

# Angular Modules

---

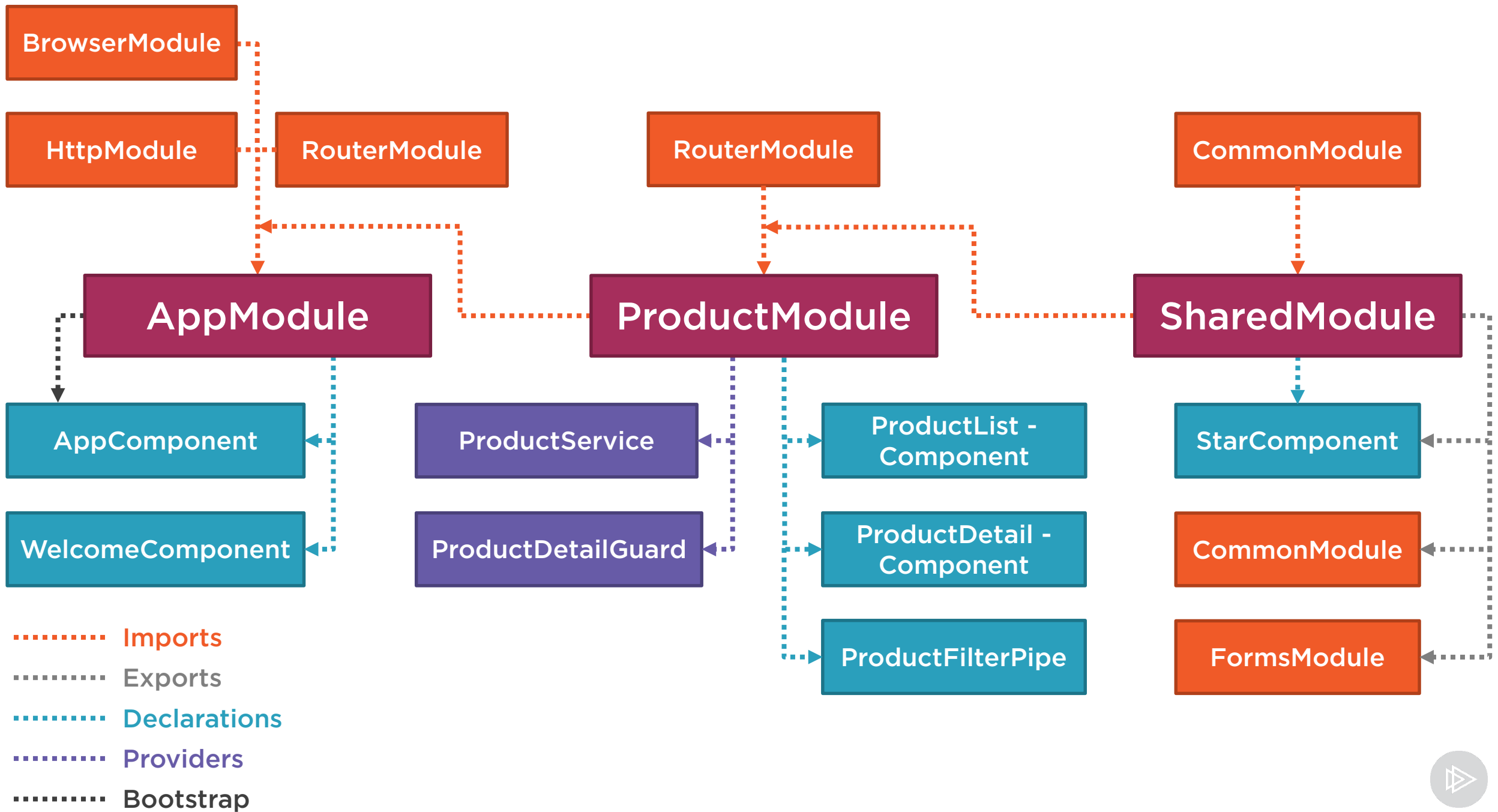


**Deborah Kurata**

CONSULTANT | SPEAKER | AUTHOR

@deborahkurata | [blogs.msmvps.com/deborahk/](https://blogs.msmvps.com/deborahk/)





# Module Overview



**What Is an Angular Module?**

**Angular Module Metadata**

**Creating a Feature Module**

**Defining a Shared Module**

**Revisiting AppModule**



# What Is an Angular Module?



Module

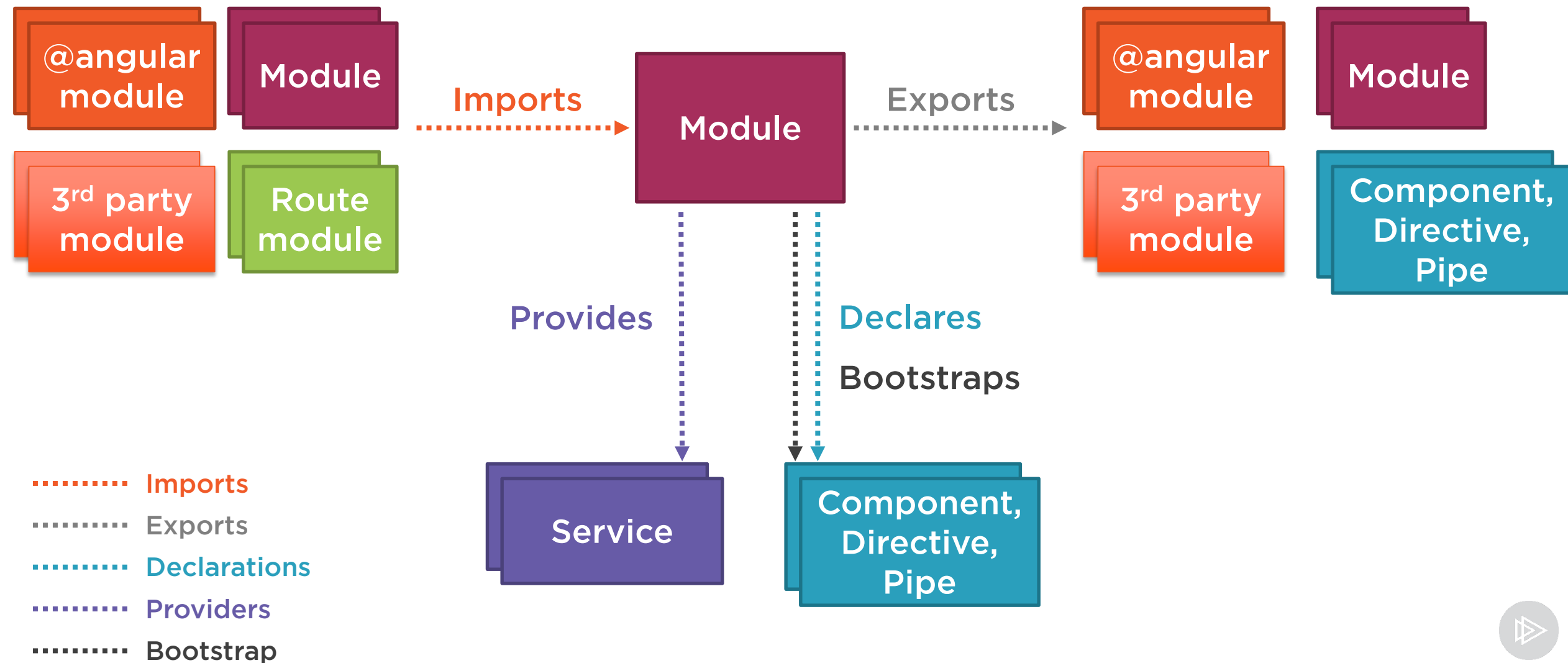
**A class with an NgModule decorator**

**Its purpose:**

- Organize the pieces of our application
- Arrange them into blocks
- Extend our application with capabilities from external libraries
- Provide a template resolution environment
- Aggregate and re-export



