Session 8

Getting to Know Angular 9

Session Objectives

- ✓ List features and enhancements in Angular 9
- ✓ Explain Angular Architecture
- ✓ Outline steps to create an Angular 9 application
- ✓ Identify the use of various types of files in an Angular application
- ✓ Define Pipes and learn their usage

Angular 9

Angular is a complete rewrite of the AngularJS Web framework.

Angular is TypeScript-based and open-source.

Angular 9 was released on February 6, 2020.

Angular has been updated to work with TypeScript 3.6 and 3.7.

In Angular 9, all applications must use Angular's next-generation compiler and runtime by default.

This compiler has been code-named **Ivy**.

Angular 9 Features and Enhancements

Angular 9 Features and Enhancements

Angular 9 Features and Enhancements

Tree shaking

Removing unused blocks of code during the bundling process.

By using tree shaking, our application will only include that code required for the application to run.

Locality

Locality refers to the process of compiling each component with only information about the component itself, except for the name and package name of its declarable dependencies.

This helps to rebuild faster by compiling partial changes and not the entire project files. Ultimately, increasing the speed of your build process.

Angular 9 Features and Enhancements

- Developers should decrease size of downloadable files, for a better mobile user experience.
- Angular 9 supports this by enabling developers to reduce bundle sizes by 25-40 percent based on the app size.
- Small apps benefit from tree shaking feature of Ivy and Angular 9 support, as they have to generate a lesser amount of code for Angular components.
- Smaller bundle means better performance and speed.

Angular 9 Features and Enhancements

Better Debugging

Angular 9 and Ivy compiler help in better debugging by providing access to instances of the components and directives, triggering change detection with **`applyChanges`**, and more.

Phantom Template Variable

Variables that are never referenced or defined in a template's associated component are called **phantom template variables**. In Angular 9, you will now get a compiler error when you create such phantom template variables.

API Extractor Updates

In Angular 9, required libraries are tracked and updated using Bazel. Bazel supports building automation and software testing. Bazel in turn references API Extractor, which is a tool invoked at build time by Angular.

Angular 9 Features and Enhancements

- The **`@Injectable()`** decorator in Angular, by default, has a **`providedIn`** property, which creates a provider for the service.
- From Angular 6 onwards, it is recommended to create a singleton service by setting **`providedIn`** property to root on the service's **`@Injectable()`** decorator.
- In Angular 9, the **`providedIn`** property has some additional options:

platform

Makes the service available in a special singleton platform injector that is shared by all applications on the page.

providedIn: 'platform' can be used for sharing services over application boundaries, such as Angular Elements. An Angular Element is an Angular component packaged as a custom element.

any

Provides a unique instance in every module that introduces the token.

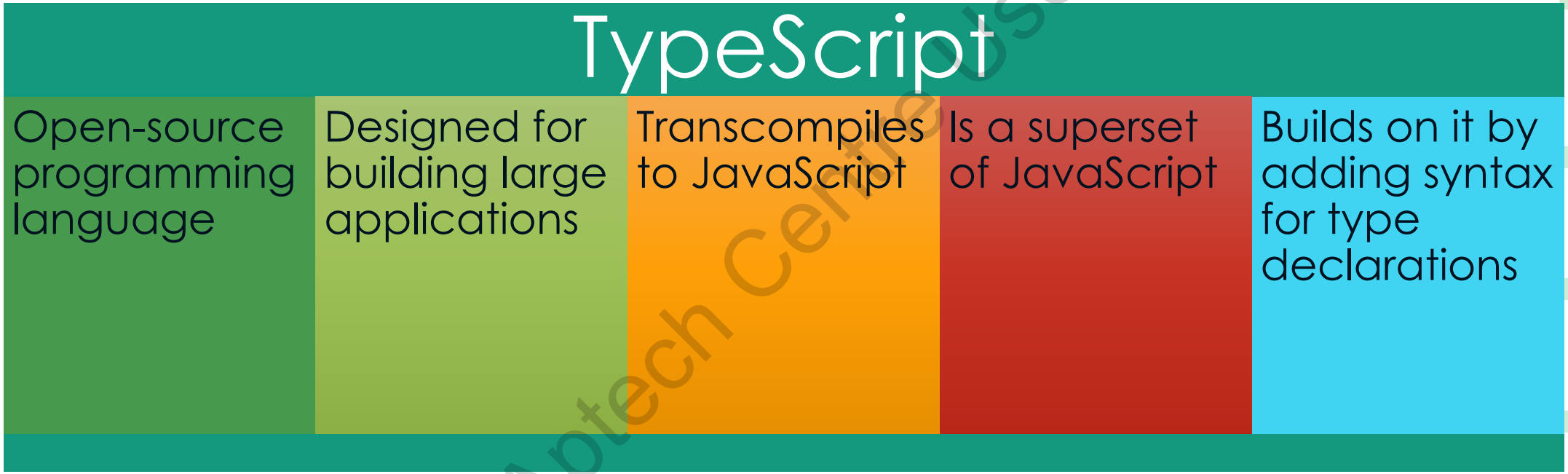
providedIn: 'any' can be used to make sure a service is a singleton within module boundaries.

