

Angular 2: 101

Deborah Kurata

Developer | Author | Consultant
InStep Technologies, Inc

Level: Beginner



Deborah Kurata

- Developer
- Angular 2 Documentation Team Member
- Pluralsight Author
 - Angular 2: Getting Started
 - Angular with TypeScript
 - Angular Front to Back with Web API
 - C# Best Practices
- Microsoft MVP

<https://github.com/DeborahK/VSLive2016-LV-101>

Rate Yourself on Angular 2

- Tried out Angular 2 - **Limited** proficiency
- New to Angular 2 - **Angular 1** proficiency
- New to Angular - **no** proficiency

Overview



Introduction to Angular 2

Components

Templates

Composable Building Blocks

Services

Routing

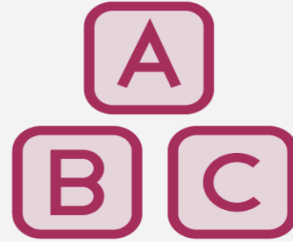
Why Angular 2?



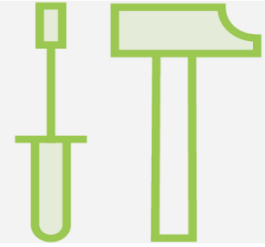
Built for
Speed



Modern



Simplified
API



Enhances
Productivity

Languages

ES 5

- Runs in the browser
- No compile required

ES 2015 (ES 6)

- Lots of new features (classes, let, fat arrow, etc.)

