Lifecycle

ngOnChanges	Called after a bound input property changes
ngOnInit	Called once the component is initialized
ngDoCheck	Called during every change detection run
ngAfterContentInit	Called after content (ng-content) has been projected into view
ngAfterContentChecked	Called every time the projected content has been checked
ngAfterViewInit	Called after the component’s view (and child views) has been initialized
ngAfterViewChecked	Called every time the view (and child views) have been checked
ngOnDestroy	Called once the component is about to be destroyed

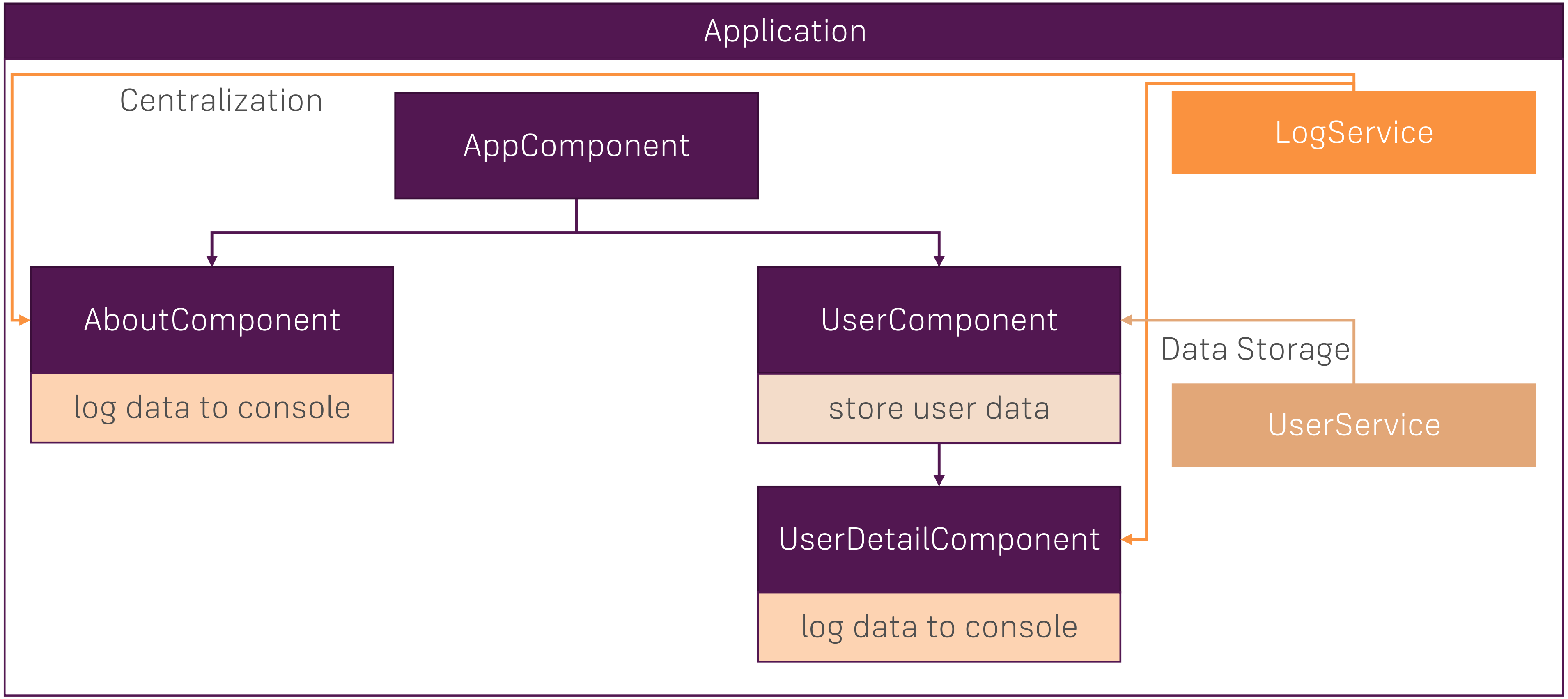
Understanding Directives

Attribute vs Structural

Attribute Directives	Structural Directives
Look like a normal HTML Attribute (possibly with databinding or event binding)	Look like a normal HTML Attribute but have a leading * (for desugaring)
Only affect/ change the element they are added to	Affect a whole area in the DOM (elements get added/ removed)

Services & Dependency Injection

What are Services?



Hierarchical Injector

AppModule	Same Instance of Service is available **Application-wide**
AppComponent	Same Instance of Service is available for **all Components** (but **not for other Services**)
Any other Component	Same Instance of Service is available for **the Component and all its child components**

Pipes

Routing

Forms

Two Approaches

Template-Driven

Angular infers the Form Object from the DOM

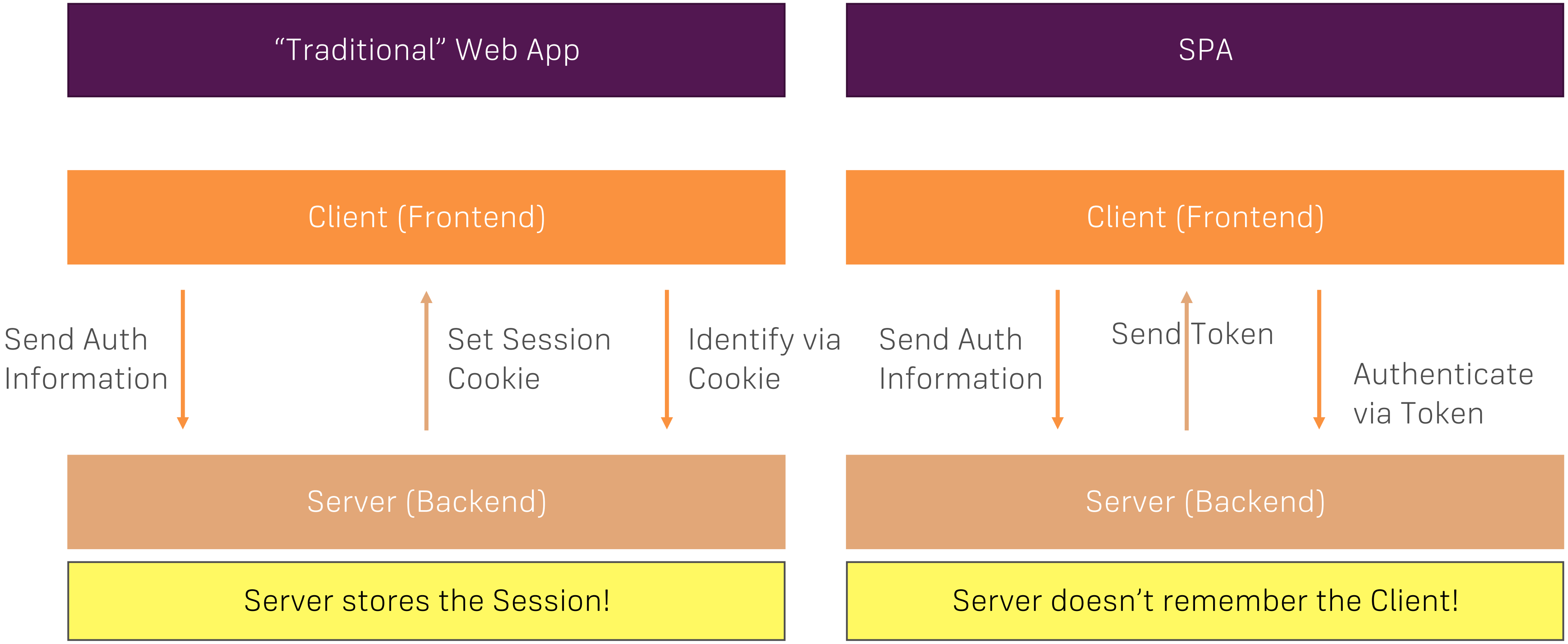
Reactive

Form is created programmatically and synchronized with the DOM

Http

Authentication

How does Authentication work?



“Traditional” Web App

Client (Frontend)

Send Auth Information

Set Session Cookie

Identify via Cookie

Server (Backend)

Server stores the Session!

SPA

Client (Frontend)

Send Auth Information

Send Token

Authenticate via Token

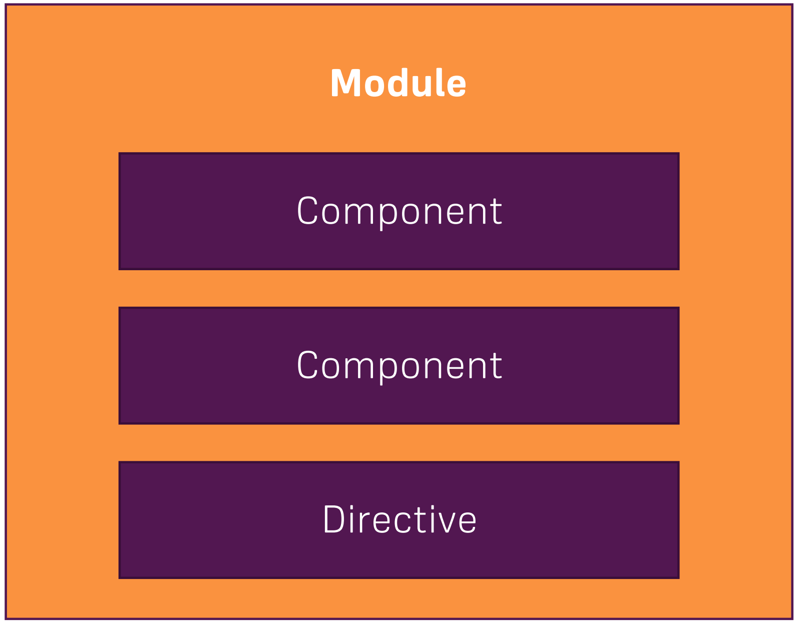
Server (Backend)

Server doesn’t remember the Client!

