Angular 8 – For Beginner's

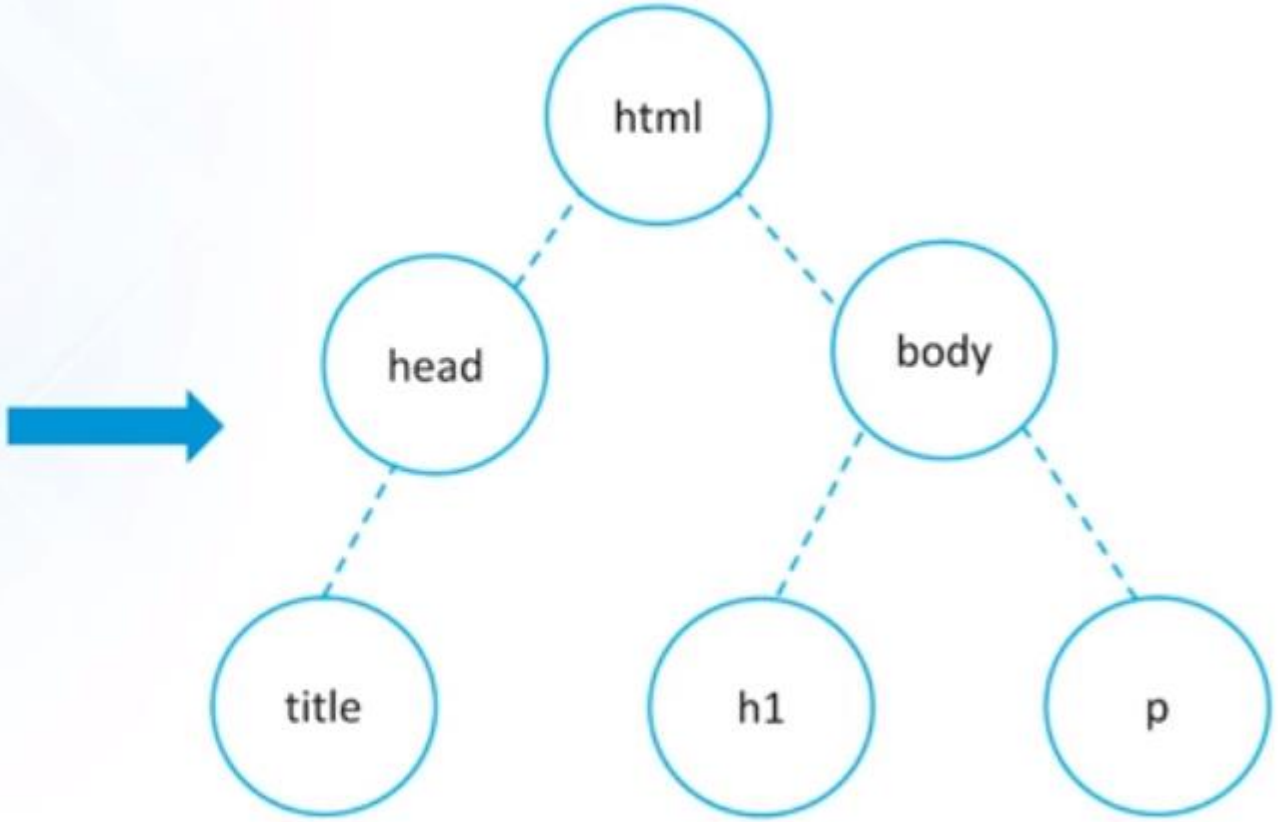
Agenda

- Webpage and DOM
- DOM Manipulation
- JavaScript and jQuery
- Why Angular?
- What is SPA?
- Angular Introduction
- Angular Features
- Angular Installation
- Basic Building Blocks of Angular
- Angular Architecture

Webpage and DOM

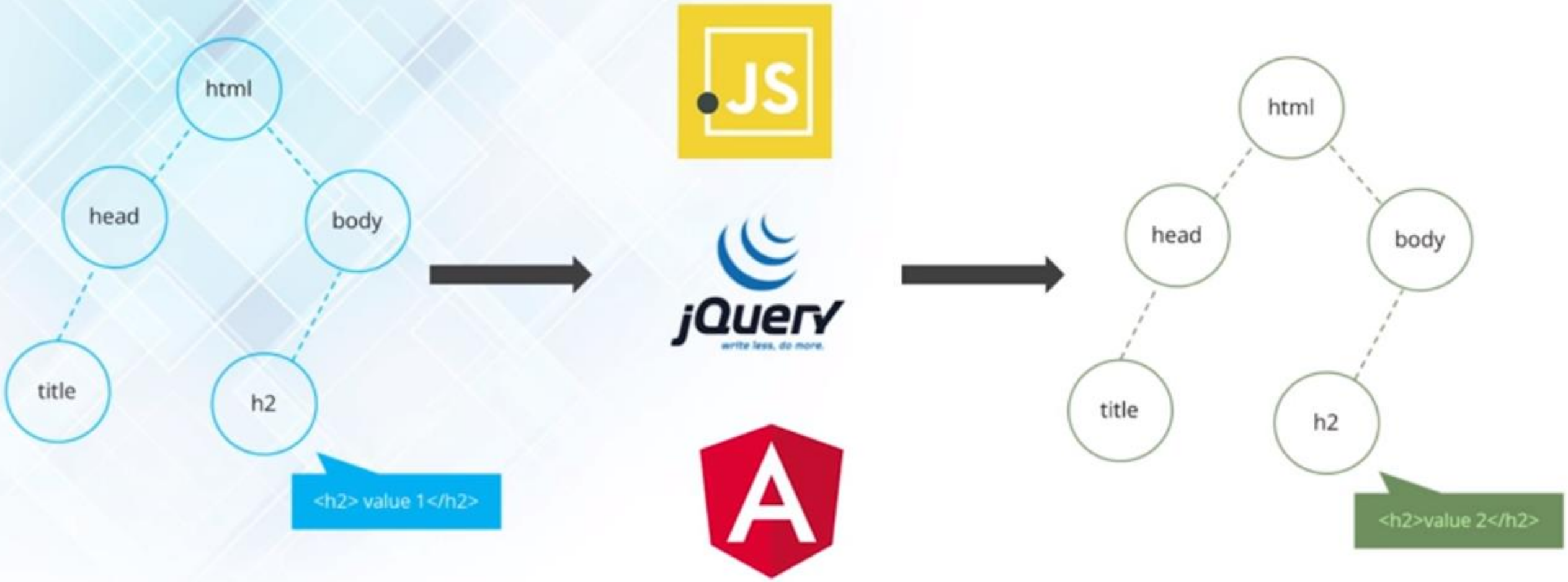
```
<html>
 <head>
 <title> Angular 4 Tutorial </title>
 </head>
 <body>
    <h1> Welcome to Angular 4 Tutorial </h1>
    <p>Angular is a development platform for creating
       applications using modern web standards.</p>
 </body>
</html>
```

