Introduction to Angular:

Angular is a platform and framework for building single-page client applications using HTML and TypeScript. Angular is written in TypeScript. It implements core and optional functionality as a set of TypeScript libraries that you import into your applications.

The architecture of an Angular application relies on certain fundamental concepts. The basic building blocks of the Angular framework are Angular components that are organized into *NgModules*. NgModules collect related code into functional sets; an Angular application is defined by a set of NgModules. An application always has at least a *root module* that enables bootstrapping, and typically has many more *feature modules*.

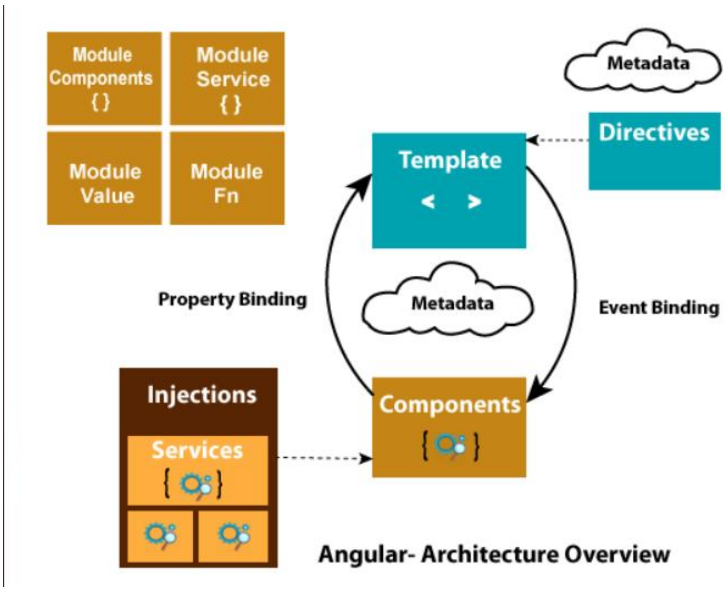
Projects built with Angular:

- Samsung Forward
- Microsoft Office Home
- Forbes
- BMW Driveaway Price Calculator

Architecture of Angular:

An Angular app always has at least a root module which enables bootstrapping, and typically has many other feature modules.

- Components define views, which are the sets of screen elements that are chosen and modified according to program logic and data by Angular.
- Components use services, which provide specific functionality not directly related to views. Service providers can be injected into components as dependencies, making your code modular, reusable, and efficient.



-**Short Description for angular architecture**

- **Modules:** AppModule, which is root module in Angular, which is used to launch the application.
- **Metadata:** As we know the template is made of ordinary HTML and binding markup that allow angular to modify the HTML before rendering it to the display.
- **Template, Directives and Data Bindings:** A template is used to combine HTML with Angular Markup and modify HTML elements before displaying them. Template directives provide program logic, and binding markup connects your application data and the DOM.

  There are two types of data binding:

- **Event Binding:** Event binding is used to bind events to your app and respond to user input in the target environment by updating your application data.
- **Property Binding:** Property binding is used to pass data from component class and facilitates you to interpolate values that are computed from your application data into the HTML.

React js

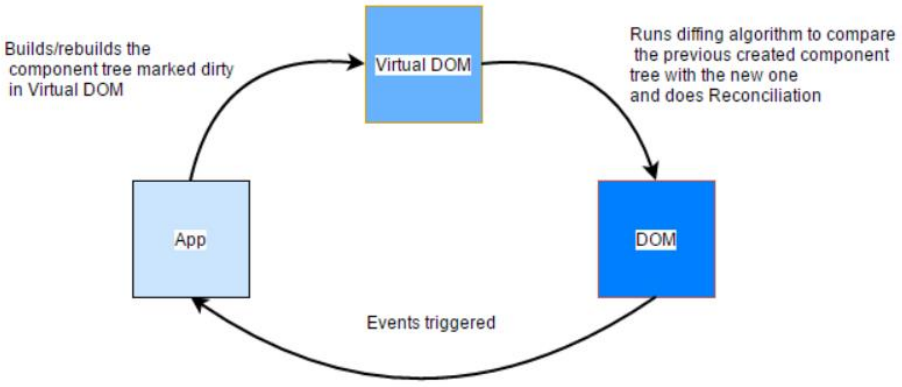
React is a JavaScript library for building user interfaces. ReactJS (or React.js/React) was developed by Facebook as a front-end JavaScript library for building user interfaces (UI). React uses a declarative style of programming to describe the UI state. React is used to build single page applications. React allows us to create reusable UI components. React uses JSX (JavaScript XML) in its component.