Animations

Using Modules & Optimizing an Angular App

The Idea behind App Modules



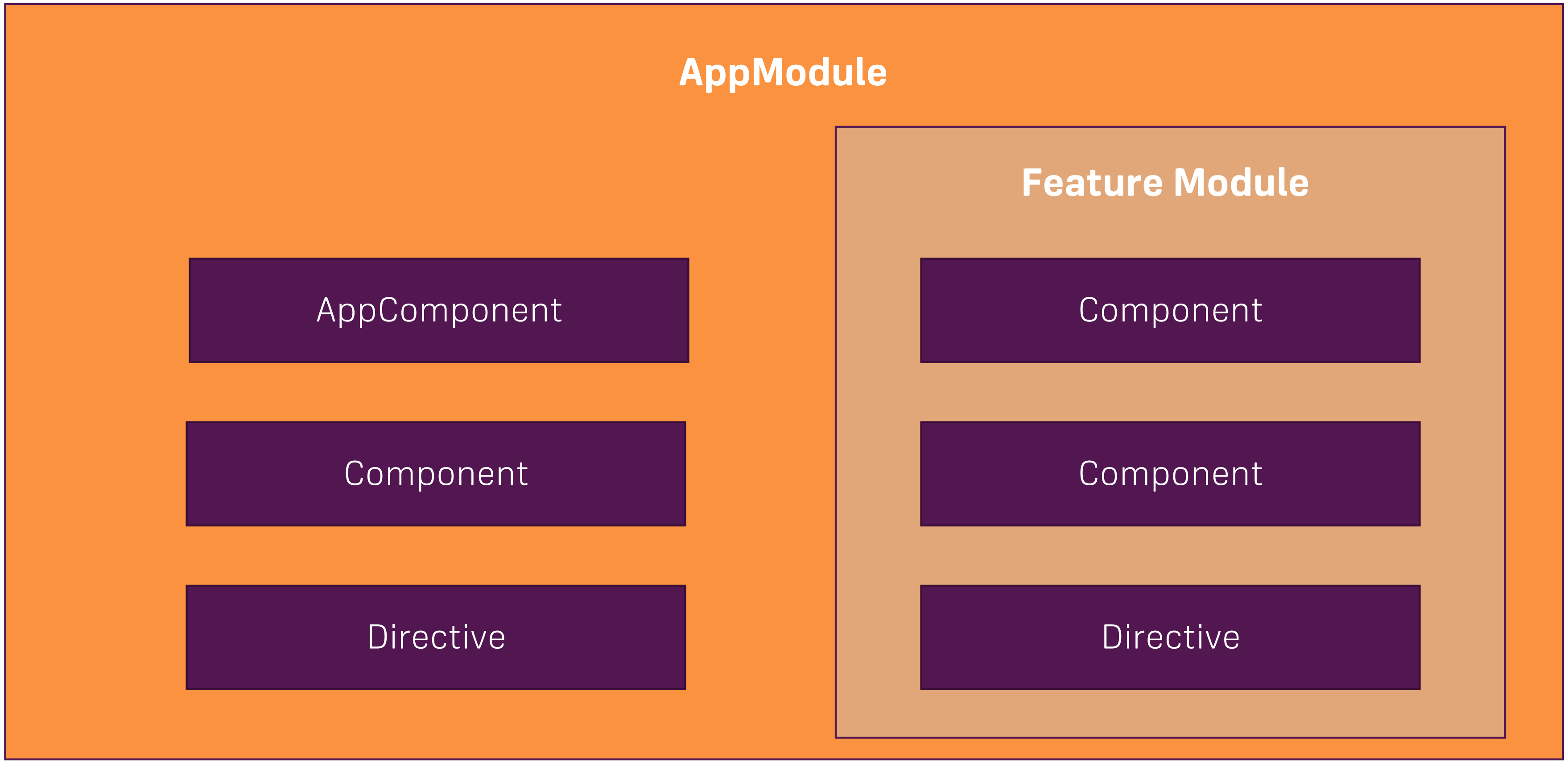
Module

Component

Component

Directive

Feature Modules



AppModule

AppComponent

Component

Directive

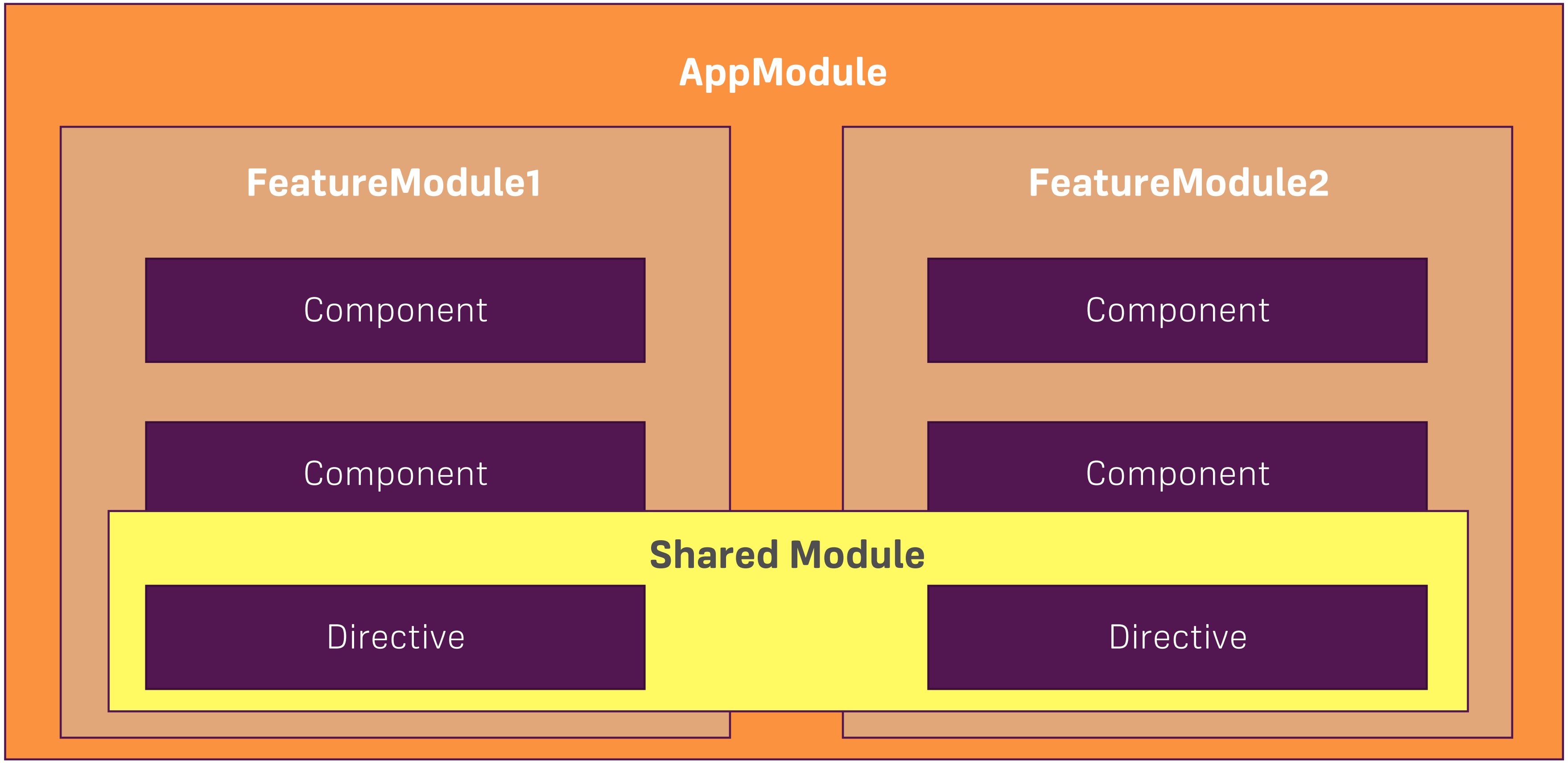
Feature Module

Component

Component

Directive

Shared Modules



AppModule

FeatureModule1

Component

Component

FeatureModule2

Component

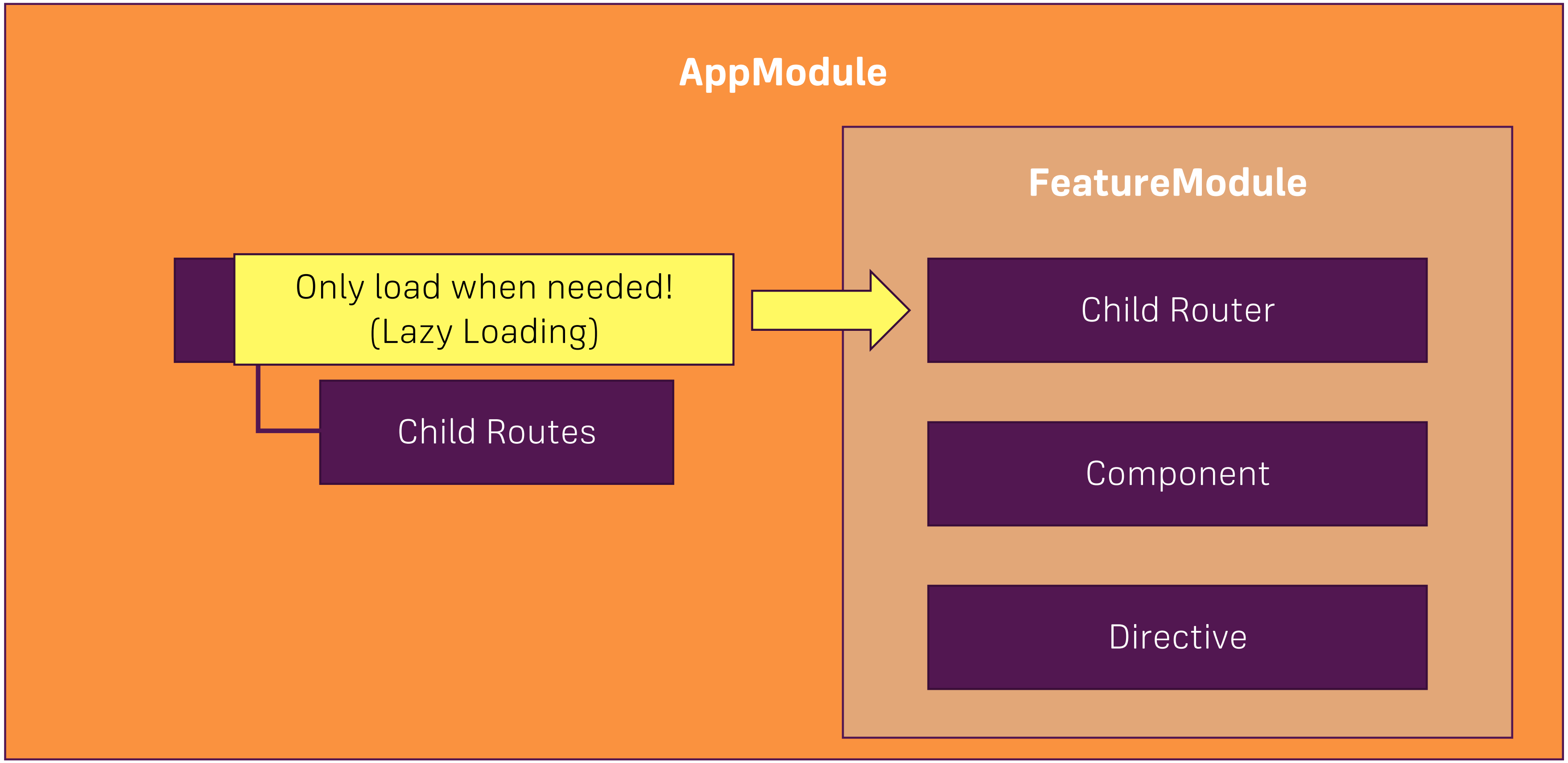
Component

Shared Module

Directive

Directive

Modules and Routing (Lazy Loading)



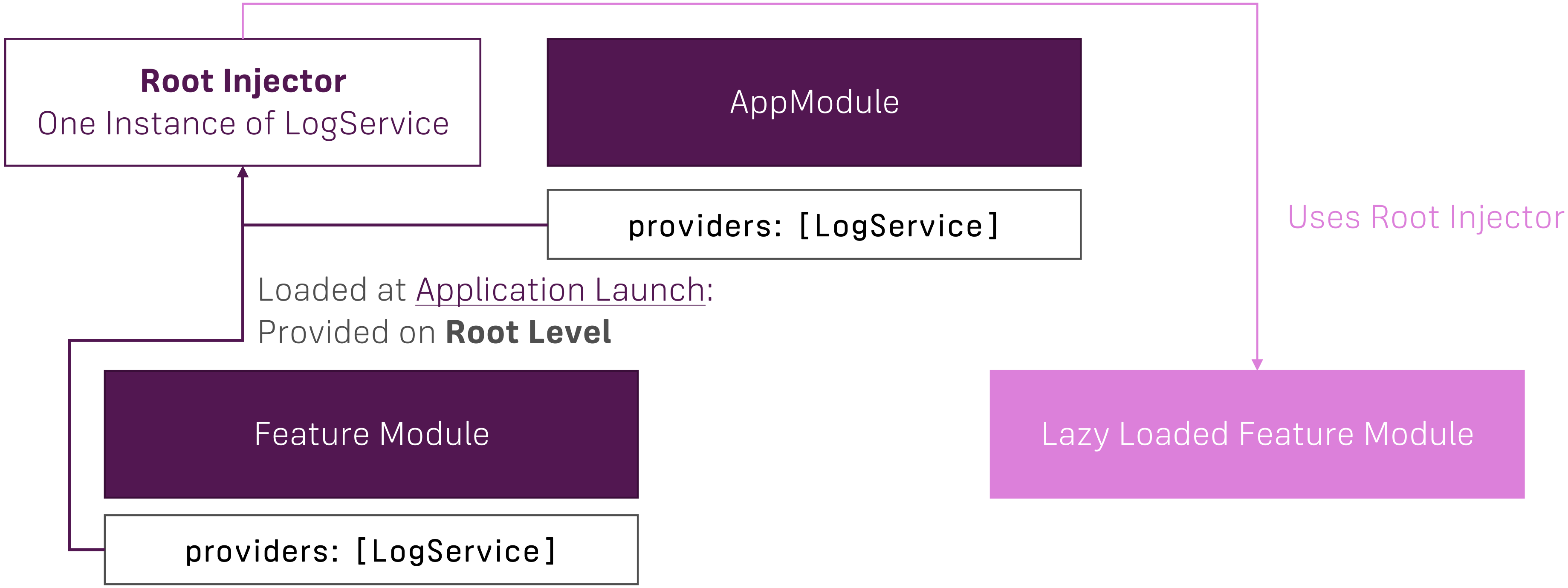
Why Modules?

