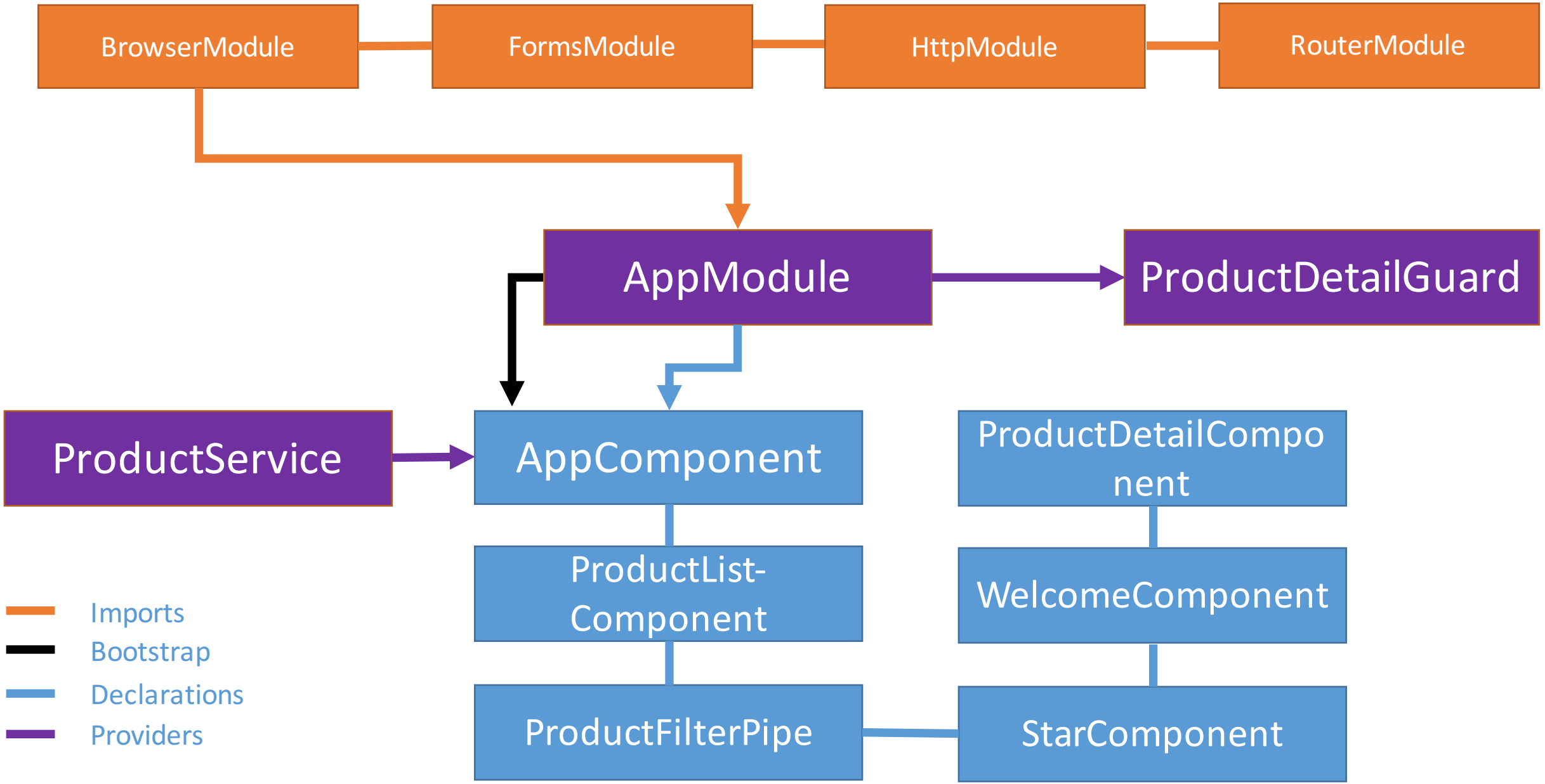
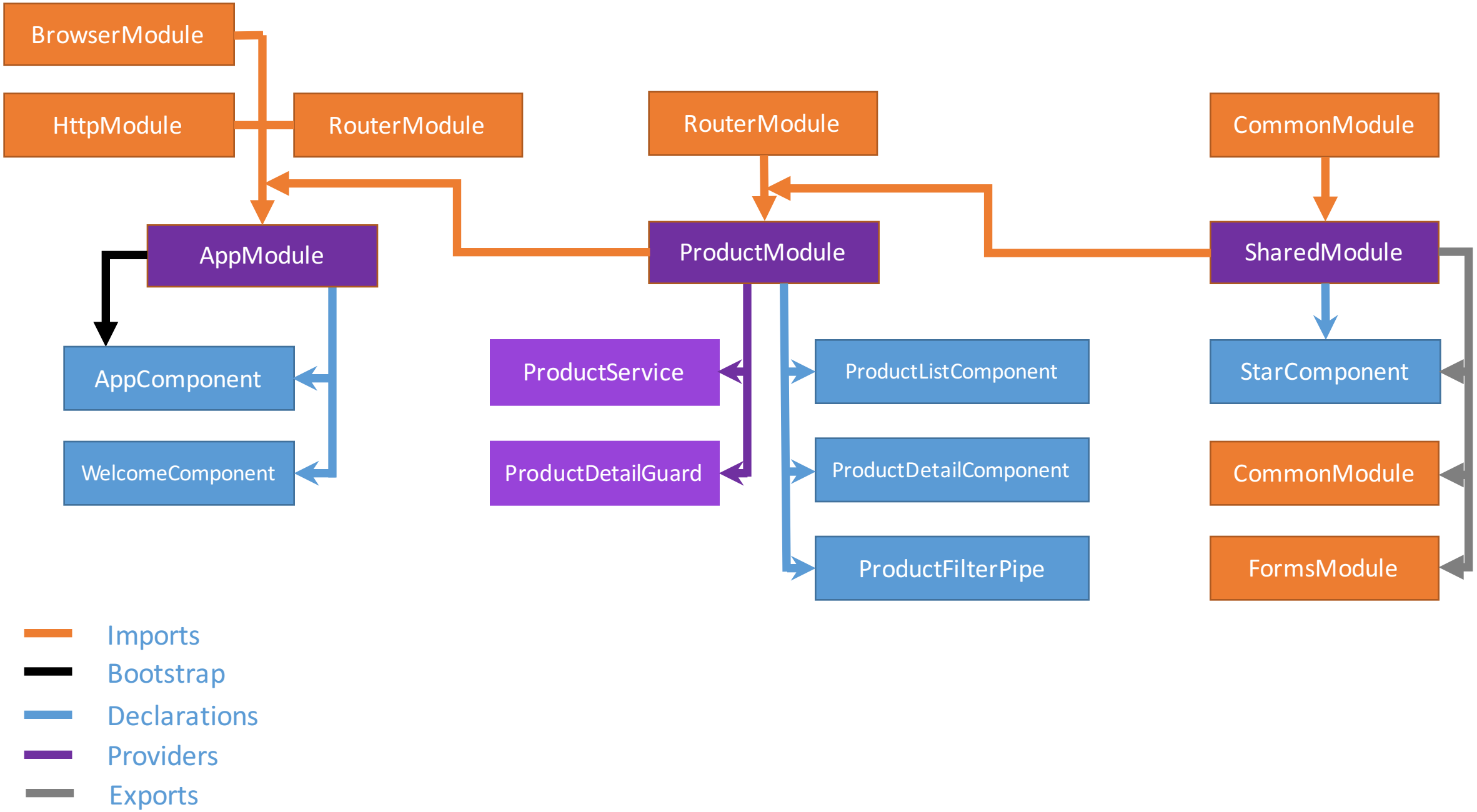
Angular 2

Angular Modules


BrowserModule
FormsModule
HttpModule
RouterModule
AppModule
ProductDetailGuard
ProductService
AppComponent
ProductDetailCompo
nent
ProductList-
Component
WelcomeComponent
ProductFilterPipe
StarComponent
Imports
Bootstrap
Declarations
Providers


BrowserModule
HttpModule
RouterModule
AppModule
AppComponent
WelcomeComponent
RouterModule
ProductModule
ProductService
ProductDetailGuard
ProductListComponent
ProductDetailComponent
ProductFilterPipe
CommonModule
SharedModule
StarComponent
CommonModule
FormsModule
Imports
Bootstrap
Declarations
Providers
Exports

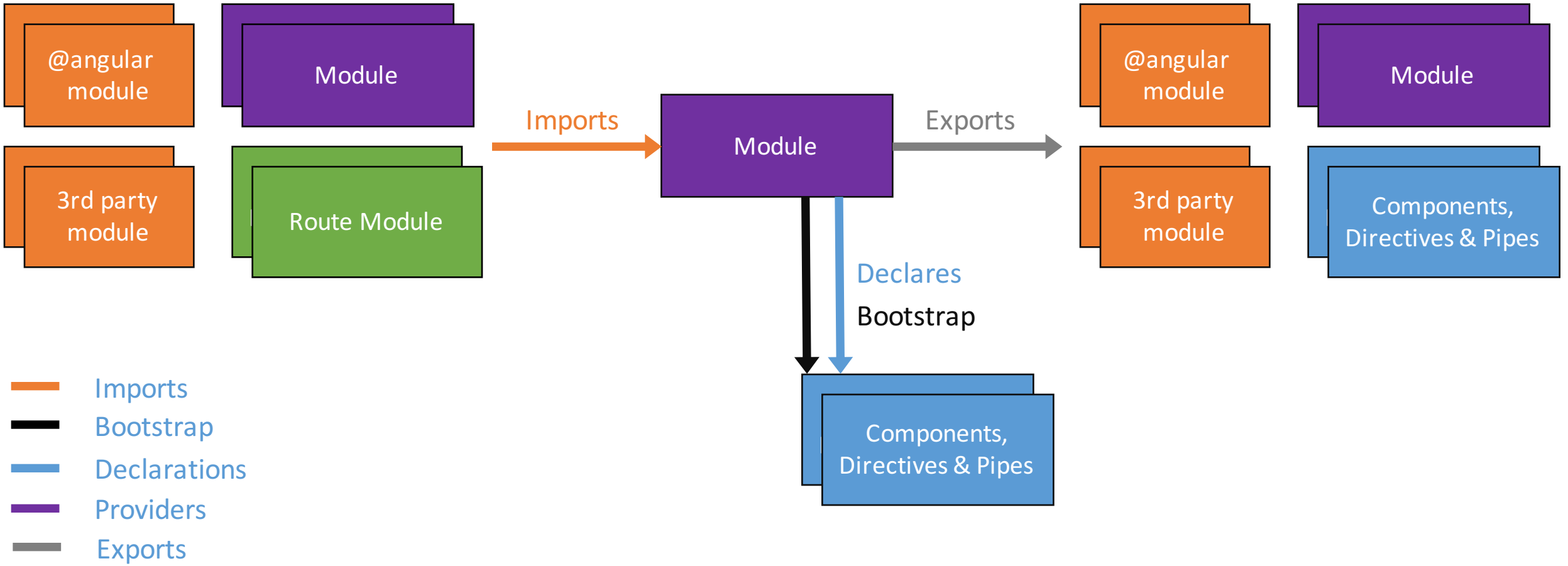
Module Overview

- What is an Angular Module?
- Angular Module Metadata
- Creating a Feature Module
- Defining a Shared Module
- Revisiting AppModule!