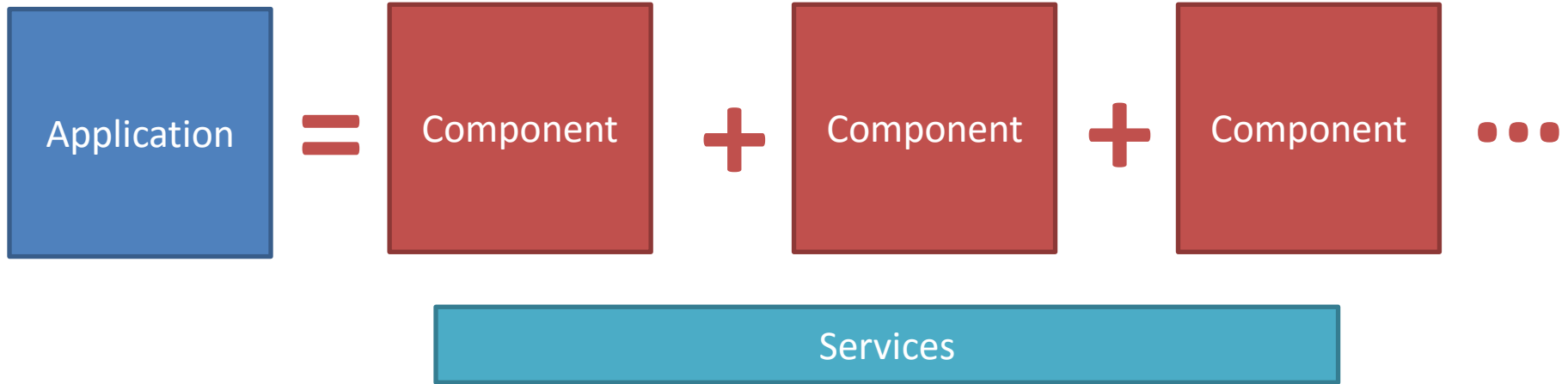
TypeScript*

- Superset of JavaScript
- Strong typing
- Great IDE tooling

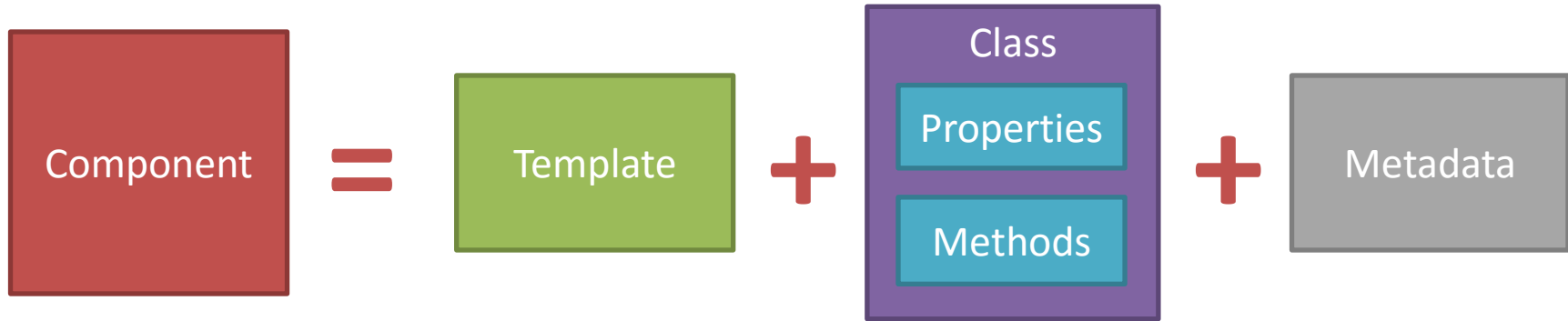
Dart

- No JavaScript

Angular 2 Application



Component

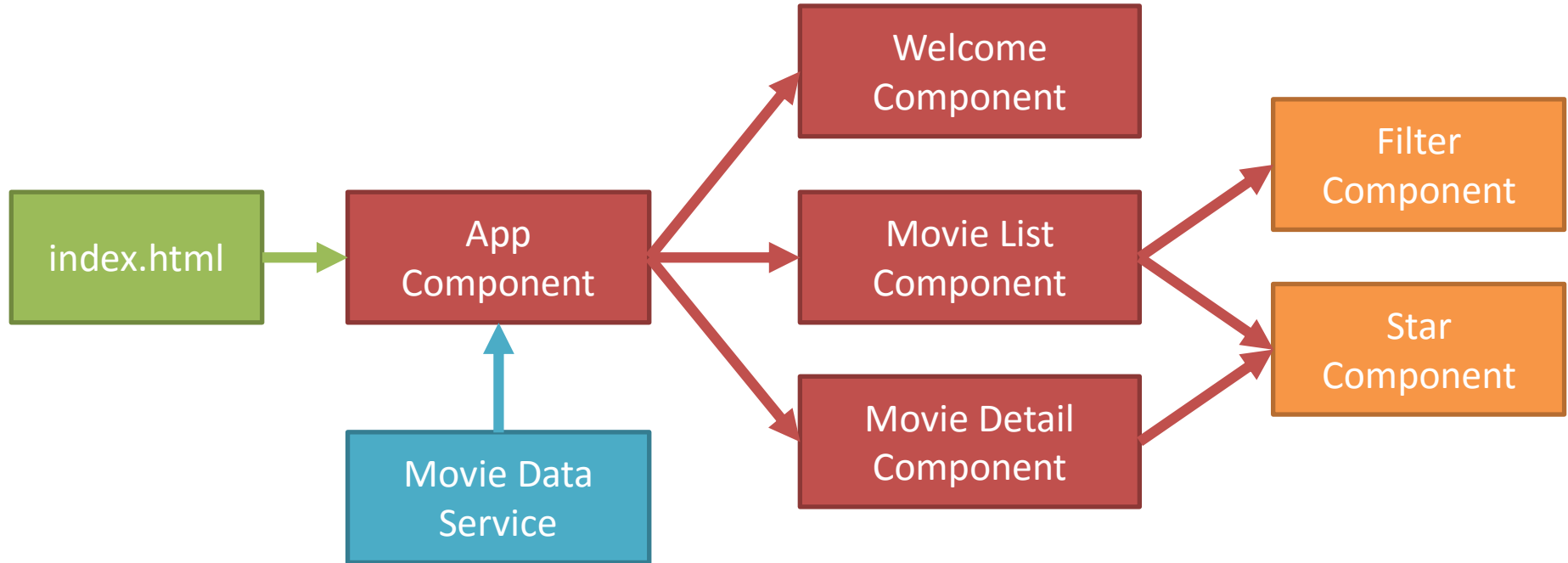


Demo



MovieHunter Sample Application

Sample Application Architecture



Component

app.component.ts

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

ES 2015
Import Syntax

```
@Component({  
  selector: 'mh-app',  
  template: `  
    <div><h1>{{pageTitle}}</h1>  
      <div>My First Component</div>  
    </div>  
  `
```

Metadata:
Component
Decorator

```
  })  
  export class AppComponent {  
    pageTitle: string = "InStep Movie Hunter";  
  }
```

ES 2015
Class Syntax

Angular 2 is Modular

angular2/
core

angular2/
animate

angular2/
http

angular2/
router

Using a Component

index.html

```
<body>  
  <mh-app>Loading App ...</mh-app>  
</body>
```

app.component.ts

```
import {Component} from 'angular2/core';  
  
@Component({  
  selector: 'mh-app',  
  template:  
    <div><h1>{{pageTitle}}</h1>  
      <div>My First Component</div>  
    </div>  
})  
export class AppComponent {  
  pageTitle: string = "InStep Movie Hunter";  
}
```