Don‘t bloat the AppModule

Be clear about Who‘s responsible for What

Allows Lazy Loading of Modules

Modules and Service Injection



Root Injector
One Instance of LogService

AppModule

```
providers: [LogService]
```

Uses Root Injector

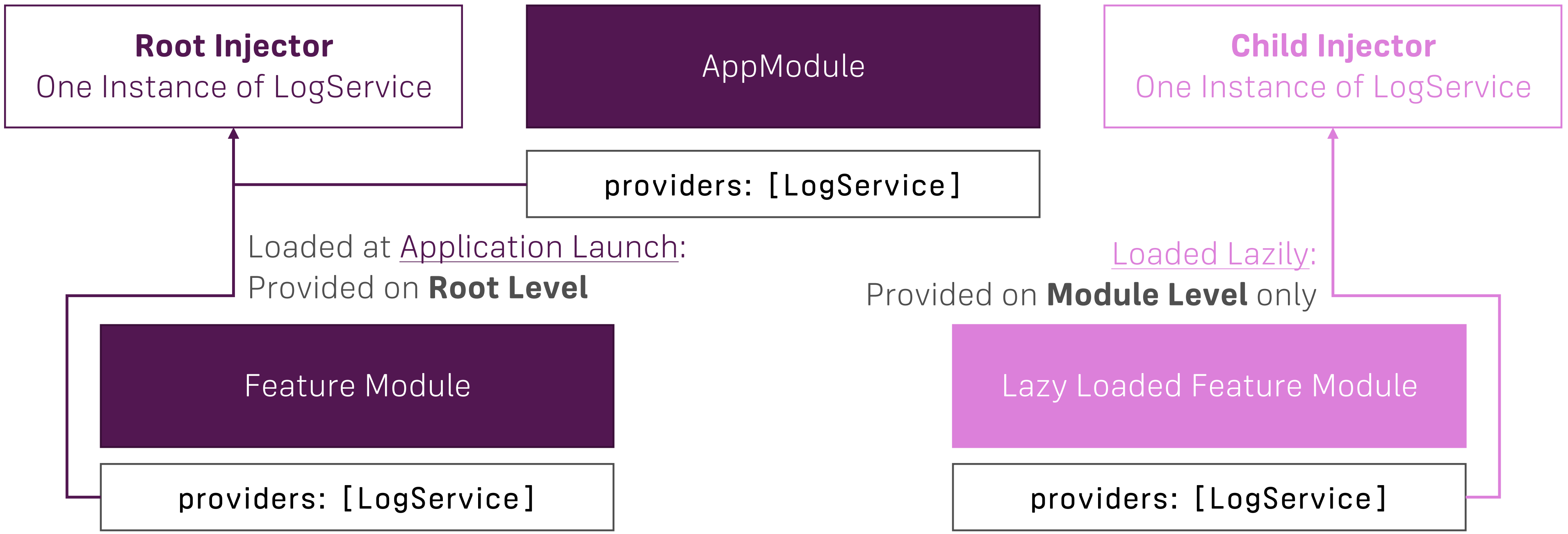
Loaded at Application Launch:
Provided on **Root Level**

Feature Module

```
providers: [LogService]
```

Lazy Loaded Feature Module

Modules and Service Injection



Root Injector
One Instance of LogService

AppModule

providers: [LogService]

Child Injector
One Instance of LogService

Loaded at Application Launch:
Provided on **Root Level**

Loaded Lazily:
Provided on **Module Level** only

Feature Module

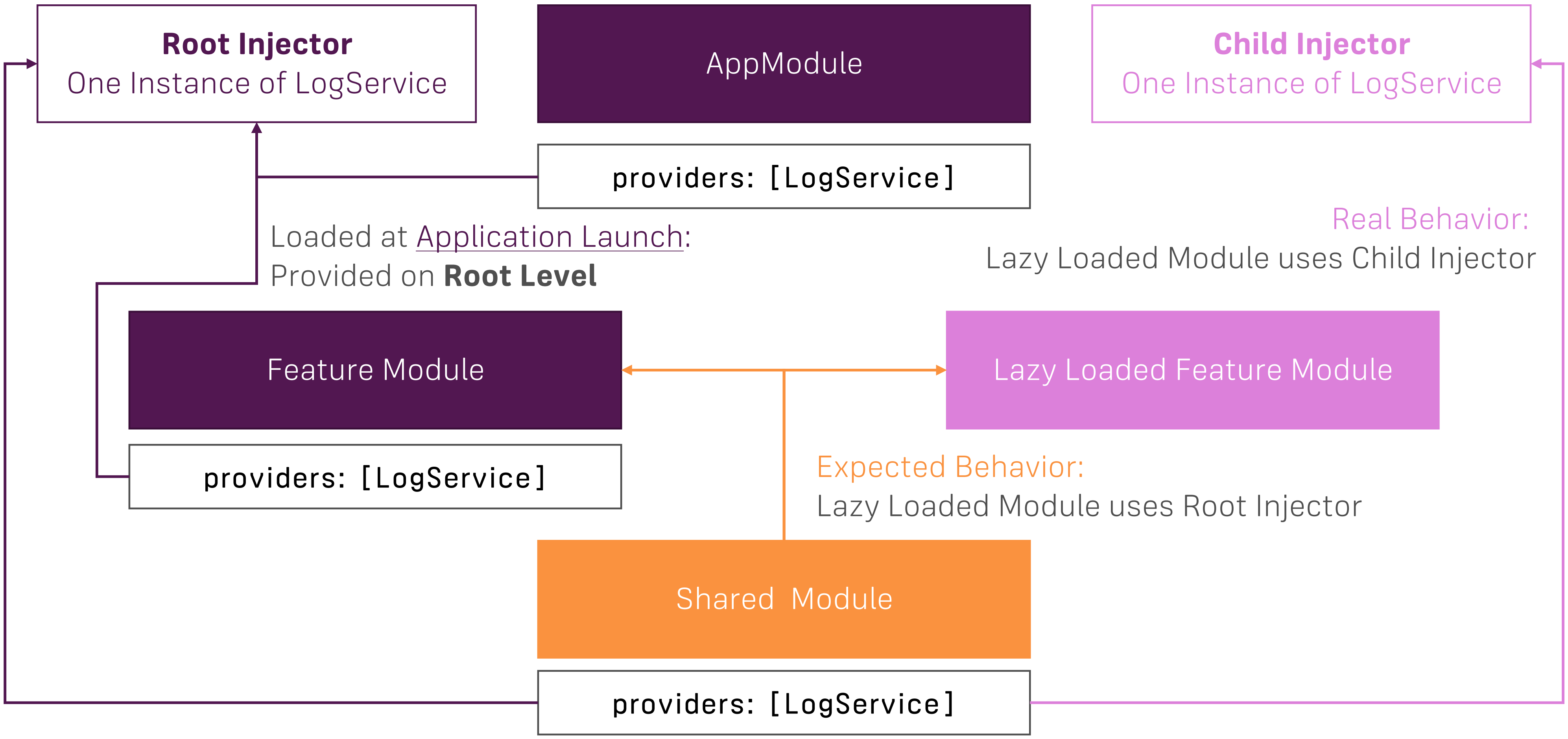
providers: [LogService]

Lazy Loaded Feature Module

providers: [LogService]

Enforce "Module Scope" by providing in a Component instead of a Module!

Modules and Service Injection



Root Injector
One Instance of LogService

AppModule

providers: [LogService]

Child Injector
One Instance of LogService

Loaded at Application Launch:
Provided on **Root Level**

Real Behavior:
Lazy Loaded Module uses Child Injector

Feature Module

providers: [LogService]

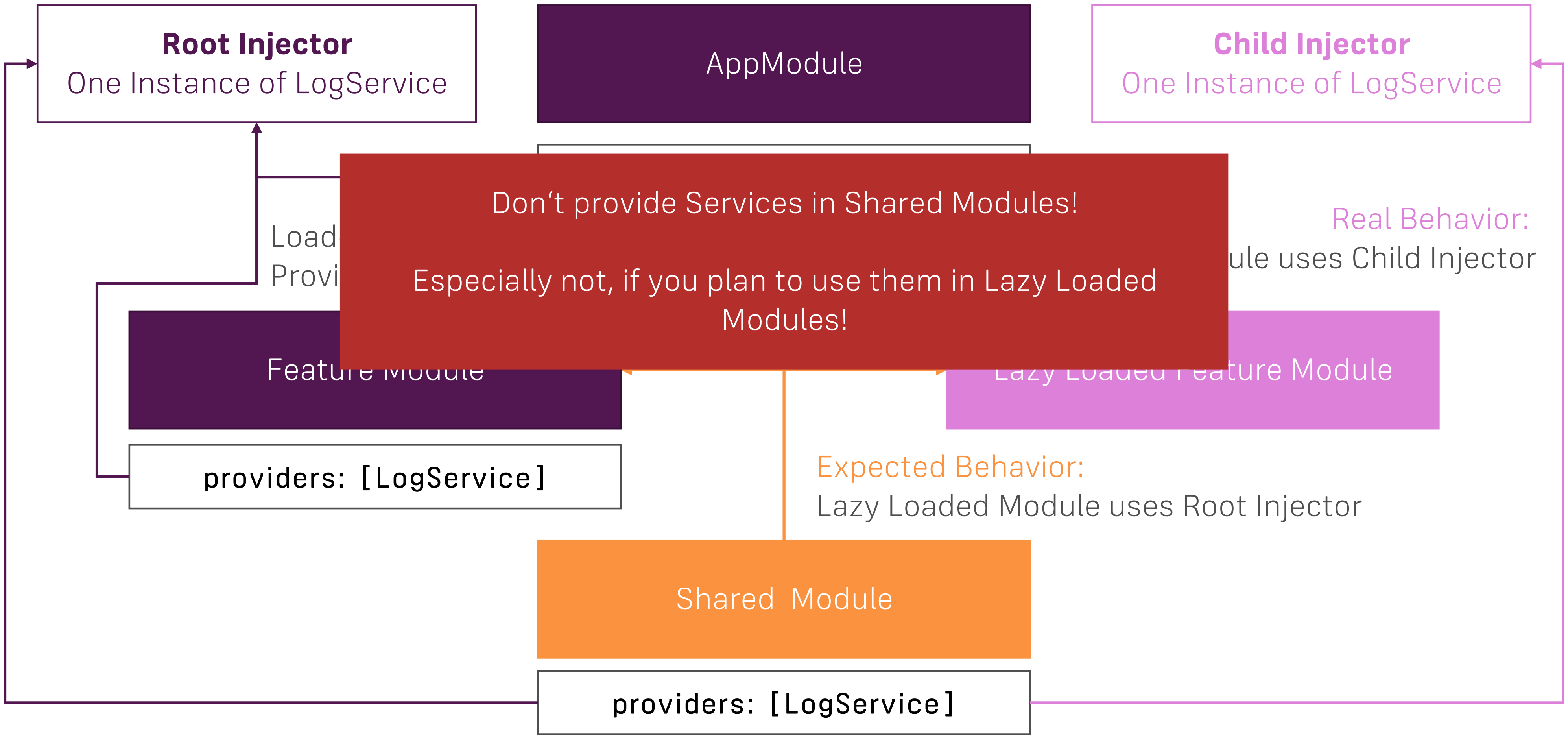
Lazy Loaded Feature Module

Expected Behavior:
Lazy Loaded Module uses Root Injector

Shared Module

providers: [LogService]

Modules and Service Injection



Root Injector
One Instance of LogService

AppModule

Child Injector
One Instance of LogService

Don‘t provide Services in Shared Modules!

