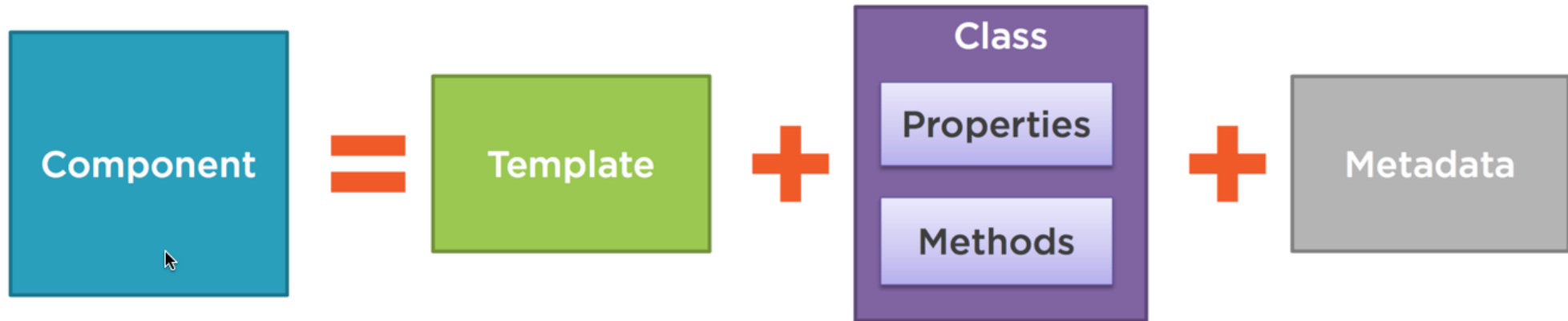


AngularJS

Introduction to Components



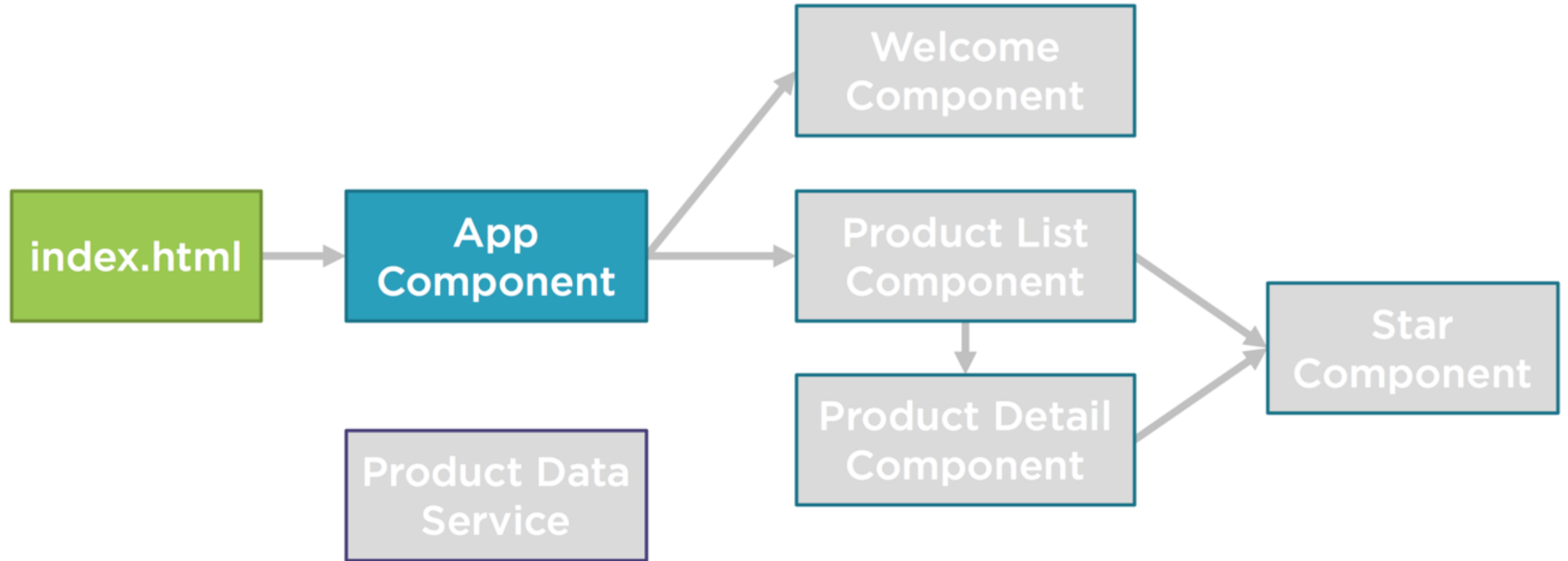
Component



Module Overview

- Building a Template
- Using a Component as a Directive
- Binding with interpolation
- Adding Logic with Directives