What is an Angular Module?

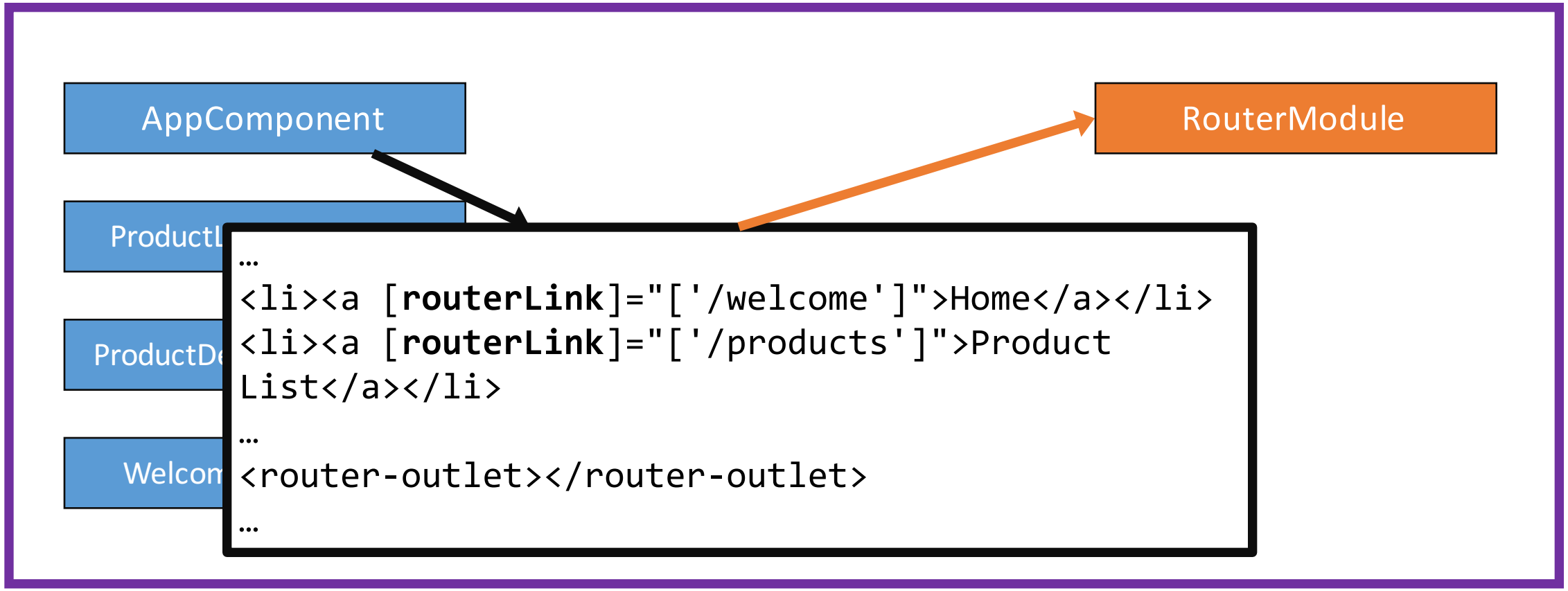
- A class with an NgModules decorator
- Its purpose:
  - Organize the pieces of our application
  - Arrange them int blocks
  - Extend our application with capabilities from external libraries
  - Provide a template resolution environment
  - Aggregate and re-export

Angular Module

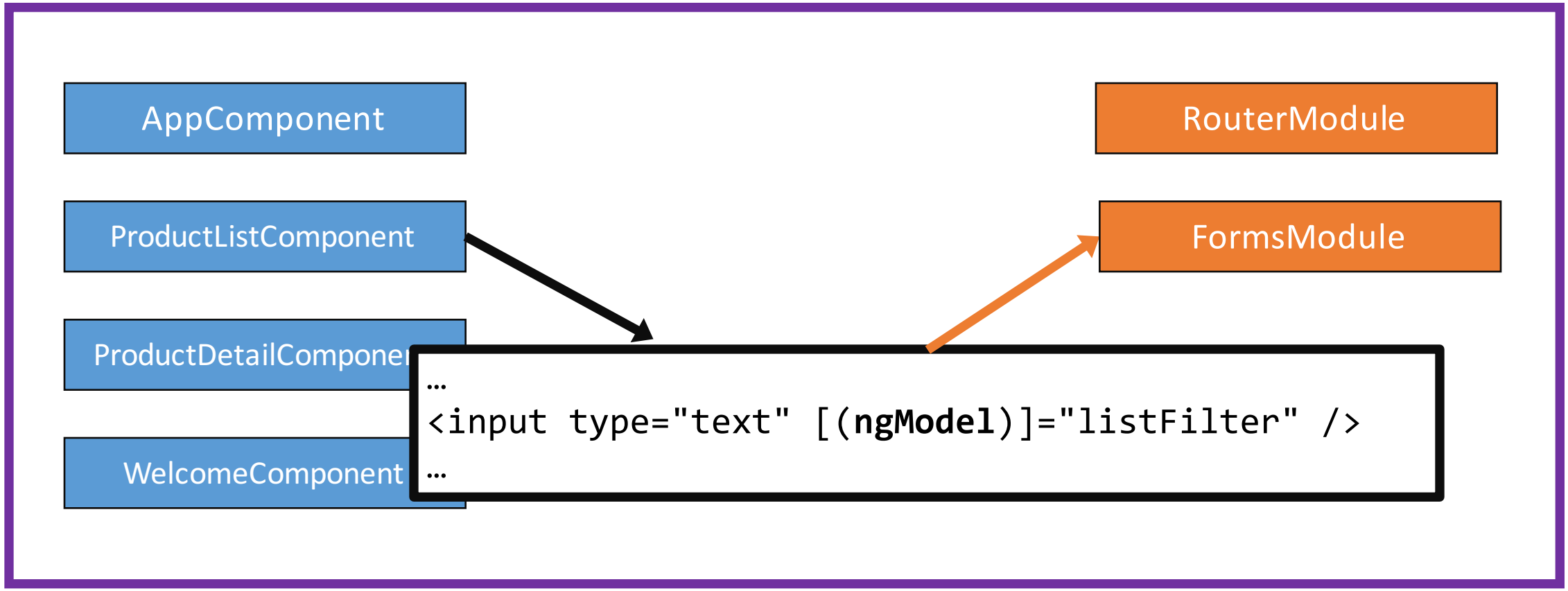


Angular Module … is a box!

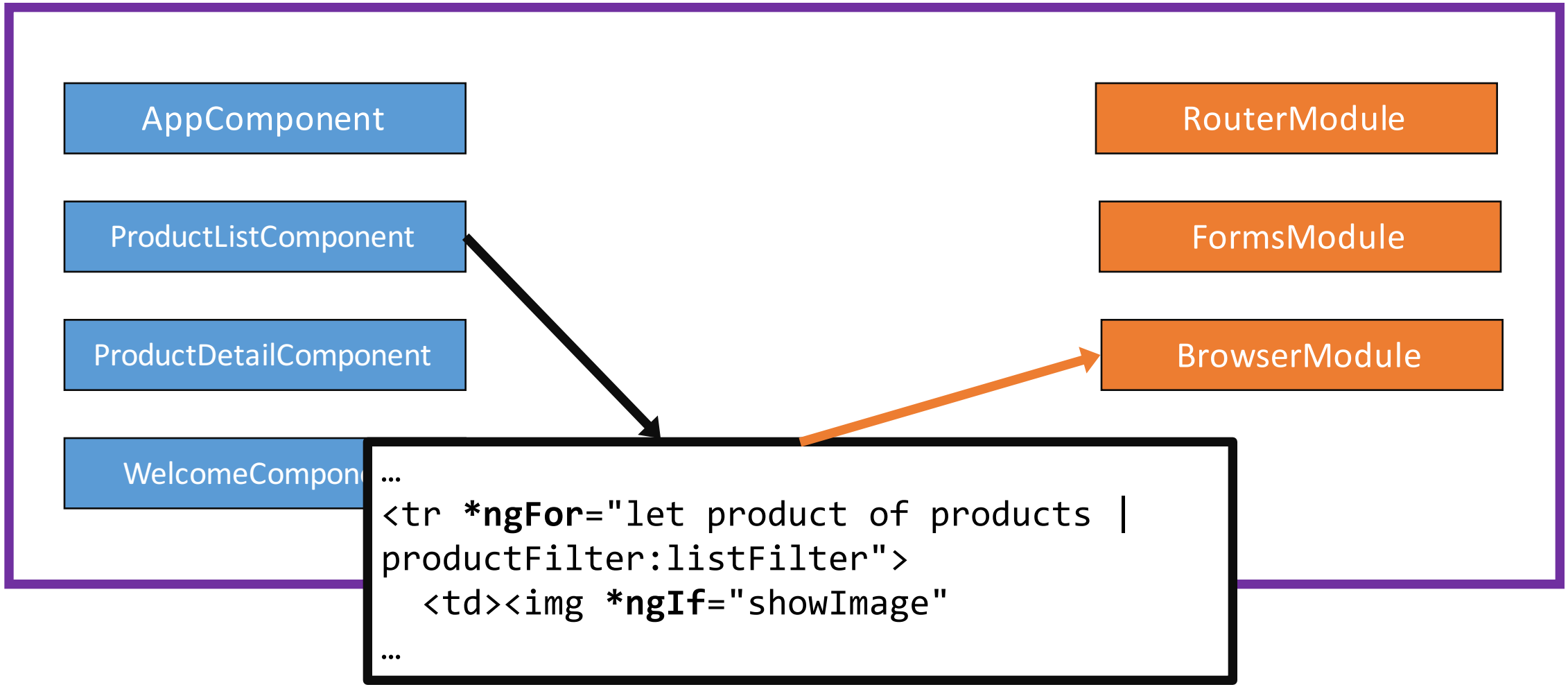
Angular Module … is a box!