Angular 9 Features and Enhancements

A component harness is a class that allows unit tests to interact with components through supported APIs.

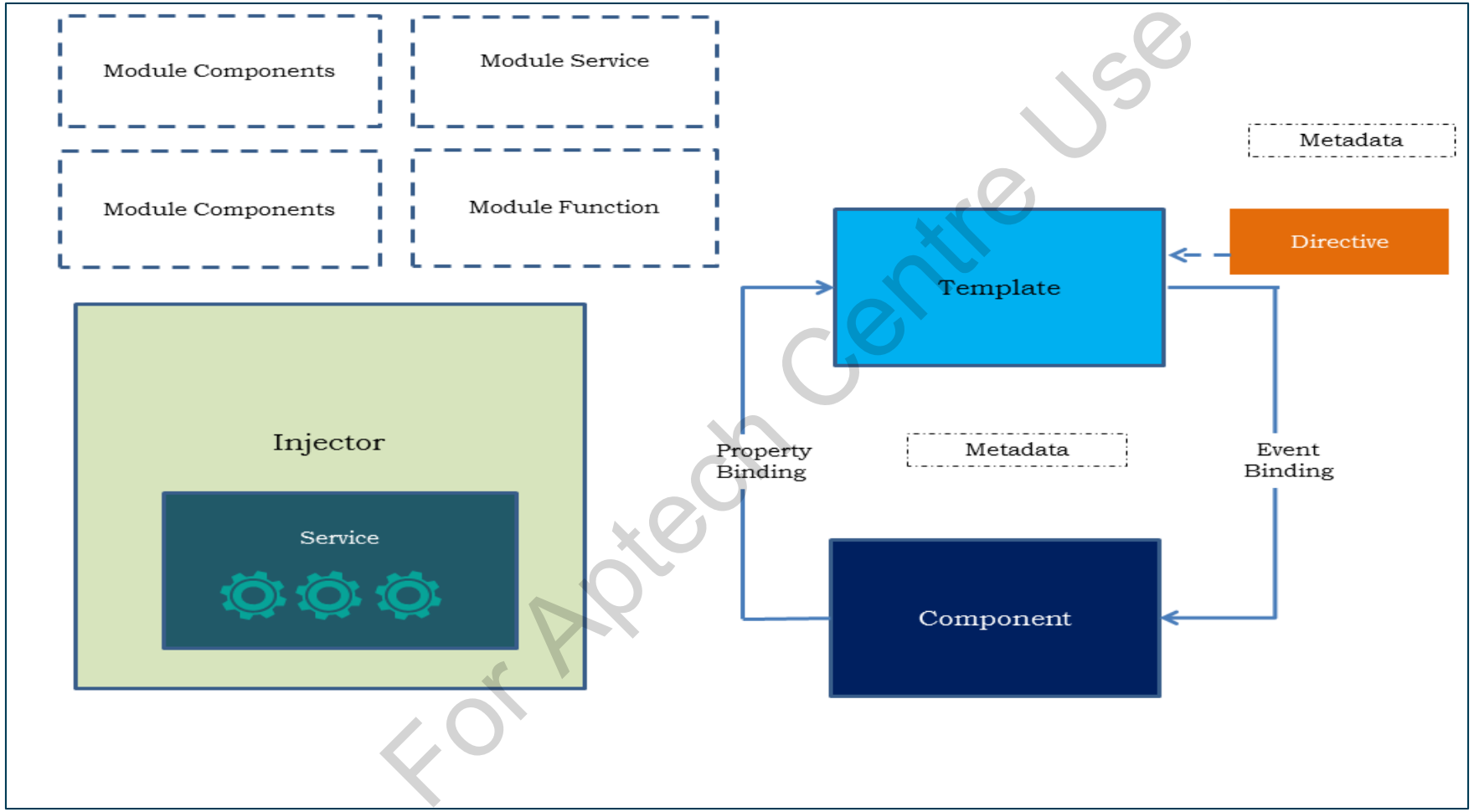
Angular 9 is making harnesses available to any component author as part of the Component Dev Kit (CDK).

Angular 9 Features and Enhancements



Angular 9 is updated to work with TypeScript 3.6 and 3.7.