Demo



MovieHunter Take 1: The Basics

Template

app.component.ts

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

```
@Component({  
  selector: 'mh-app',  
  template: `  
    <div><h1>{{pageTitle}}</h1>  
      <div>My First Component</div>  
    </div>  
  `,  
})  
export class AppComponent {  
  pageTitle: string = "InStep Movie Hunter";  
}
```


Template

Simple HTML

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

Multi-line HTML

```
template:  
<div>  
  <h1>{{pageTitle}}</h1>  
  <div class="container">  
    <mh-movies></mh-movies>  
  </div>  
</div>
```

ES 2015
Back Ticks

URL

```
templateUrl:  
"app/movies/movie-  
list.component.html"
```

Binding

Binding	Example
Interpolation model -> view	<code>{{ movie.mpaa uppercase }}</code>

[()] -> Banana in a Box

Directives

Structural Directives

```
<table *ngIf="movies.length">
```

```
<tr *ngFor="#movie of movies">
```

#

Local Variable

Custom Directives

- Defined by our components

Styling

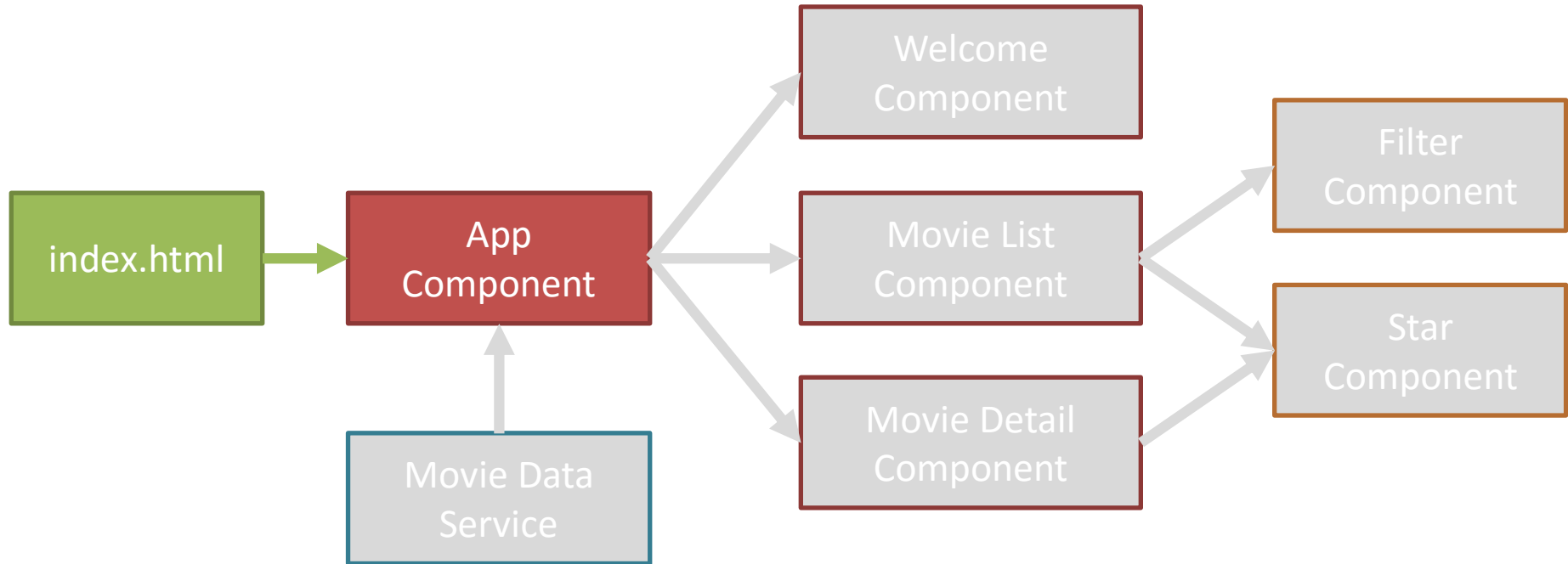
The application

```
<link href="app/app.component.css"  
      rel="stylesheet" />
```