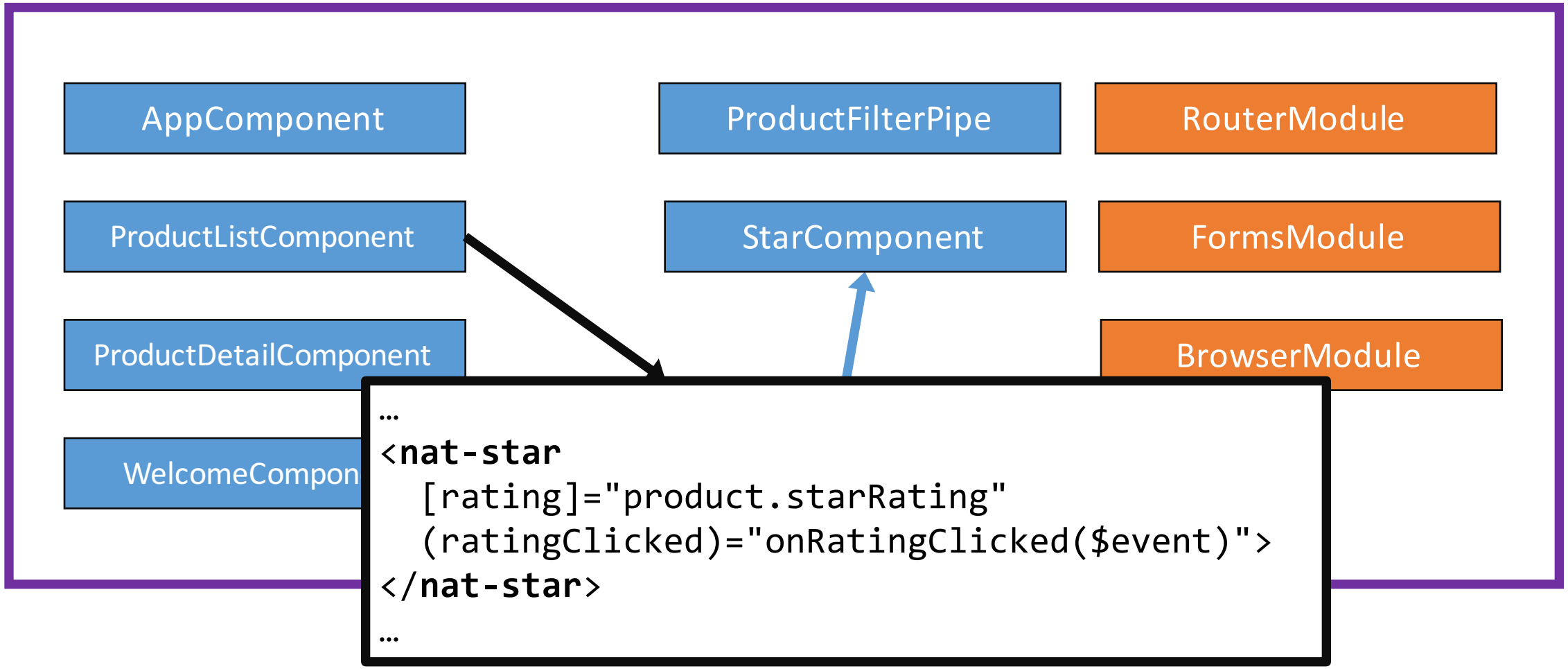
Angular Module … is a box!



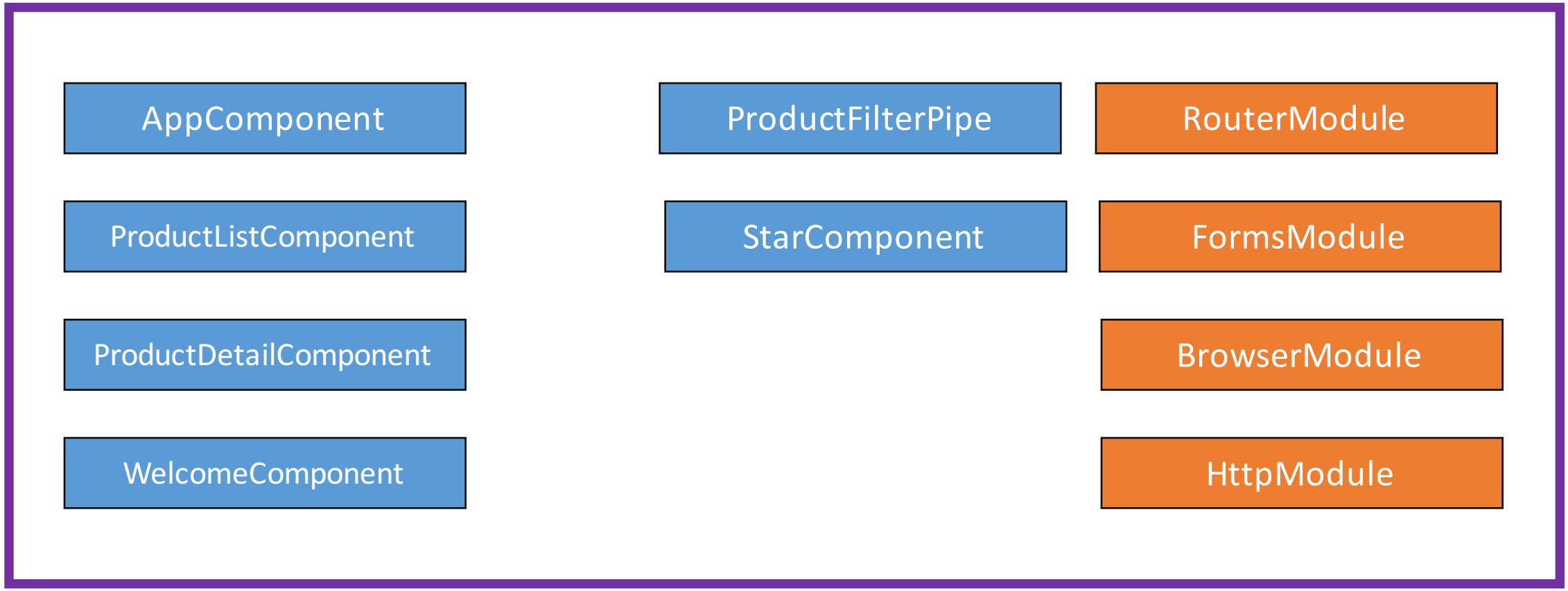
Angular Module … is a box!



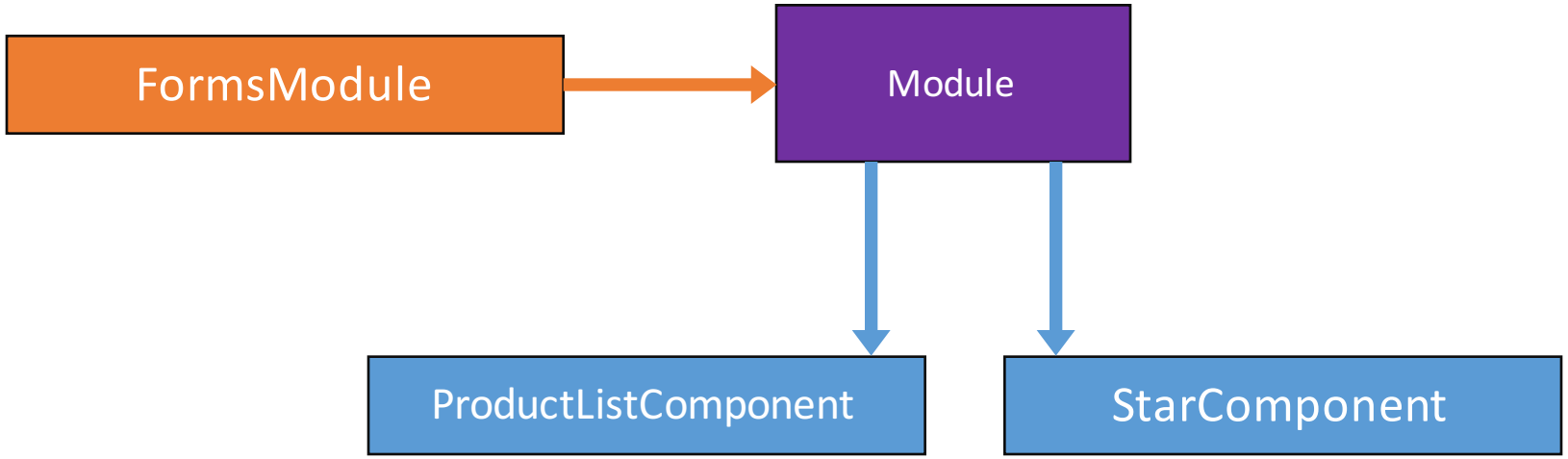
Angular Module … is a box!