Application Architecture



Component

```
// app.component.ts

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

@Component({
  selector: 'nat-app',
  template: `
    <h1>Angular2: {{pageTitle}}</h1>
  `
})

export class AppComponent {
  pageTitle: string = 'Product Management';
}
```

Defining a Template in a Component

Inline Template

```
template:  
"<h1>{{pageTitle}}</h1>"
```

Inline Template ES2015

```
template: `  
<div>  
  <h1>{{pageTitle}}</h1>  
  <div>  
    My First Component  
  </div>  
</div>  
`
```

Linked Template

```
templateUrl:  
'product-list.component.html'
```

Product List View

Product List					
Filter by:	<input type="text"/>				
Show Image	Product	Code	Available	Price	5 Star Rating
	Leaf Rake	GDN-0011	Mar 19, 2016	\$19.95	★★★
	Garden Cart	GDN-0023	Mar 18, 2016	\$32.99	★★★★
	Hammer	TBX-0048	May 21, 2016	\$8.99	★★★★★
	Saw	TBX-0022	May 15, 2016	\$11.55	★★★★
	Video Game Controller	GMG-0042	Oct 15, 2015	\$35.95	★★★★★

Defining a Template in a Component

Product List

Filter by:

Filtered by: am

Show Image

Product	Code	Available	Price	5 Star Rating
Hammer	TBX-0048	May 21, 2016	\$8.99	★★★★★
Video Game Controller	GMG-0042	Oct 15, 2015	\$35.95	★★★★★

Building the Component

```
// product-list.components.ts
```

```
import { Component } from '@angular/core';  
@Component({  
  selector: 'nat-products',  
  templateUrl: 'app/products/product-list.component.html'  
})  
export class ProductListComponent {  
  pageTitle: string = 'Product List';  
}
```