# Angular Module



AppComponent

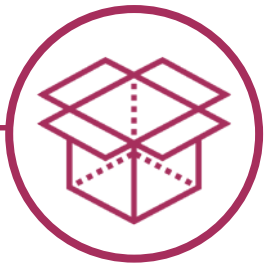
ProductList -  
Component

ProductDetail -  
Component

WelcomeComponent

```
...  
<li> <a [routerLink]="['/welcome']">  
  Home</a> </li>  
<li> <a [routerLink]="['/products']">  
  Product List</a> </li>  
...  
<router-outlet> </router-outlet>  
...
```

RouterModule



Angular Module



AppComponent

ProductList -  
Component

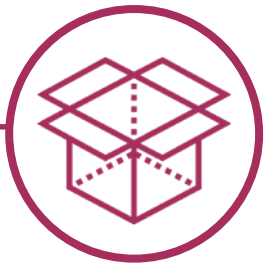
ProductDetail -  
Component

WelcomeComponent

RouterModule

Module

```
...  
<input type='text'  
      [(ngModel)]='listFilter' />  
...
```



Angular Module



AppComponent

ProductList -  
Component

ProductDetail -  
Component

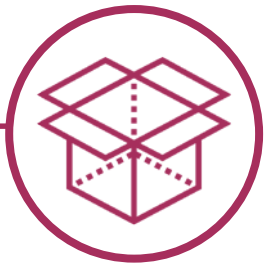
WelcomeComponent

RouterModule

sModule

erModule

```
...  
<tr *ngFor='let product of products |  
    productFilter:listFilter'>  
  <td>  
    <img *ngIf='showImage'  
  ...
```



# Angular Module





AppComponent

ProductFilterPipe

RouterModule

ProductList -  
Component

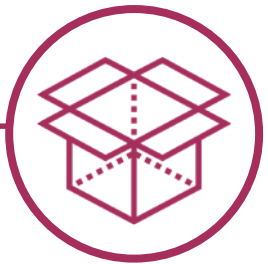
ProductDetail -  
Component

WelcomeComponent

```
...  
<tr *ngFor='let product of products |  
    productFilter:listFilter'>  
  <td>  
    <img *ngIf='showImage'  
  ...
```

RouterModule

RouterModule



# Angular Module



AppComponent

ProductList -  
Component

ProductDetail -  
Component

WelcomeComponent

ProductFilterPipe

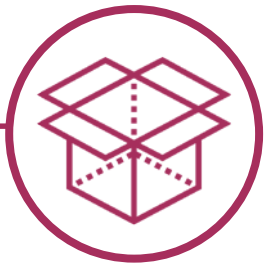
RouterModule

StarComponent

Module

Module

```
...  
<ai-star  
  [rating]='product.starRating'  
  (ratingClicked)='onRatingClicked($event)'  
</ai-star>  
...
```



# Angular Module



AppComponent

ProductList -  
Component

ProductDetail -  
Component

WelcomeComponent

ProductFilterPipe

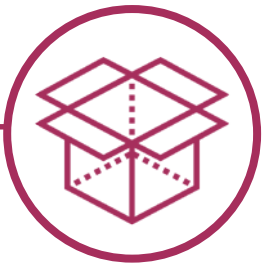
StarComponent

RouterModule

FormsModule

BrowserModule

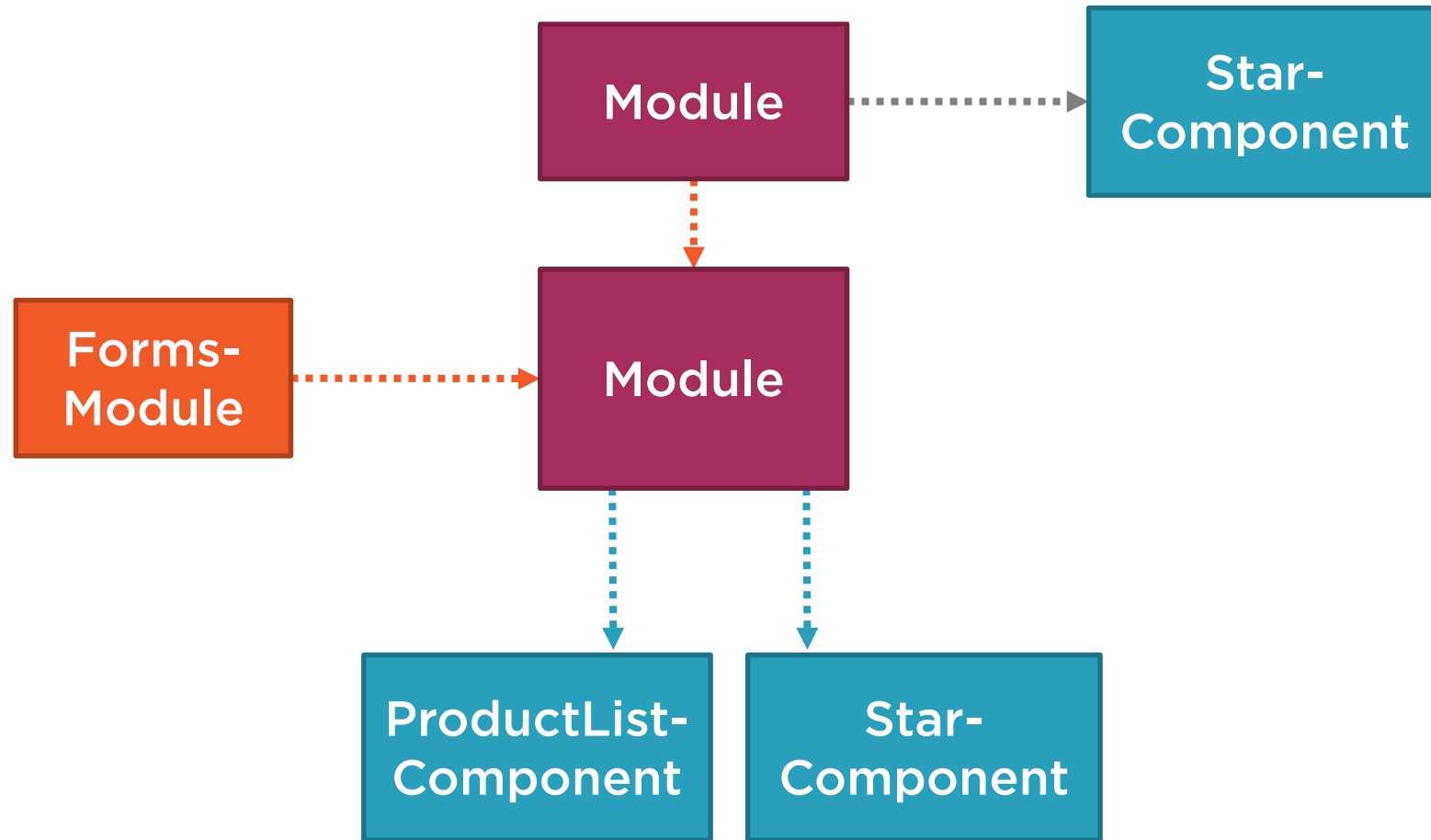
HttpModule



Angular Module



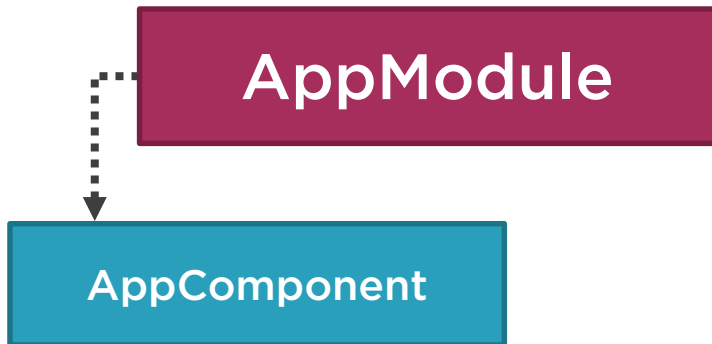
# Template Resolution Environment



- ..... Imports
- ..... Exports
- ..... Declarations
- ..... Providers
- ..... Bootstrap



# Bootstrap Array



**app.module.ts**

```
...  
bootstrap: [ AppComponent ]  
...
```

# Bootstrap Array Truth #1

Every application must bootstrap at least one component, the root application component.