Especially not, if you plan to use them in Lazy Loaded Modules!

Load

Provi

Real Behavior:

ule uses Child Injector

Feature Module

providers: [LogService]

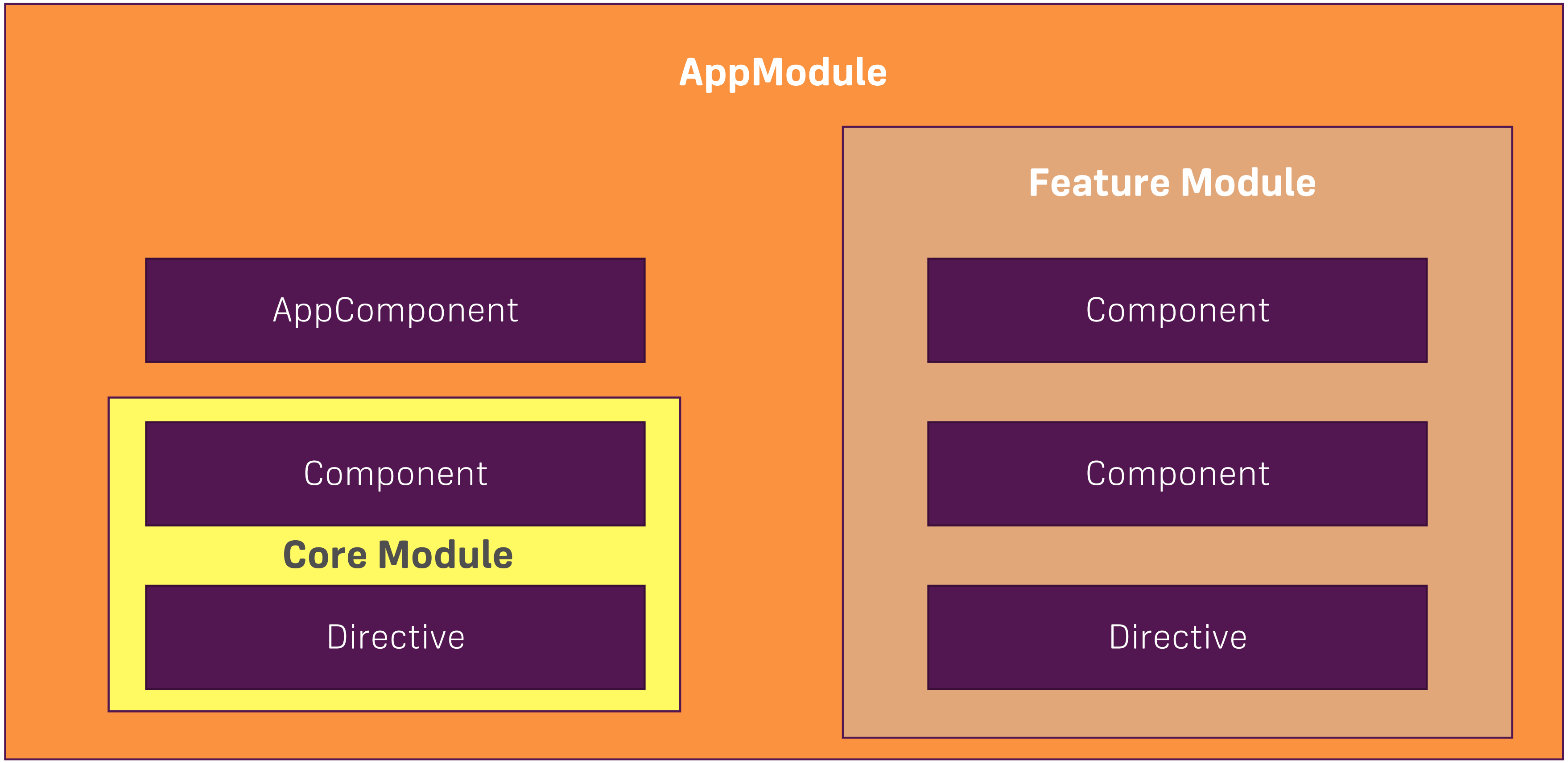
Lazy Loaded Feature Module

Expected Behavior:
Lazy Loaded Module uses Root Injector

Shared Module

providers: [LogService]

Core Module



AppModule

AppComponent

Component

Core Module

Directive

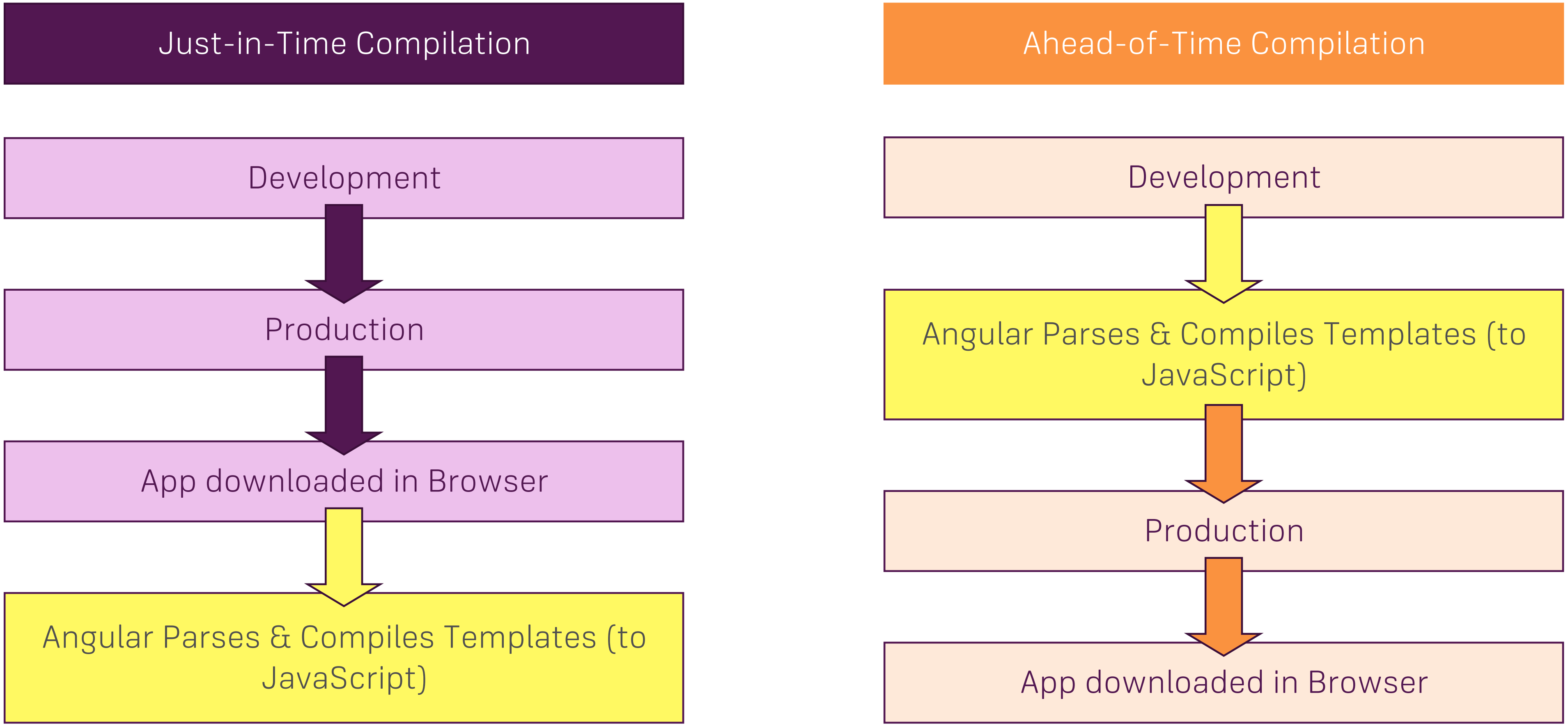
Feature Module

Component

Component

Directive

Ahead-of-Time Compilation



Just-in-Time Compilation

Development

Production

App downloaded in Browser

Angular Parses & Compiles Templates (to JavaScript)

Ahead-of-Time Compilation

Development

Angular Parses & Compiles Templates (to JavaScript)

Production

App downloaded in Browser

Advantages of AoT Compilation

Faster Startup sind Parsing and Compilation doesn‘t happen in Browser

Templates get checked during Development

Smaller File Size as unused Features can be stripped out and the Compiler itself isn‘t shipped

Testing

Deployment

Deployment Steps & Things to Keep in Mind

Build your App for Production

Consider AoT Compilation

Set the correct <base> element

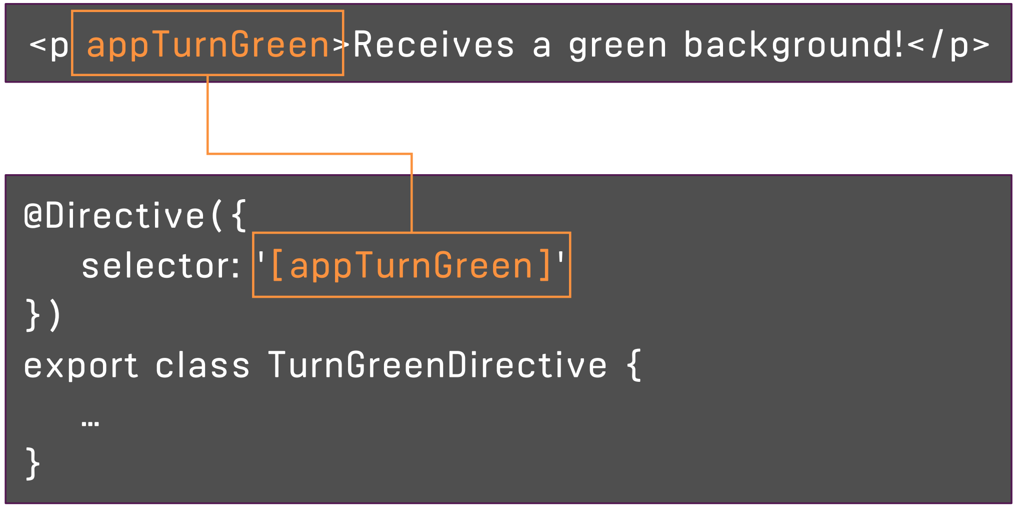
For example.com/my-app you should have <base href="/my-app/">

Make sure your Server ALWAYS returns index.html

Routes are registered in Angular App, so the server won't know your routes! Return index.html in case of 404 errors!