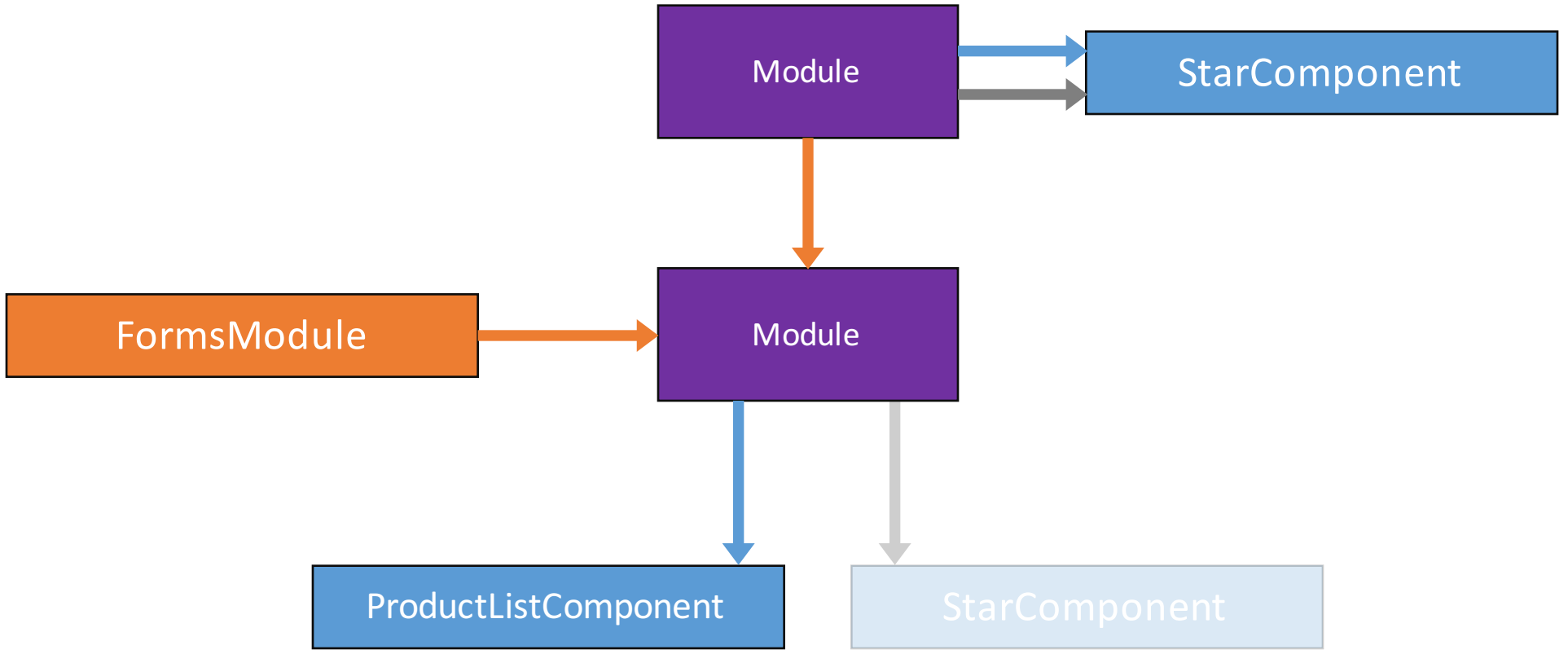
Angular Module … is a box!



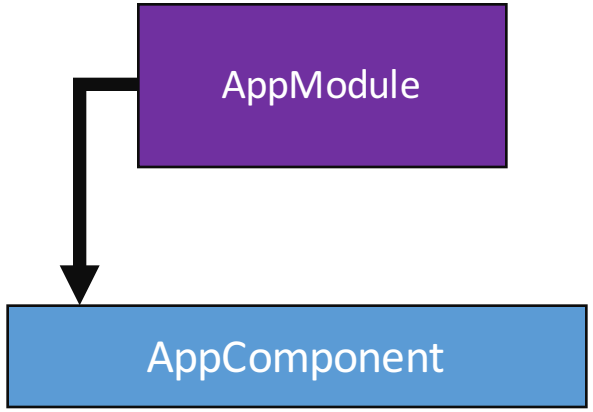
Template resolution environment



Template resolution environment



Bootstrap Array

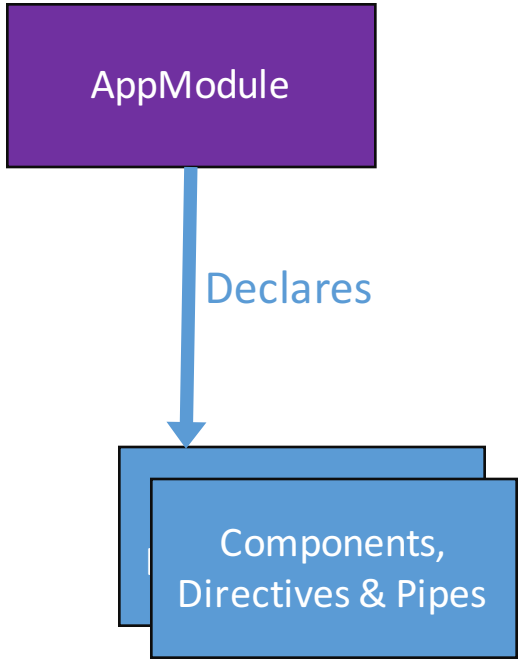



```
...
bootstrap: [AppComponent],
...
```

Bootstrap Array

- ✓ Every application must bootstrap at least one component, the root application component
- ✓ The bootstrap array must be only use in the RootModule

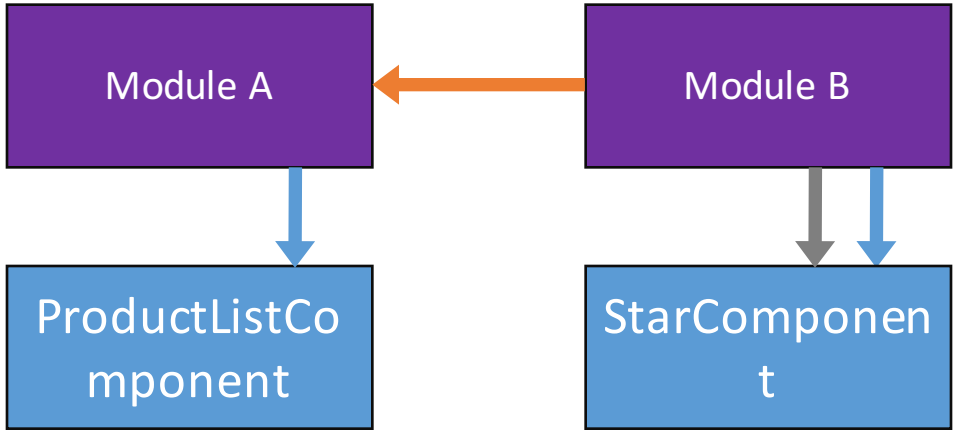
Declarations Array




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

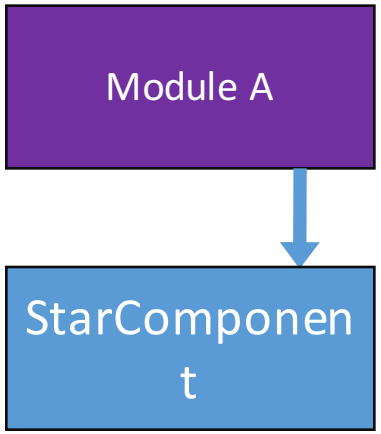
Declarations Array

- ✓ Every component, directive and pipe we creat must belong to **one and only one** Angular Module
- ✓ Only declare components, directive and pipe
- ✓ Never re-declare components, directives and pipes that belong to another modules (cf #1)



Declarations Array

- ✓ 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.

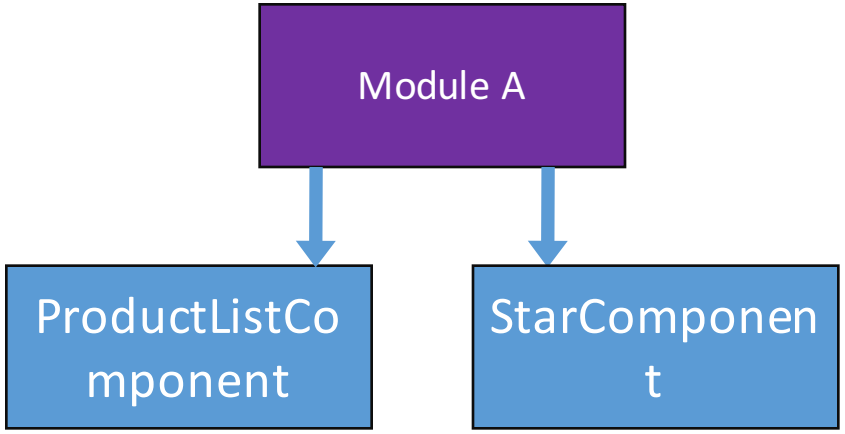


Declarations Array

- ✓ The Angular Module provides the template resolution environment for its component templates.

```
// product-list.component.html
<td>
  <nat-star ...>
  </nat-star>
</td>
```

```
// star.component.ts
...
@Component({
  selector: 'nat-star',
  templateUrl: 'app/shared/star.component.html'
})
```