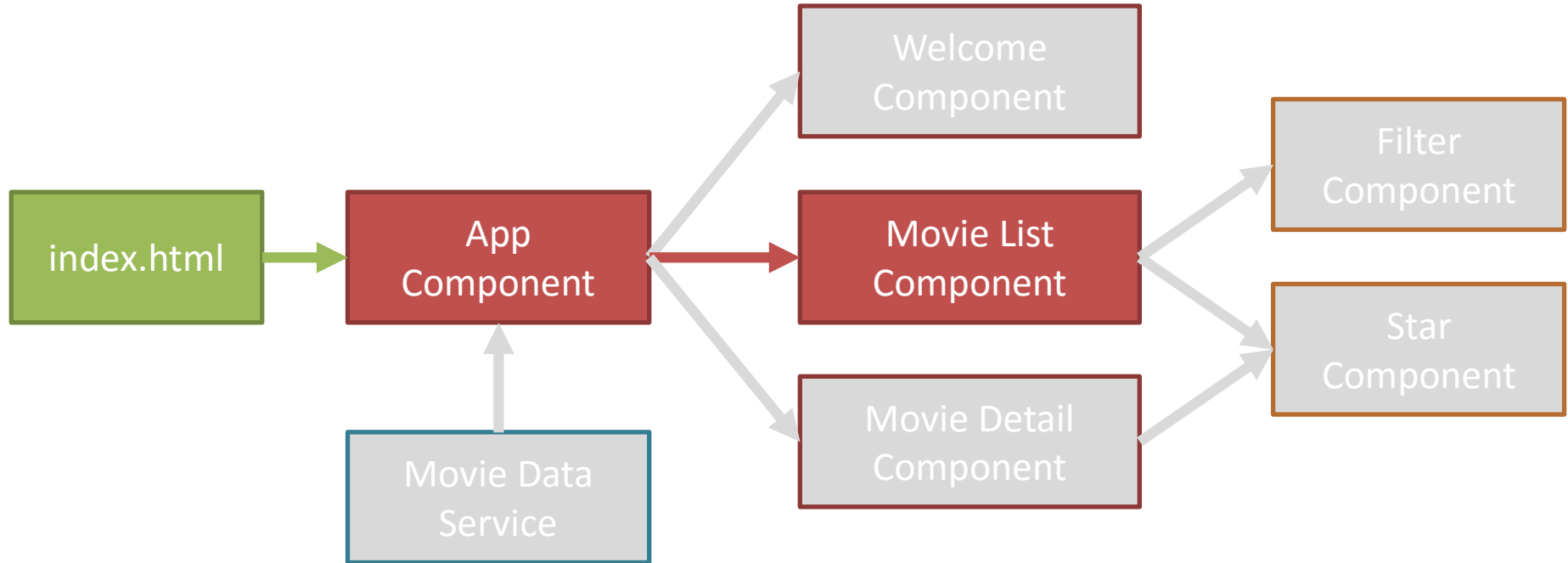
The component

```
@Component({  
  selector: 'mh-movies',  
  templateUrl: 'app/movies/  
               movie-list.component.html',  
  styleUrls: ['app/movies/  
             movie-list.component.css']})
```

Sample Application Architecture



Sample Application Architecture



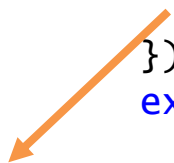
Composable Building Blocks

app.component.ts

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

movie-list.component.ts

```
@Component({
  selector: 'mh-movies',
  templateUrl: "app/movies/movie-list.component.html"
})
export class MovieListComponent { }
```



Composable Building Blocks

app.component.ts

1

```
@Component({
  selector: 'mh-app',
  template: `
    <div><h1>{{pageTitle}}</h1>
    <mh-movies></mh-movies>
  </div>

  `
})
export class AppComponent { }
```

movie-list.component.ts

```
@Component({
  selector: 'mh-movies',
  templateUrl: "app/movies/movie-list.component.html"
})
export class MovieListComponent { }
```



Composable Building Blocks

app.component.ts

```
1
2
@Component({
  selector: 'mh-app',
  template: `
    <div><h1>{{pageTitle}}</h1>
      <mh-movies></mh-movies>
    </div>
  `,
  directives: [MovieListComponent]
})
export class AppComponent { }
```

movie-list.component.ts

```
@Component({
  selector: 'mh-movies',
  templateUrl: "app/movies/movie-list.component.html"
})
export class MovieListComponent { }
```