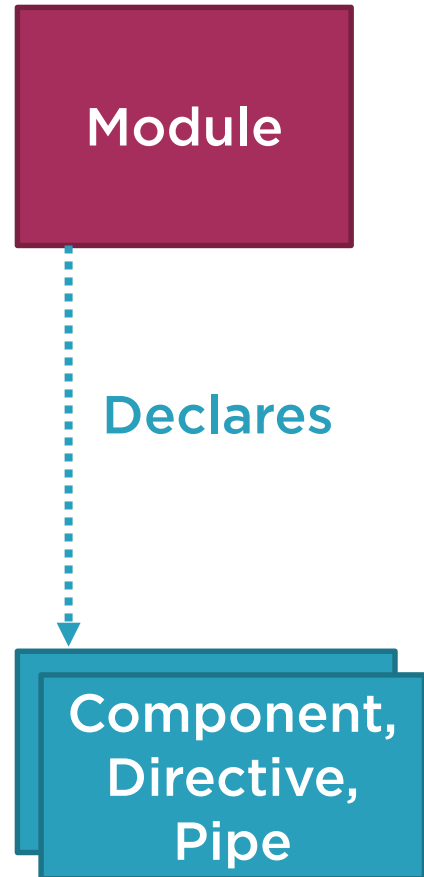


# Bootstrap Array Truth #2

The bootstrap array should only be used in the root application module, AppModule.



# Declarations Array



**app.module.ts**

```
...  
declarations: [  
  AppComponent,  
  WelcomeComponent,  
  ProductListComponent,  
  ProductDetailComponent,  
  ProductFilterPipe,  
  StarComponent  
]  
...
```

..... Declarations





# Declarations Array Truth #1

Every component, directive, and pipe we create must belong to **one and only one** Angular module.



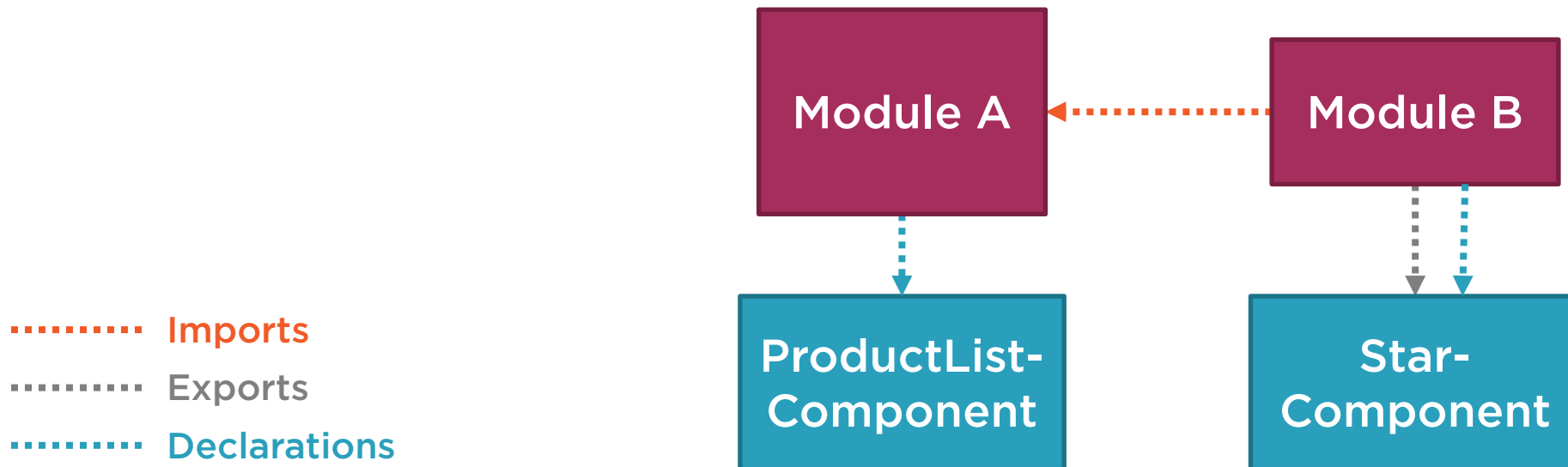
# Declarations Array Truth #2

Only declare components, directives and pipes.



# Declarations Array Truth #3

Never re-declare components, directives, or pipes that belong to another module

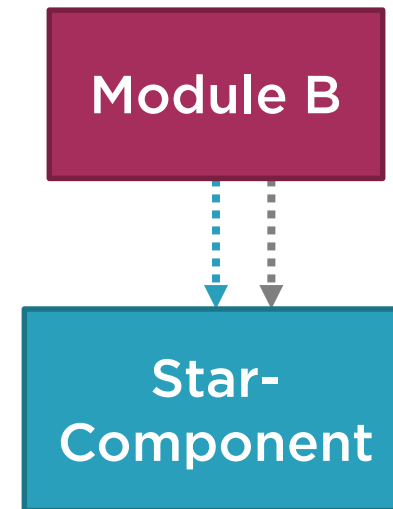


# Declarations Array Truth #4

All declared components, directives, and pipes are private by default.

They are only accessible to other components, directives, and pipes declared in the same module.

..... Exports  
..... Declarations



# Declarations Array Truth #5

The Angular module provides the template resolution environment for its component templates.

product-list.component.html

```
<ai-star ...>  
</ai-star>
```

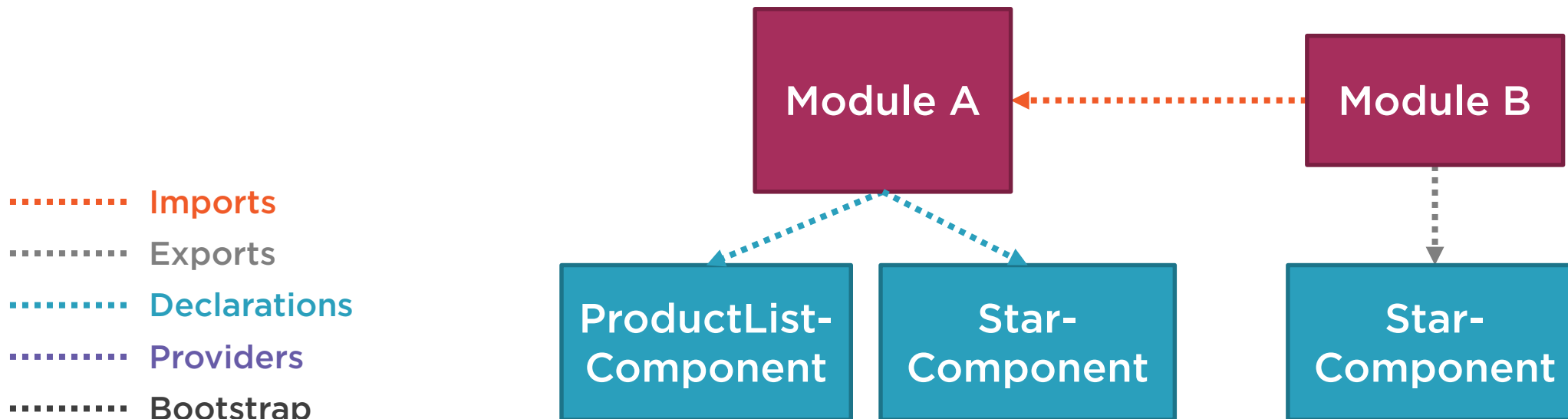
star.component.ts

```
...  
@Component({  
  selector: 'ai-star',  
  template: ...  
})  
...
```