What are Directives?

Directives are Instructions in the DOM!

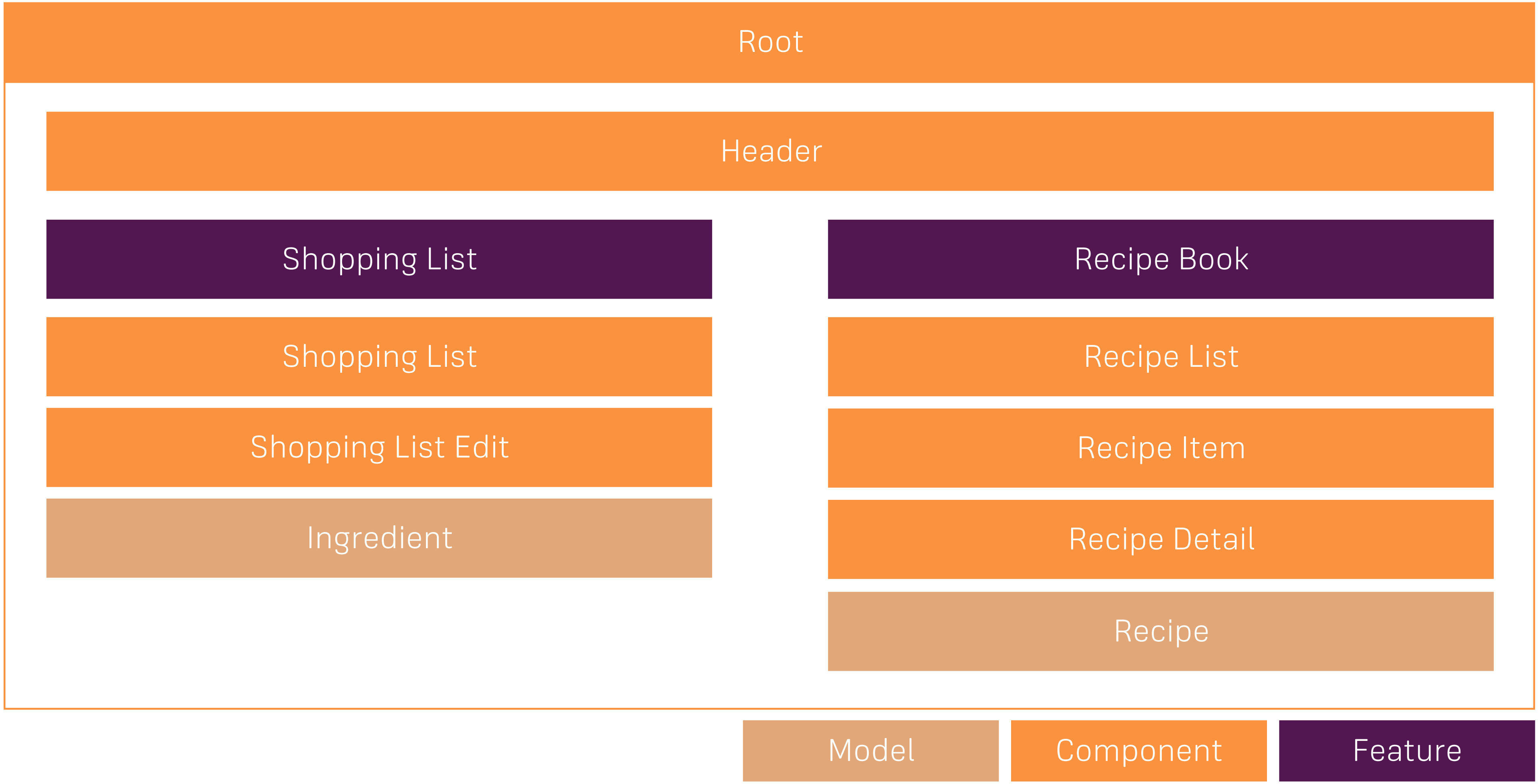


```
<p appTurnGreen>Receives a green background!</p>
```

```
@Directive({
   selector: '[appTurnGreen]'
})
export class TurnGreenDirective {
   …
}
```