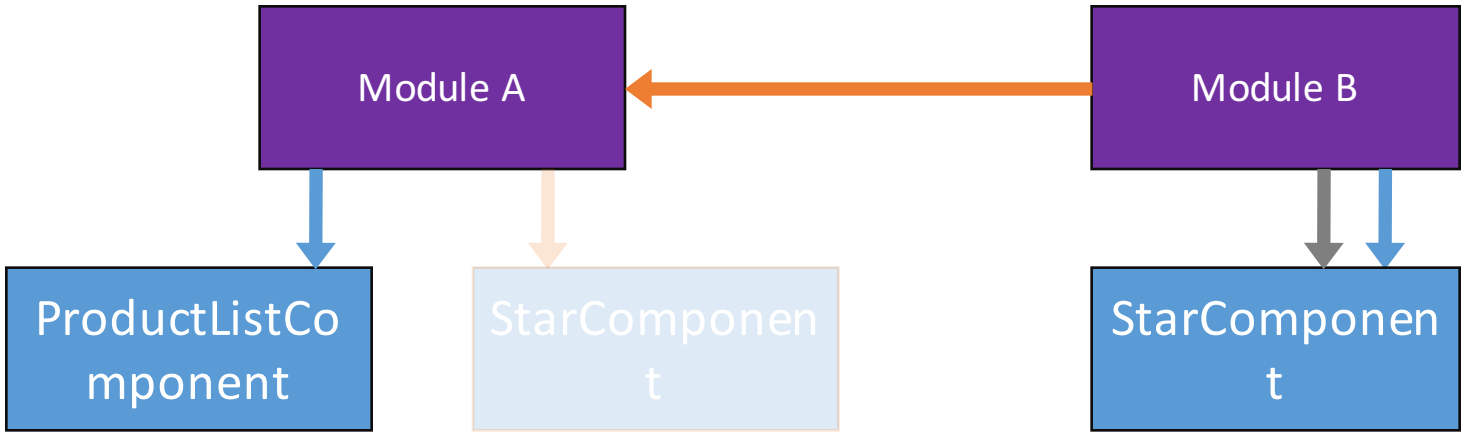


Declarations Array

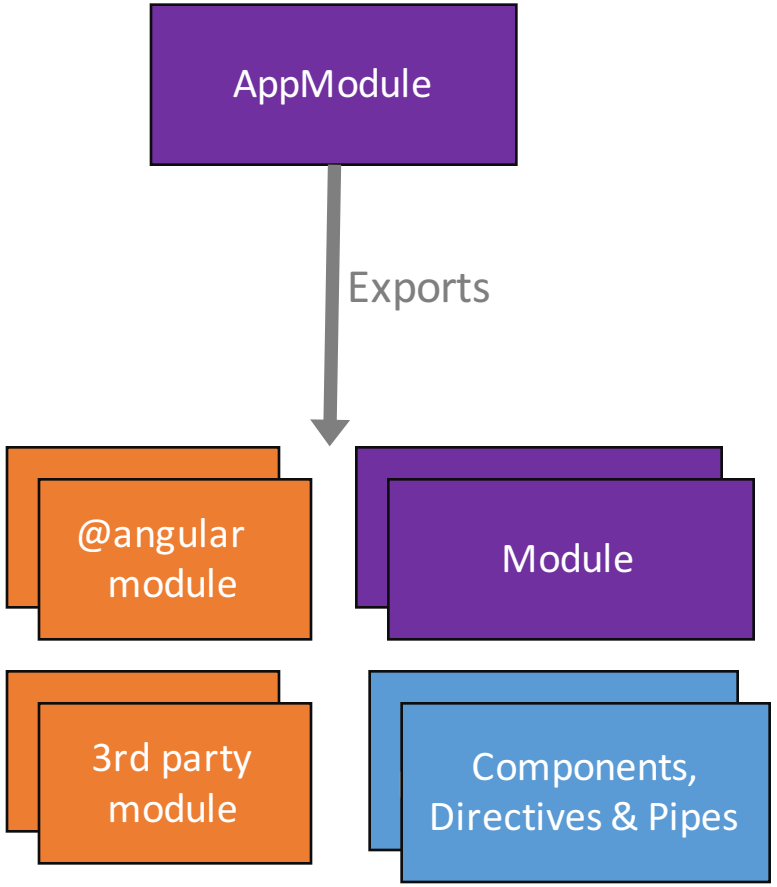
- ✓ The Angular Module provides the template resolution environment for its component templates.

```
// product-list.component.html
<td>
  <nat-star ...>
  </nat-star>
</td>
```

```
// star.component.ts
...
@Component({
  selector: 'nat-star',
  templateUrl: 'app/shared/star.component.html'
})
```

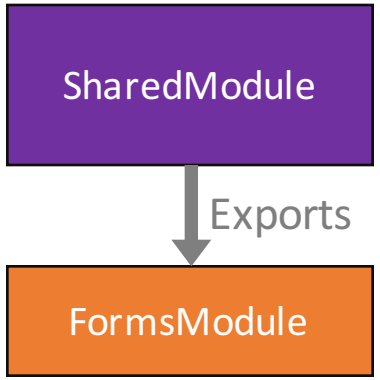


Exports Array

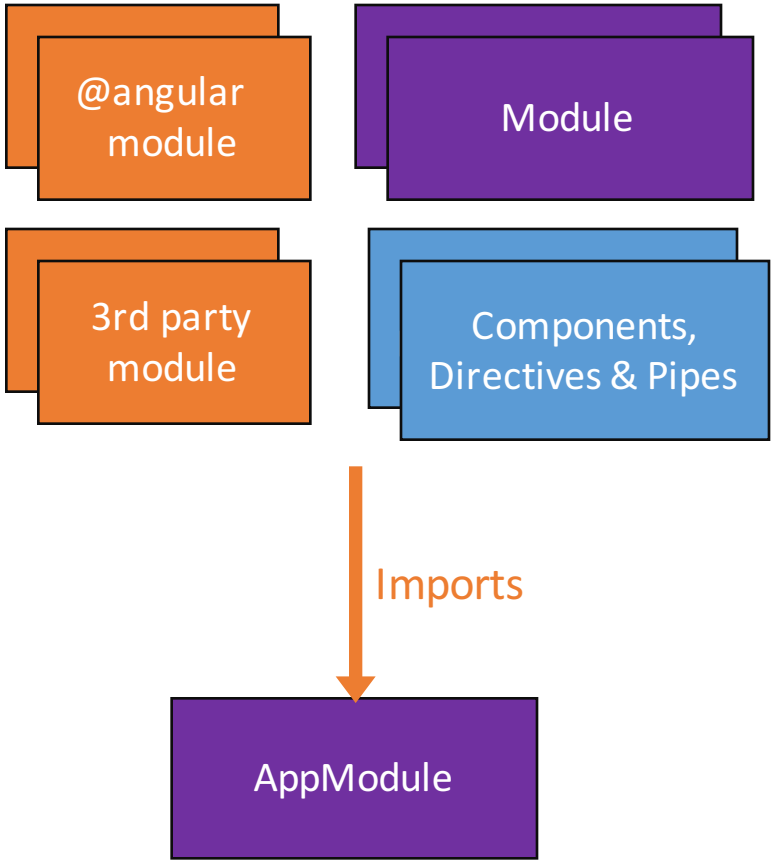


Exports Array

- ✓ Export any components, directives, or pipes if another component needs it.
- ✓ Re-export modules to re-exports their components, directives, and pipes.
- ✓ We can re-export something without importing it first.
- ✓ Never export a service



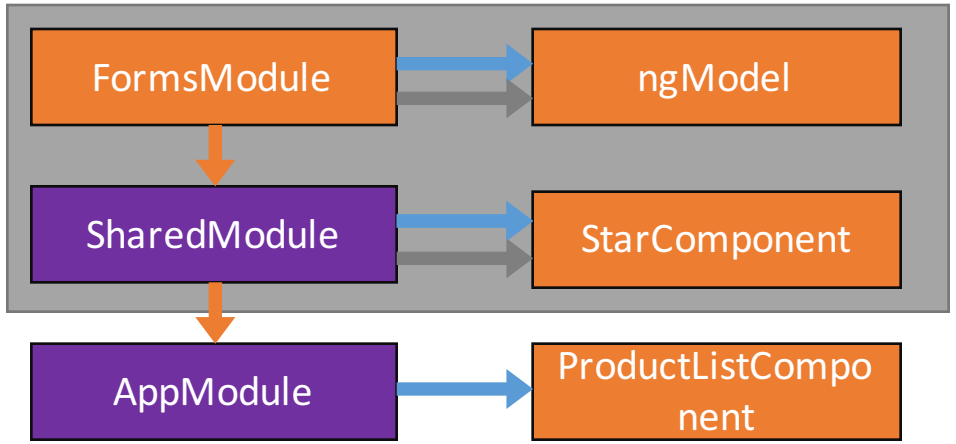
Imports Array




```
...
imports: [
  BrowserModule,
  HttpModules,
  FormsModules,
  RouterModule.forRoot([...])
],
...
```

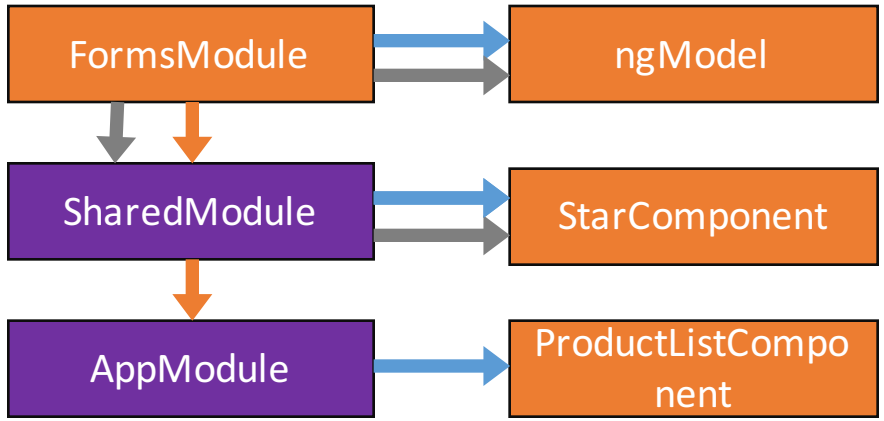
Imports Array

- ✓ Importing a module makes available **any exported** components, directives, and pipes from that module.
- ✓ Only import what this module needs.
- ✓ Importing a module does NOT provide access to its imported modules (if not exported).