Using a component as a Directive

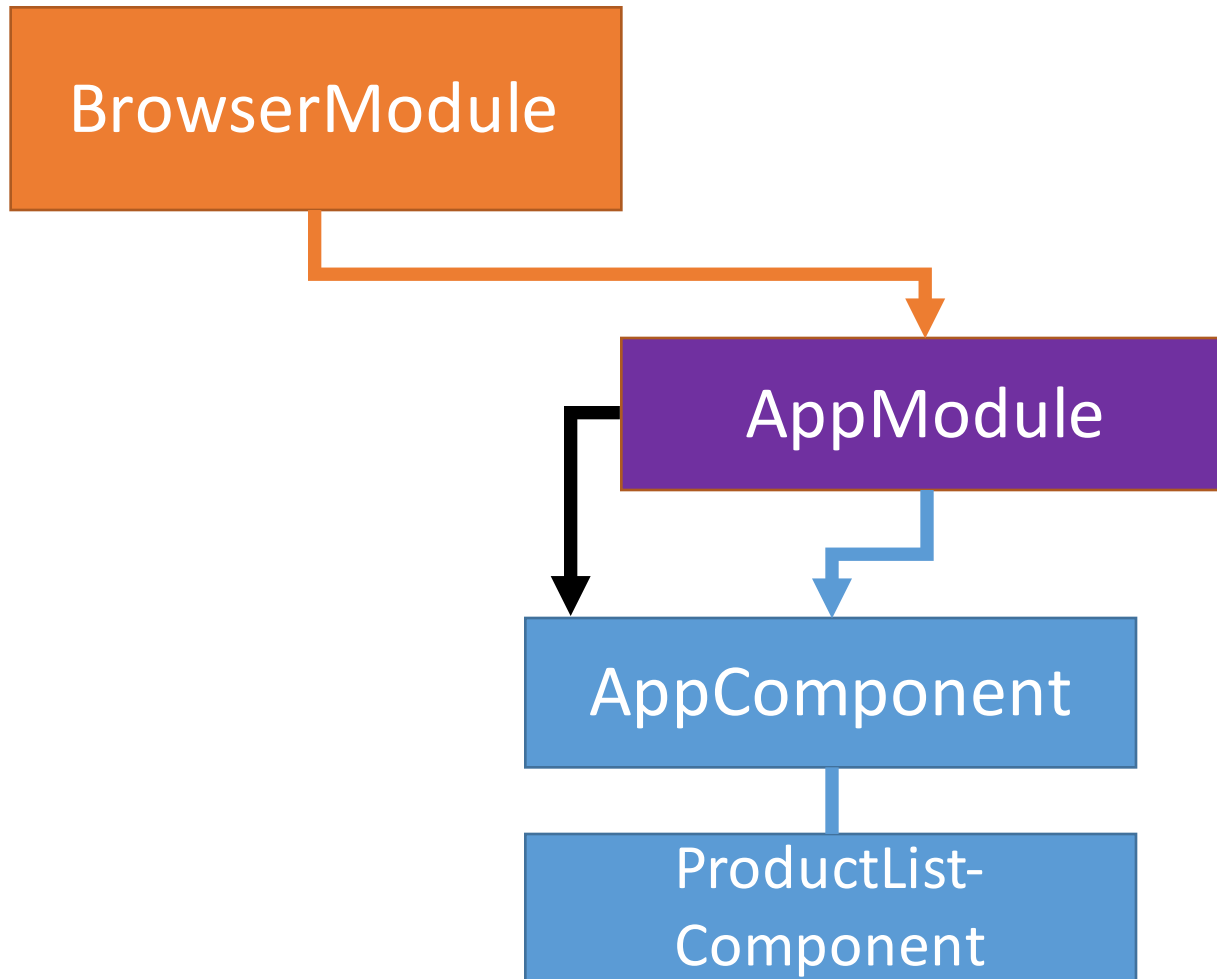
// app.components.ts

```
import { Component } from '@angular/core';
@Component({
  selector: 'nat-app',
  templateUrl: `
    <div><h1>{{pageTitle}}</h1></div>
    <nat-products></nat-products>
  `
})
export class ProductListComponent {
  pageTitle: string = 'Product List';
}
```

// product-list.components.ts

```
import { Component } from '@angular/core';
@Component({
  selector: 'nat-products',
  templateUrl: 'app/products/product-
list.component.html'
})
export class ProductListComponent {
  pageTitle: string = 'Product List';
}
```

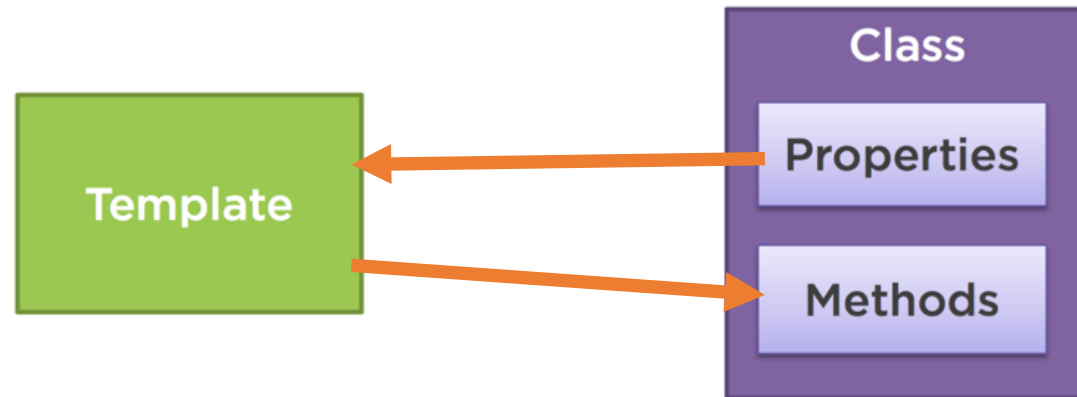




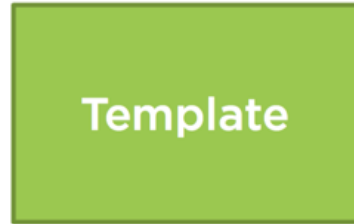
- Imports
- Bootstrap
- Déclarations

Binding

Coordinates communication between the component's class and its template and often involves passing data



