

ANGULAR MODULARITY

(NgModules)

Naveen Pete

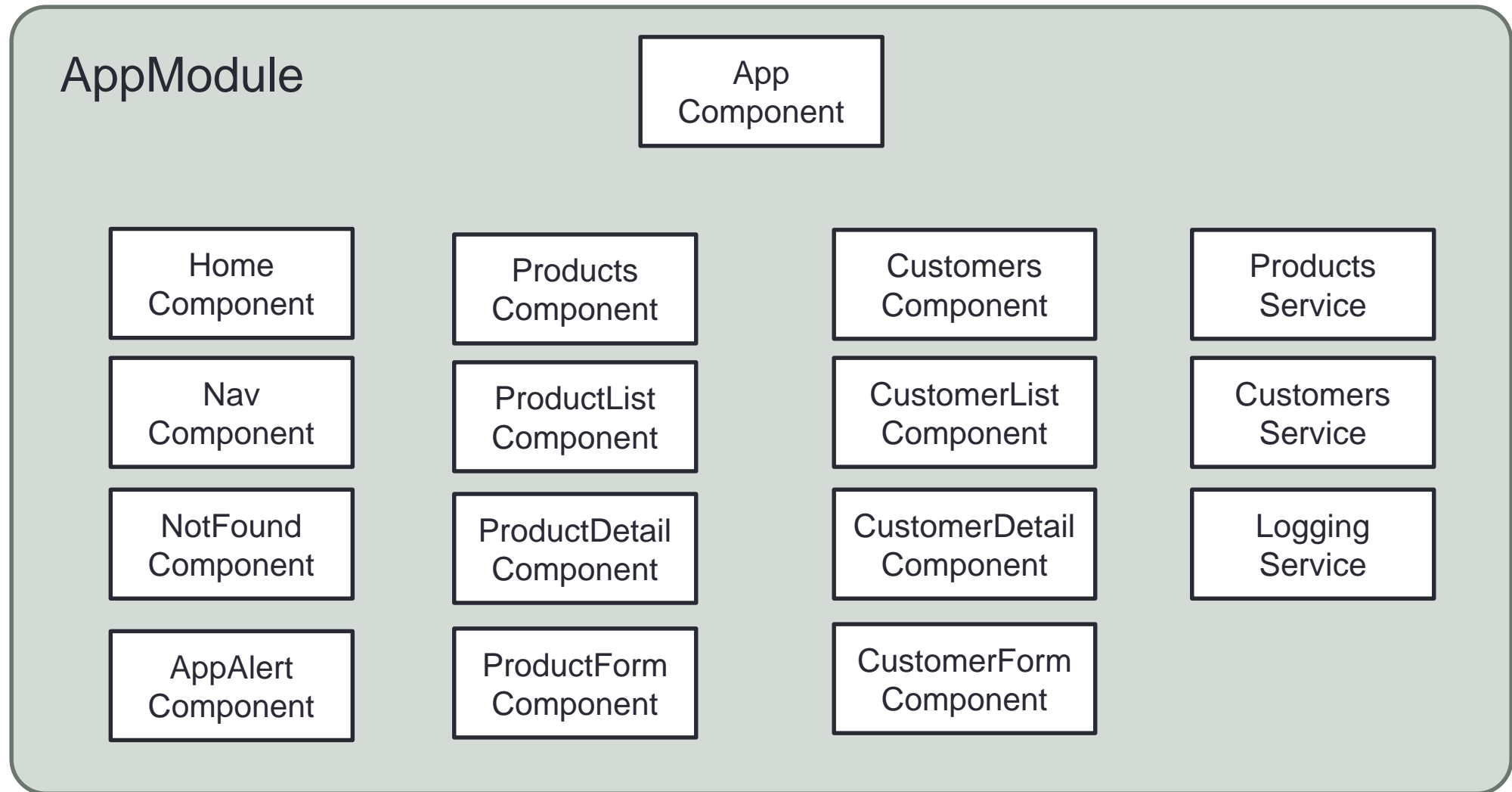
Agenda

- Modules
- A Typical Angular App
- Angular Modules
- @NgModule Decorator
- Routing Modules
- Feature Modules
- Sharing Components across Modules
- Organizing Core Components of an App
- Q & A