Course Project

Planning the App



Root

Header

Shopping List

Shopping List

Shopping List Edit

Ingredient

Recipe Book

Recipe List

Recipe Item

Recipe Detail

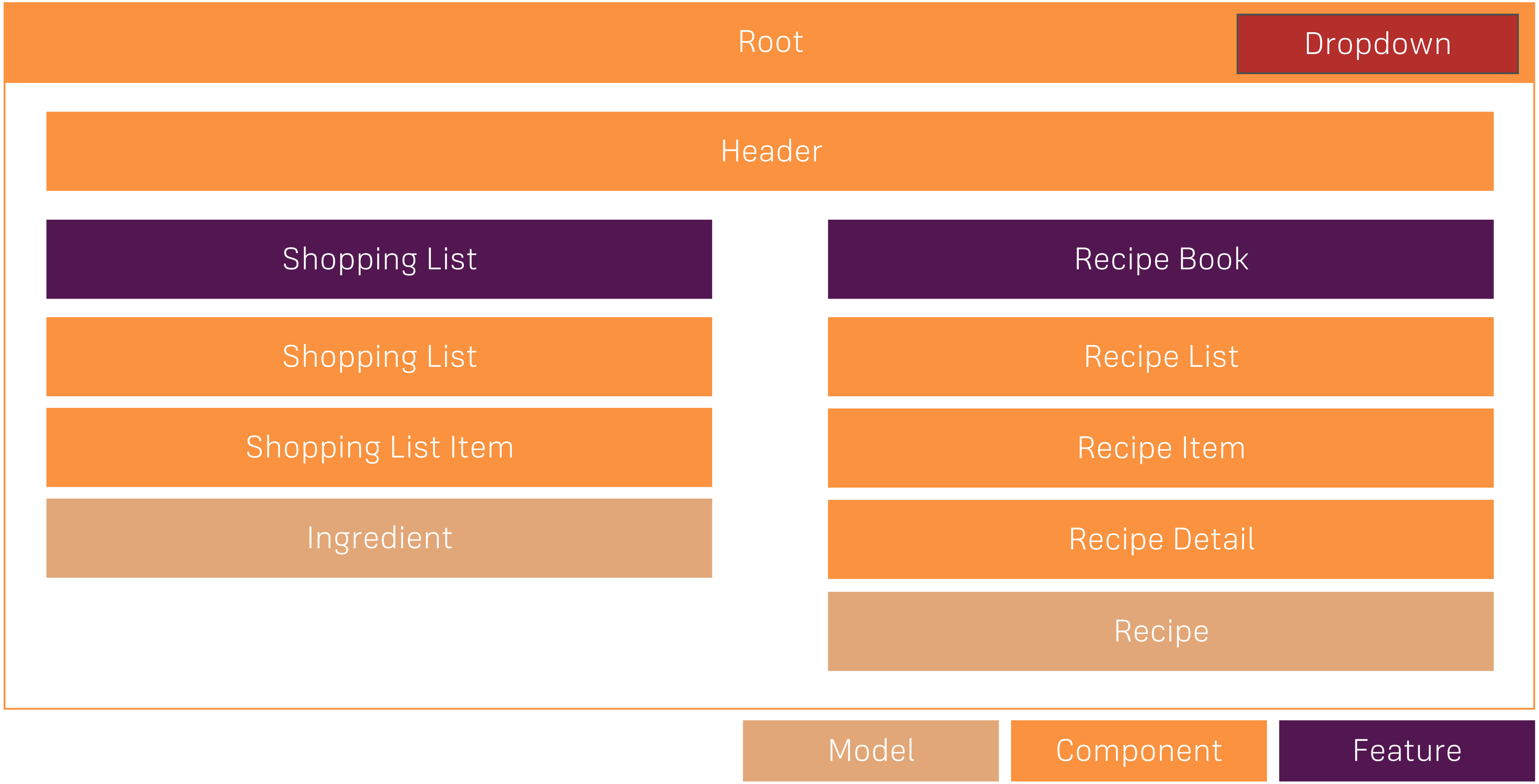
Recipe

Model

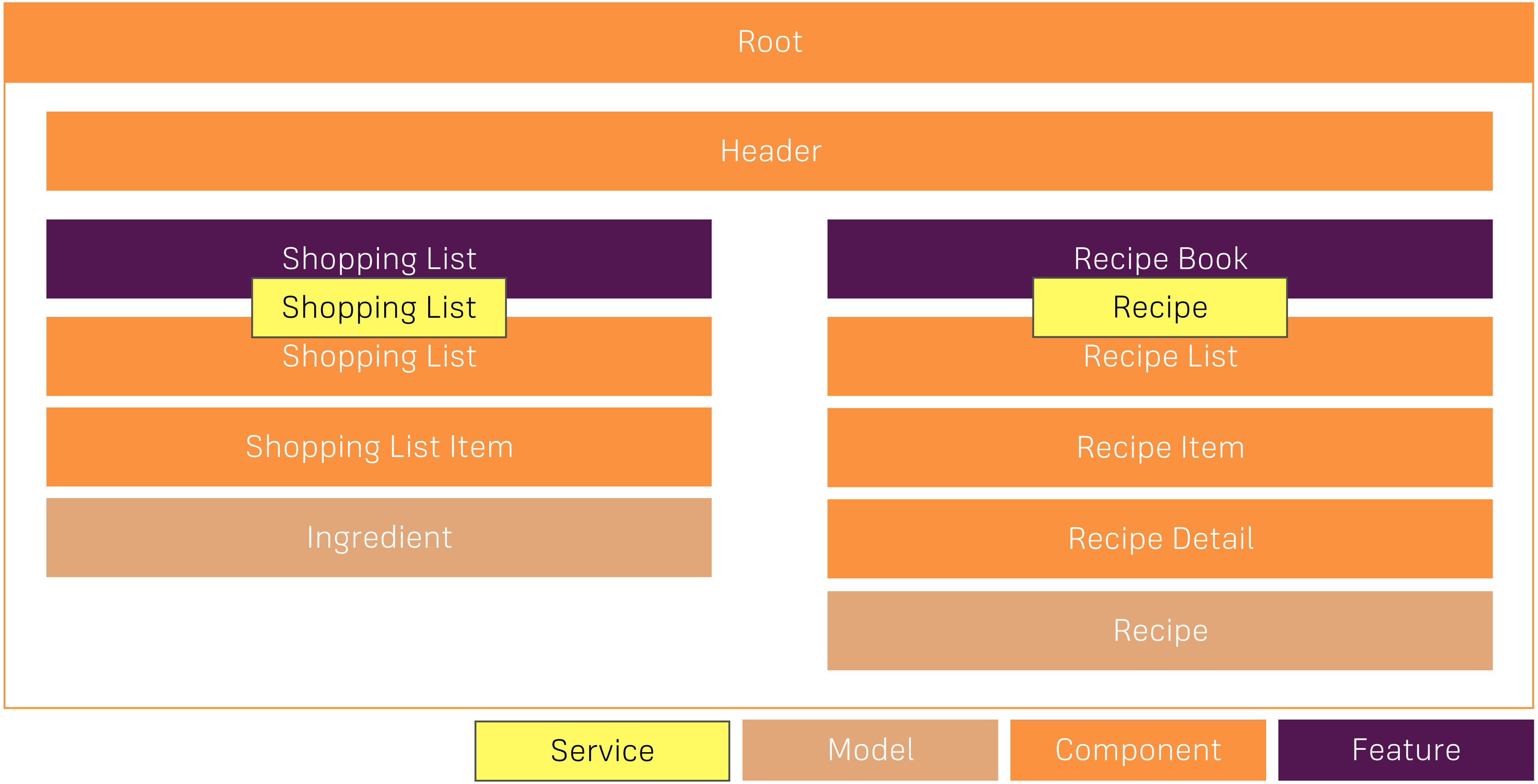
Component

Feature

Adding Directives



Adding Services



Root

Header

Shopping List

Shopping List

Shopping List

Shopping List Item

Ingredient

Recipe Book

Recipe

Recipe List

Recipe Item

Recipe Detail

Recipe

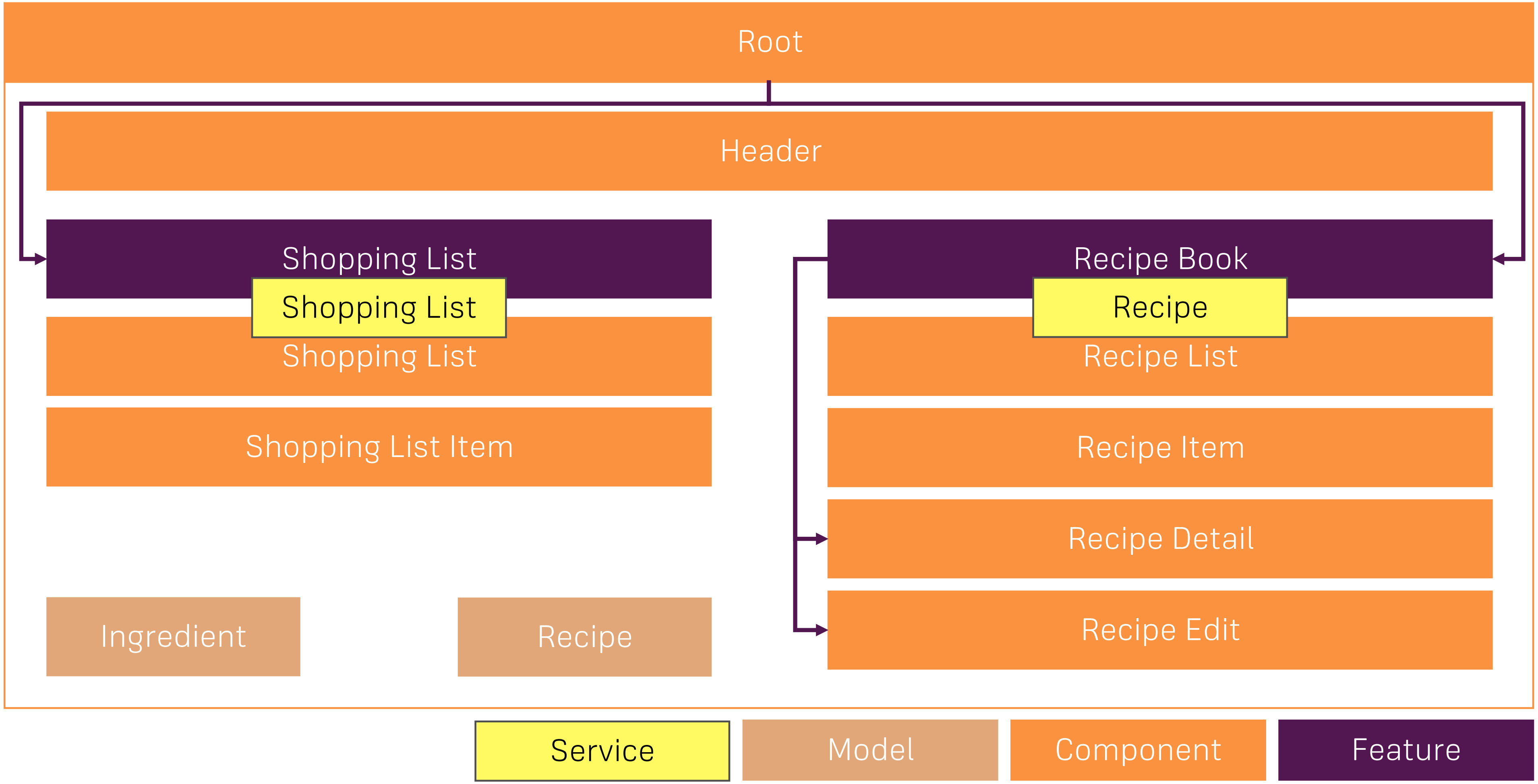
Service

Model

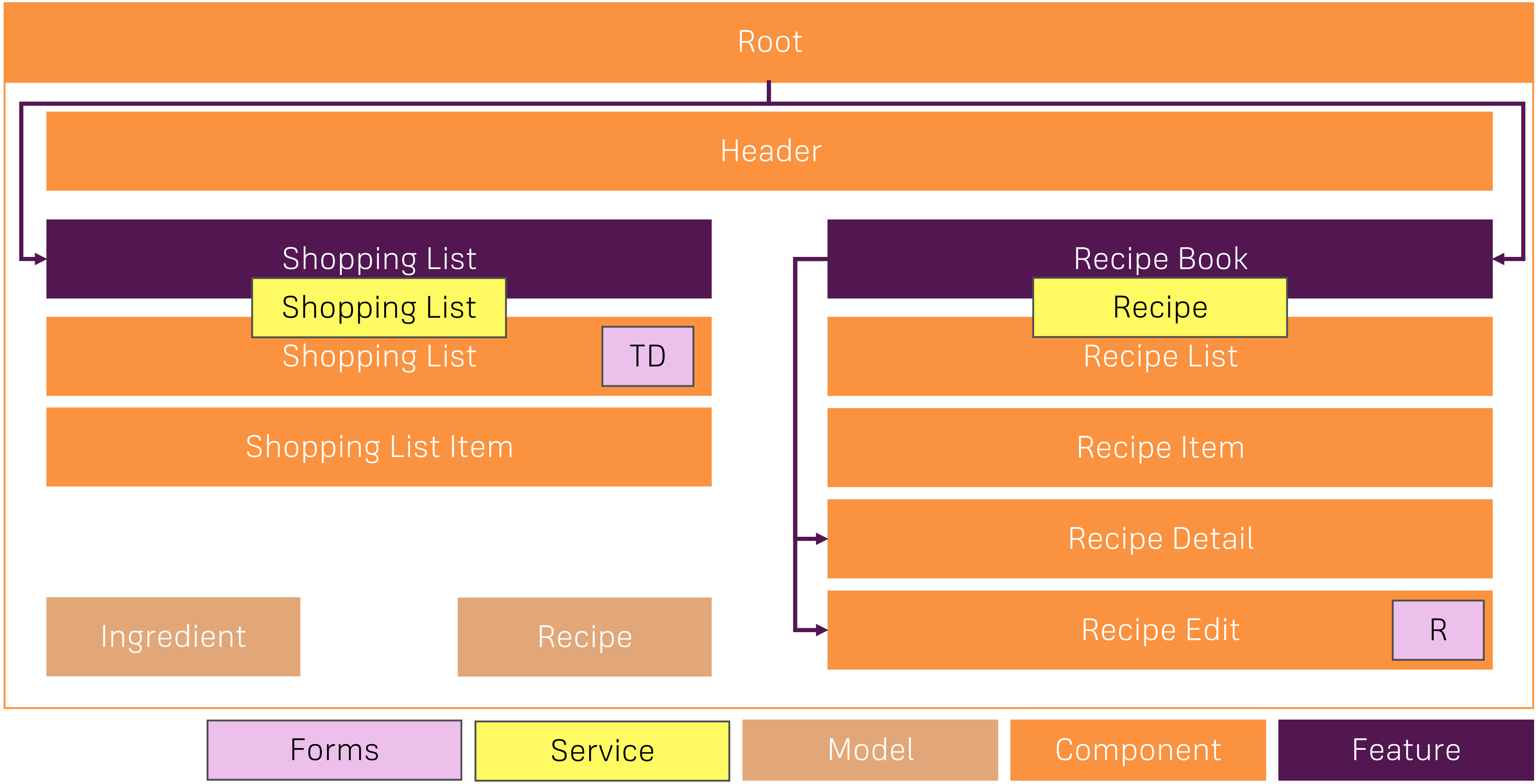
Component

Feature

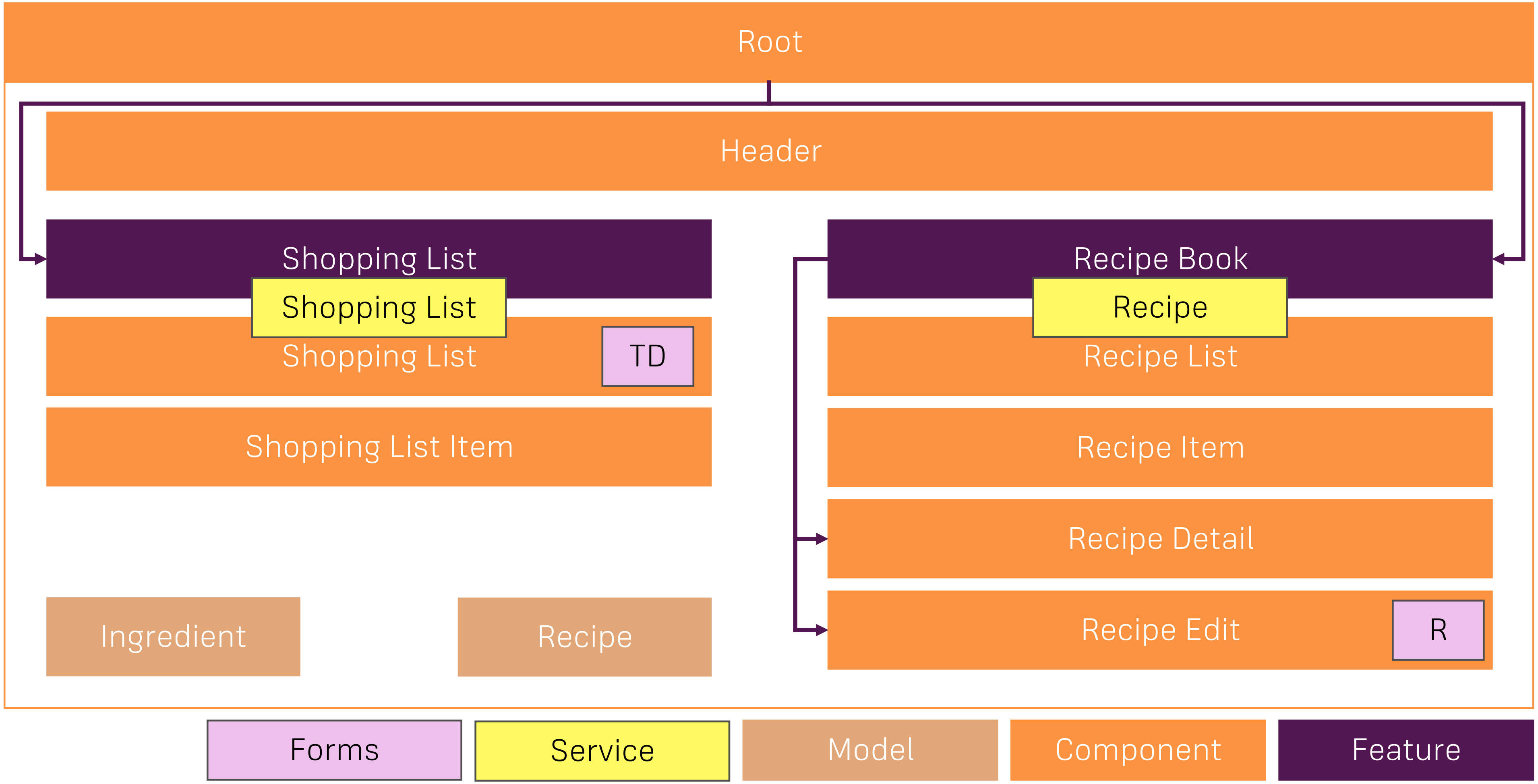
Adding Routing



Adding Forms



Adding Modules



What changed in the Course Update?