Important concept in React

- **Component:** Component in react is nothing but just a JavaScript methods or class, which takes input (props) from user and returns JSX.
- **Props:** This are the arguments we pass from parents' component to child component.
- **State:** State are nothing but a variable used for different purpose. When we want to re-render the component on basis of change in some variable value we use State.
- **React Hooks:** To use lifecycle methods in React functional components, the concept named hooks was introduced. Ex: useState, useEffect. UseDispatch, useContext

Why React is Fast?

-Reacts uses concept of virtual DOM.



- As we know, all the nodes are stored in windows document object model. So, whenever there is an update in site, we have to update the DOM. SO this process works differently in react then in other frameworks.
- **Reconciliation:** React always keep a copy of previous virtual DOM. So, when there is a change in any place in the site (States) react will compare the current snapshot of the virtual DOM with the previous snapshot and will find the difference. So, this process of finding the changes in DOM is called Reconciliation in React
- **Diffing in React:** After comparing the snapshot, react will only render the component which are updated in the DOM. This is called Diffing in React.

Difference between Angular and React

Dom Manipulation: React works on virtual DOM while Angular operates on real DOM.

Data Binding:

Data binding is about synchronizing data between business logic and UI.

The difference between Angular and React.js is that Angular 2 uses both **one- and two-way data binding**: changing data impacts view and changing view triggers changes in data.

React uses **one-way binding**: when designing a React app developers often nest child components within higher-order parent components.

Component Architecture

React approaches building UIs by breaking them into components. Here's how it works: you build components that manage their own state and structure them together into more complex UIs.

Still, if the project architecture is based on React, we need multiple integrations and supporting tools.

Some of them are:

- **Redux** (a state container which speeds up the work of React in large apps)
- **Babel** (converts JSX into JavaScript for the app to be understood by browsers)

- **Webpack** (a standard module bundler)

Unlike React, Angular is a pure full-fledged framework which comes with many out-of-the-box features like:

- **RxJS** introduces the concept of reactive programming to JS
- **Angular CLI** is a powerful command-line interface
- **Angular Universal** used for server-side rendering

Scalability

When choosing a tool, keep in mind the future scope of the project.

As I've said, react (versus Angular) relies heavily on third-party tools. Same in case you need to scale your app.

Yet a maintainable architecture with server-side rendering can still be easily developed with React. Facebook, WhatsApp, and Instagram all make use of React, and each of them is a large app at a global scale.

On the other hand, Angular (versus React) comes packed with all the core features developers may need for scaling an existing app by adding new functionality.

Advantage of React over Angular

React is better than Angular due to its virtual DOM implementation and rendering optimizations. Migrating between Reacts versions is quite easy, too; you don't need to install updates one by one, as in the case of Angular.

- **React has an easier learning curve**, so the ramp-up time is much shorter.
- **React offers a better mobile cross-platform framework** solution than Angular.

Advantage of Angular over React

- **Angular allows you to detect errors in compile-time**, as TypeScript is a statically typed language, instead of runtime (like in JavaScript).

- **Programmers that come from statically typed languages tend to prefer Angular as a framework**, as TypeScript lets you have explicit types in your code.
- **Angular ensures that data is always sync at all levels**, with the two-way data binding, that contrasts with one-way data binding of React.