HTML Markup



DOM Tree of the HTML document

DOM Manipulation



JavaScript & jQuery

- JavaScript is a programming language designed for use in a web browser.
- Used for manipulating DOM
- Example:

Document.body.style.background = red;

- jQuery is a library built in JavaScript to automate and simplify common tasks.
- Used for manipulating DOM
- Example:

$ ('body') .css ('background', '#ccc');

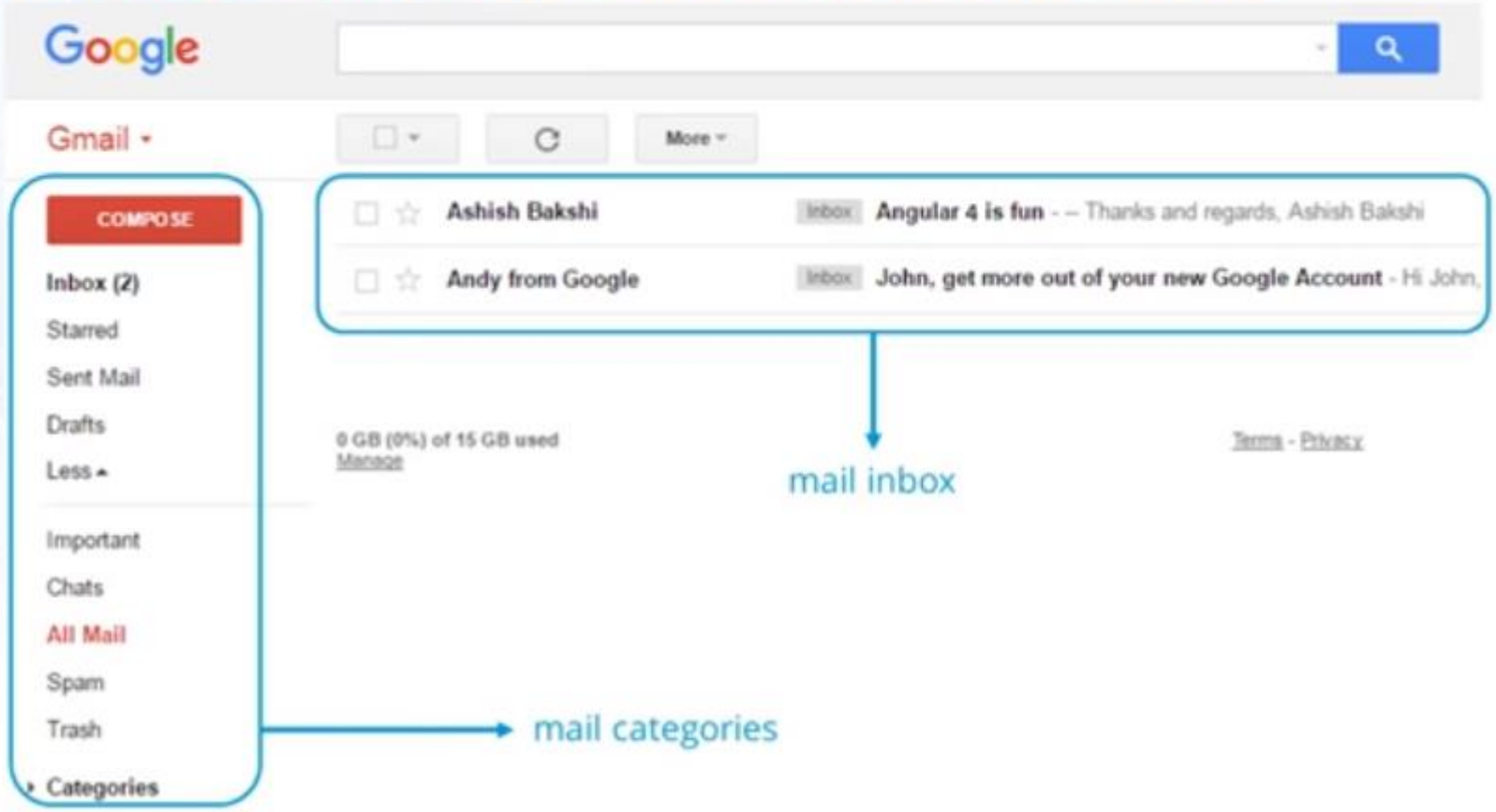
Why Angular?

	jQuery	Angular
DOM Manipulation	✓	✓
RESTful API	✗	✓
Animation Support	✓	✓
Deep Linking Routing	✗	✓
Form Validation	✗	✓
2 Way Data Binding	✗	✓
AJAX/JSONP	✓	✓