Modules

- Great way to organize an app
- Allow an app to extend with capabilities from external libraries
- Increase performance of the app
- Decrease file size
- Structure code in a better way
- Make the app code more maintainable

A Typical Angular App



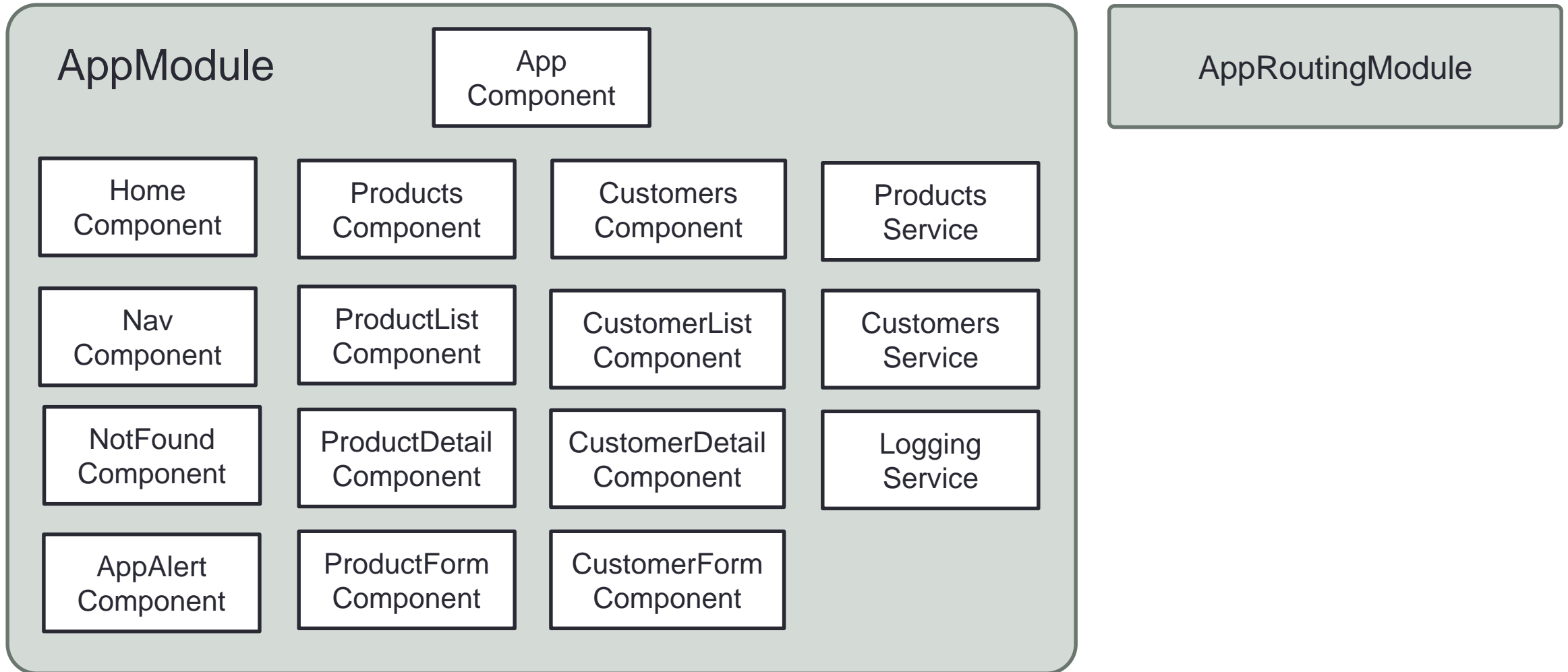
Angular Modules

- By default, all Angular apps contain one module – AppModule
- An Angular module
 - consolidates components, directives, and pipes into cohesive blocks of functionality
 - bundles related functionalities that can be imported and reused
 - can be eagerly loaded or lazy loaded asynchronously
 - is a simple TypeScript class that is decorated using @NgModule decorator