Interpolation



```
<h1>{{pageTitle}}</h1>
{{'Title: ' + pageTitle}}
{{2*20+1}}
{{'Title: ' + getTitle()}}
<h1 innerText={{pageTitle}}></h1>
```



```
export class ProductListComponent {
  pageTitle: string = 'Product List';
  getTitle(): string { ... };
}
```

Directive

Custom HTML Element or attribute used to power up and extend our HTML

- Custom
- Built-in

Custom Directive

```
// app.components.ts
```

```
import { Component } from '@angular/core';
@Component({
  selector: 'nat-app',
  templateUrl: `
    <div><h1>{{pageTitle}}</h1></div>
    <nat-products></nat-products>
  `
})
export class ProductListComponent {
  pageTitle: string = 'Product List';
}
```

```
// product-list.components.ts
```

```
import { Component } from '@angular/core';
@Component({
  selector: 'nat-products',
  templateUrl: 'app/products/product-
list.component.html'
})
export class ProductListComponent {
  pageTitle: string = 'Product List';
}
```



Angular Built-in Directives

Structural Directives

***ngIf: If logic**

***ngFor: For loops**

*ngIf Built-In Directive

```
<div class='table-responsive'>  
  <table class='table' *ngIf='products && products.length'>  
    <thead> ...  
  </thead>  
  <tbody> ...  
  </tbody>  
</table>  
</div>
```

*ngIf Built-In Directive

```
<div class='table-responsive'>
  <table class='table' *ngIf='products && products.length'>
    <thead> . .
    </thead>
    <tbody> ...
    </tbody>
  </table>
</div>
```

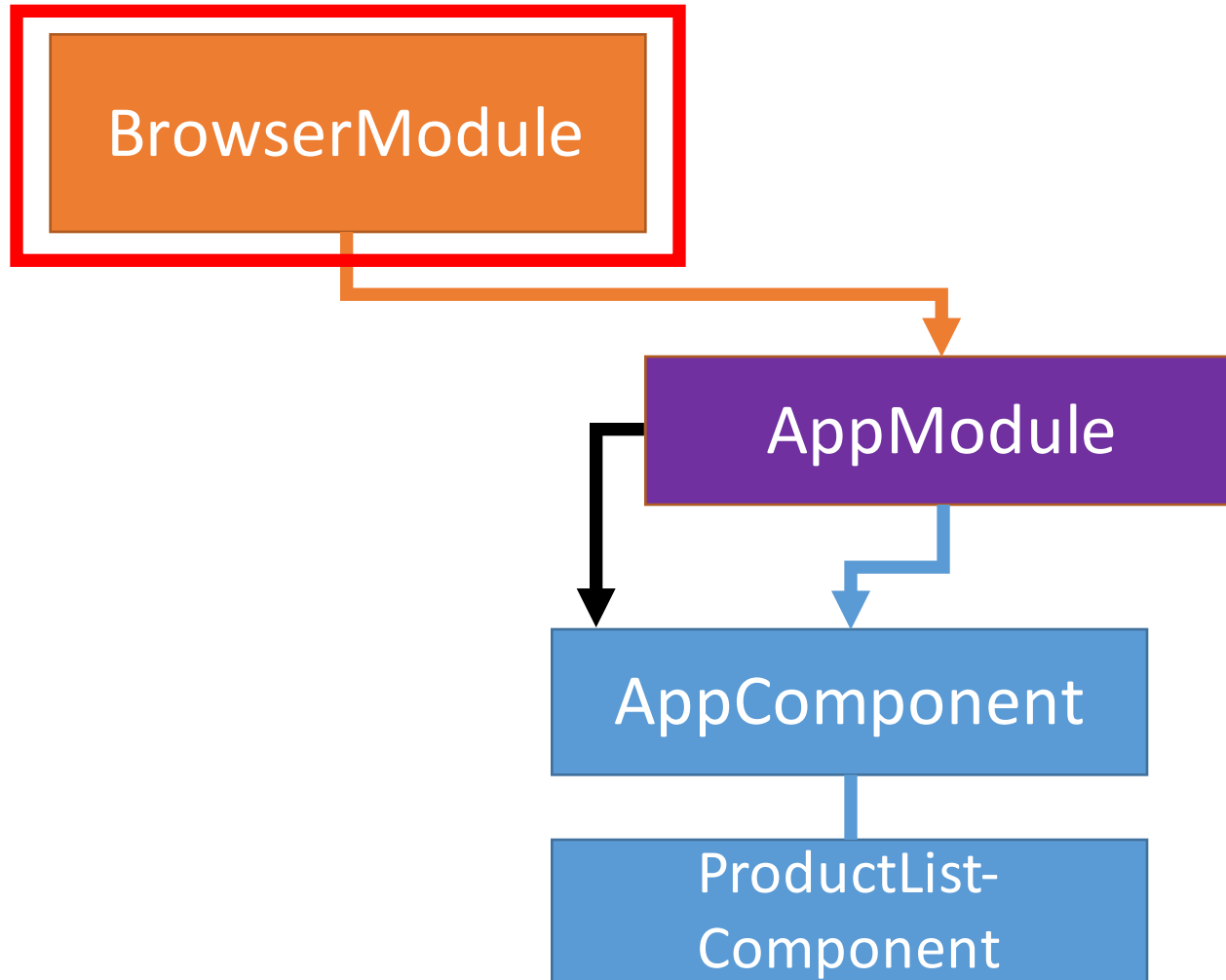


```
...<!DOCTYPE html>
<html>
  <head>...</head>
  <body>
    <script id="__bs_script__">...</script>
    <script async src="/browser-sync/browser-sync-client.2.11.2.js"></script>
    <pm-app>
      <div>
        <h1>Acme Product Management</h1>
        <pm-products>
          <div class="panel panel-primary">
            <div class="panel-heading">
              Product List
            </div>
            <div class="panel-body">
              ::before
              <div class="row">...</div>
              <div class="row">...</div>
              <div class="table-responsive">
                <!--template bindings={}-->
                <table class="table">
                  <thead>...</thead>
                  <tbody>...</tbody>
                </table>
              </div>
              ::after
            </div>
          </div>
        </pm-products>
      </div>
    </pm-app>
  </body>
</html>
```