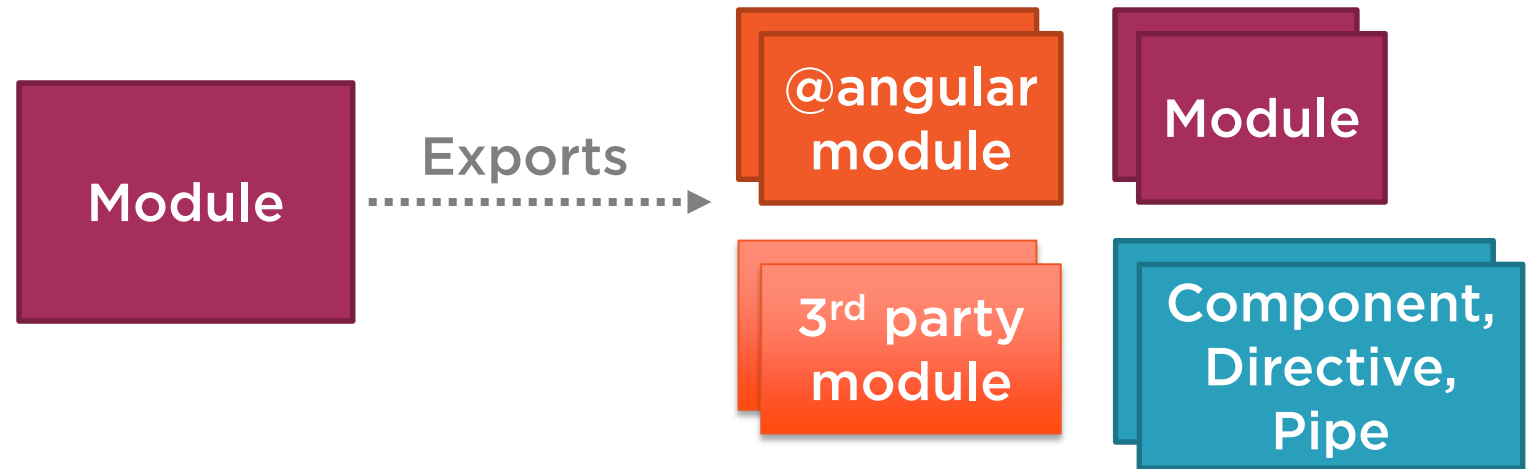


# Declarations Array Truth #5

The Angular module provides the template resolution environment for its component templates.



# Exports Array



- ..... Imports
- ..... Exports
- ..... Declarations
- ..... Providers
- ..... Bootstrap



# Exports Array Truth #1

Export any component, directive, or pipe if another components need it.





# Exports Array Truth #2

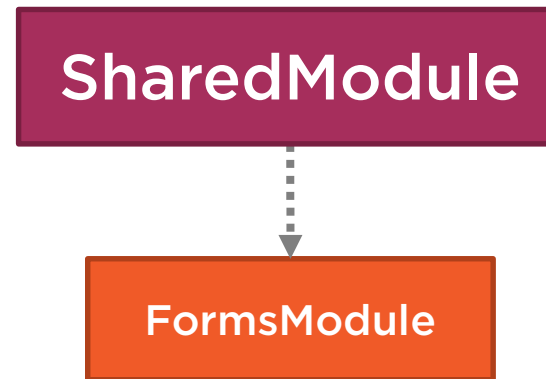
Re-export modules to re-export their components, directives, and pipes.



# Exports Array Truth #3

We can re-export something without importing it first.

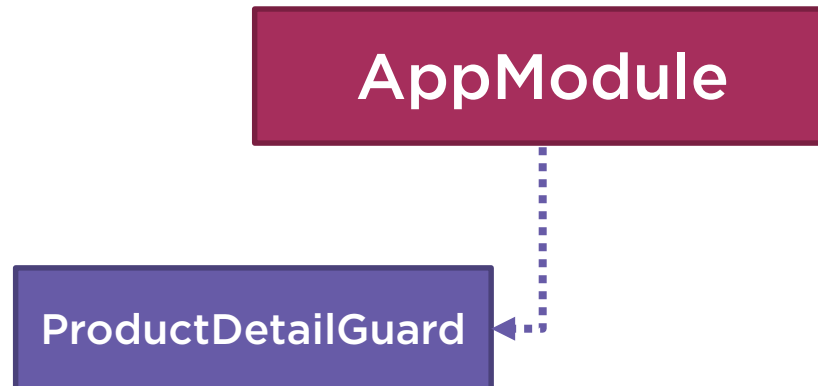
- ..... Imports
- ..... Exports
- ..... Declarations
- ..... Providers
- ..... Bootstrap



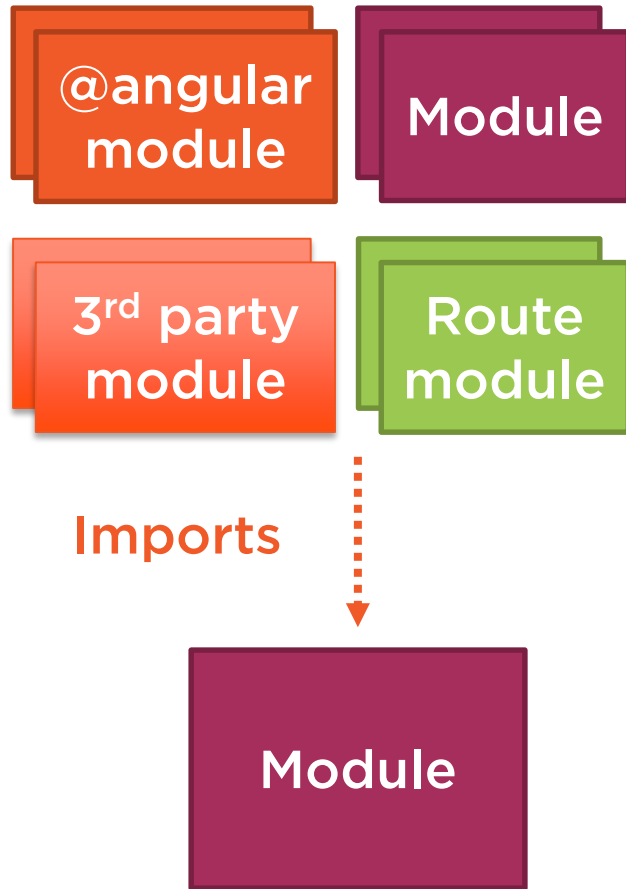
# Exports Array Truth #4

Never export a service.

- ..... Imports
- ..... Exports
- ..... Declarations
- ..... Providers
- ..... Bootstrap



# Imports Array



app.module.ts

```
...
imports: [
  BrowserModule,
  FormsModule,
  HttpClientModule,
  RouterModule.forRoot([...])
]
...
```



# Imports Array Truth #1

Importing a module makes available **any exported** components, directives, and pipes from that module.



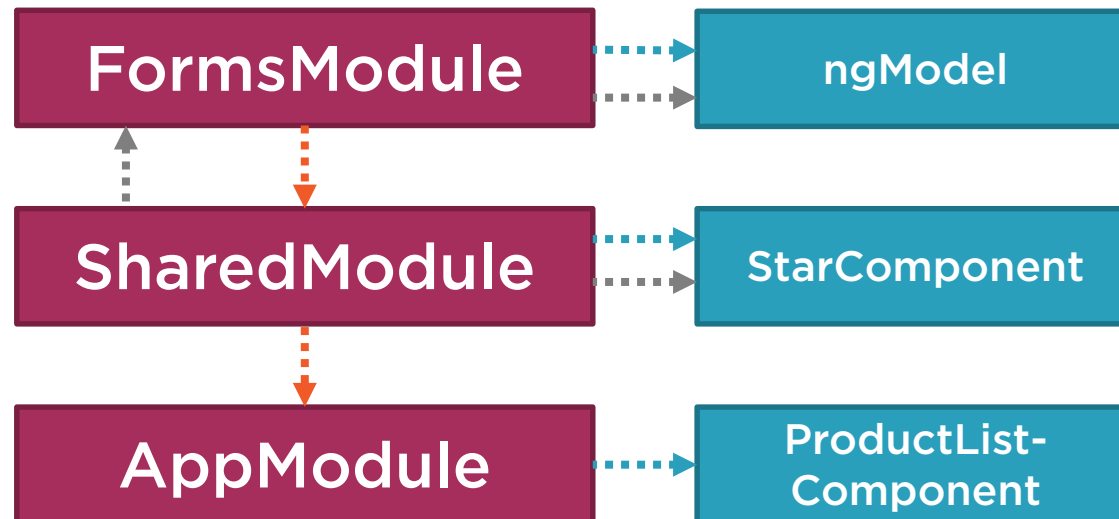
# Imports Array Truth #2

Only import what this module needs.

