Why frameworks ?

BUT FIRST... WHAT IS A FRAMEWORK ?

A software framework provides a generic software structure, which can be customized.

WHY FRAMEWORKS?

01

Manual data updates and DOM manipulation can get complex

02

They use well defined application architectures

– Model View Controller / Model View View Model / Model View Whatever

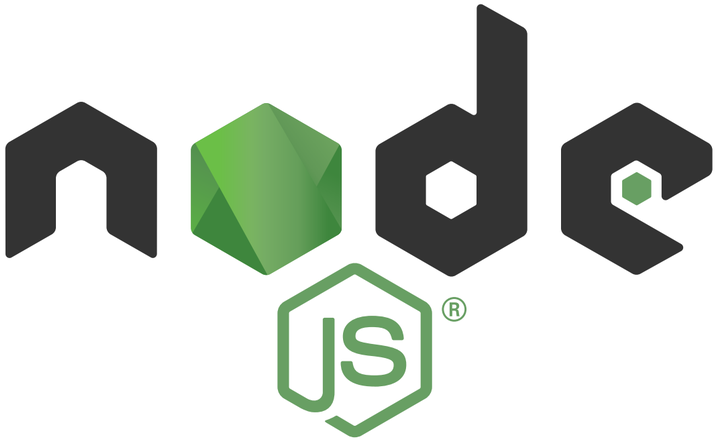
ABOUT FRAMEWORKS

01

Framework is always in control of your code. Meaning, the framework decides when and how to call your code.

02

Other terms - Declarative programming, Inversion of control etc.



01

JavaScript runtime built on Chrome V8 JavaScript Engine

02

Allows you to run JavaScript outside the browser

Node is used to build:

- Web Servers
- Command Line Tools
- Native Apps (VSCode is a Node app!)
- Video Games
- Drone Software
- A Whole Lot More!