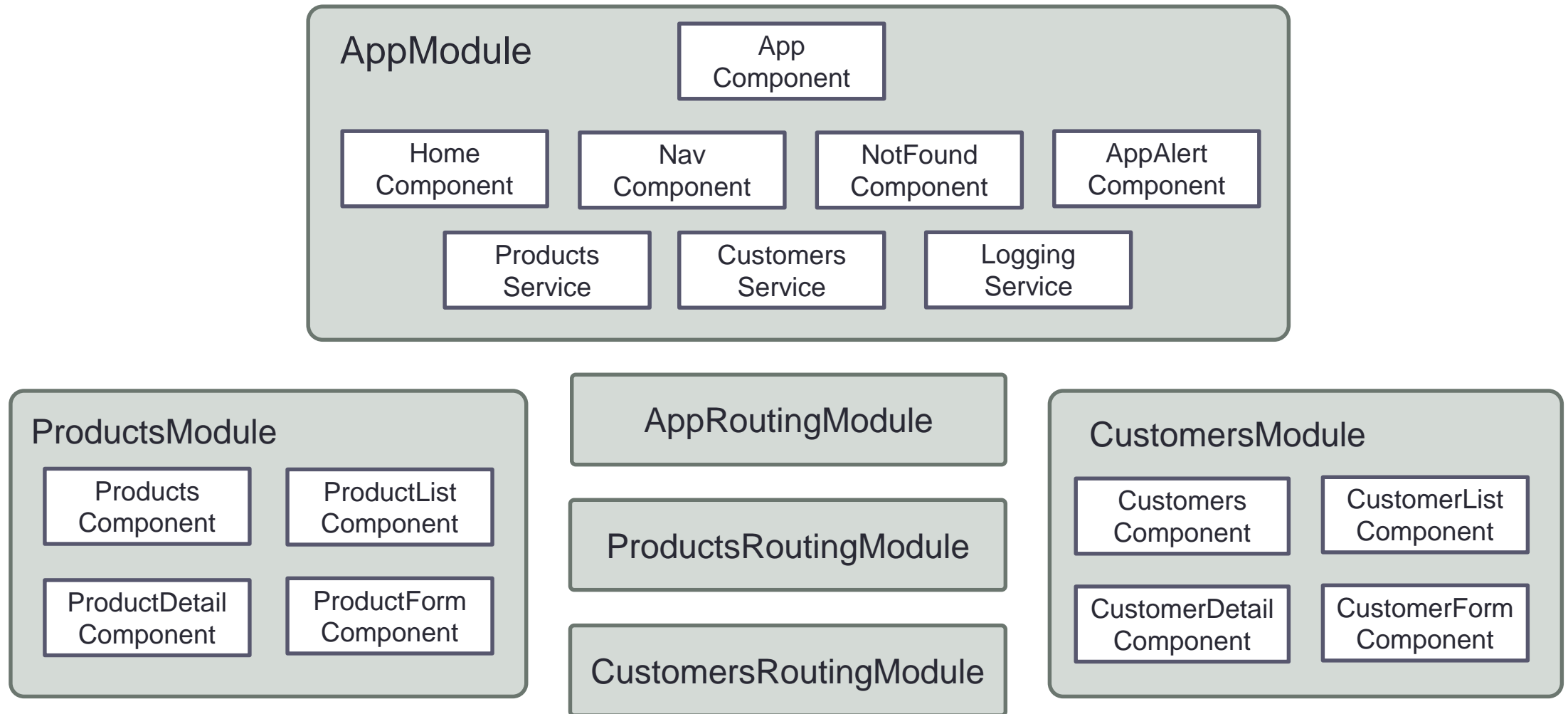
@NgModule Decorator

- Specifies metadata for an Angular module using a plain JS object
- Properties
 - declarations: []
 - specifies a list of components, directives, and pipes that belong to this module
 - imports: []
 - specifies a list of modules whose exported components, directives, pipes should be available to templates in this module
 - providers: []
 - specifies services that the components, directives and pipes can use within the app
 - bootstrap: []
 - defines the components that should be bootstrapped when this module is bootstrapped
 - defines the root component of the app
 - exports: []
 - makes some of the components, directives, and pipes public so that other module's component templates can use them