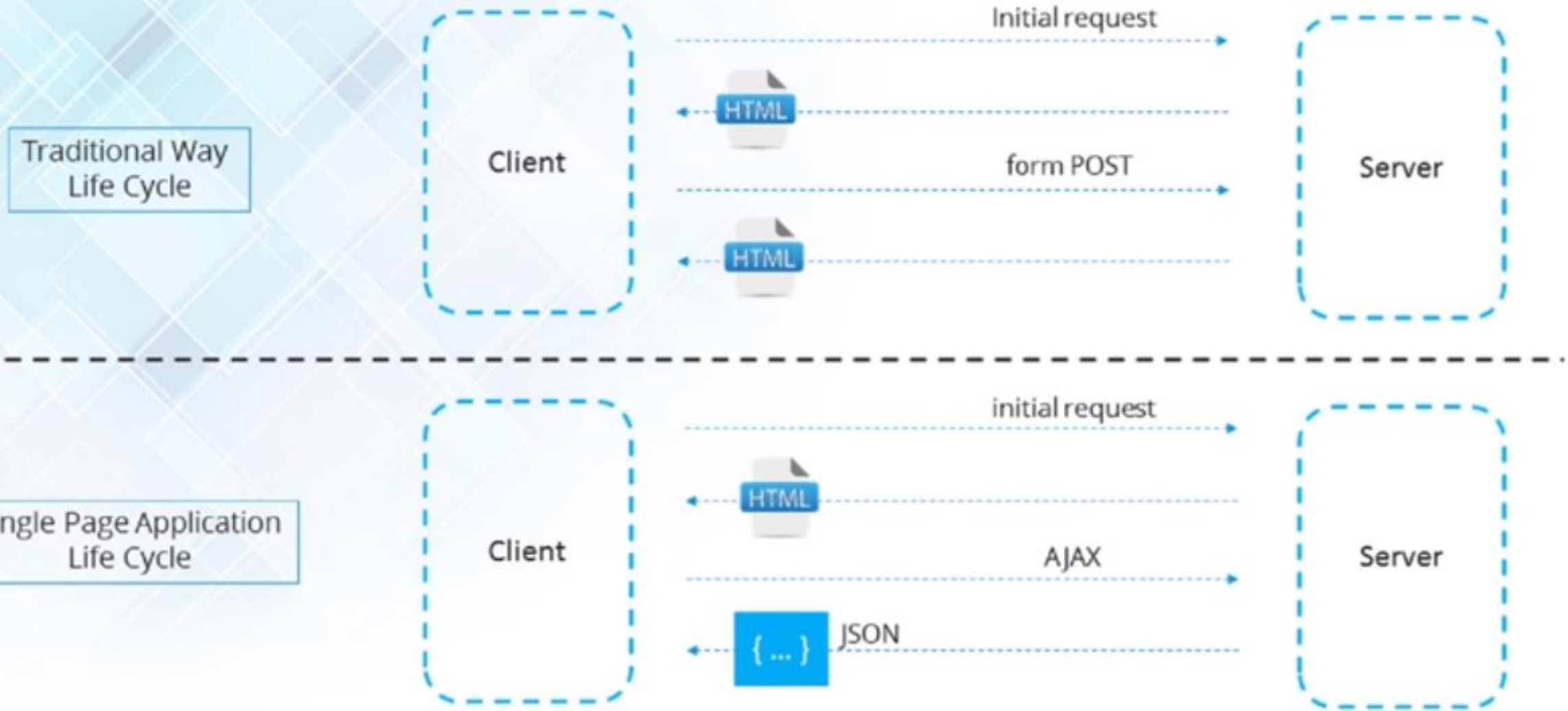
What is SPA?

A Single Page Application is a web application that requires only a single page load in a web browser.

- Whole page is not reloaded every time
- Your browser fully renders the DOM once
- Later any server interactions is performed by JavaScript which modifies the view



Traditional Way Vs Single Page Application



Angular Introduction

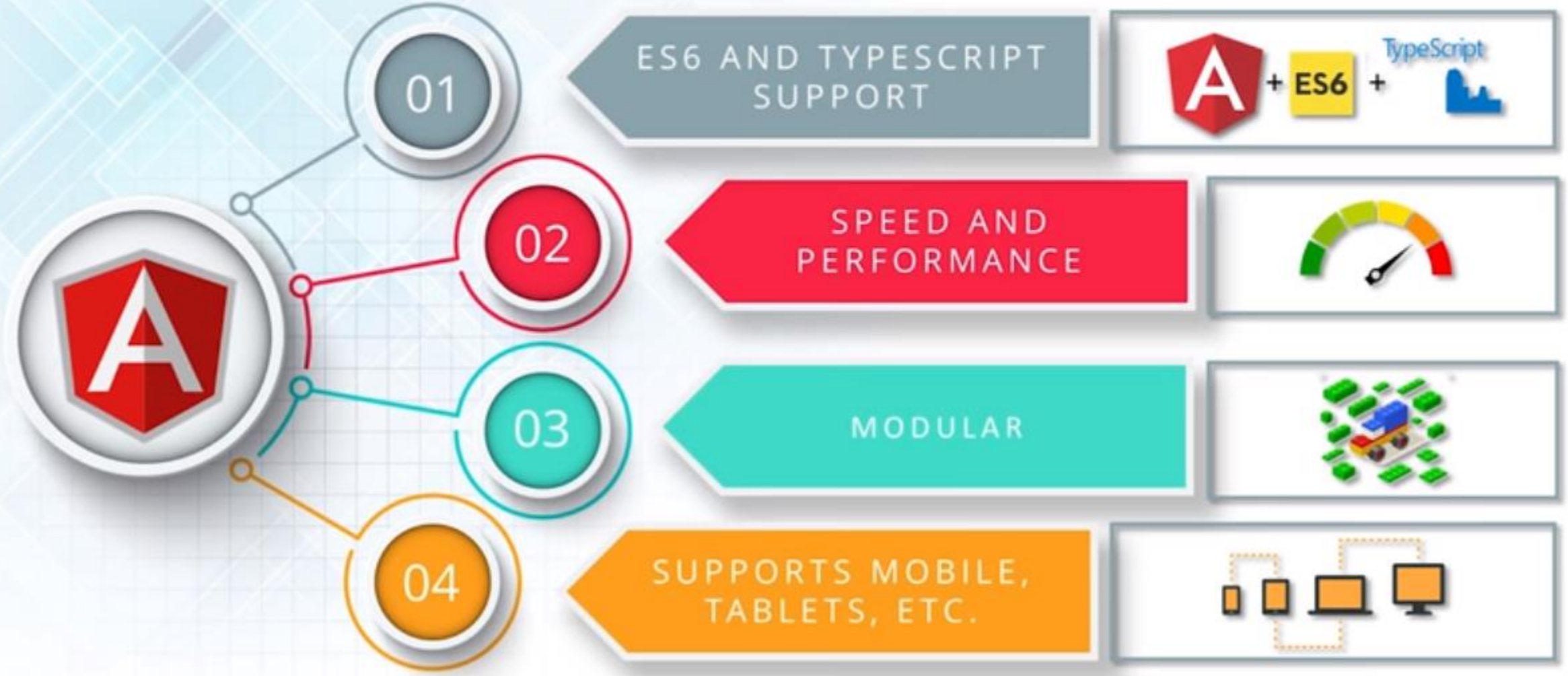
Useful for building complex applications in the browser (and beyond).