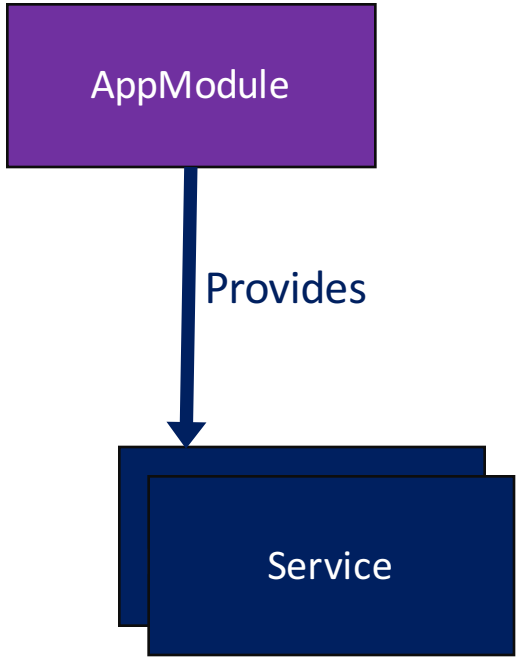


Imports Array

- ✓ Importing a module makes available **any exported** components, directives, and pipes from that module.
- ✓ Only import what this module needs.
- ✓ Importing a module does NOT provide access to its imported modules (if not exported).



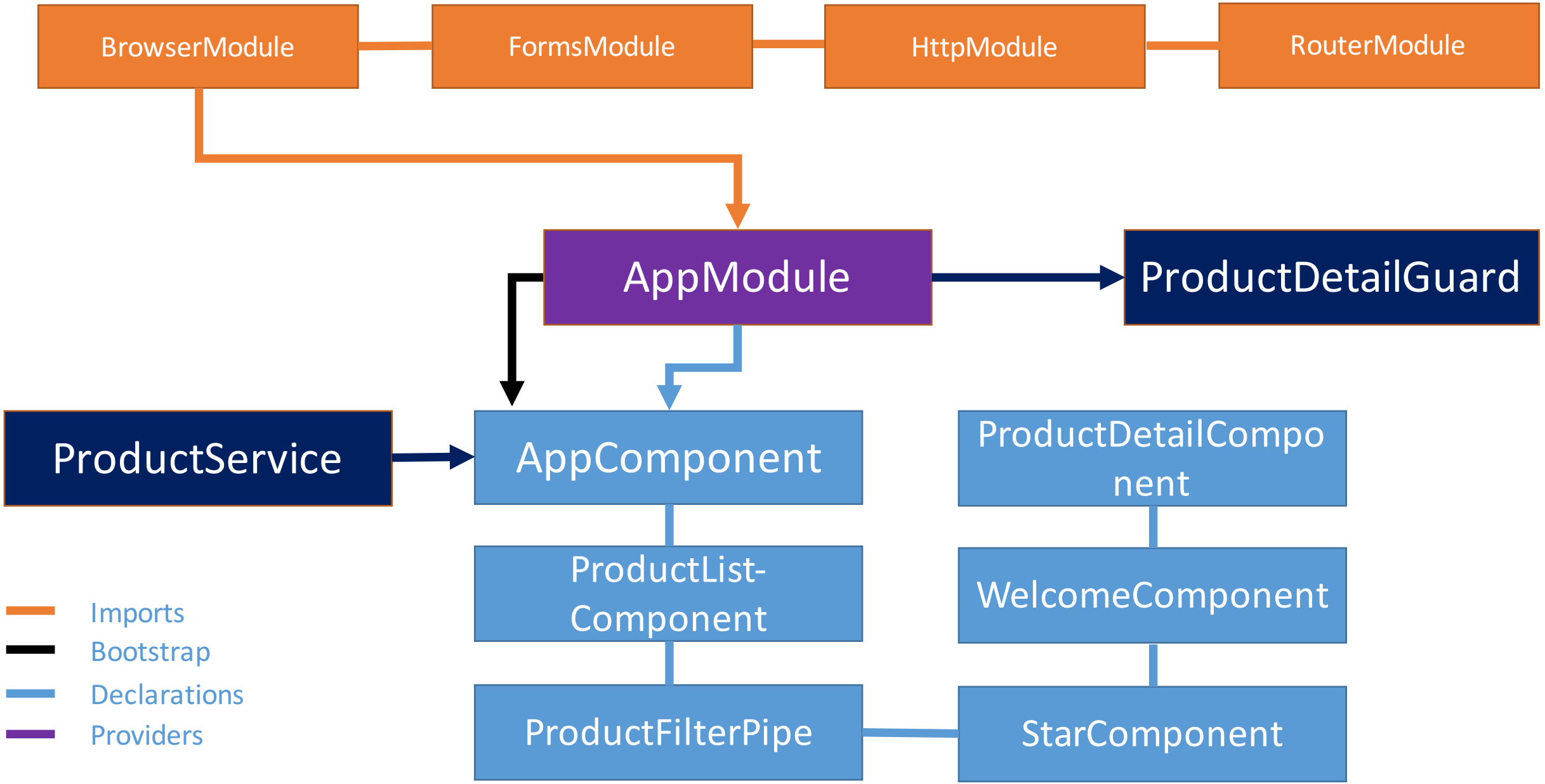
Providers Array

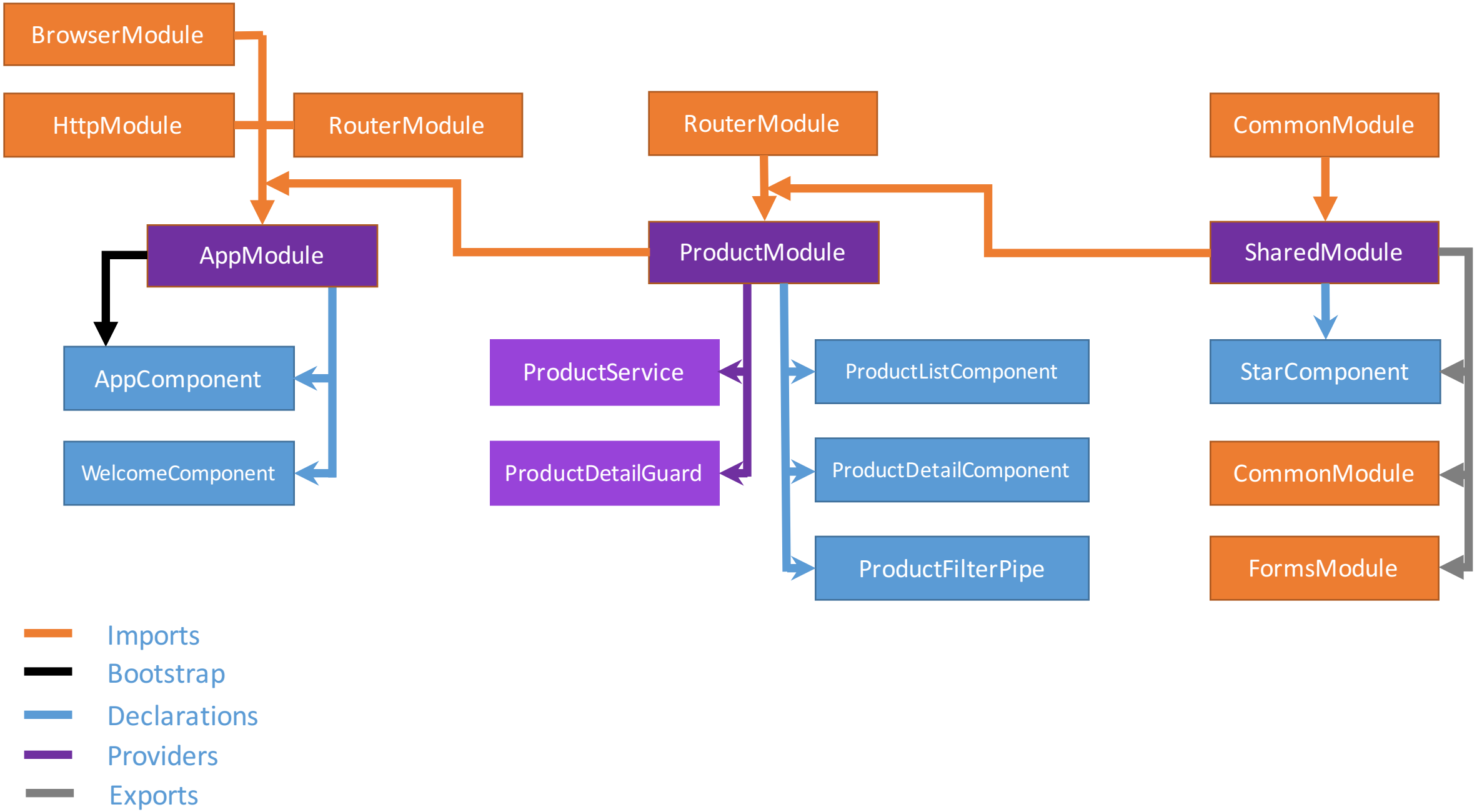



```
...
providers: [
  ProductDetailGuard
],
...
```