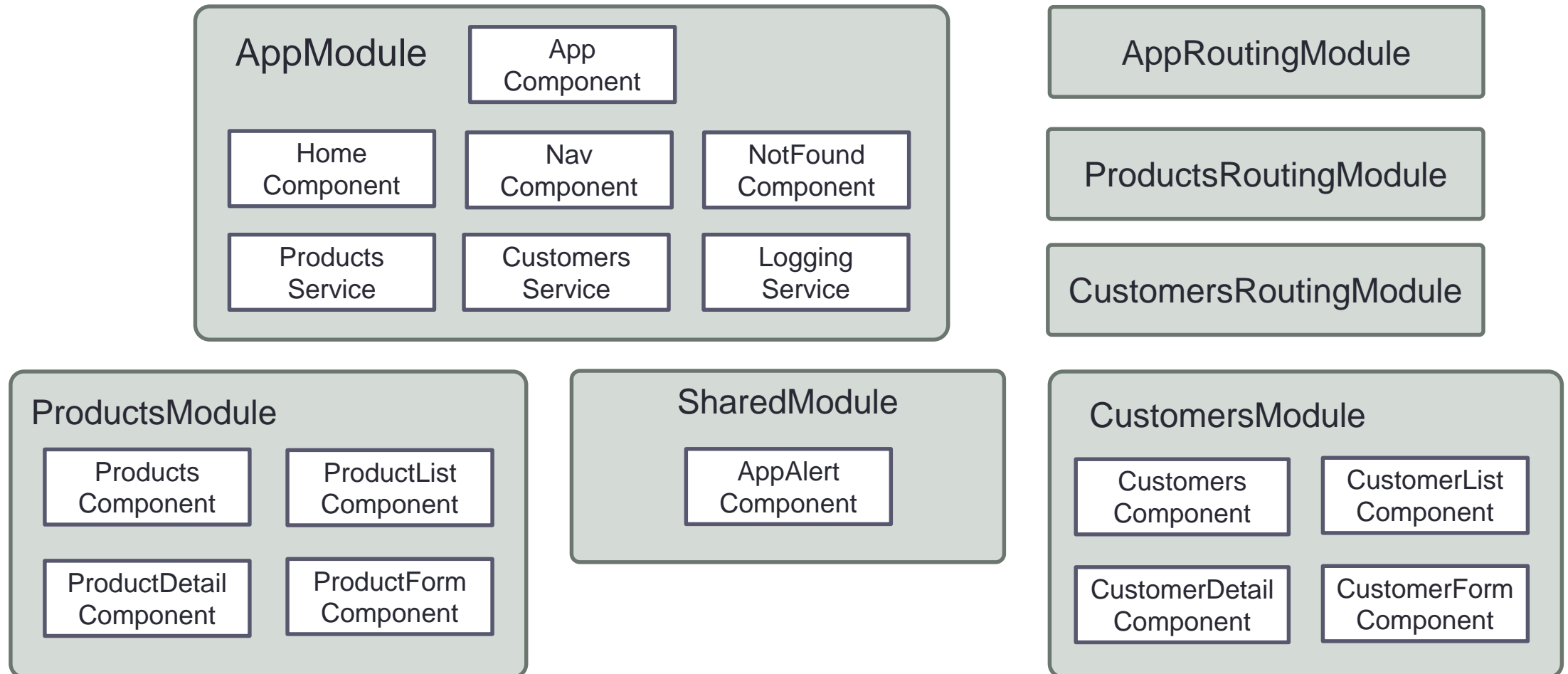
Routing Modules



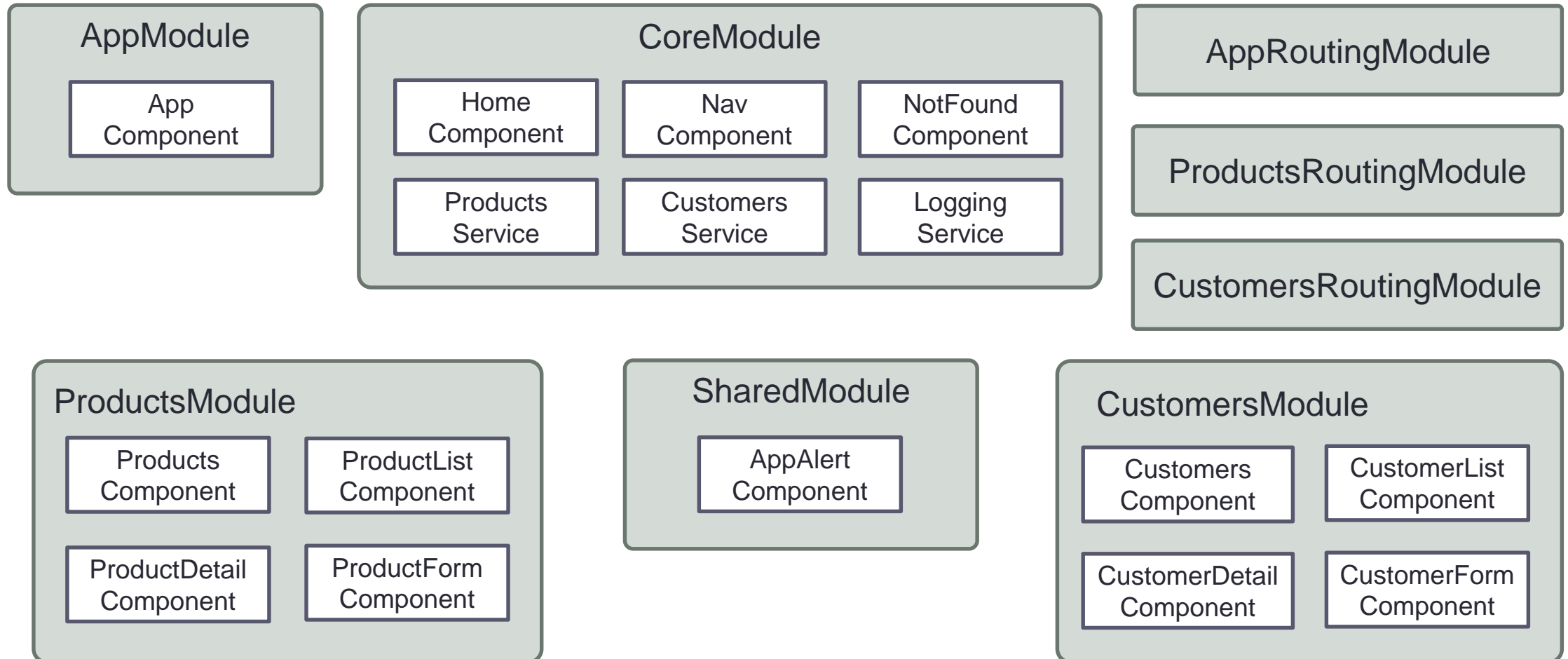
Feature Modules



Sharing Components Across Modules



Organizing Core Components of an App



Q & A

- Thank you!