Composable Building Blocks

app.component.ts

movie-list.component.ts

3

```
import {MovieListComponent}
  from './movies/movie-list.component';
```

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

1

2

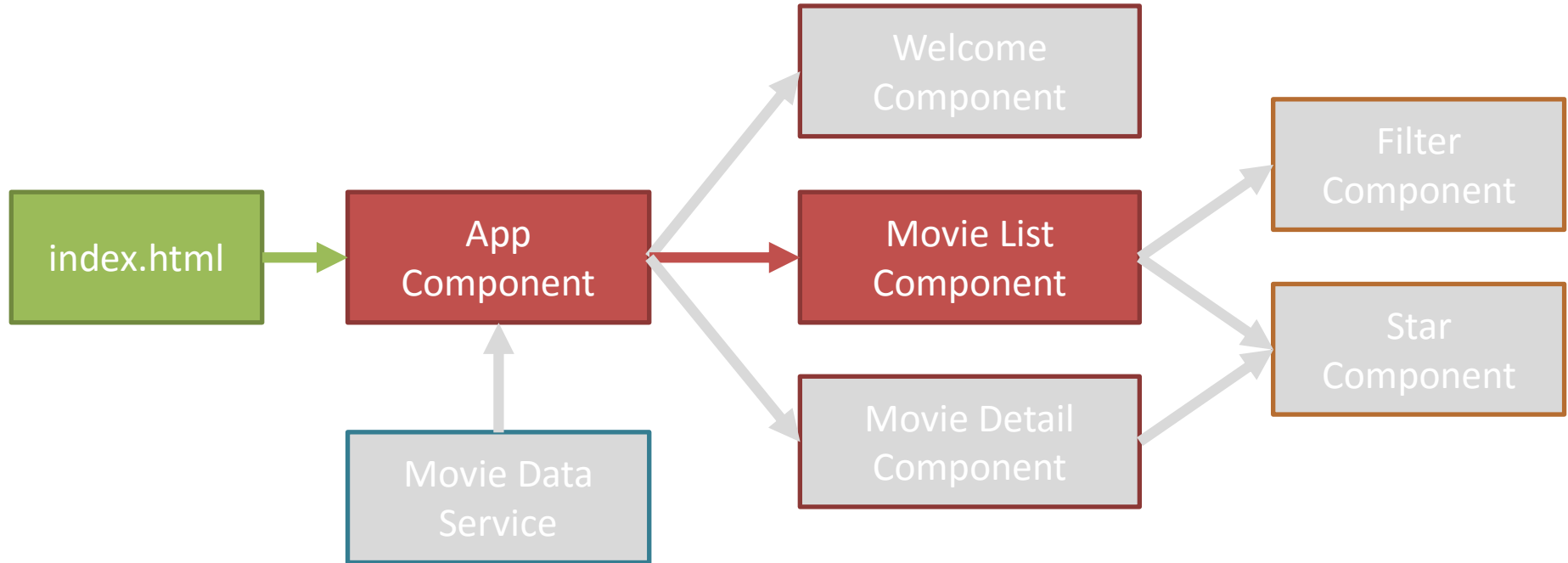
```
@Component({
  selector: 'mh-movies',
  templateUrl: "app/movies/movie-list.component.html"
})
export class MovieListComponent { }
```