OpenSource Client Side JavaScript framework created by Google

Introduction of Object Oriented Web Development using TypeScript

Angular accomplishes a lot by embracing HTML, JavaScript & CSS

Angular Features



Angular Installation

```
// Basic data types of JS, null, undefined
// Array and Manipulation
// Object and Object Manipulation

// Design Patterns - singleton, factory, constructor patterns


// Typescript
// Superset of JS - Valid JS is valid TS, transpile TS into JS
// Types, Interfaces, Classes, Decorator, and Import/Export OR Modularity
```

Building Blocks of Angular

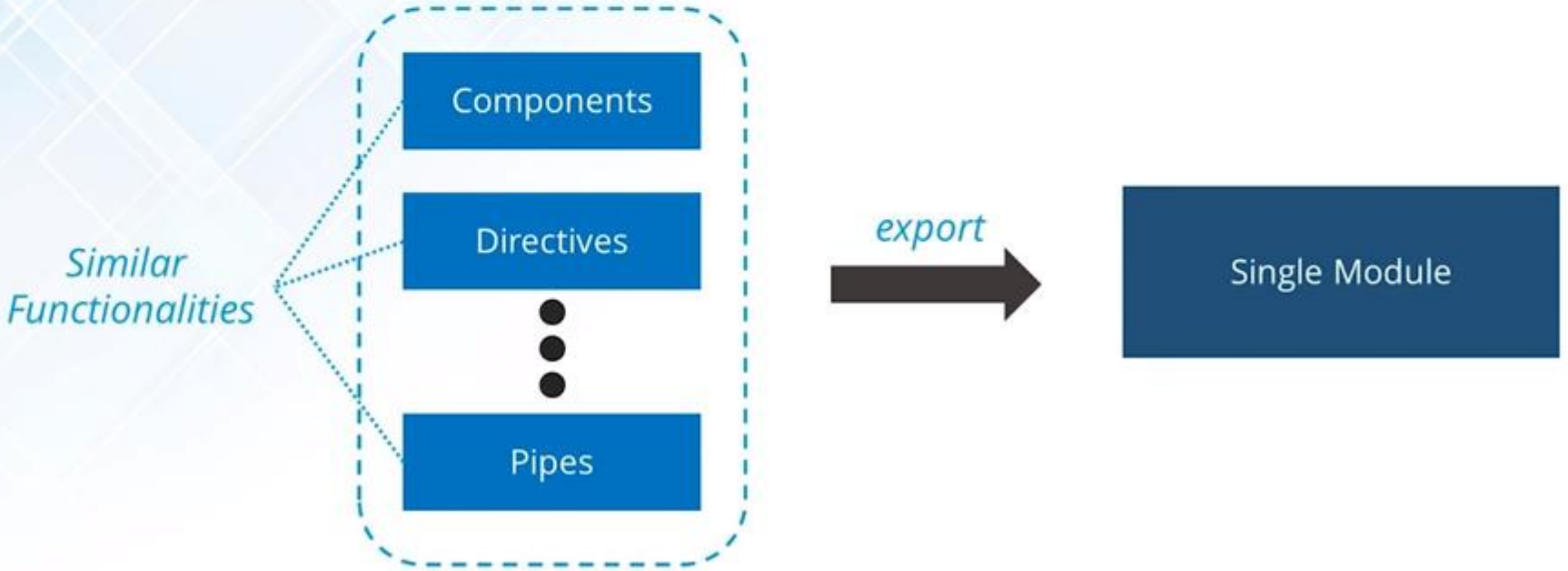
Building Blocks of Angular

- **Module**
- Component
- Metadata
- Template
- Data Binding
- Services
- Directives

Module is a class with @NgModule metadata

Every Angular app has at least one root module

Encapsulation of different similar functionalities



Building Blocks of Angular

- **Module**
- Component
- Metadata
- Template
- Data Binding
- Services
- Directives

```
import { BrowserModule } from '@angular/platform-browser';
import { NgModule } from '@angular/core';
import { FormsModule } from '@angular/forms';
import { HttpModule } from '@angular/http';

import { AppComponent } from './app.component';
import { TaskComponent } from './task/task.component';

@NgModule({

  declarations: [
    AppComponent,
    TaskComponent
  ],
  imports: [
    BrowserModule,
    FormsModule,
    HttpModule
  ],
  providers: [],
  bootstrap: [AppComponent]
})
export class AppModule { }
```