Angular 4 is NOT a re-write of Angular 2!

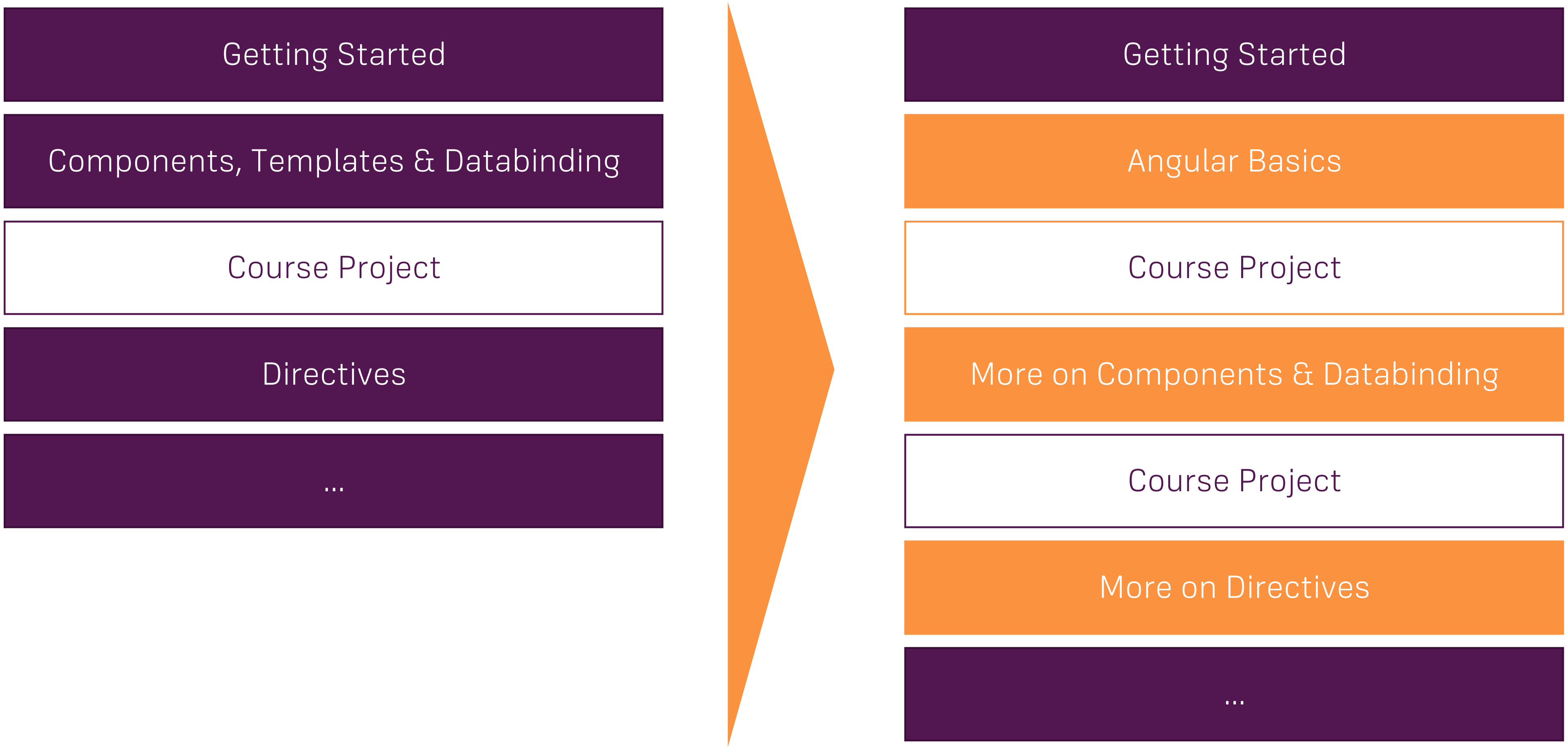
Angular 4 is only an update to Angular 2 (now only "Angular")

Angular Team switched to Semantic Versioning and renamed Angular 2 to Angular
(Angular 1 will be AngularJS)

Semantic Versioning

MAJOR.MINOR.BUGFIX
e.g. 2.4.3
e.g. 4.0.1

Basics First



Getting Started

Components, Templates & Databinding

Course Project

Directives

...

Getting Started

Angular Basics

Course Project

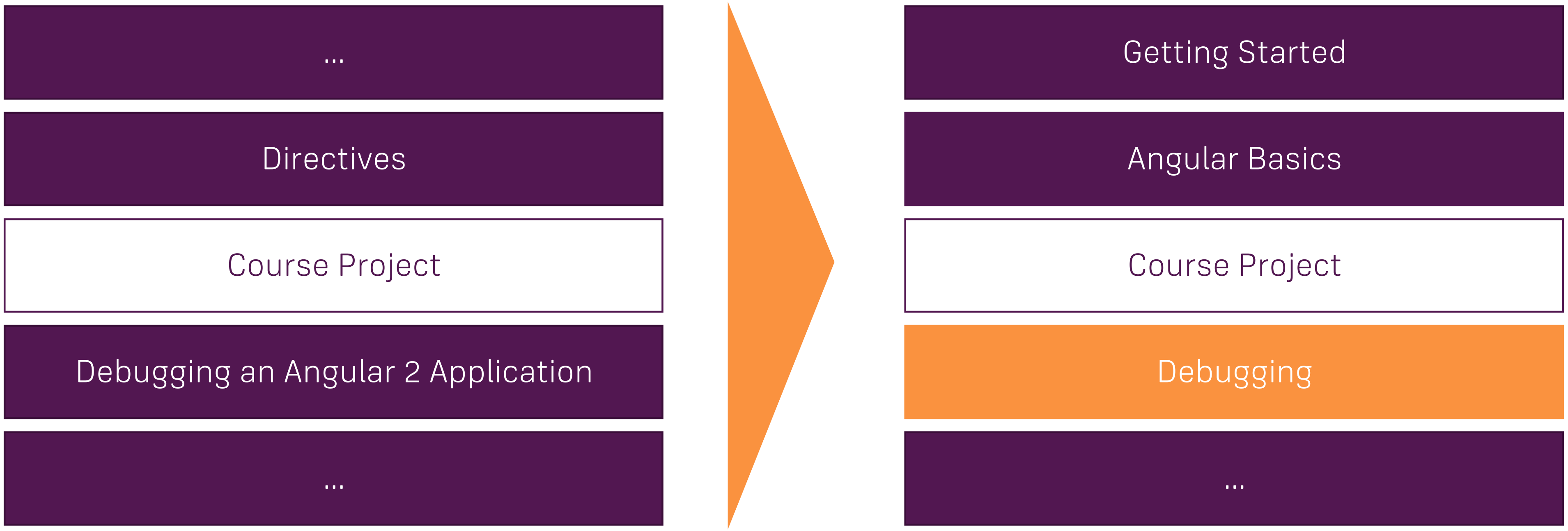
More on Components & Databinding

Course Project

More on Directives

...

Debug Eearlier



...

Directives

Course Project

Debugging an Angular 2 Application

...

Getting Started

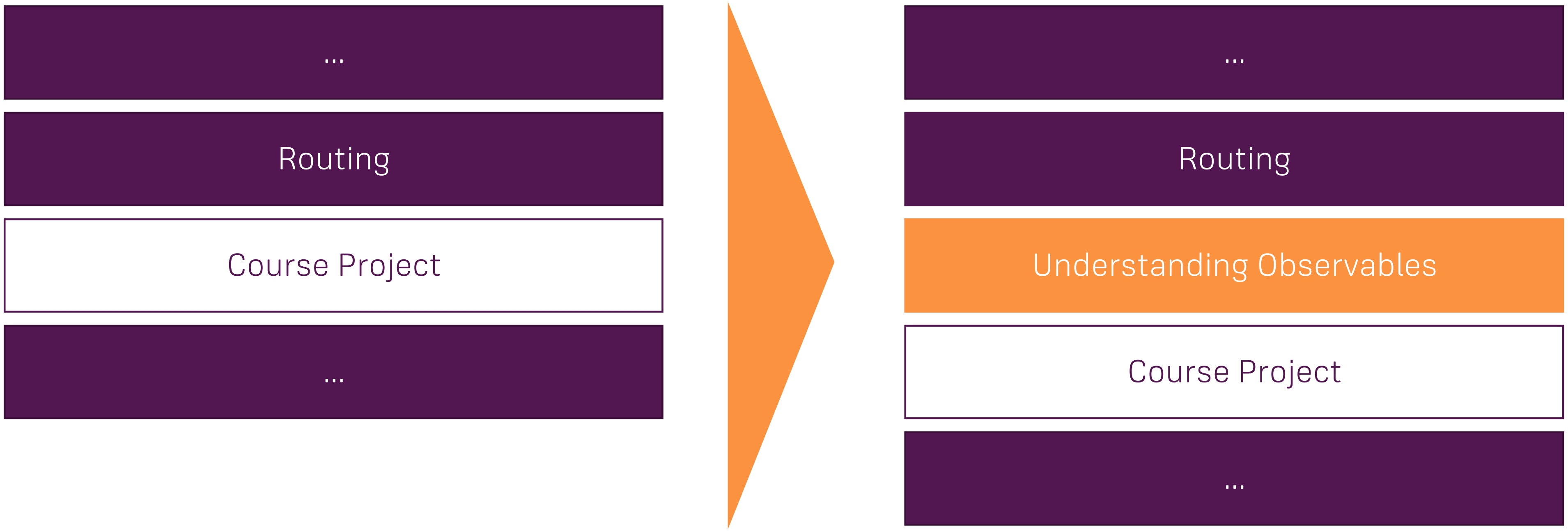
Angular Basics

Course Project

Debugging

...

Observables



...

Routing

Course Project

...

...

Routing

Understanding Observables

Course Project

...

Improved Content

Tried to **sharpen Explanations** in **all Modules**

Added a **Basics** Section to learn the Core Features Faster

More Details in **Routing**, **Forms** **and** **Optimizations** Module

Changed **Authentication** Module to now use the **Recipe Book**

The **Deployment** Section now uses the **Recipe Book** as an Example

How to Continue

"Old Content" is kept around until end of April
(I will inform you before it‘s removed!)