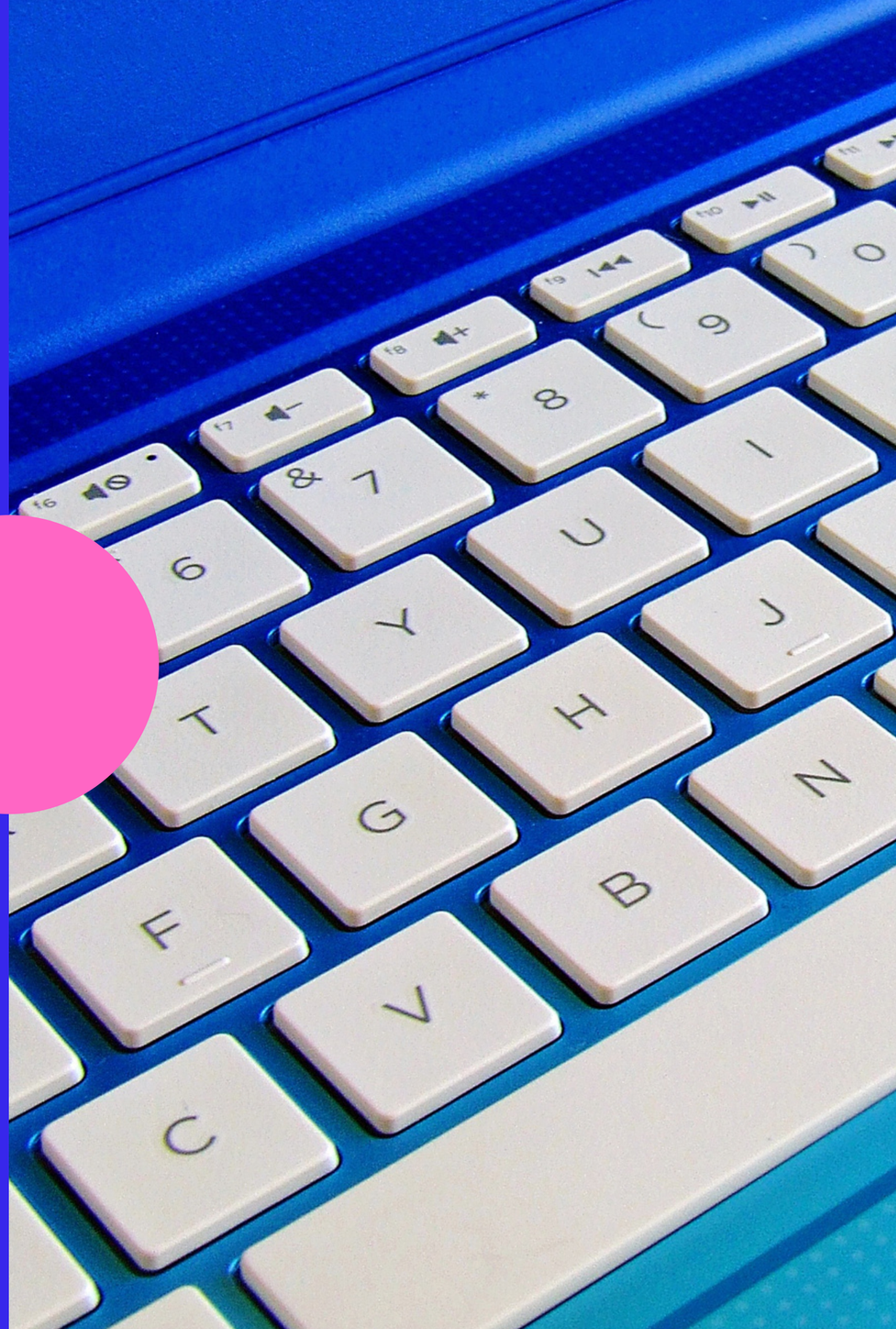
01

A library of thousands of packages published by other developers that we can use for free!

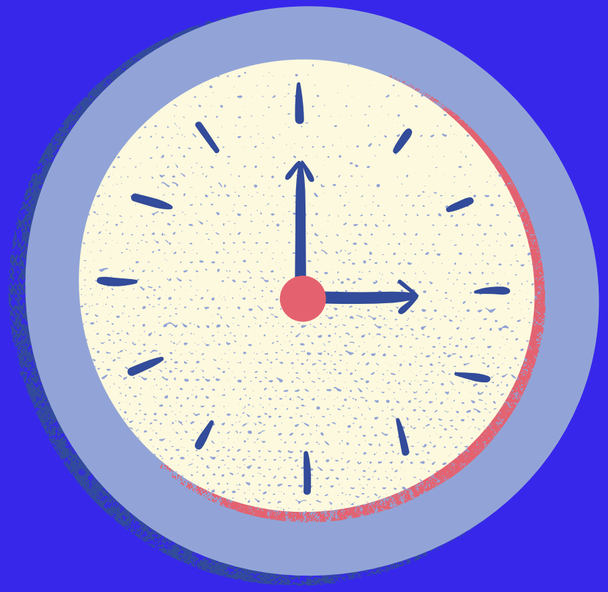
02

A command line tool to easily install and manage those packages in our Node projects

The World of JAVASCRIPT Frameworks



FRONT-END JAVASCRIPT FRAMEWORKS:



HISTORY

01

AngularJS was released in 2010 and quickly became one of the most popular front-end JS frameworks

02

In 2016 Google completely rewrote it and released it as Angular

SERVER-SIDE RENDERING

COMPONENT-BASED

MOBILE SUPPORT

POWERFUL TEMPLATES

SEMANTIC VERSIONING

<Major Version>.<Minor Version>.<Patch>

is Angular

Structural framework for dynamic web applications:

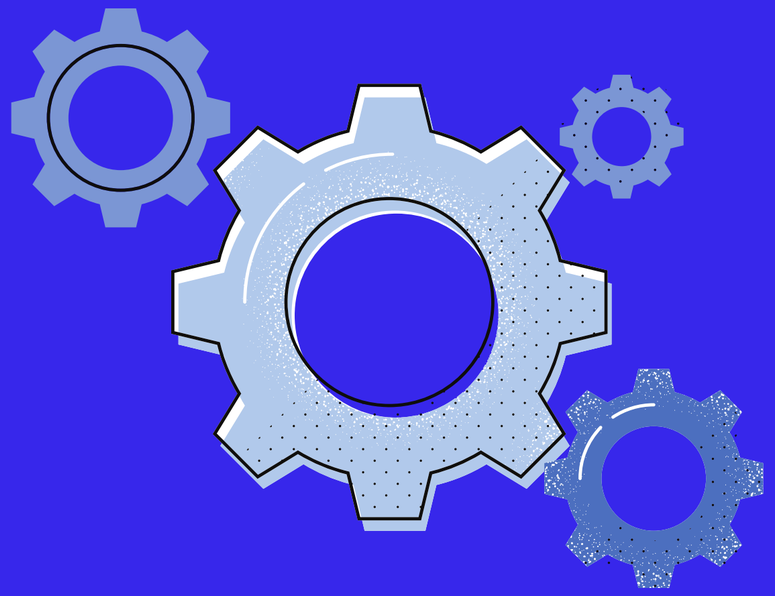
- HTML only does static documents

– Angular fills in the gap to support dynamic applications

• Designed with CRUD applications (data-driven) in mind

ANGULAR VOCABULARY

- One-way/Two-way Data Binding
- Components
- Directives
- Service
- Testing
- Templates
- Routing
- Modules
- Provider