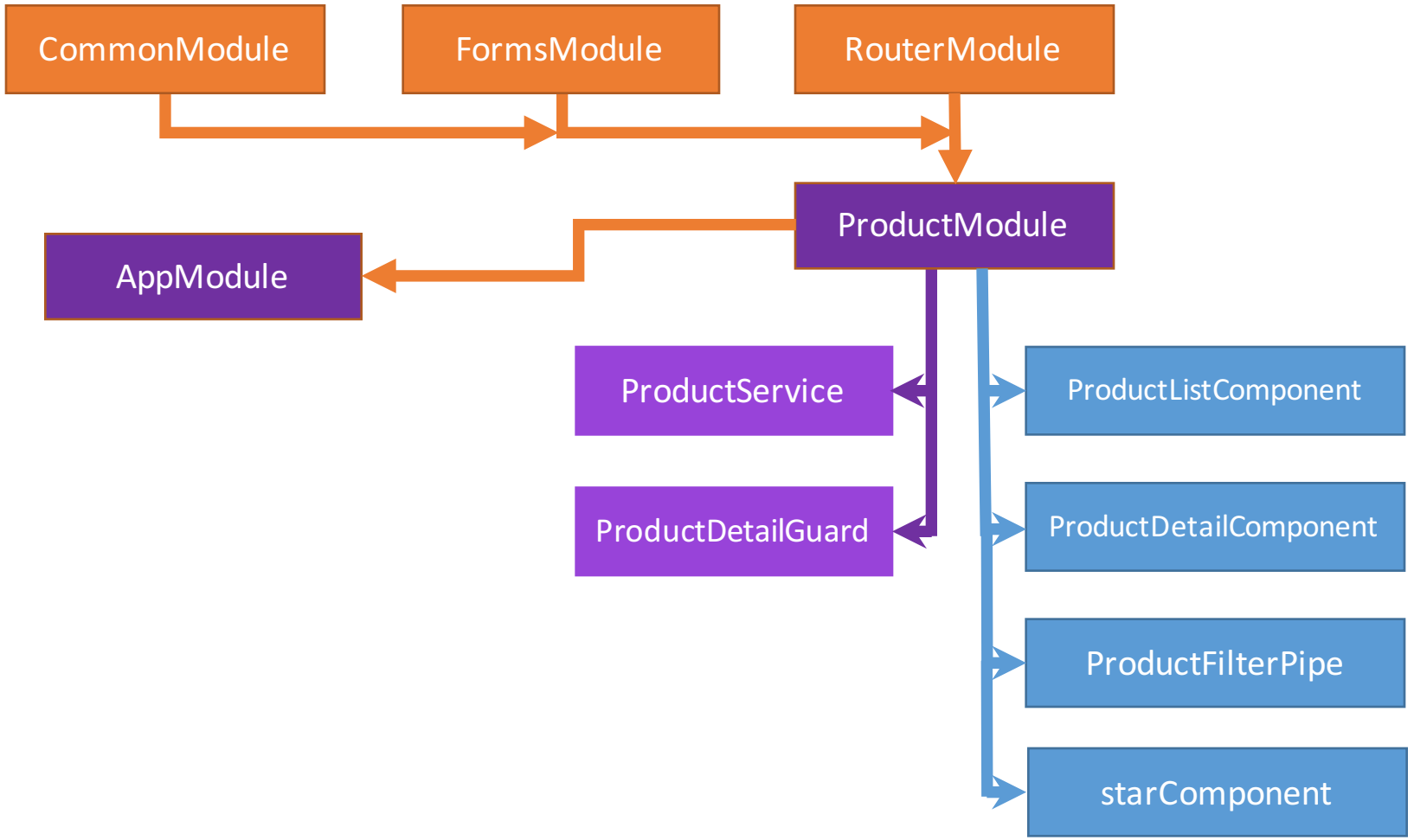
Providers Array

- ✓ Any service provider added to the providers array is registered at the **root** level of the application.
- ✓ Don’t add services to the providers array of a shared module (think about a CoreModule)
- ✓ Guard services must be defined on an Angular Module Level

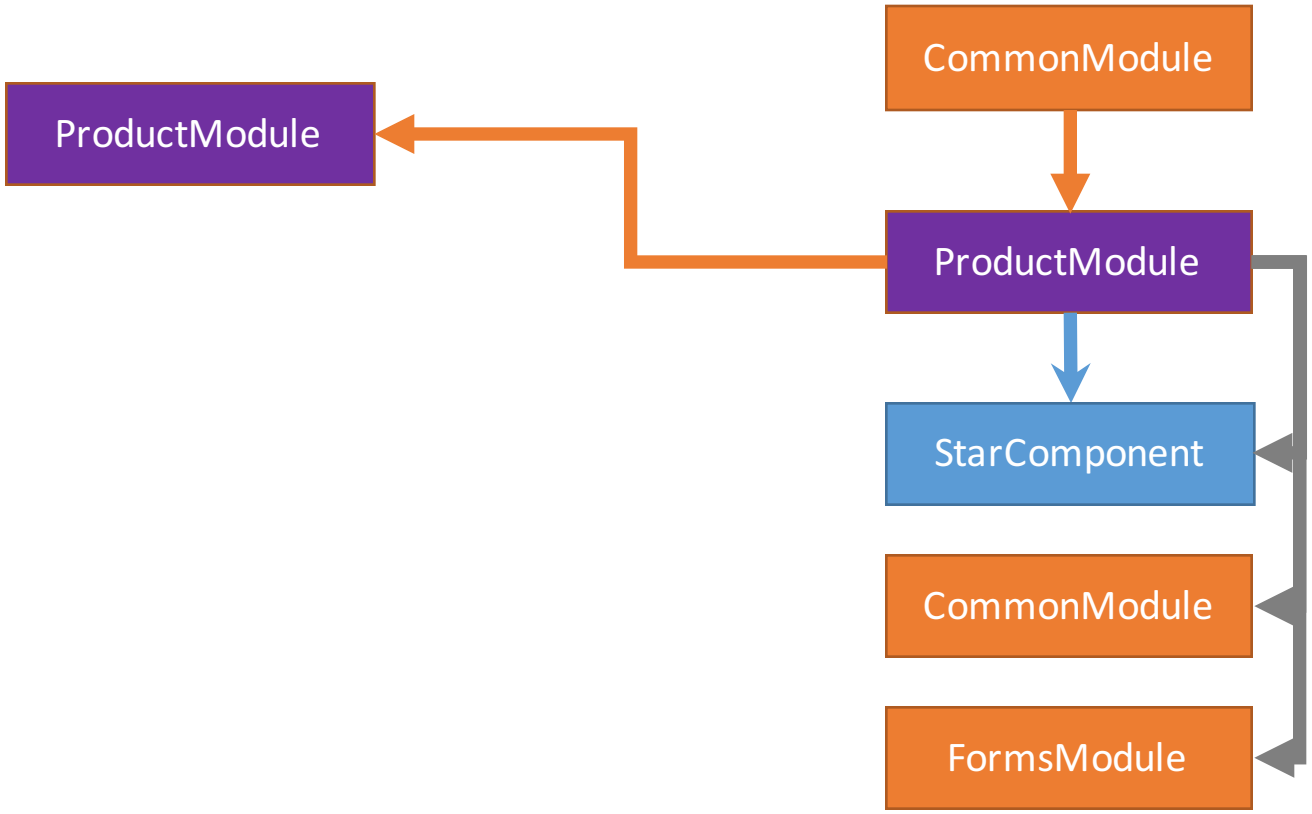

BrowserModule
FormsModule
HttpModule
RouterModule
AppModule
ProductDetailGuard
ProductService
AppComponent
ProductDetailCompo
nent
ProductList-
Component
WelcomeComponent
ProductFilterPipe
StarComponent
Imports
Bootstrap
Declarations
Providers


BrowserModule
HttpModule
RouterModule
AppModule
AppComponent
WelcomeComponent
RouterModule
ProductModule
ProductService
ProductDetailGuard
ProductListComponent
ProductDetailComponent
ProductFilterPipe
CommonModule
SharedModule
StarComponent
CommonModule
FormsModule
Imports
Bootstrap
Declarations
Providers
Exports

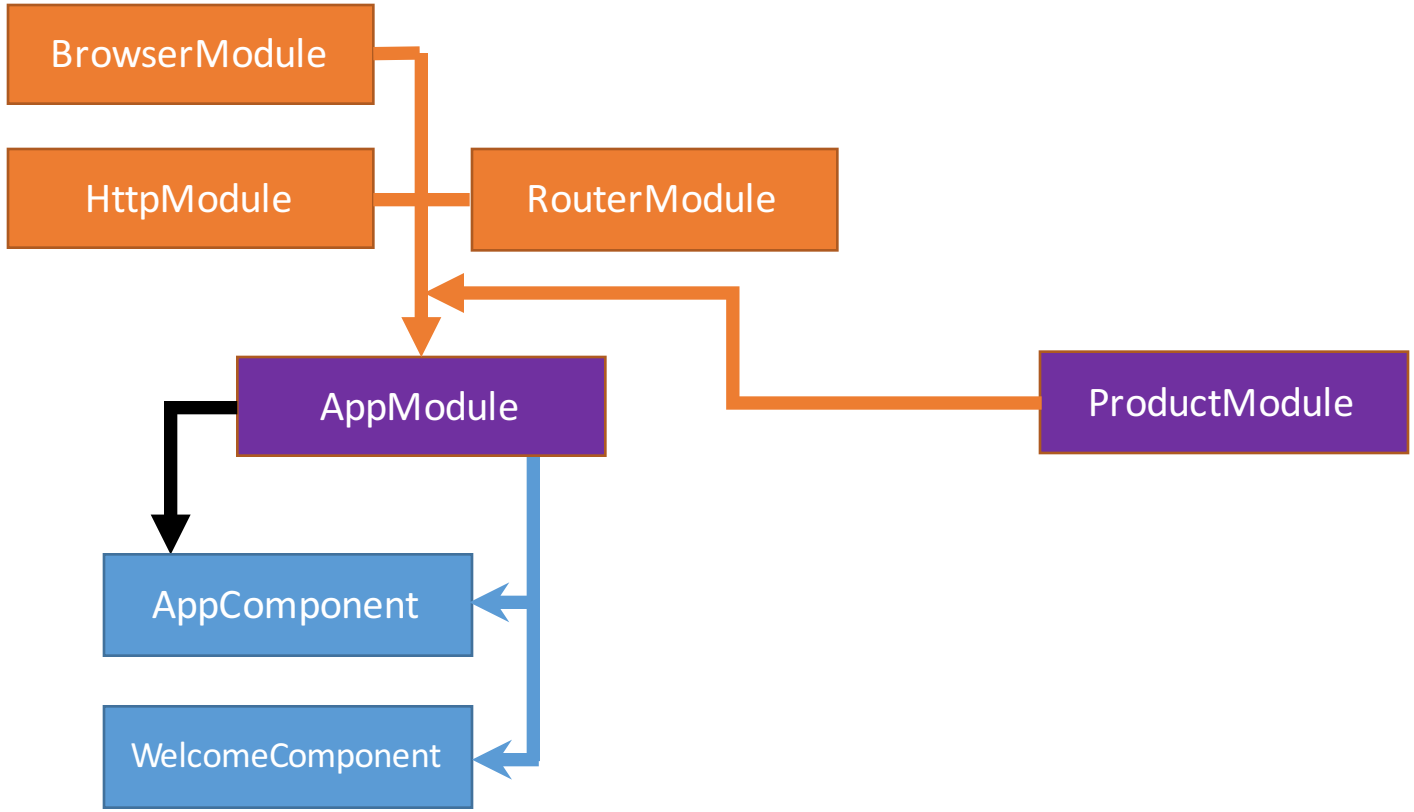
Defining a Feature Module



Defining a Shared Module



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()
```

Application Routing Module