*ngIf Built-In Directive

```
<div class='table-responsive'>
  <table class='table' *ngIf='products && products.length'>
    <thead> ...
  </thead>
  <tbody> ...
</tbody>
</table>
</div>
```

```
...<!DOCTYPE html>
<html>
  <head>...</head>
  <body>
    <script id="__bs_script__">...</script>
    <script async src="/browser-sync/browser-sync-client.2.11.2.js"></script>
    <pm-app>
      <div>
        <h1>Acme Product Management</h1>
        <pm-products>
          <div class="panel panel-primary">
            <div class="panel-heading">
              Product List
            </div>
            <div class="panel-body">
              ::before
              <div class="row">...</div>
              <div class="row">...</div>
              <div class="table-responsive">
                <!--template bindings={}-->
              </div>
              ::after
            </div>
          </div>
        </pm-products>
      </div>
    </pm-app>
  </body>
</html>
```



- Imports
- Bootstrap
- Déclarations

*ngFor Built-In Directive

```
<tr *ngFor='let product of products'>  
  <td></td>  
  <td>{{ product.productName }}</td>  
  <td>{{ product.productCode }}</td>  
  <td>{{ product.releaseDate }}</td>  
  <td>{{ product.price }}</td>  
  <td>{{ product.starRating }}</td>  
</tr>
```

*ngFor Built-In Directive

Local variable

```
<tr *ngFor='let product of products'>
  <td></td>
  <td>{{ product.productName }}</td>
  <td>{{ product.productCode }}</td>
  <td>{{ product.releaseDate }}</td>
  <td>{{ product.price }}</td>
  <td>{{ product.starRating }}</td>
</tr>
```

For...of vs for...in

for...of

- Iterates over iterable objects, such as an array.
- Result: di, boo, punkeye

```
let nicknames= ['di', 'boo', 'punkeye'];  
  
for (let nickname of nicknames) {  
  console.log(nickname);  
}
```

for...in

- Iterates over the properties of an object.
- Result: 0, 1, 2

```
let nicknames= ['di', 'boo', 'punkeye'];  
  
for (let nickname in nicknames) {  
  console.log(nickname);  
}
```