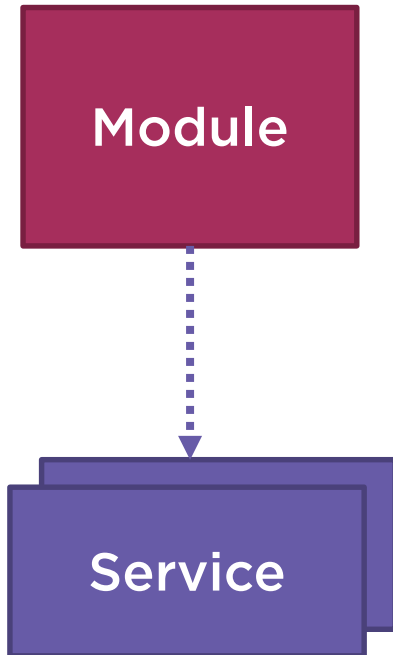


# Imports Array Truth #3

Importing a module does NOT provide access to its imported modules



# Providers Array



**app.module.ts**

```
...  
providers: [ ProductDetailGuard ]  
...
```

..... Imports

..... Exports

..... Declarations

..... Providers

..... Bootstrap

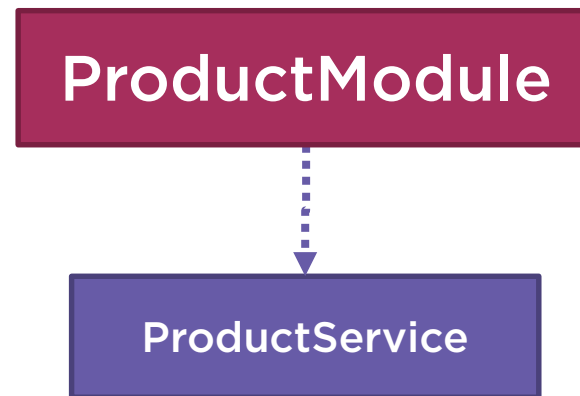




# Providers Array Truth #1

Any service provider added to the providers array is registered at the **root** of the application.

- ..... Imports
- ..... Exports
- ..... Declarations
- ..... Providers
- ..... Bootstrap



## Providers Array Truth #2

Don't add services to the providers array of a shared module.

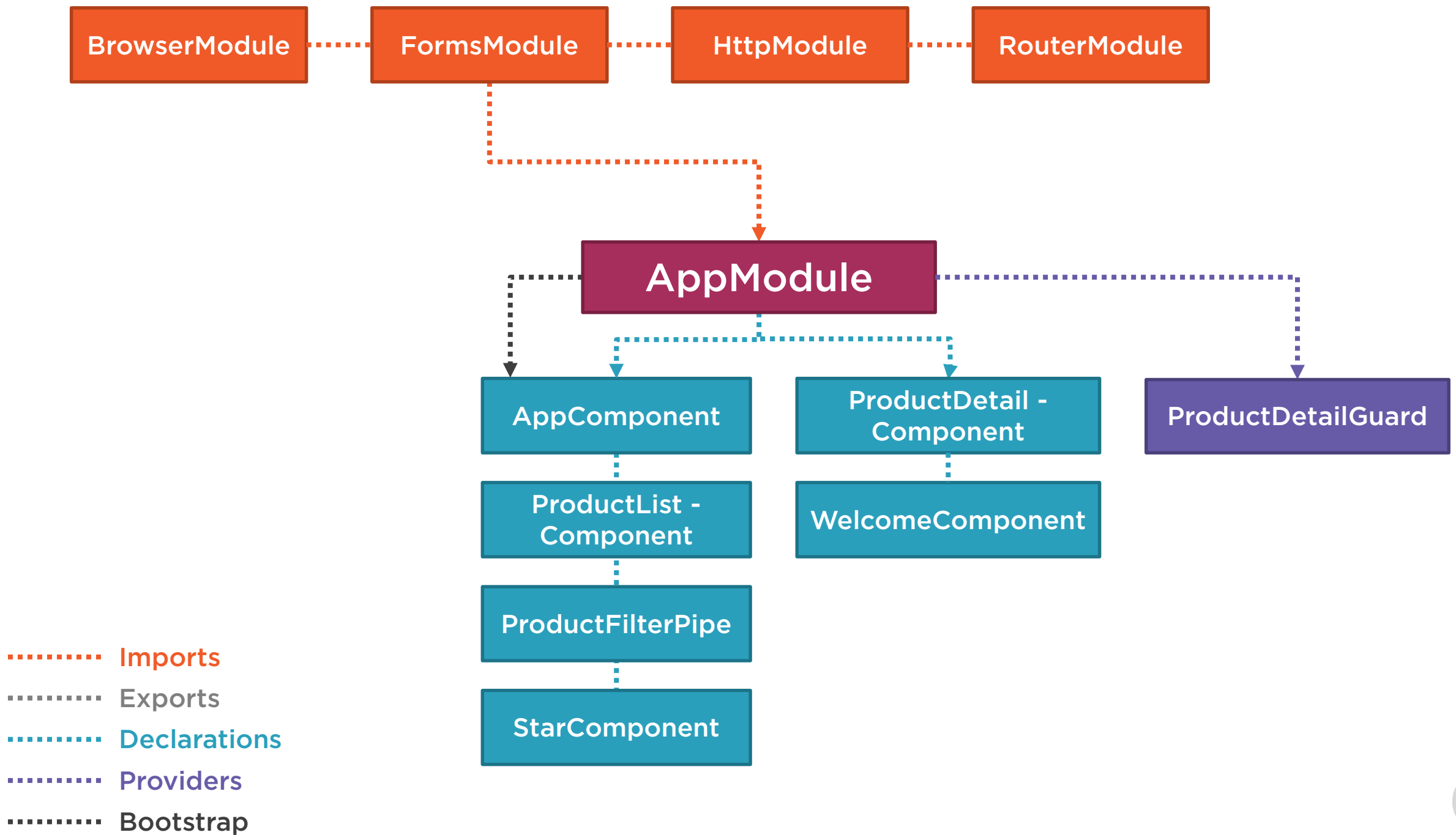
Consider building a CoreModule for services and importing it once in the AppModule.