```
// products/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.server';

@NgModule({
  imports: [
    RouterModule.forRoot([
      { path: 'products', component: ProductListComponent },
      { path: 'product/:id', component: ProductDetailComponent,
        canActivate: [ ProductDetailGuard ] },
    ])
  ],
  exports: [ RouterModule]
})
export class ProductRoutingModule()
```

Angular Module Checklist: Module Structure

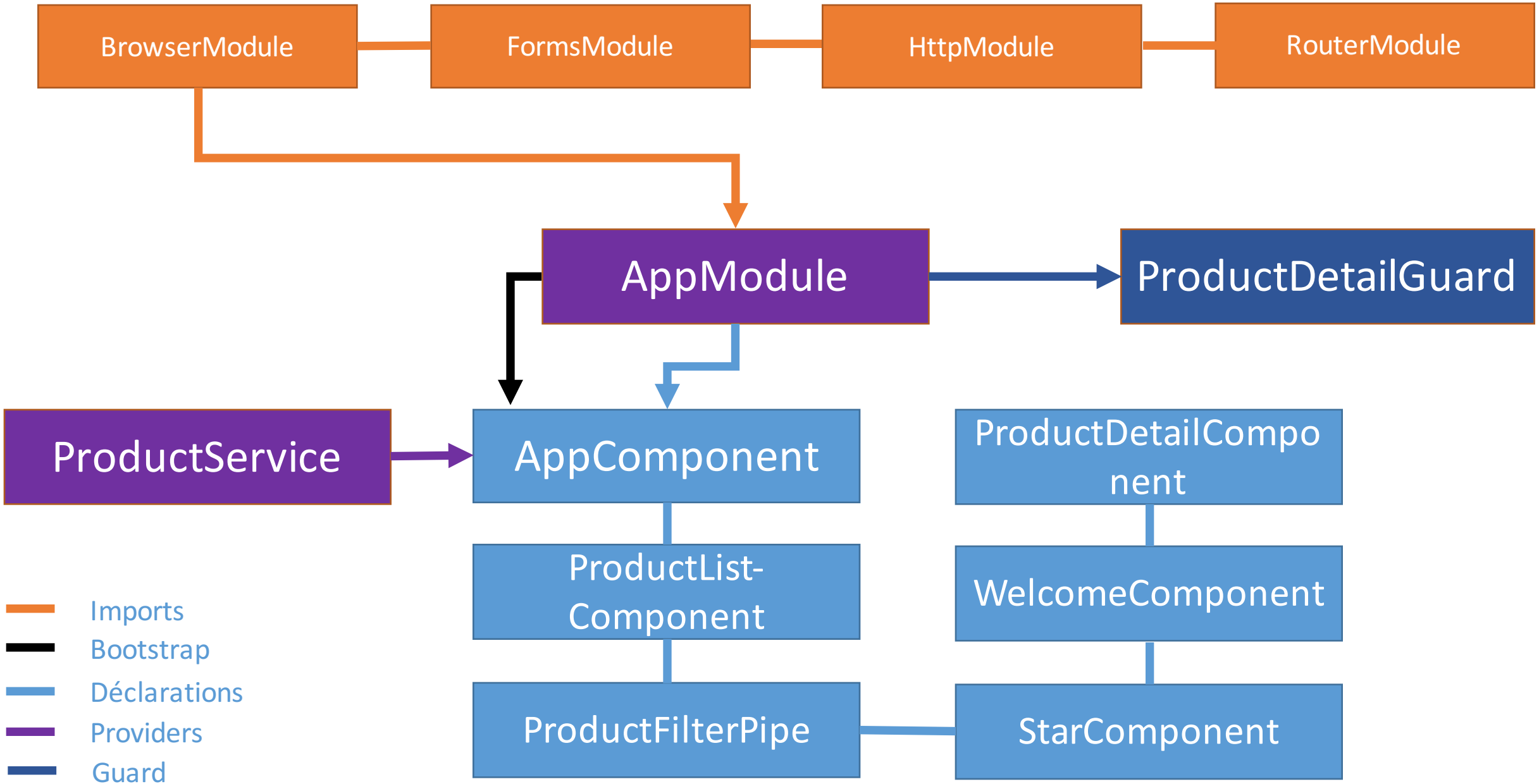
- Root application module (AppModule)
- Feature modules
- Shared modules (SharedModule)
- Core module (CoreModule)
- Routing modules

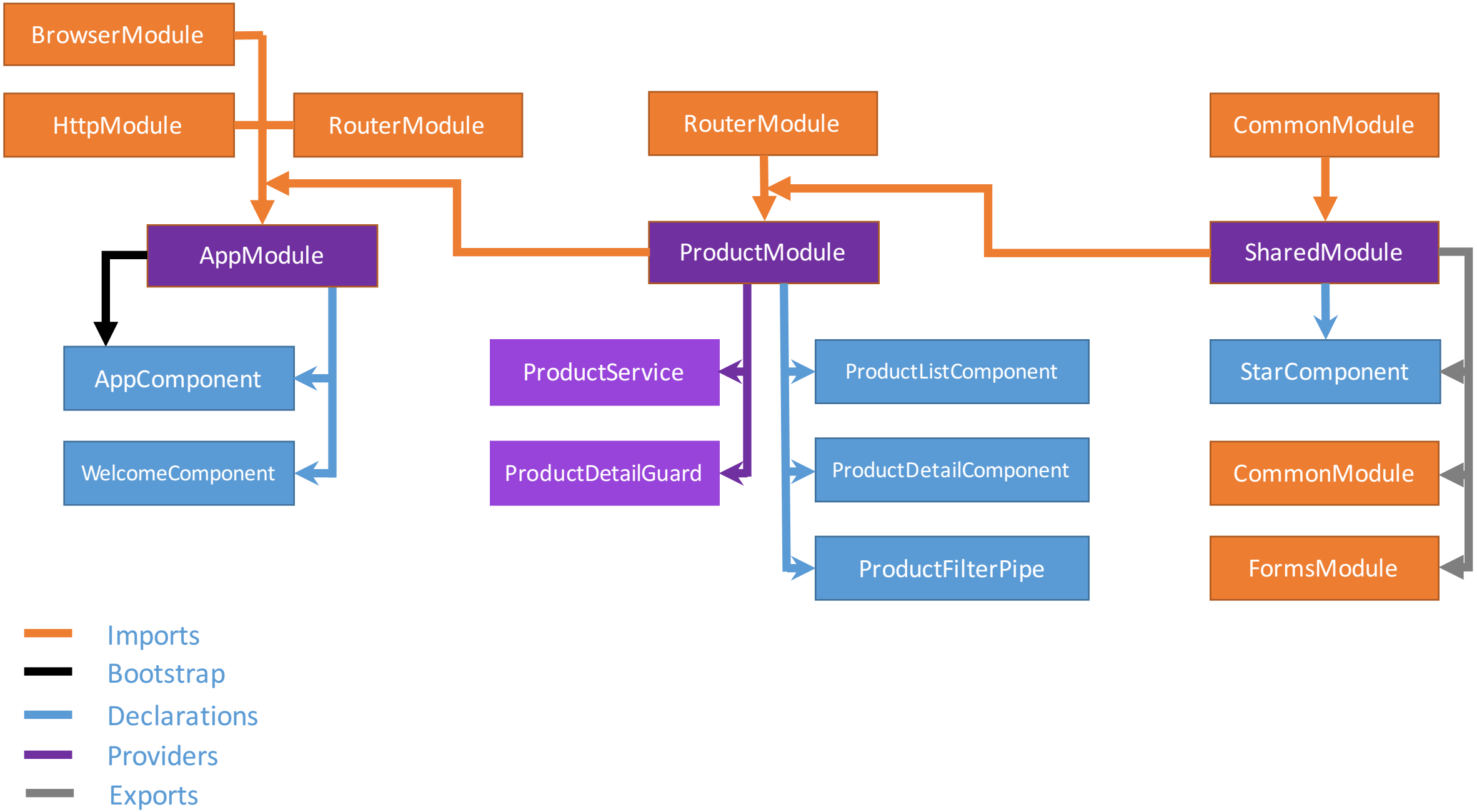
Angular Module Checklist: NgModule Metadata

- Bootstrap: Startup component(s)
- Declarations: What belongs to this module
- Exports: What an importing module can use
- Imports: Supporting modules
- Providers: Service providers

Module Overview

- What is an Angular Module?
- Angular Module Metadata
- Creating a Feature Module
- Defining a Shared Module
- Revisiting AppModule!


BrowserModule
FormsModule
HttpModule
RouterModule
AppModule
ProductDetailGuard
ProductService
AppComponent
ProductDetailCompo
nent
ProductList-
Component
WelcomeComponent
ProductFilterPipe
StarComponent
Imports
Bootstrap
Déclarations
Providers
Guard


BrowserModule
HttpModule
RouterModule
AppModule
AppComponent
WelcomeComponent
RouterModule
ProductModule
ProductService
ProductDetailGuard
ProductListComponent
ProductDetailComponent
ProductFilterPipe
CommonModule
SharedModule
StarComponent
CommonModule
FormsModule
Imports
Bootstrap
Declarations
Providers
Exports