Angular 9 Architecture



Angular 9 Architecture - Key Components

Modules	In Angular 9, apps are modular. Angular has its own modularity system called `NgModules`. Angular 9 apps comprise a root module called `AppModule`. In addition, an application can contain other modules.
Components	In Angular 9, component-based architecture allows to use components to compose applications. Each component defines a class containing feature application data and logic.
Template, Directives, and Data Binding	Templates in Angular 9 combine HTML with Angular markup and modify HTML elements before displaying them.
Services and Dependency Injection	Dependency injection is a technique to pass a dependent object into another object to make all functionality of former available to latter. Using DI, you can inject a service into a component, giving the component access to that service class.

Creating Angular 9 Applications

Angular CLI

Angular Command-Line Interface (CLI) is the command-based interface that helps you build Angular applications and is considered the official tool to work with Angular projects.

Node.js

Similar to many other modern front-end tools available nowadays, Angular CLI is built on top of Node.js, which is a server technology. Node.js is open-source, cross-platform, and provides a runtime environment to run JavaScript on the server and build server-side Web applications.

npm

npm stands for Node Package Manager and is a package manager program for JavaScript. It is the default package manager for Node.js. It comprises:

- Command line client, also called npm
- npm registry, an online database of packages

Creating Angular 9 Applications

Step 1 - Install Node and npm

Visit the link https://nodejs.org/en/download/ to download and install Node.js. Follow the setup wizard instructions.

Step 2 - Install Angular CLI 9

Open Command Prompt and give the following command to install Angular 9 CLI tool:

```
npm install -g @angular/cli
```

Step 3 - Verify installation

Once it is installed successfully, verify the version with the following command: `ng --version`

Step 4 - Initialize a new Angular 9 Project

```
ng new first-ang9-project
```

Step 5 - Serve the Project

After the project has been created, change the current working directory to the project directory and then, type the following command:

```
ng serve
```

Creating Angular 9 Applications

File/Folder Name	Purpose
/e2e/	Is a folder containing end-to-end tests of the Website
/node_modules/	Is a folder containing all third party libraries installations through npm install
/src/	Is a folder containing the source code of the application. Most work will be done here
/app/	Is a folder containing modules and components
/assets/	Is a folder containing static assets like images, icons, and styles
/environments/	Is a folder containing environment (production and development) specific configuration files
browserslist	Is a file required by autoprefixer for CSS support
favicon.ico	Is a file that represents the favicon
index.html	Is the main HTML file
karma.conf.js	Is the configuration file for Karma (a testing tool)
main.ts	Is the main starting file from where the AppModule is bootstrapped
styles.css	Is a global stylesheet file for the project
test.ts	Is a configuration file for Karma
tsconfig.*.json	Is the configuration file for TypeScript
angular.json	Contains the configuration for CLI
package.json	Contains basic information of the project (name, description, and dependencies)
README.md	Is a markdown file that contains a description of the project
tsconfig.json	Is the configuration file for TypeScript
tslint.json	Is the configuration file for TSlint (a static analysis tool)

NgModules

Some key properties of NgModule are:

imports

Represents modules whose exported classes are referred to by component templates declared within current `NgModule`.

providers

Represents creators of services that current `NgModule` contributes to the global set of services. They can be specified at the component level too.

declarations

Components, directives, and pipes belonging to the `NgModule`.

bootstrap

The main application view, called the `root` component, which hosts all other app views. Only the root `NgModule` should set the `bootstrap` property.

Generating Angular Items

Using **`ng generate`** command of Angular CLI, developers can generate basic Angular items such as modules, components, directives, and services.

For example:

```
ng generate component ang9-component
```

Angular Pipes

- An Angular pipe takes in data as input and transforms it to a desired output.
- Angular provides several built-in pipes such as: `DatePipe`, `DecimalPipe`, `TitleCasePipe`, `UpperCasePipe`, `LowerCasePipe`, `CurrencyPipe`, and `PercentPipe`.
- They are all available for use in any template.

Summary

- Angular is a complete rewrite of the AngularJS Web framework and is TypeScript-based and open-source.
- Angular 9 makes the next-generation compiler and runtime, Ivy, as default for all applications.
- Angular 9 has several new features and enhancements, including support for smaller bundle sizes, better debugging, phantom template variable menace, API extractor updates, and so on.
- Key components of the architecture include Modules, Components, Template, Directives, Data Binding, Services, and Dependency Injection.
- Angular Command-Line Interface (CLI) is the command-based interface and official tool that helps you build Angular applications.
- A module is a means by which one can group components, directives, services, and so on. To define a module, one should apply @NgModule() decorator to a class.
- Angular pipes are similar to filters in earlier versions of AngularJS and transform a value for display before it is rendered on the browser.