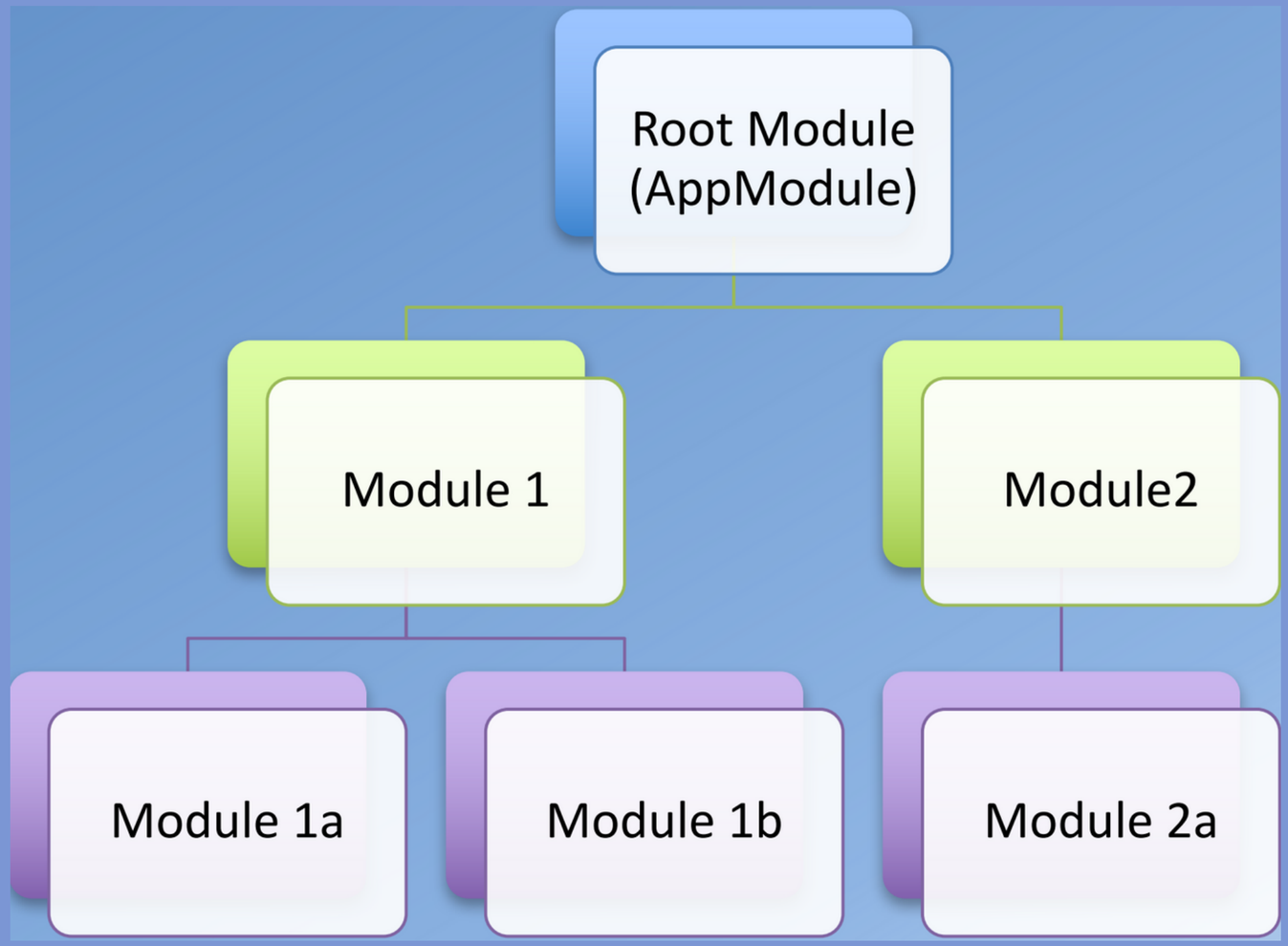


ARCHITECTURE

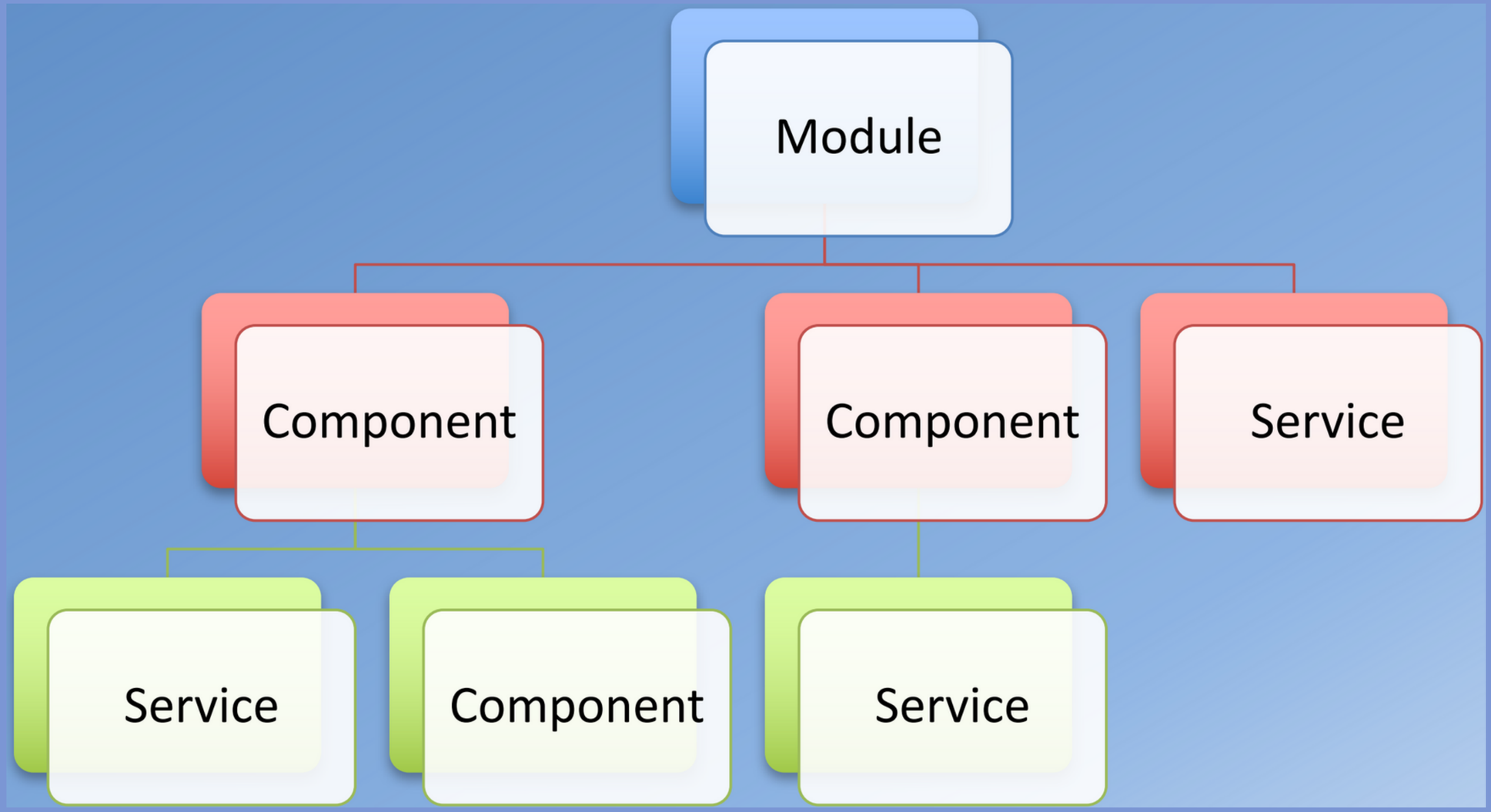
01 Modular

02 Component-based with Templates

MODULAR ARCHITECTURE



MODULE


Module
Component
Component
Service
Service
Component
Service

Template

.component.ts

.component.html

Metadata

Properties

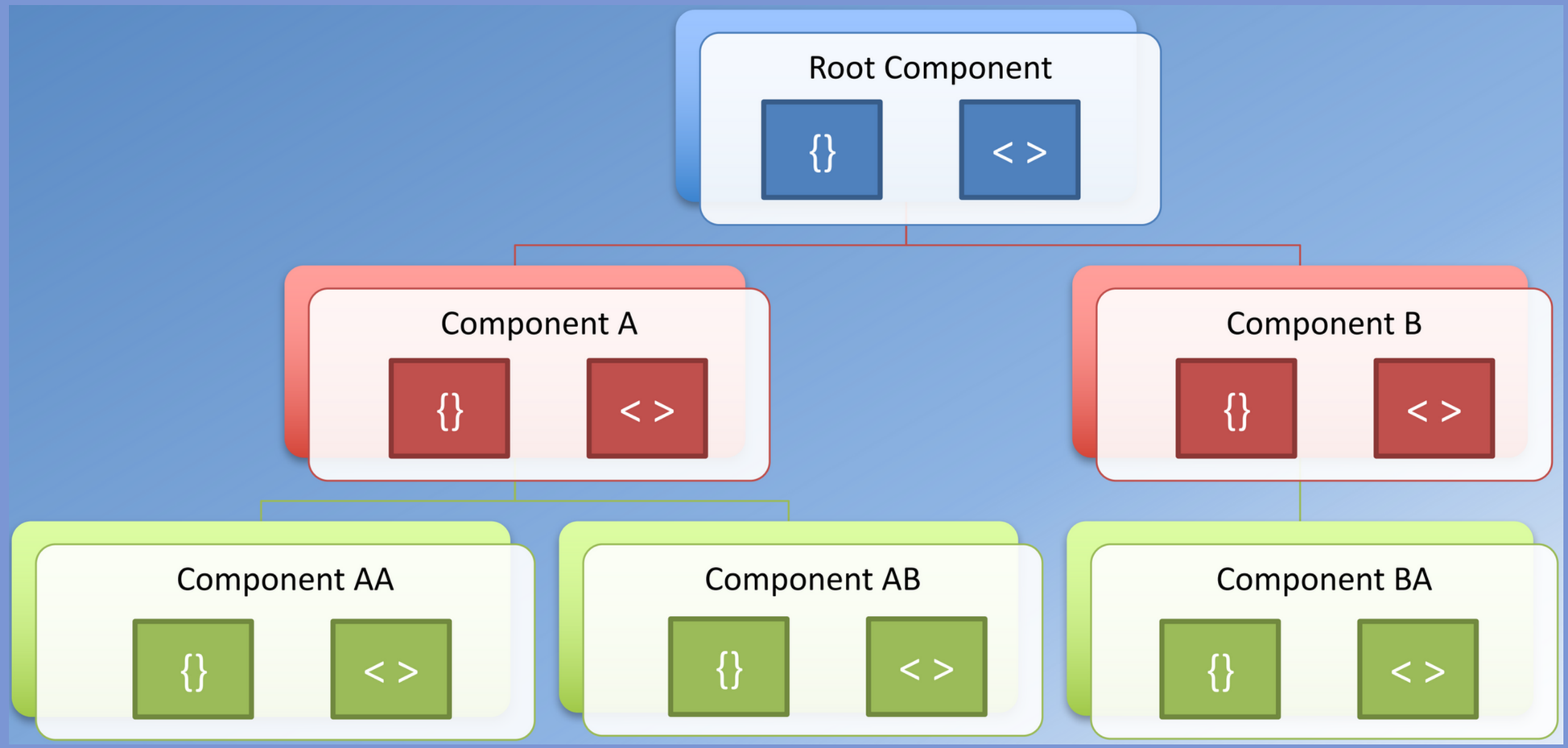
Methods

Property binding

Event binding

< >

COMPONENT HIERACHY



DIRECTIVES

01

Directives give instructions to Angular on how to render the templates to the DOM

02

A component is a special kind of directive with a template associated to it

03

Two other kinds of directives in Angular:
Structural and Attribute

STRUCTURAL DIRECTIVES

01 Allows you to alter the layout by adding, removing and replacing elements in DOM

02 Apply a structural directive to a host element in the DOM and its descendants

COMMON STRUCTURAL DIRECTIVES

01

NgIf

```
<div *ngIf=“selectedDish”> . . . </div>
```

02

NgFor

```
<mat-list-item *ngFor="let dish of dishes">
```

03

NgSwitch