Component

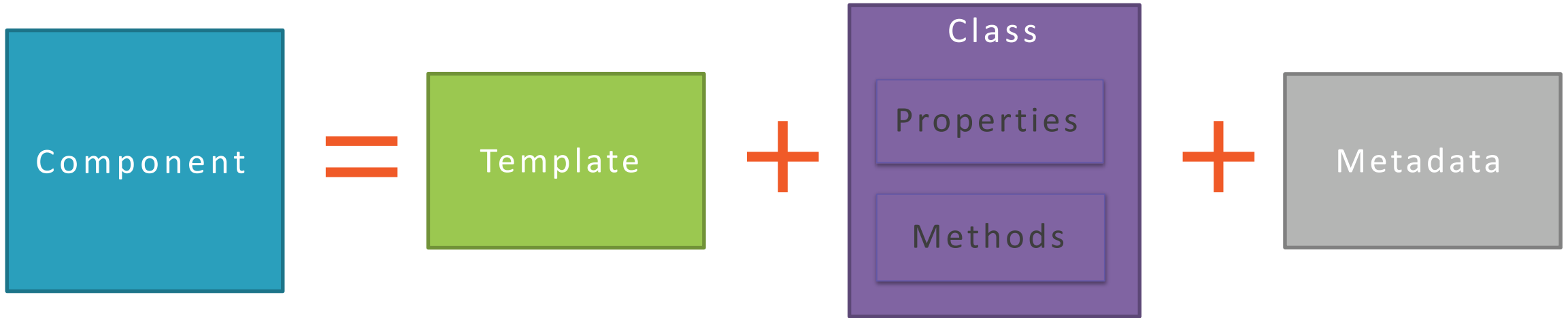Component = Template + Class
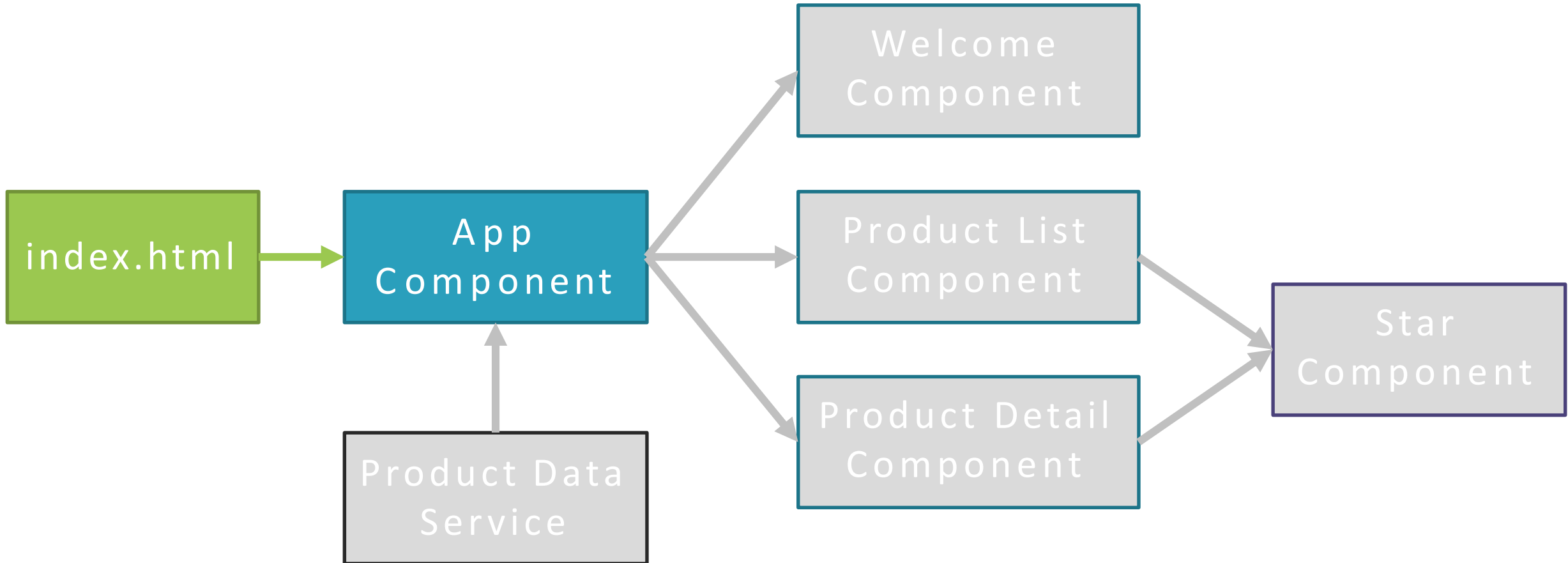(Properties, Methods) + Metadata