Decorator

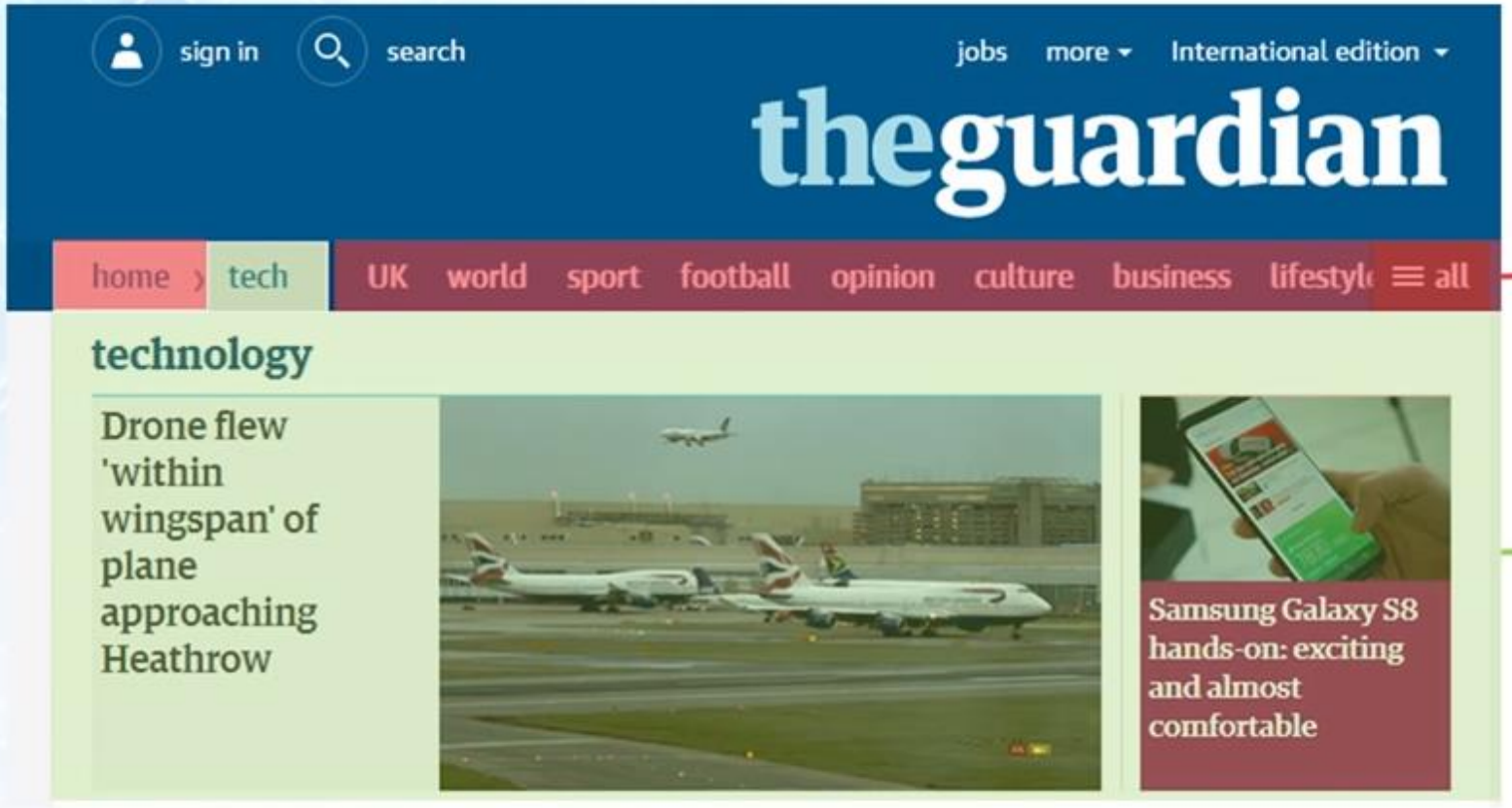
Declaring all the components

Importing Modules

Provide Services to all module's component

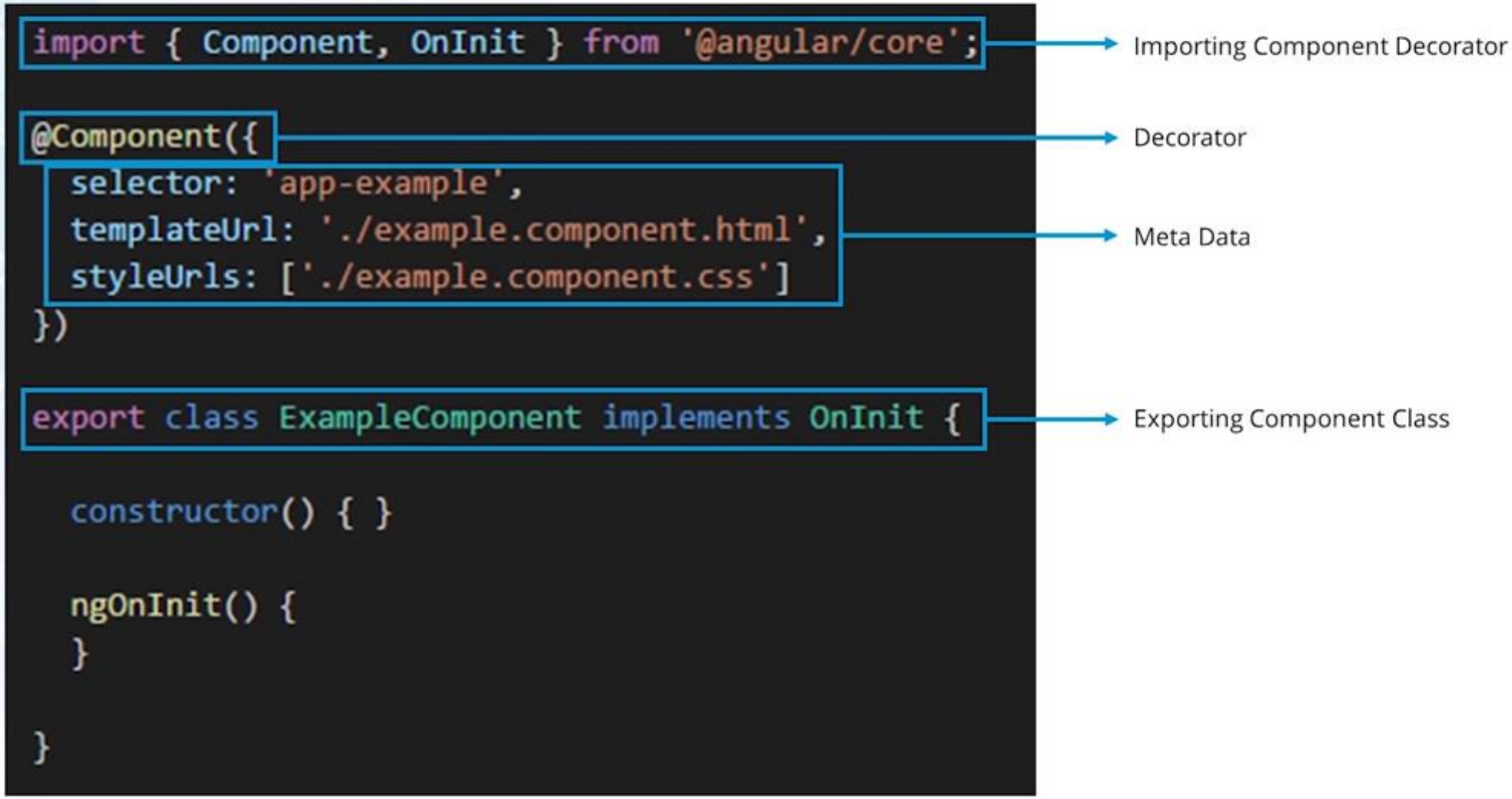
Building Blocks of Angular

- Module
- **Component**
- Metadata
- Template
- Data Binding
- Services
- Directives



Building Blocks of Angular

- Module
- **Component**
- Metadata
- Template
- Data Binding
- Services
- Directives