Demo

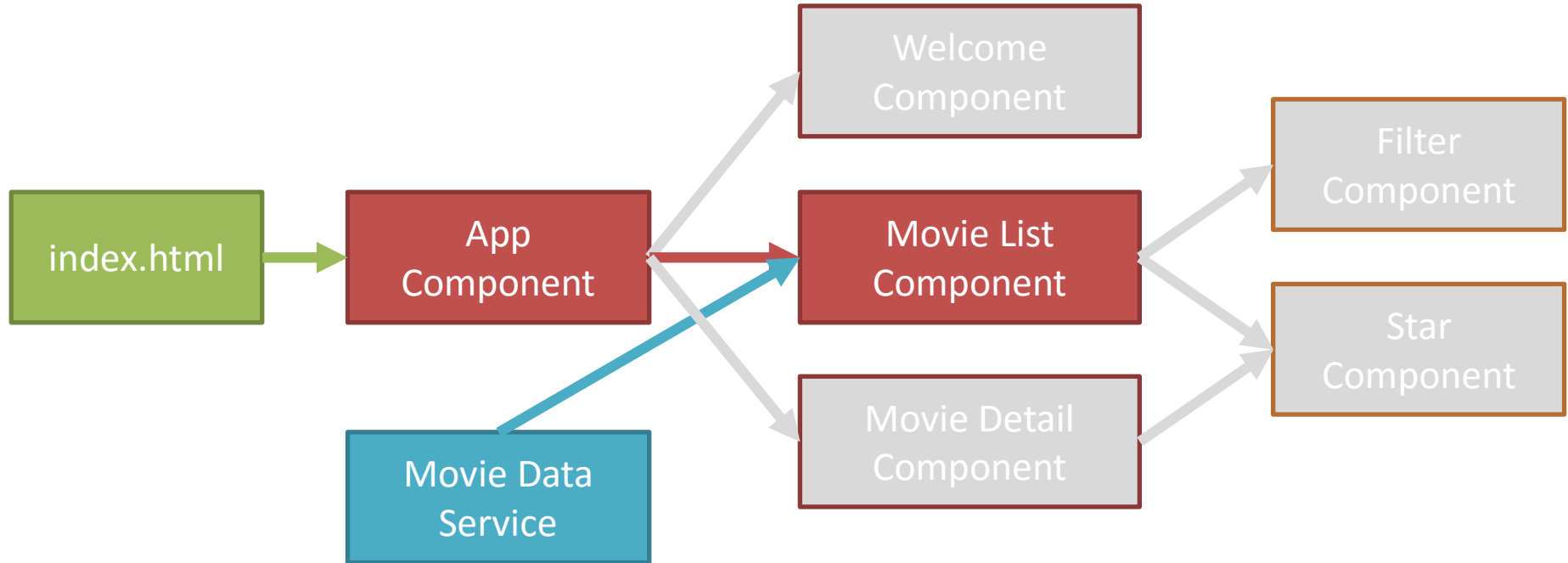


MovieHunter Take 2: Templates & Components

Sample Application Architecture



Sample Application Architecture



Services

Provide
functionality
across
components

Examples:

- Logging
- Calculations
- Data access
- ...

Injected into
any
component
that needs it

Singletons

Services

movie-list.component.ts

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

@Component({
  selector: 'mh-movies',
  templateUrl: "app/movies/movie-list.component.html"
})
export class MovieListComponent {
  constructor() { }
}
```

movie.service.ts

```
import {Injectable}
      from 'angular2/core';

@Injectable()
export class MovieService {
  getMovies() { }
}
```

Services

movie-list.component.ts

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

@Component({
  selector: 'mh-movies',
  templateUrl: "app/movies/movie-list.component.html",
  providers: [MovieService]
})
export class MovieListComponent {
  constructor() { }
}
```

1

movie.service.ts

```
import {Injectable}
      from 'angular2/core';

@Injectable()
export class MovieService {
  getMovies() { }
}
```


Services