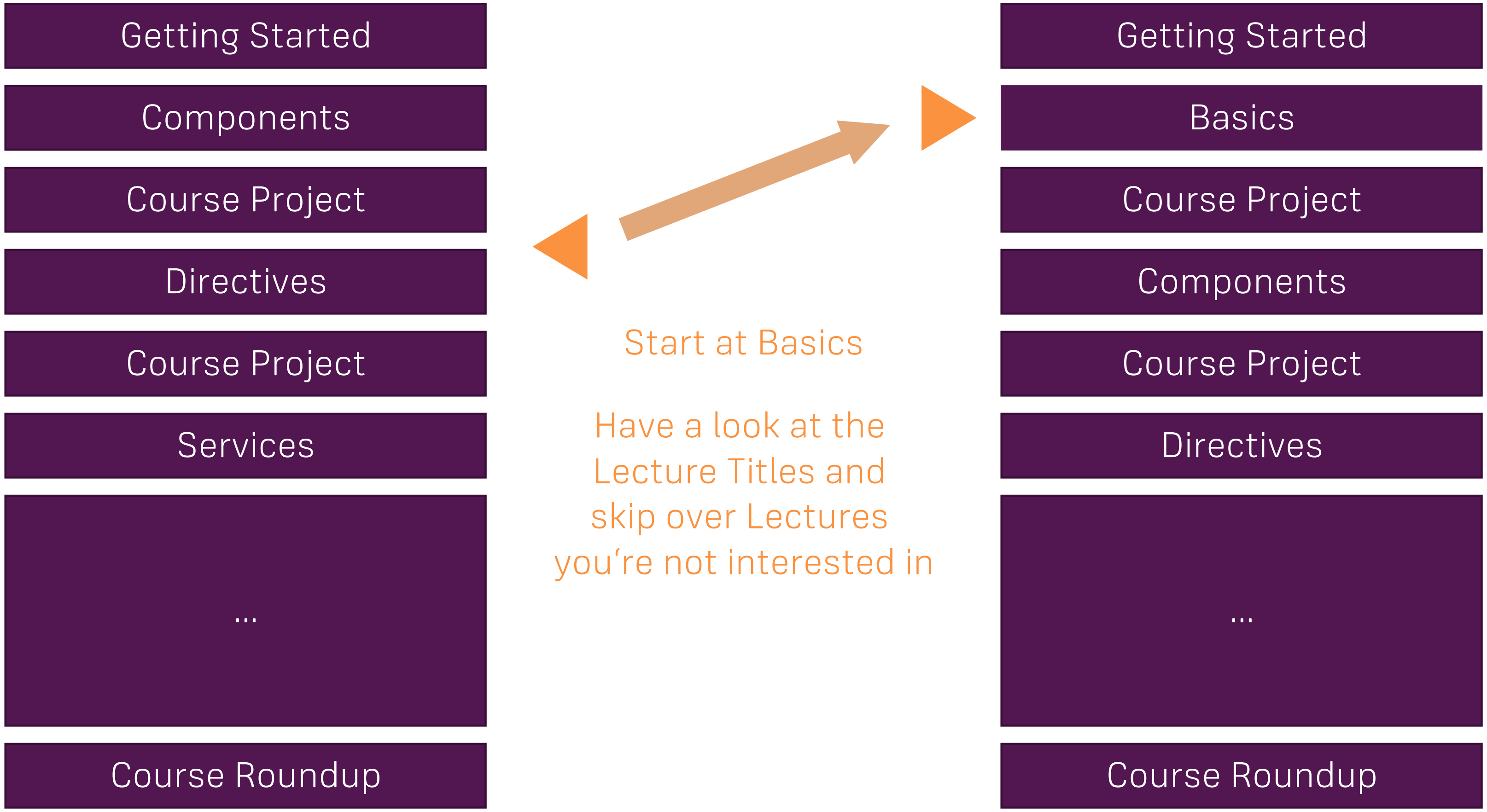
Download "old" lectures!

Keep in mind: Angular 2 Syntax = Angular 4 Syntax

The "old Content" is still correct!

You don‘t have to go through the updated Content at all!

How to Continue (if you want to use the new Content)



Getting Started

Components

Course Project

Directives

Course Project

Services

...

Course Roundup

Start at Basics

Have a look at the Lecture Titles and skip over Lectures you‘re not interested in

Getting Started

Basics

Course Project

Components

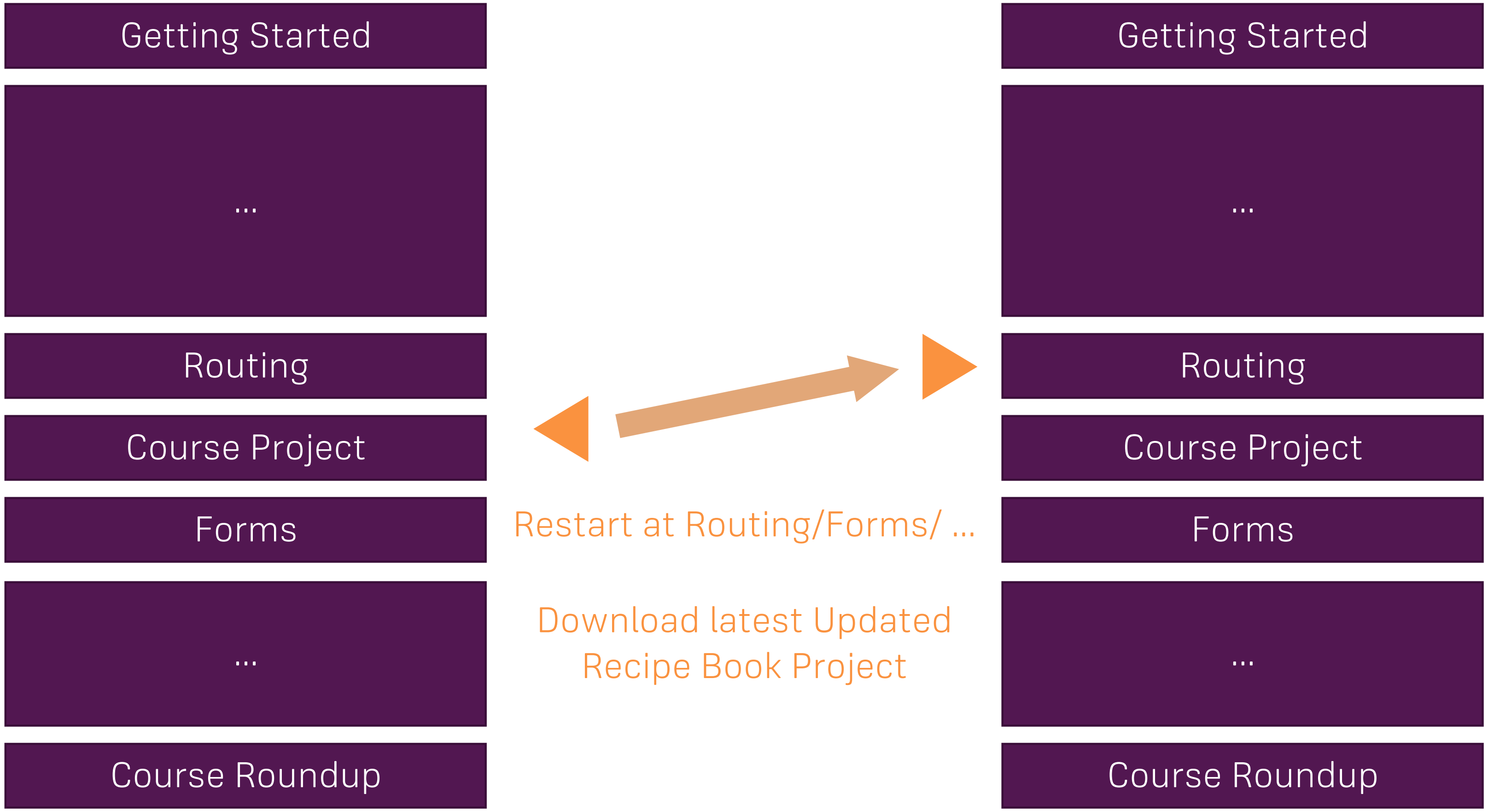
Course Project

Directives

...

Course Roundup

How to Continue (if you want to use the new Content)



Getting Started

...

Routing

Course Project

Forms

...

Course Roundup

Restart at Routing/Forms/ ...

Download latest Updated Recipe Book Project

Getting Started

...

Routing

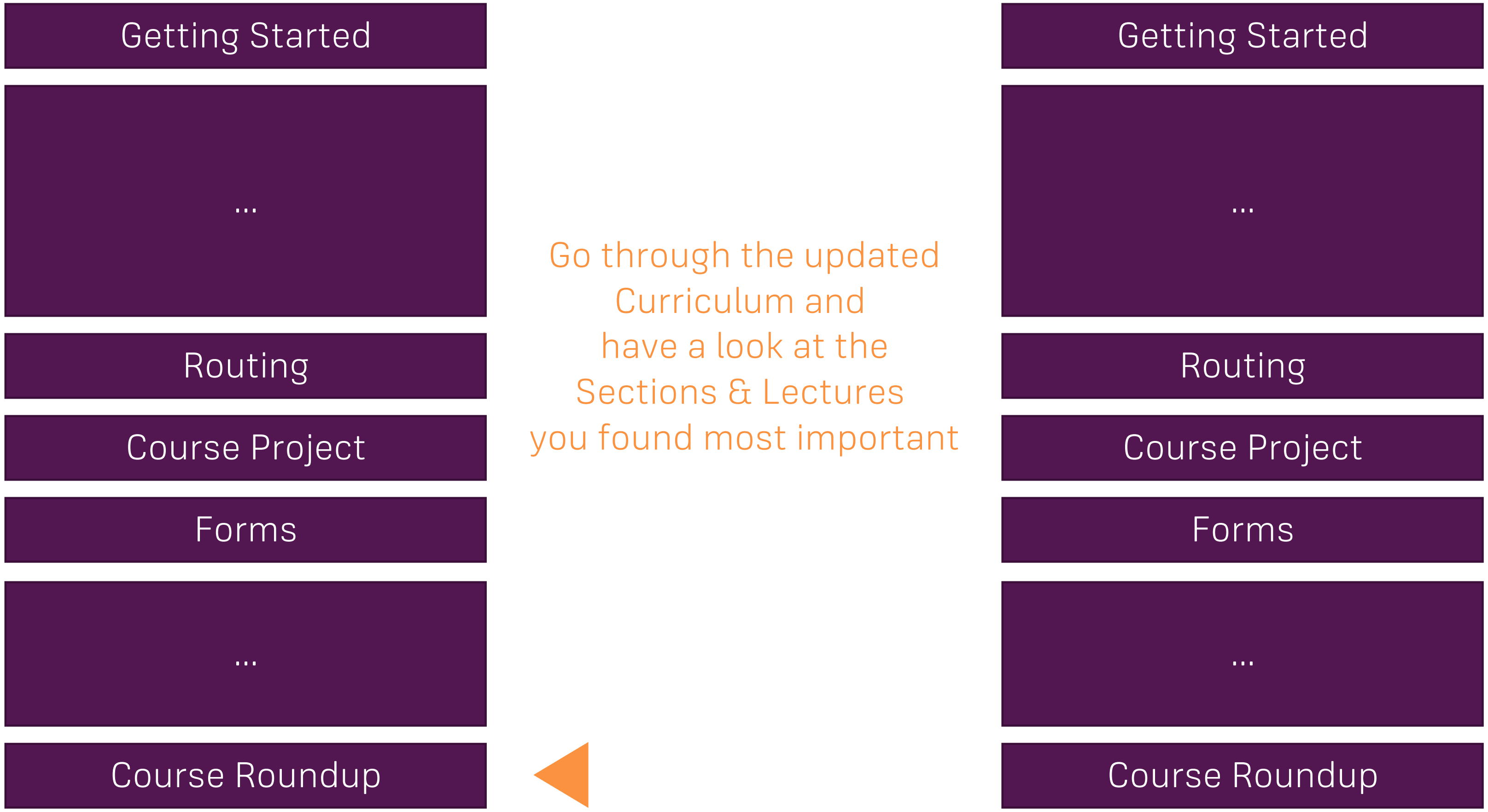
Course Project

Forms

...

Course Roundup

How to Continue (if you want to use the new Content)



Getting Started

...

Routing

Course Project

Forms

...

Course Roundup

Go through the updated Curriculum and have a look at the Sections & Lectures you found most important

Getting Started

...

Routing

Course Project

Forms

...

Course Roundup