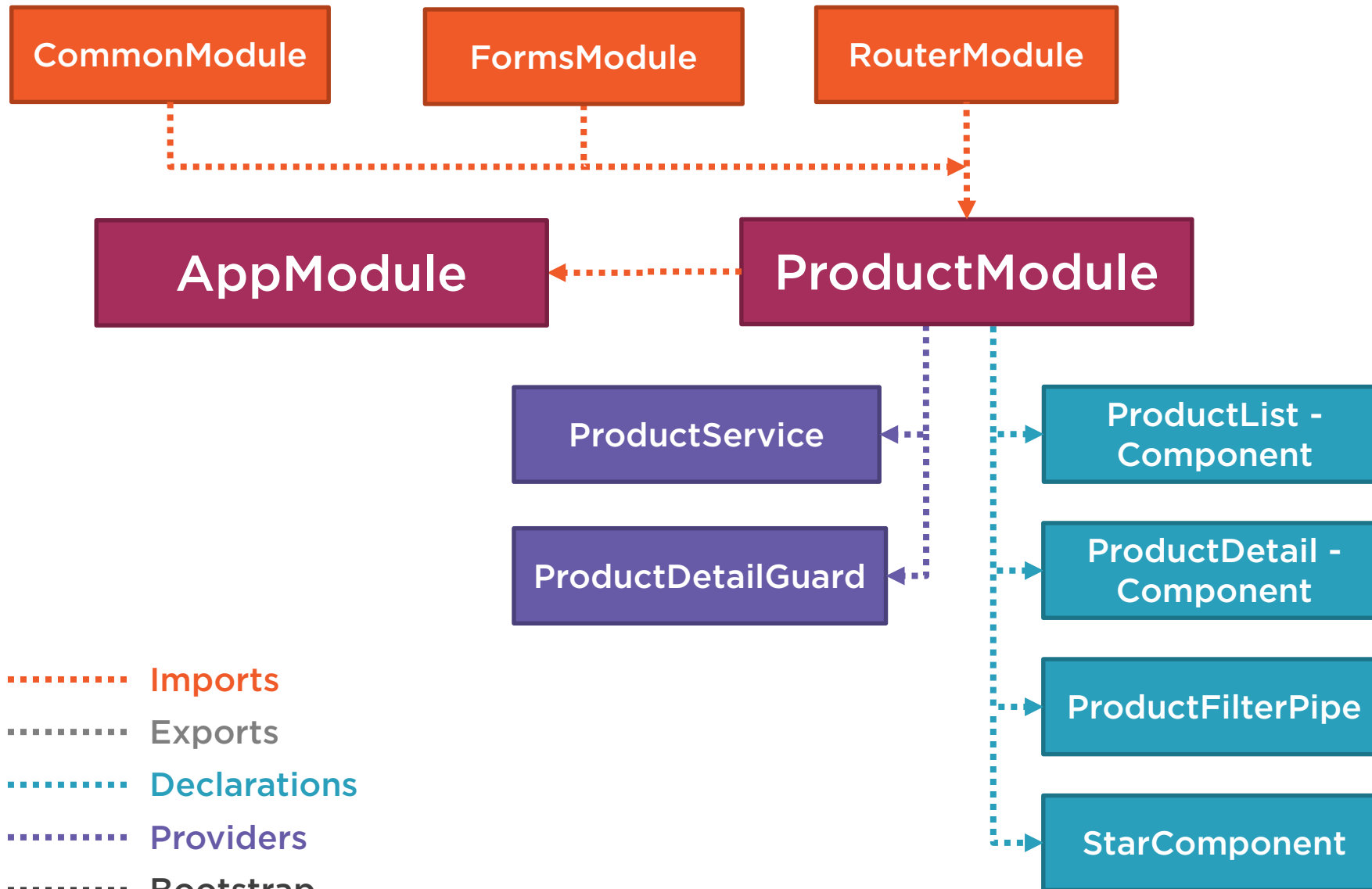
# Providers Array Truth #3

Routing guards must be added to the providers array of an Angular module.