movie-list.component.ts

```
2 import {Component} from 'angular2/core';  
import {MovieService} from "../movie.service";  
  
@Component({  
  selector: 'mh-movies',  
  templateUrl: "app/movies/movie-list.component.html",  
  providers: [MovieService]  
})  
1 export class MovieListComponent {  
  constructor() { }  
}
```

movie.service.ts

```
import {Injectable}  
      from 'angular2/core';  
  
@Injectable()  
export class MovieService {  
  getMovies() { }  
}
```

Services

movie-list.component.ts

```
2 import {Component} from 'angular2/core';
import {MovieService} from "../movie.service";

@Component({
  selector: 'mh-movies',
  templateUrl: "app/movies/movie-list.component.html",
  providers: [MovieService]
})
1 export class MovieListComponent {
3   constructor(private _movieService: MovieService) { }

  private _movieService: MovieService;
  constructor(movieService: MovieService) {
    this._movieService = movieService;
  }
}
```



movie.service.ts

```
import {Injectable}
  from 'angular2/core';

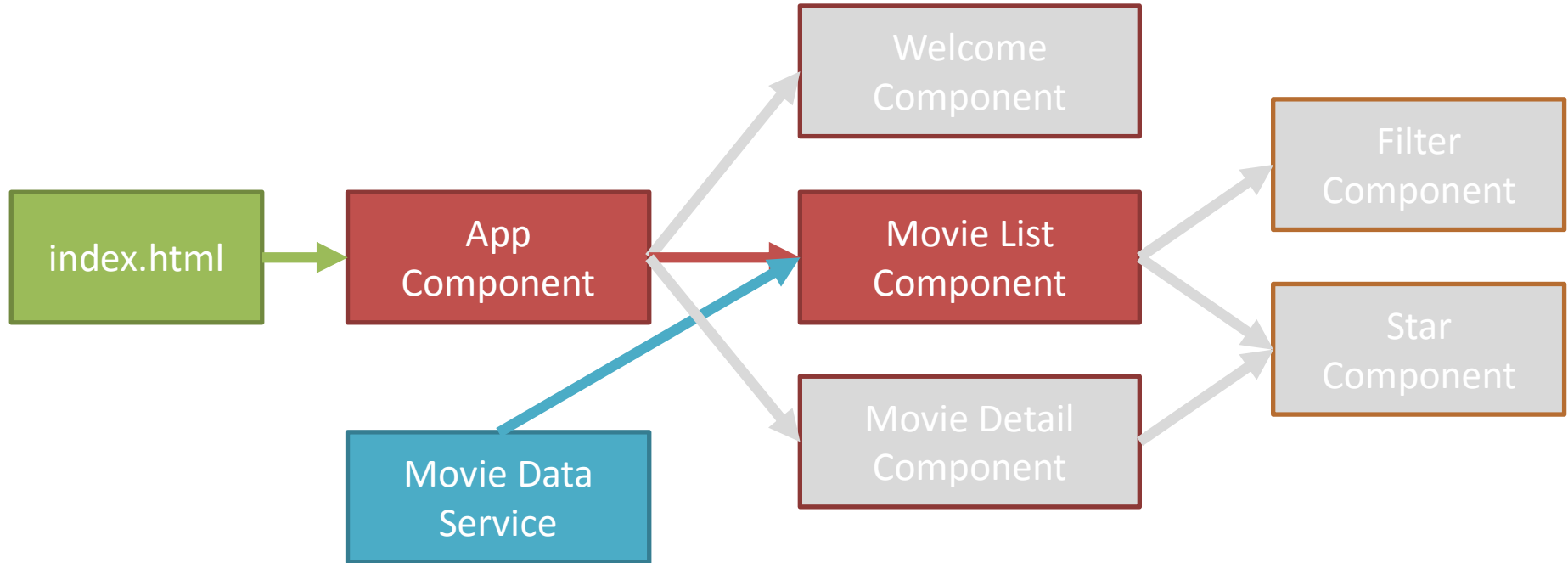
@Injectable()
export class MovieService {
  getMovies() { }
}
```

