**Angular:**

1. **What is Angular?**

* It is Open source.
* It is a client-side web application framework.
* Mainly used to build dynamic & SPAs(Single Page applications)
* Developed & maintained by Google.
* Written in TypeScript (a superset of JavaScript).
* Component-based architecture.
* AngularJS or Angular 1.0 released on October 20, 2010.
* Angular 2 released on September 14, 2016.

1. **Additional features**

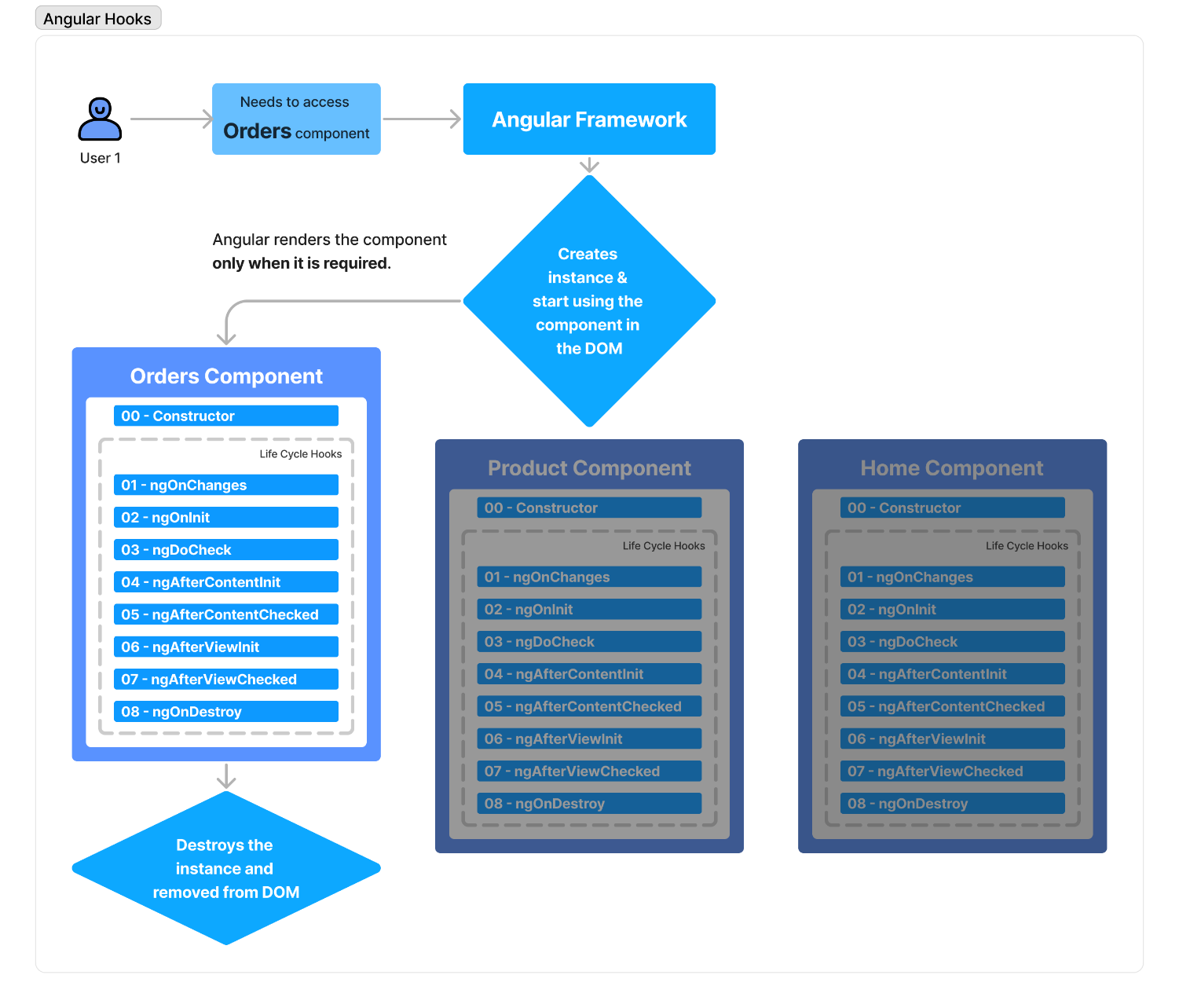
* Handles both **Presentation logic** (managing UI, rendering views, handling user interactions, displaying data) & **Business logic** (data processing, communication with servers by using services).
* Two-way data binding (changes in DOM also gets reflected in program data).
* Dependency Injection
* Directives for extending HTML(i.e., Managing DOM).
* Routing
* Handles asynchronous operations.
* Strong support for testing.