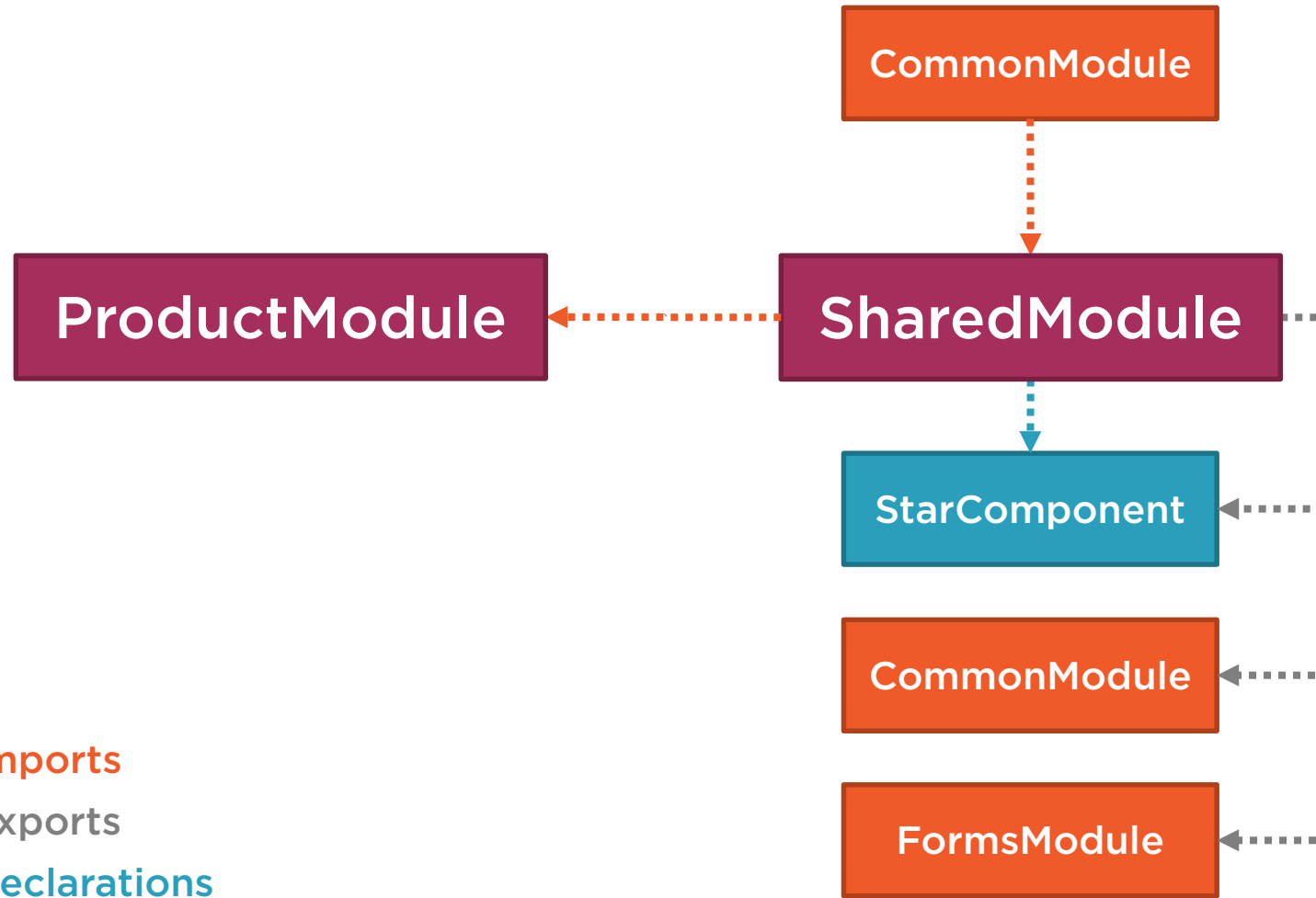




# Defining a Feature Module



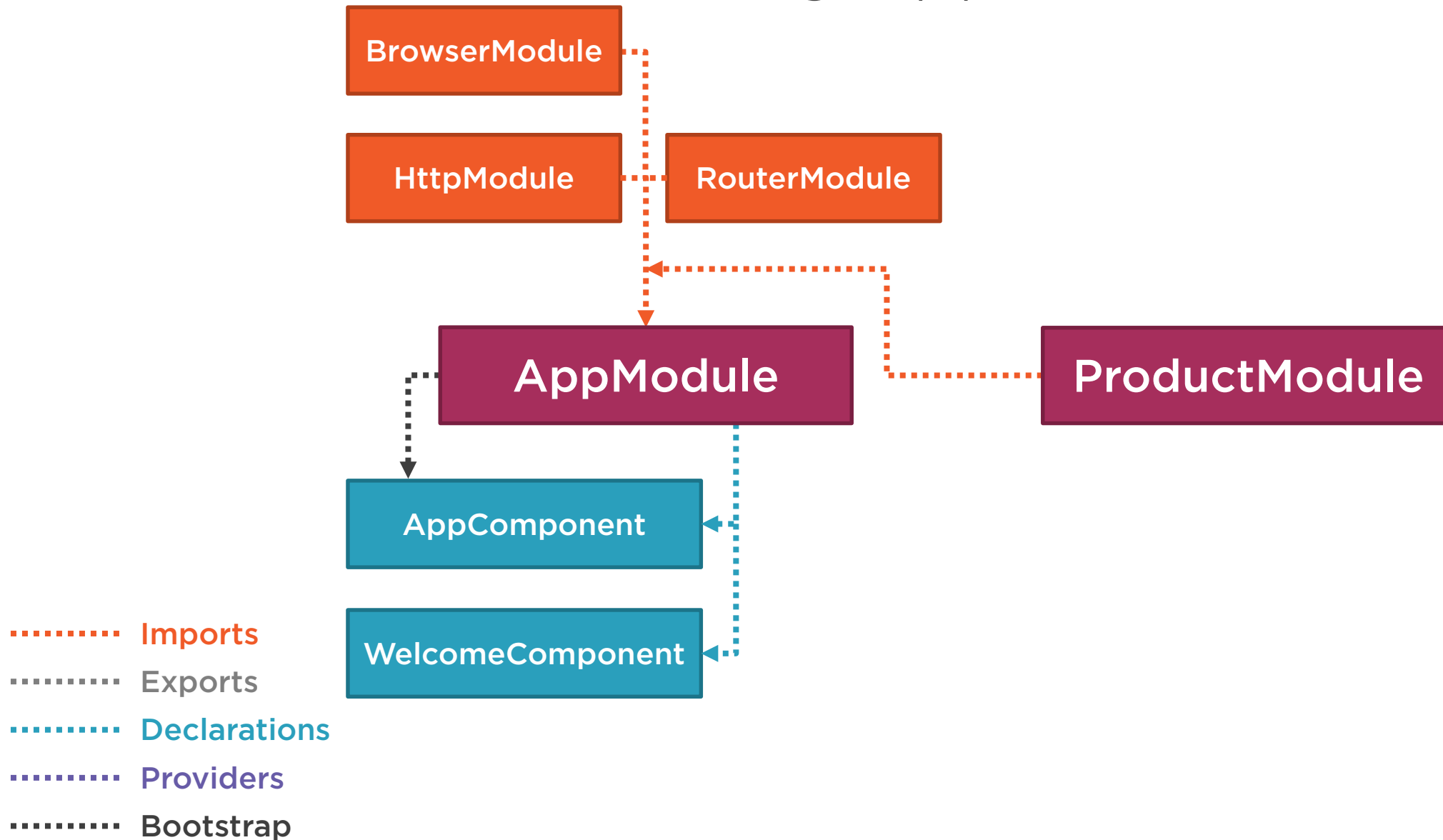
# Defining a Shared Module

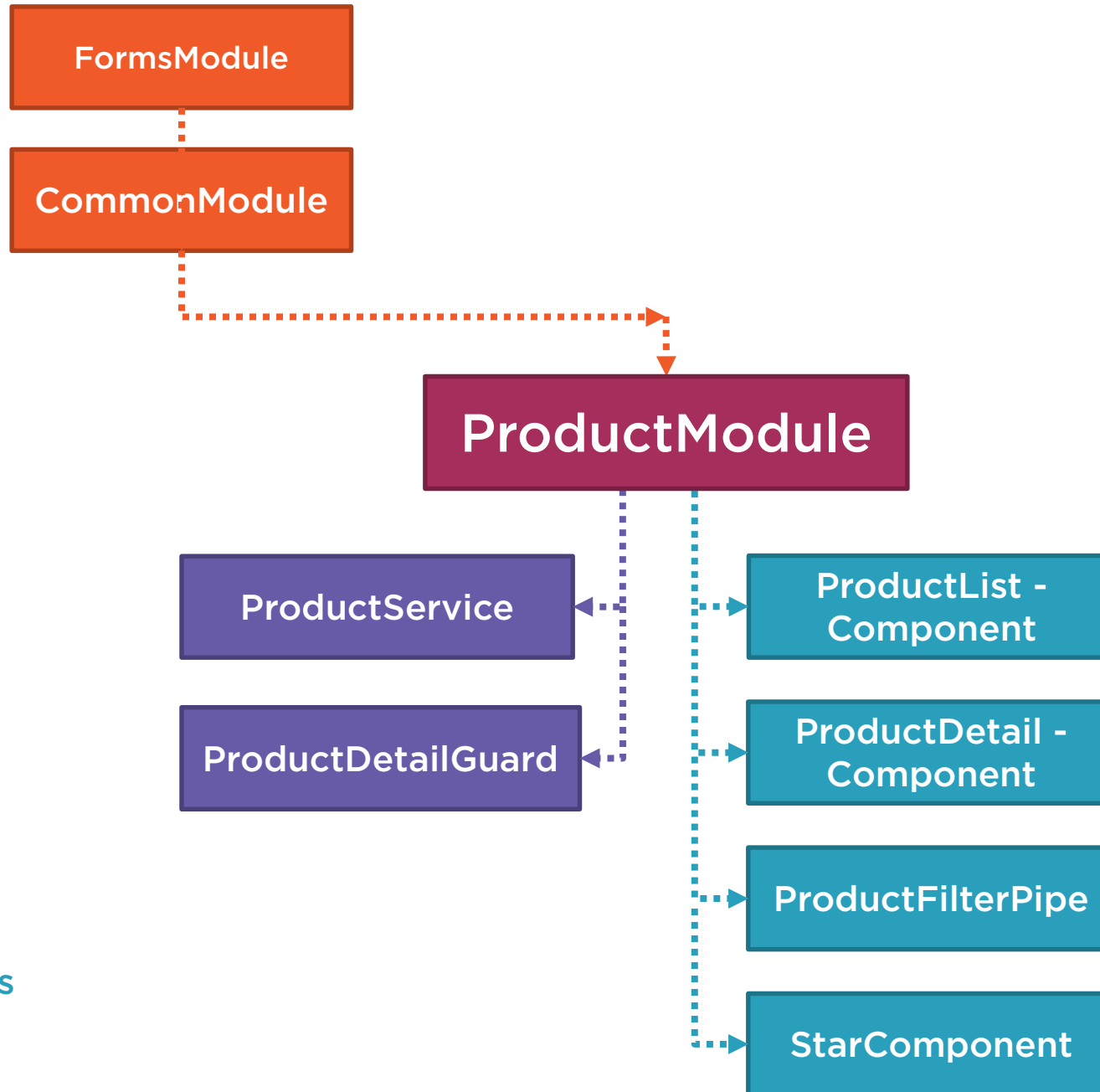


- ..... Imports
- ..... Exports
- ..... Declarations
- ..... Providers
- ..... Bootstrap



# Revisiting AppModule







# Application Routing Module

## app-routing.module.ts

```
import { NgModule } from '@angular/core';
import { RouterModule } from '@angular/router';

import { WelcomeComponent } from './home/welcome.component';

@NgModule({
  imports: [
    RouterModule.forRoot([
      { path: 'welcome', component: WelcomeComponent },
      { path: '', redirectTo: 'welcome', pathMatch: 'full'},
      { path: '**', redirectTo: 'welcome', pathMatch: 'full' }
    ])
  ],
  exports: [ RouterModule ]
})
export class AppRoutingModule { }
```



# Using the Routing Module

**app.module.ts**

```
@NgModule({  
  imports: [  
    BrowserModule,  
    HttpClientModule,  
    ProductModule,  
    AppRoutingModule  
  ],  
  declarations: [ AppComponent, WelcomeComponent ],  
  bootstrap: [ AppComponent ]  
})  
export class AppModule { }
```



# Feature Routing Module

## product-routing.module.ts

```
import { NgModule } from '@angular/core';
import { RouterModule } from '@angular/router';
import { ProductListComponent } from './product-list.component';
import { ProductDetailComponent } from './product-detail.component';
import { ProductDetailGuard } from './product-guard.service';
@NgModule({
  imports: [
    RouterModule.forChild([
      { path: 'products', component: ProductListComponent },
      { path: 'product/:id', canActivate: [ ProductDetailGuard ],
        component: ProductDetailComponent }
    ])
  ],
  exports: [ RouterModule ]
})
export class ProductRoutingModule { };
```



# Using the Routing Module

## product.module.ts

```
@NgModule({
  imports: [
    SharedModule,
    ProductRoutingModule
  ],
  declarations: [
    ProductListComponent,
    ProductDetailComponent,
    ProductFilterPipe
  ],
  providers: [
    ProductService,
    ProductDetailGuard
  ]
})
export class ProductModule {}
```



# Angular Module Checklist: Module Structure



Root application module (AppModule)

Feature modules

Shared module (SharedModule)

Core module (CoreModule)