Module Overview

Building a Template

Using a Component as a Directive

Binding with Interpolation

Adding Logic with Directives

Application Architecture

Component

```typescript
import { Component } from 'angular2/core';

@Component({
    selector: 'pm-app',
    template: `
    <div><h1>{{pageTitle}}</h1>
        <div>My First Component</div>
    </div>
    `
})
export class AppComponent {
 pageTitle: string = 'Acme Product Management';
}
```

Defining a Template in a Component

Inline Template

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

Inline Template

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

ES 2015 Back Ticks

Linked Template

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

Product List View

Product List

Filter by: [   ]

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	★★★★

Product List View

Product List

Filter by: [am   ×]

Filtered by: am

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

Product List View

Product List

Filter by: [am]

Filtered by: am

Hide 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	★★★★⯪

Product List View

Product List

Filter by: [   ]

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	★★★★

http://getbootstrap.com/

Building the Component

```typescript
import { Component } from 'angular2/core';

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

Using a Component as a Directive

```html
<body>
  <pm-app>Loading App...</pm-app>
</body>
```

product-list.component.ts

```typescript
@Component({
    selector: 'pm-products',
  templateURL: 'product-list.component.html'
})
export class ProductListComponent { }
```

app.component.ts

```typescript
@Component({
    selector: 'pm-app',
    template: `
      <div><h1>{{pageTitle}}</h1>
       <div>My First Component</div>
      </div>`
})
export class AppComponent { }
```

Using a Component as a Directive

```html
<body>
  <pm-app>Loading App...</pm-app>
</body>
```

```typescript
@Component({
    selector: 'pm-products',
  templateURL: 'product-list.component.html'
})
export class ProductListComponent {  }
```

1

```typescript
@Component({
    selector: 'pm-app',
    template: `
      <div><h1>{{pageTitle}}</h1>
       <pm-products></pm-products>
      </div>`
})
export class AppComponent {  }
```

Using a Component as a Directive

index.html

```html
<body>
  <pm-app>Loading App...</pm-app>
</body>
```

product-list.component.ts

```typescript
@Component({
    selector: 'pm-products',
  templateURL: 'product-list.component.html'
})
export class ProductListComponent {  }
```

app.component.ts

```typescript
@Component({
    selector: 'pm-app',
    template: `
      <div><h1>{{pageTitle}}</h1>
        <pm-products></pm-products>
      </div>`,
    directives: [ProductListComponent]
})
export class AppComponent {  }
```

1

2

Using a Component as a Directive

```html
<body>
  <pm-app>Loading App...</pm-app>
</body>
```

product-list.component.ts

```typescript
@Component({
    selector: 'pm-products',
   templateURL: 'product-list.component.html'
})
export class ProductListComponent {  }
```

app.component.ts

```typescript
3  import {ProductListComponent} from 'product-list.component';
   @Component({
       selector: 'pm-app',
       template: `
         <div><h1>{{pageTitle}}</h1>
1          <pm-products></pm-products>
         </div>`,
2      directives: [ProductListComponent]
   })
   export class AppComponent {  }
```

Binding

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

Template

Class

Properties

Methods

Interpolation

```
<h1>{{pageTitle}}</h1>
```

```
{{'Title: ' + pageTitle}}
```

```
{{2*20+1}}
```

```
{{'Title: ' + getTitle()}}
```

```
<h1 innerText={{pageTitle}}></h1>
```

```
export class AppComponent {
  pageTitle: string =
      'Acme Product Management';
}getTitle(): string {...};
}
```

Directive

Custom HTML element or attribute used to power up and extend our HTML.

- Custom

- Built-In

Custom Directives

```
@Component({
  selector: 'pm-app',
  template: `
    <div><h1>{{pageTitle}}</h1>
      <pm-products></pm-products>
    </div>
    `,
  directives: [ProductListComponent]
})
export class AppComponent { }
```

```
@Component({
  selector: 'pm-products',
  templateURL: 'product-list.component.html'
})
export class ProductListComponent { }
```

Angular Built-in Directives

Structural
Directives

*ngIf: If logic

*ngFor: For loops

*ngIf Built-In Directive