Demo

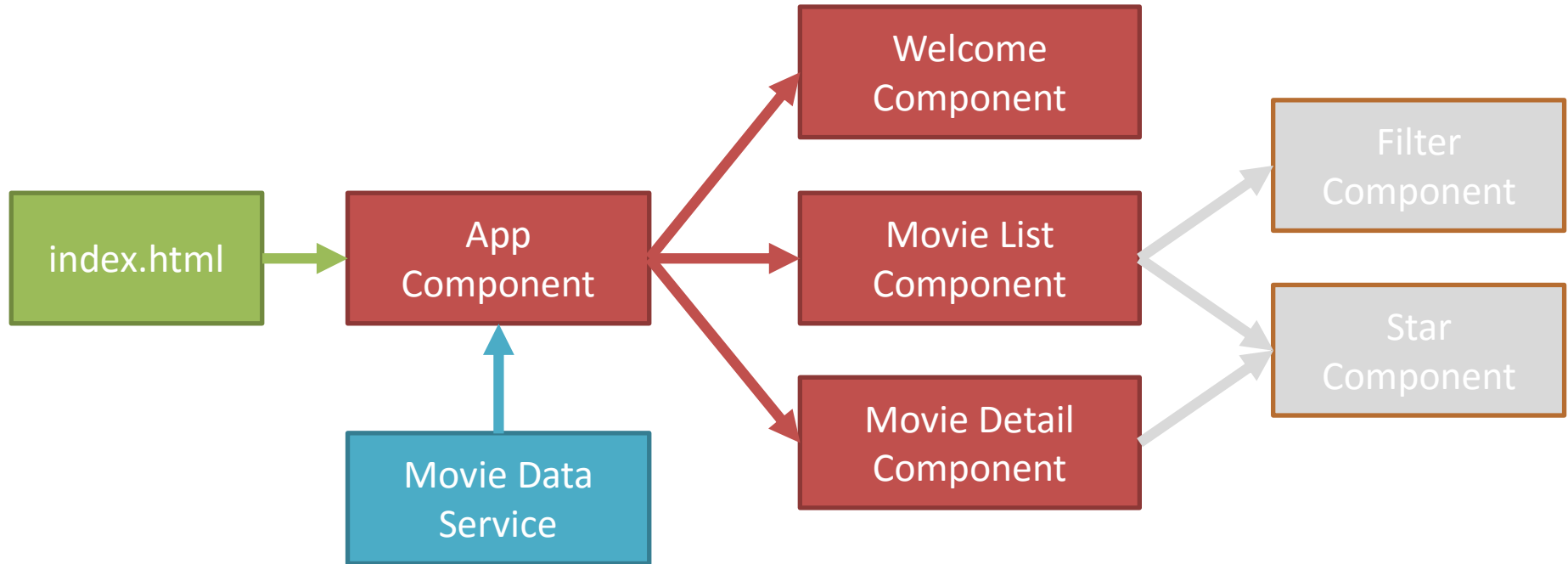


MovieHunter Take 3: Services

Sample Application Architecture



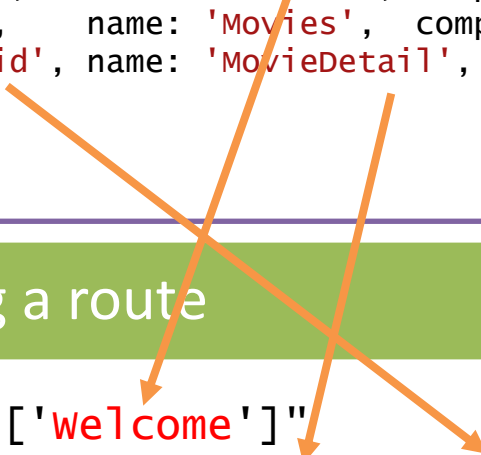
Sample Application Architecture



Routing

Configuration (app.components.ts)

```
@RouteConfig([
  { path: '/welcome',    name: 'welcome', component: welcomeComponent, useAsDefault: true },
  { path: '/movies',    name: 'Movies',  component: MovieListComponent },
  { path: '/movie/:id', name: 'MovieDetail', component: MovieDetailComponent }
])
```



Activating a route

```
[routerLink]="['welcome']"
[routerLink]="['MovieDetail', { id: movie.movieId }]"
```

Router Displays the View

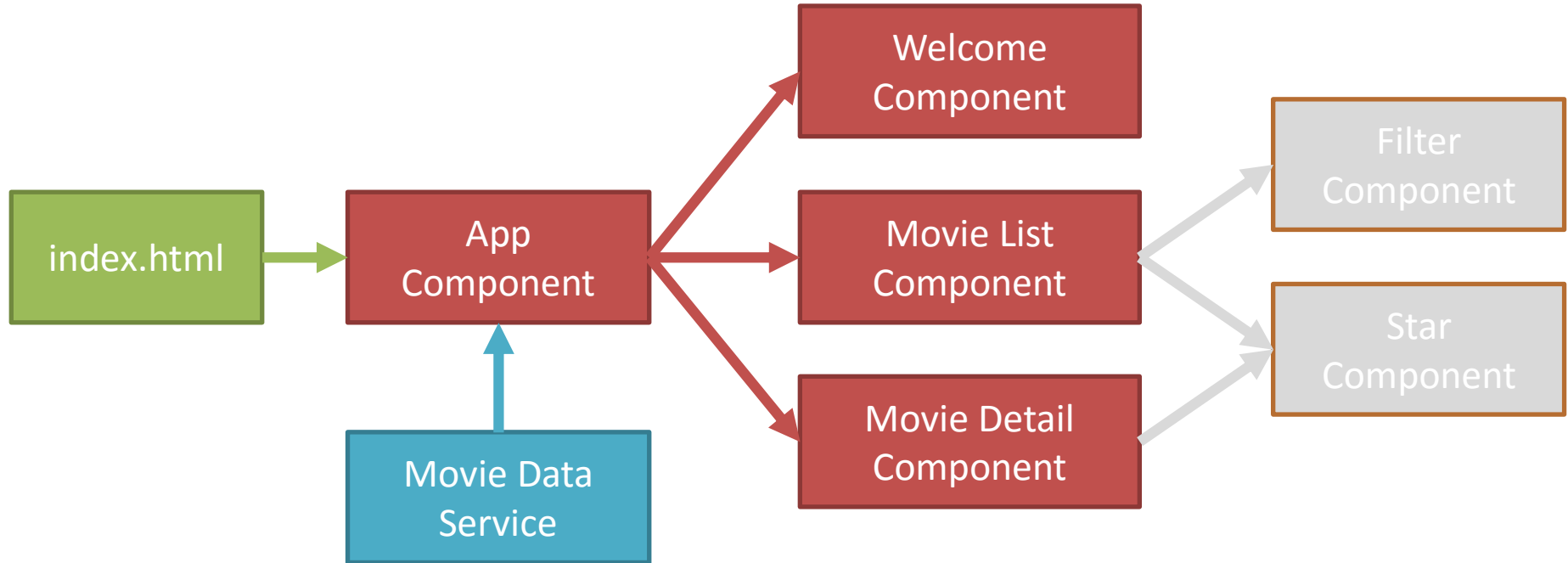
```
<div class="container">  
  <router-outlet></router-outlet>  
</div>
```

Demo



MovieHunter Take 4: Routing

Sample Application Architecture





Moving from A1 to A2

ngUpgrade

- Run A1 and A2 side by side
- Migrate pieces of an app from A1 to A2

Or don't

- A1 will be supported for while
- Wait for that next project

Take Away

Components

Templates

Composable
Blocks

Services

Routing

Thank You!

@deborahkurata

deborahk@insteptech.com

<http://blogs.msmvps.com/deborahk>

<https://github.com/DeborahK/VSLive2016-LV-101>