Routing modules



# Angular Module Checklist: NgModule Metadata



**Bootstrap:** Startup component(s)

**Declarations:** What belongs to this module

**Exports:** What an importing module can use

**Imports:** Supporting modules

**Providers:** Service providers



# Summary



**What Is an Angular Module?**

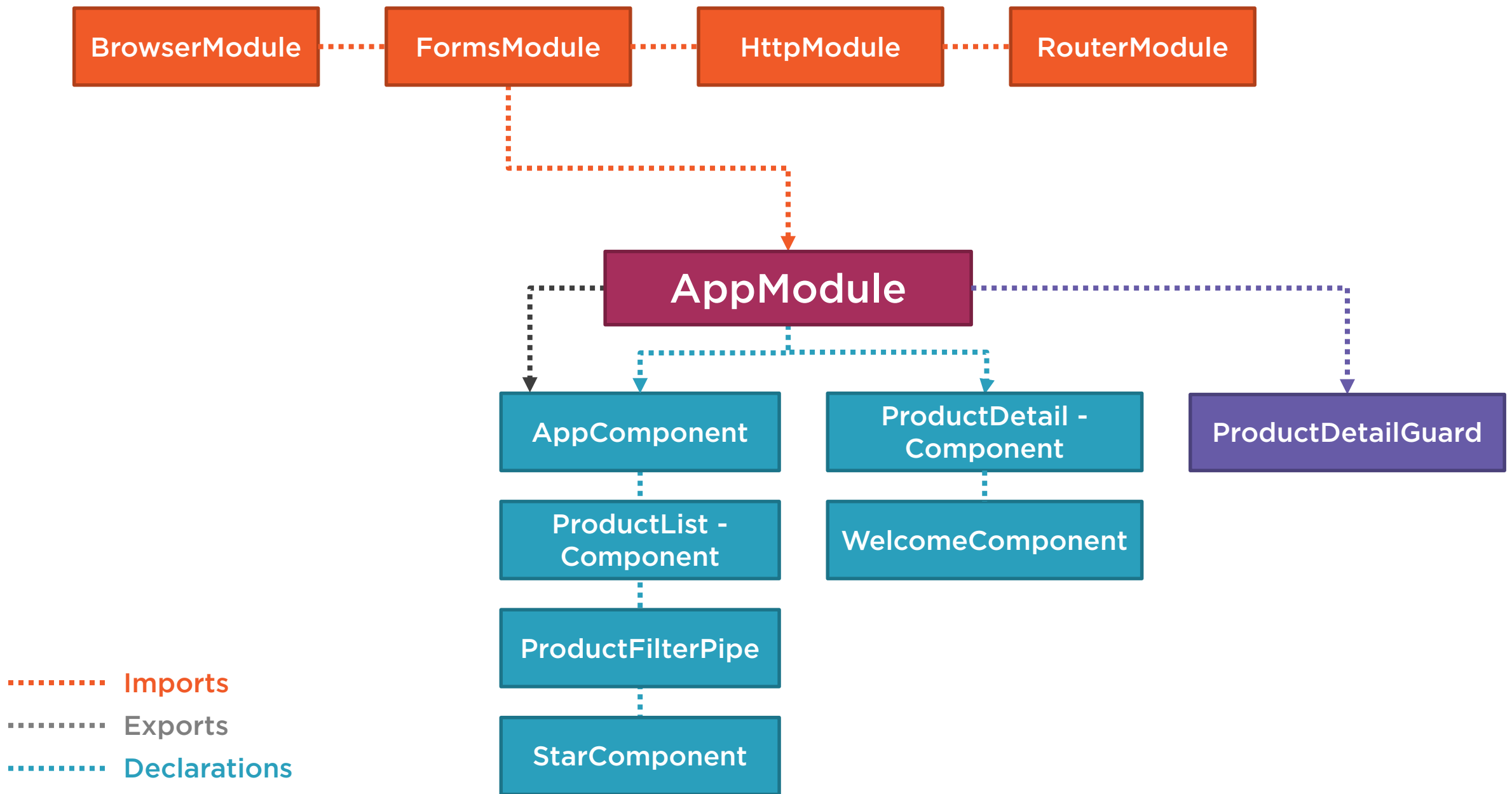
**Angular Module Metadata**

**Creating a Feature Module**

**Defining a Shared Module**

**Revisiting AppModule**

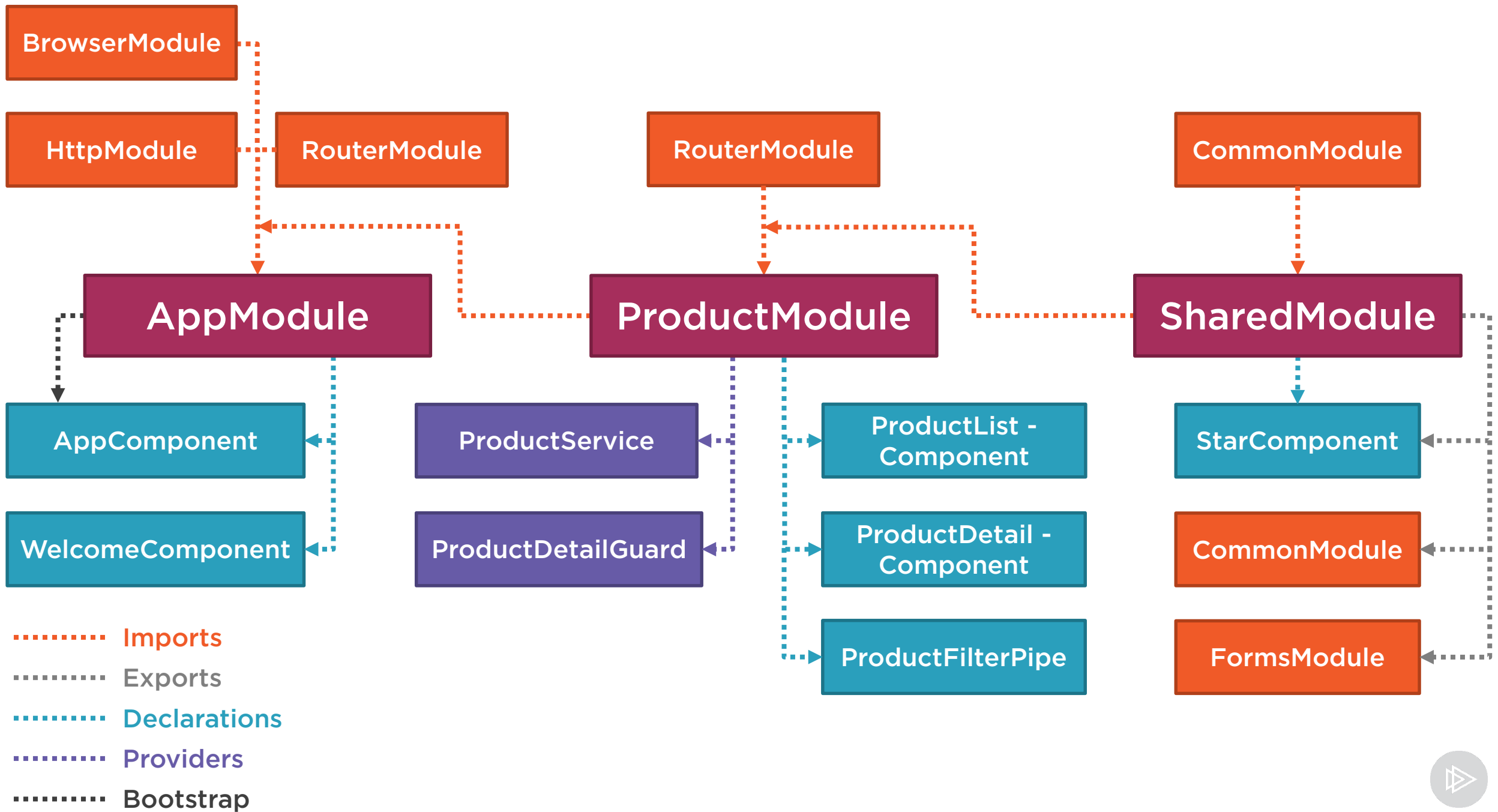




- ..... Imports
- ..... Exports
- ..... Declarations
- ..... Providers
- ..... Bootstrap







# Up Next