Checklist: Template

- ✓ Inline template
 - ✓ For short templates
 - ✓ Specify the template property
 - ✓ Use the ES2015 back ticks for multiple lines
 - ✓ Watch syntax
- ✓ Linked template
 - ✓ For longer template
 - ✓ Specify the templateUrl property
 - ✓ Define the path to the HTML file

Checklist: Interpolation

- ✓ One way binding
 - ✓ From component class property to an element property
- ✓ Defined with double curly braces
 - ✓ Contains a template expression
 - ✓ No quote needed

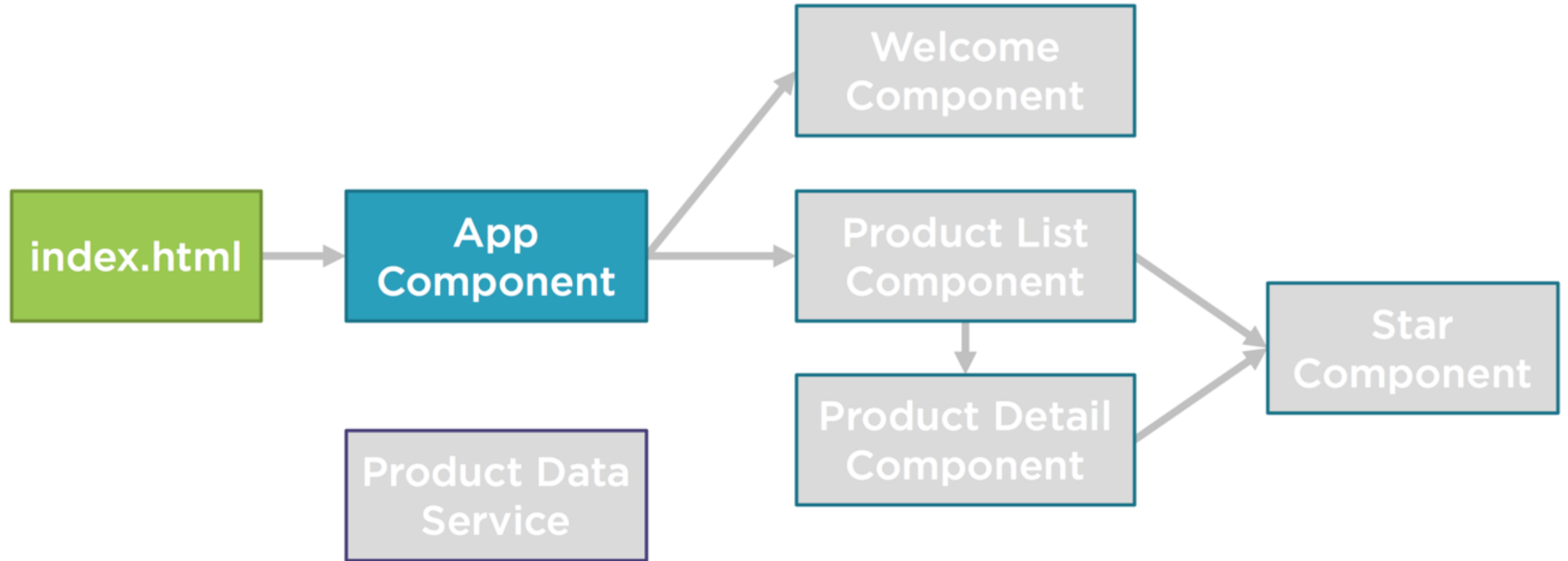
Checklist: Structural Directives

- ✓ `*ngIf` and `*ngFor`
 - ✓ Prefix with an asterisk
 - ✓ Assign to a quoted string expression
- ✓ `*ngIf`
 - ✓ Expression is evaluated as a boolean value
- ✓ `*ngFor`
 - ✓ Define the local variable with `let`
 - ✓ Specify 'of': `'let product of products'`

Module Overview

- Building a Template
- Using a Component as a Directive
- Binding with interpolation
- Adding Logic with Directives

Application Architecture



Application Architecture