```html
<table *ngIf='products.length'>
  <thead>
  </thead>
  <tbody>
  </tbody>
</table>
```

*ngFor Built-In Directive

```html
<tr *ngFor='#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>
```

#
Local Variable

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);
}
```

*ngFor Built-In Directive

```html
<tr *ngFor='#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>
```

Checklist: Template

Inline template
- For short templates
- Specify the `template` property
- Use the ES 2015 back ticks for multiple lines
- Watch syntax

Linked template
- For longer templates
- Specify the `templateUrl` property
- Define the path to the HTML file

Checklist: Component as a Directive

product-list.component.ts

```
@Component({
    selector: 'pm-products',
  templateURL: 'product-list.component.html'
})
export class ProductListComponent { }
```

app.component.ts

```
3  import {ProductListComponent} from 'product-list.component';
   @Component({
     selector: 'pm-app',
     template: `
       <div><h1>{{pageTitle}}</h1>
1        <pm-products></pm-products>
       </div>`,
2    directives: [ProductListComponent]
   })
   export class AppComponent { }
```

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 true or false value

`*ngFor`

- Mark the local variable with #
- Specify 'of': `'#product of products'`
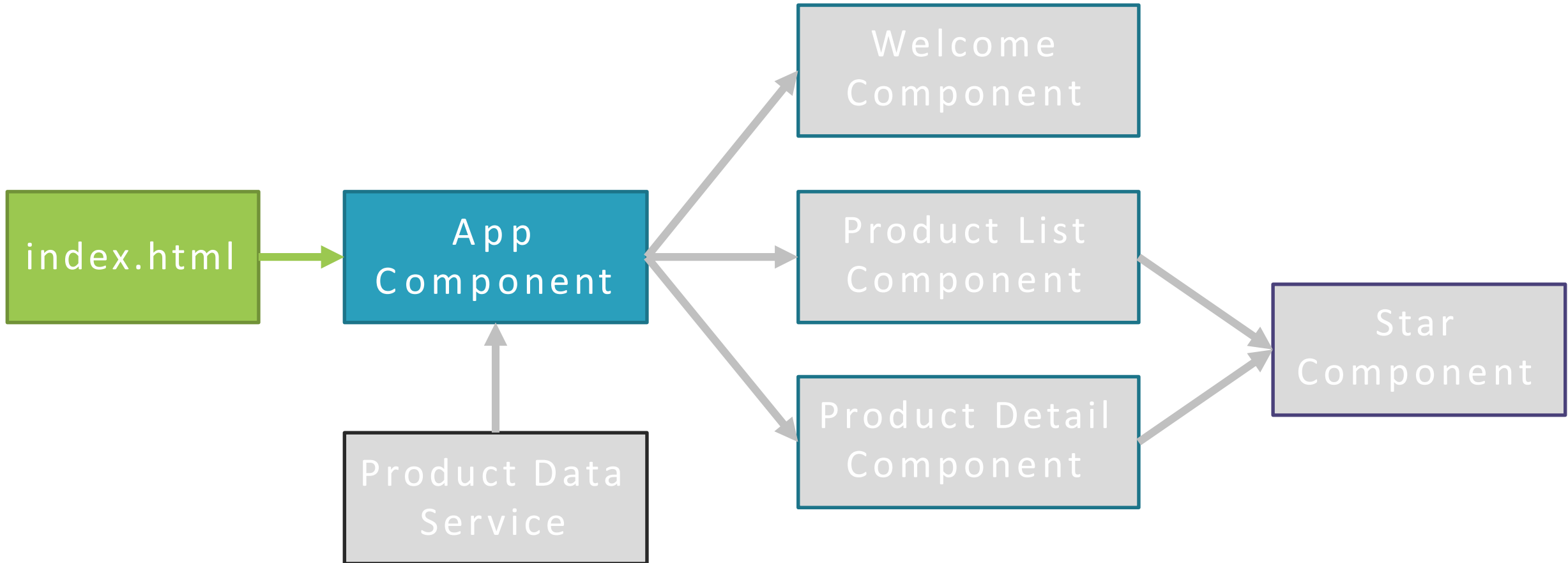
Summary

Building a Template

Using a Component as a Directive

Binding with Interpolation

Adding Logic with Directives

Application Architecture

Application Architecture