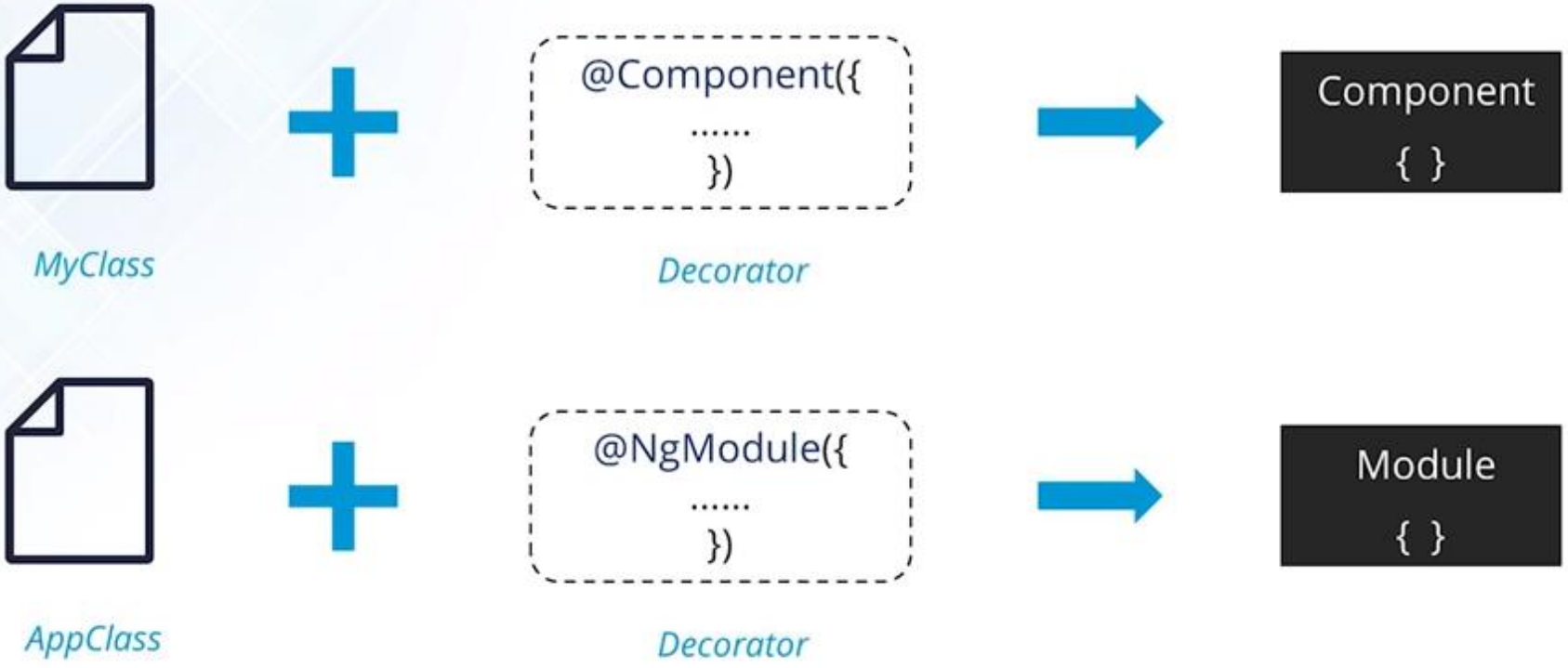
Building Blocks of Angular

- Module
- Component
- **Metadata**
- Template
- Data Binding
- Services
- Directives

Metadata describes how to process the class

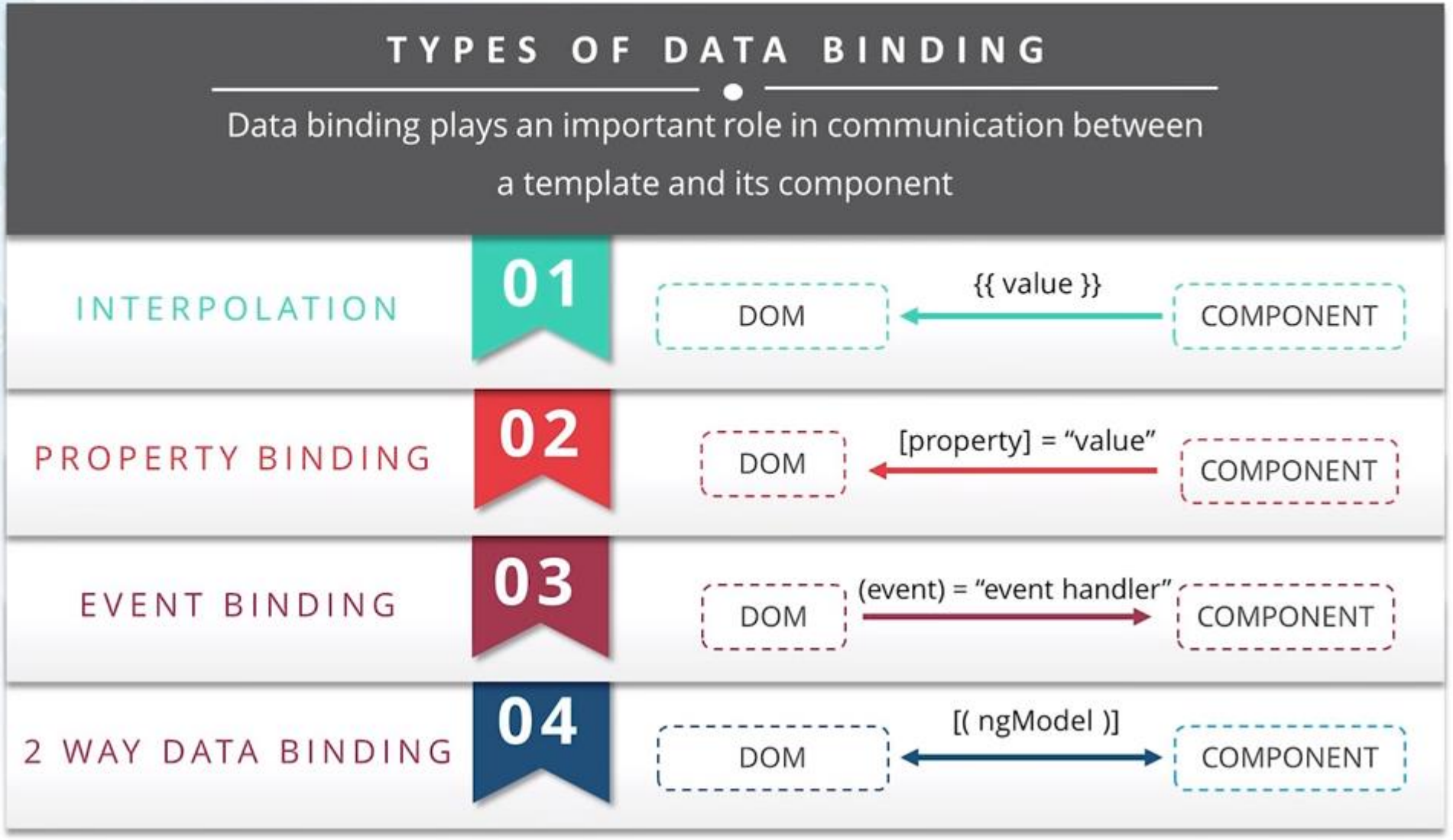
Decorator is used to attach metadata

Example:



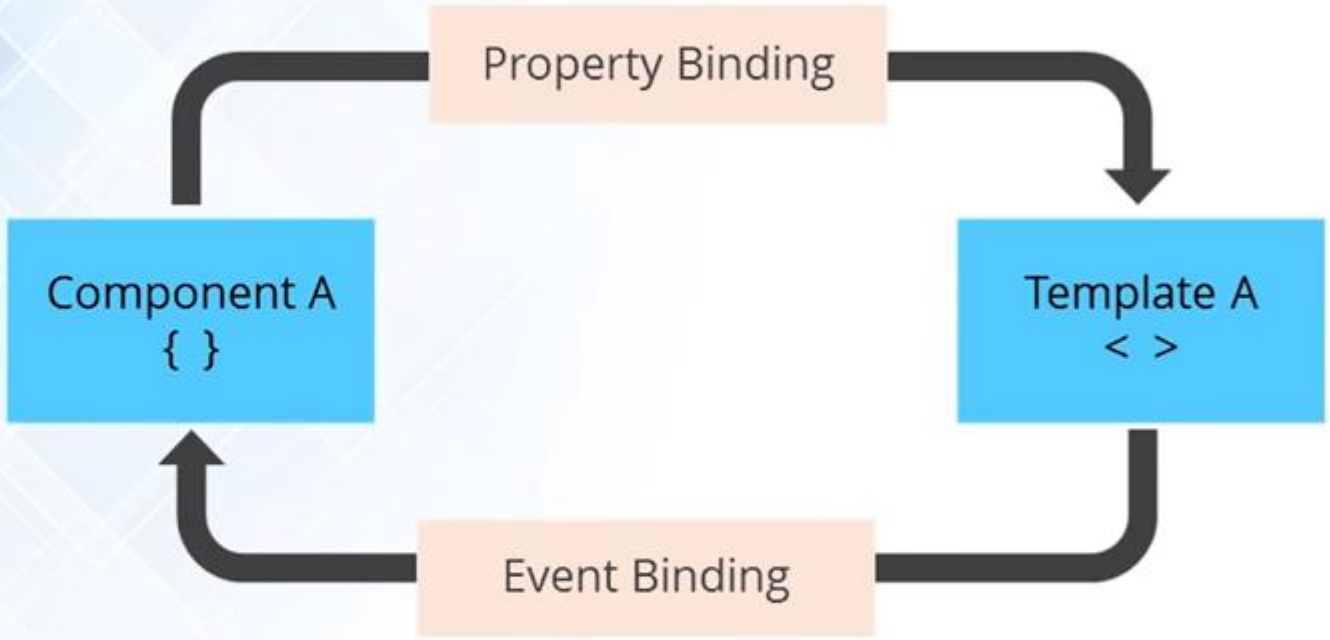
Building Blocks of Angular

- Module
- Component
- Metadata
- Template
- **Data Binding**
- Services
- Directives



Building Blocks of Angular

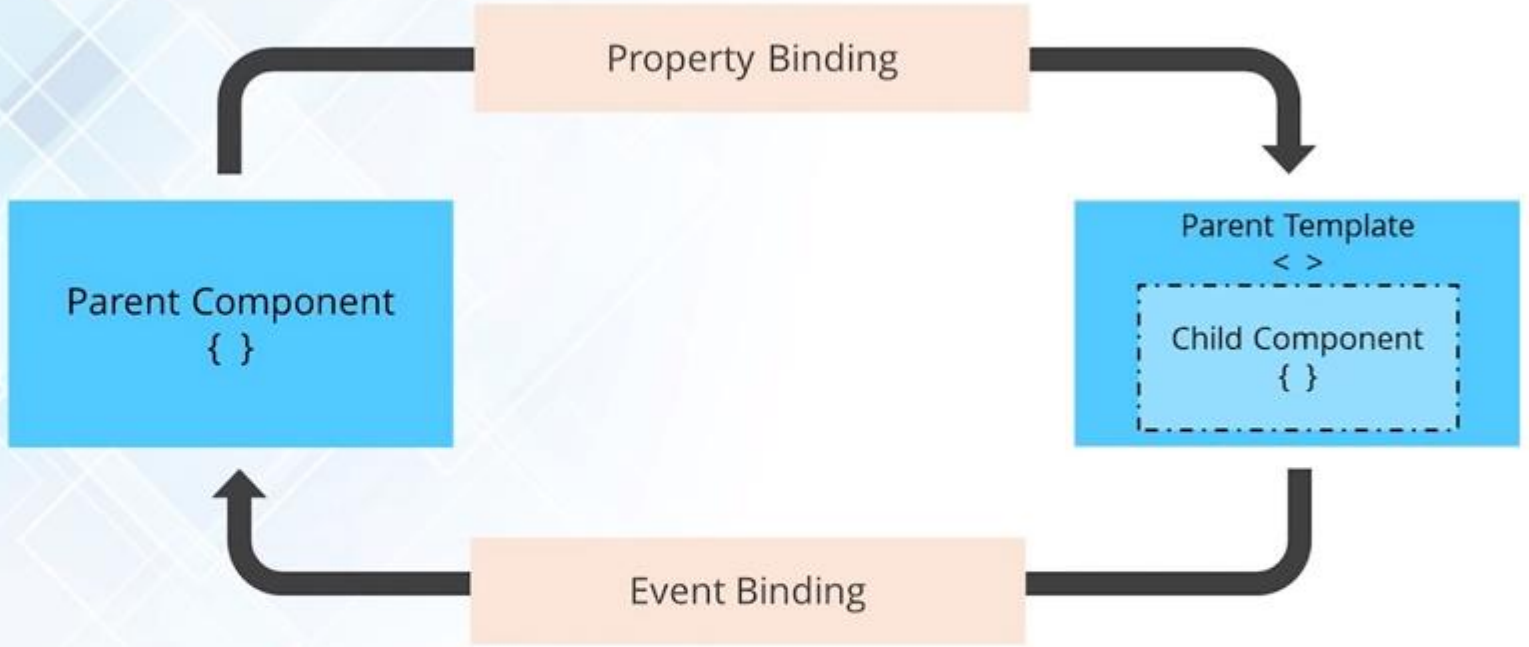
- Module
- Component
- Metadata
- Template
- **Data Binding**
- Services
- Directives



Data binding plays an important role in communication between a template and its component.

Building Blocks of Angular

- Module
- Component
- Metadata
- Template
- **Data Binding**
- Services
- Directives



Data binding is also important for communication between parent and child components.

