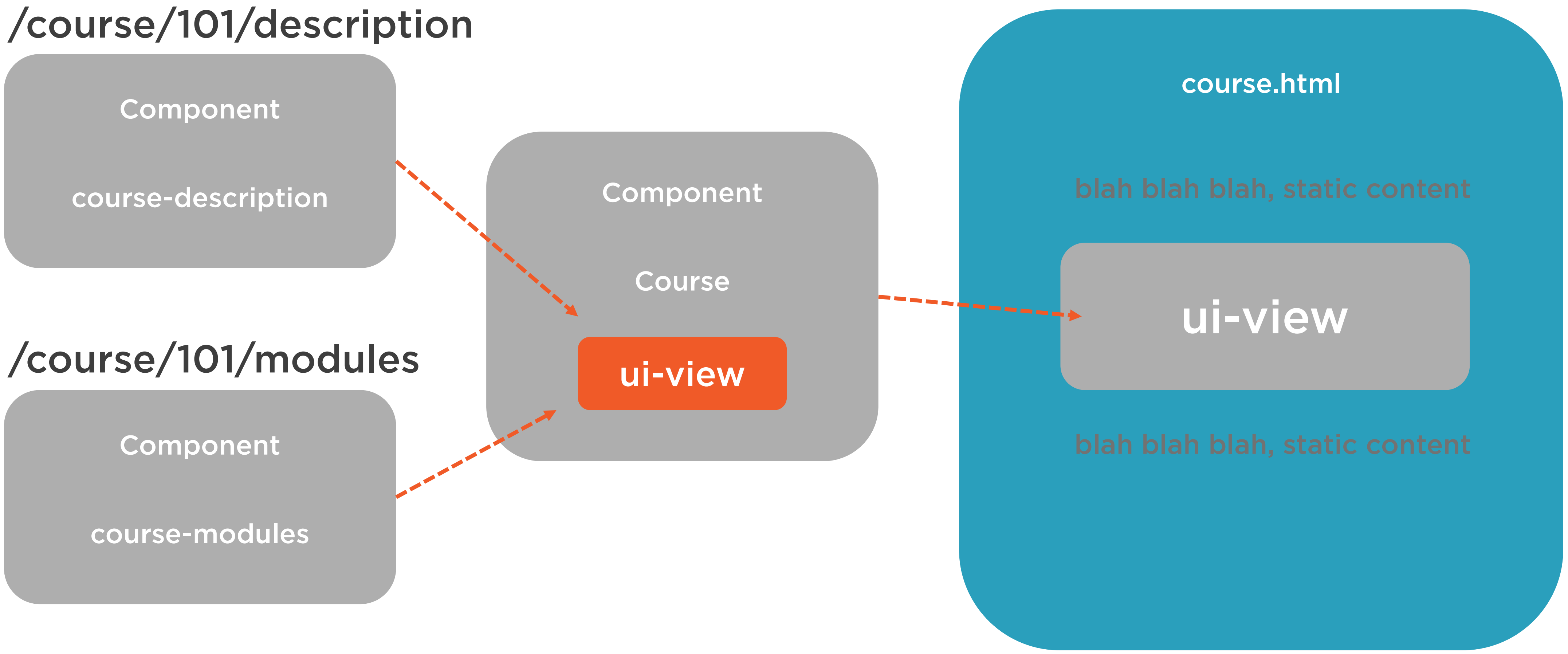
AngularJS

```
angular.module('courseViewer').component('course', {
    bindings: bindings,
    controller: controller,
    template: template
});
angular.module('courseViewer').component('courseModules', {
    bindings: bindings,
    controller: controller,
    template: template
});
angular.module('courseViewer').component('courseDiscusion', {
    bindings: bindings,
    controller: controller,
    template: template
});
```

Component Design vs. Traditional Design



Component Design vs. Traditional Design



Summary

Component overview

Dissecting component into its parts

Component-based design
vs.
Traditional design