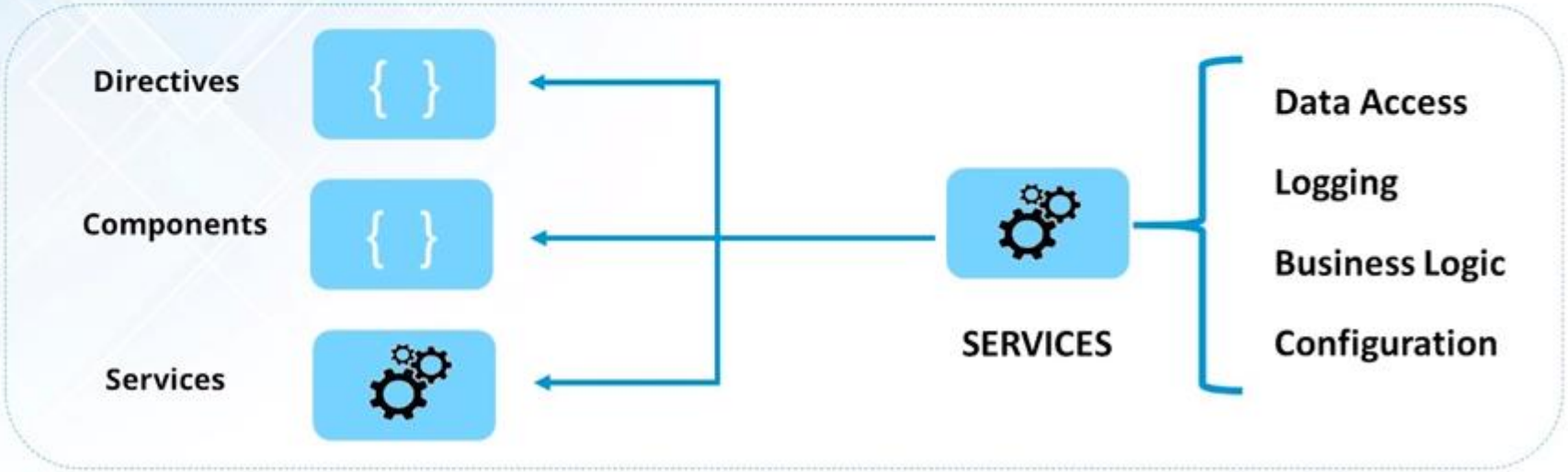
Building Blocks of Angular

- Module
- Component
- Metadata
- Template
- Data Binding
- **Services**
- Directives

Service is a broad category encompassing any value, function, or feature that your application needs.

Example:

logging service | data service | message bus | tax calculator



Building Blocks of Angular

- Module
- Component
- Metadata
- Template
- Data Binding
- **Services**
- Directives

```
import { Injectable } from '@angular/core';

@Injectable()
export class ExampleService {

  movies: string[] = ["Inception", "Dark Knight", "Shutter Island"];

  constructor() { }

  getMovies(): string[]
  {
    return this.movies;
  }

}
```

Service Class

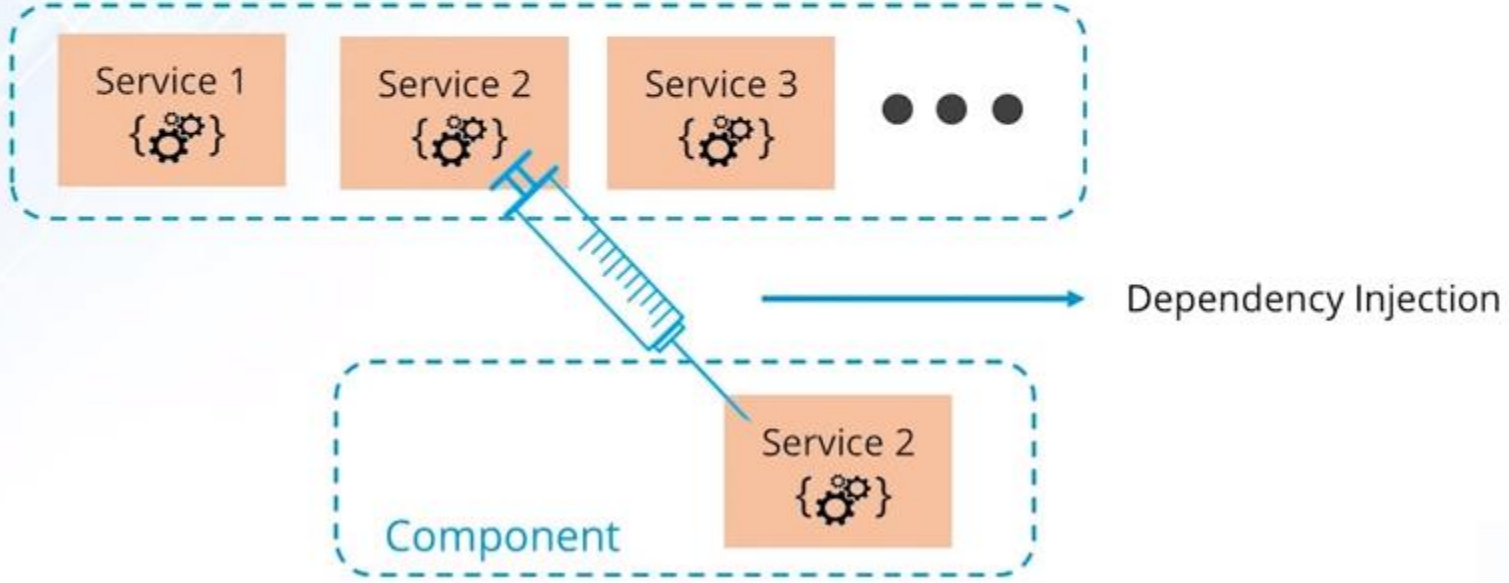
Service Method for retrieving data

Building Blocks of Angular

- Module
- Component
- Metadata
- Template
- Data Binding
- **Services**
- Directives

Creates a new instance of class along with its required dependencies

Used to provide services to a component



Building Blocks of Angular

- Module
- Component
- Metadata
- Template
- Data Binding
- Services
- **Directives**

Changes the appearance or behavior of a DOM element

1	COMPONENTS	Directives with a template
2	STRUCTURAL DIRECTIVE	Adds & removes DOM elements to change DOM layout
3	ATTRIBUTE DIRECTIVE	Changes the appearance or behavior of an element

Building Blocks of Angular

- Module
- Component
- Metadata
- Template
- Data Binding
- Services
- **Directives**

3 ATTRIBUTE DIRECTIVE

Changes the appearance or behavior of an element

```
import { Directive, ElementRef, HostListener } from '@angular/core';

@Directive({
  selector: '[appBoldText]'
})
export class BoldTextDirective {

  constructor(private elementRef: ElementRef) { }

  @HostListener('mouseenter') onMouseEnter() {
     this.elementRef.nativeElement.style.fontWeight = 'bold';
  }

  @HostListener('mouseleave') onMouseLeave() {
   this.elementRef.nativeElement.style.fontWeight = null;
  }
}
```

- Importing Directive, ElementRef & HostListener
- Directive Metadata
- Injecting ElementRef to access the DOM elements
- Bold the text on cursor hover
- Un-bold the text

Angular Architecture