1. **Why Angular?**
   1. **For Developers**
      1. Clean structure, well organized & documented.
      2. Highly reusable code.
      3. Easy to learn and code.
      4. Provide libraries and 3rd party integrations.
      5. High testing support and support automated testing
      6. Two-way data binding, Dependency Injection.
      7. **Loosely coupled** (Independent, Flexible, Reusable, Scalable).
   2. **For Users**
      1. Fast loading (using Ahead-of-Time (AOT) compilation)
      2. Built-in security like **Cross-Site Scripting (XSS) Protection, Cross-Site Request Forgery (CSRF) Protection, Content Security Policy (CSP) Support.**
      3. Offline Support - **Progressive Web App (PWA)** can be integrated with Angular, enable users to access data even offline.
      4. High Performance (No Page refresh due to SPA).
2. **What is Angular CLI?**
   1. Angular Command Line Interface to scaffold angular applications.
   2. Create, build, test, run & deploy angular applications.
   3. Generates components, modules, services, models, etc.…
   4. **ng** command is used to perform such tasks.
   5. Angular CLI is built using Node.js.
3. **What is TypeScript?**
   1. Superset of JavaScript.
   2. Highlights errors during development.
   3. It has its own compiler called **tsc(TypeScript Compiler) –** which compiles the TypeScript code into JavaScript code to run into browser. This process called **transpilation** (source to source translator).
4. **Setup Development Environment for Angular:**
   1. Need latest version of node from nodejs.org.
   2. Need to install Angular CLI using npm.
      1. npm i - g @angular/cli
5. **What is Node.js**
   1. Node.js is an open source, cross-platform JavaScript runtime environment that allows developers to run JavaScript code on the server side.
   2. **Node.js acts as a development server for serving Angular applications to web browsers** with help of Angular CLI.
   3. Built on V8 JavaScript engine from Google.
   4. Used to build server-side applications (like web servers, APIs, real-time applications etc...)
   5. Has large ecosystem of packages and libraries available through npm (Node Package Manager).
6. **What is npm?**
   1. Npm is a package manager for Node.js.
   2. It is a tool for managing/sharing libraries and tools.
   3. Used to install, share, and manage project dependencies.
7. **New Angular project structure**

**Angular concepts:**

**Life Cycle Hooks:**

* Angular Lifecycle hooks are a **set of pre-defined methods** in Angular that are executed at specific lifecycle of component, directives, and services.
* Inherited from interface.
* Ex. **ngOnInit()** method inherited from **OnInit** interface.]
* Angular renders the specific component only when it is required as shown below



**Types**

1. ngOnChanges()
2. ngOnInit()
3. ngDoCheck()
4. ngAfterContentInit()
5. ngAfterContentChecked()
6. ngAfterViewInit()
7. ngAfterViewChecked()
8. ngOnDestroy()

**ngOnChanges()**

Change detection in Angular is a mechanism by which Angular keeps the view template in sync with the component class. Ex. {{name}}.

* ngOnChanges() will be executed when there is **Change detection** occurs.
  + Only called when **@input property** of component changes.
  + 
* Can assign parameter like **SimpleChanges** which holds **previous value, currentValue, firstChange** properties.

**ngOnInit()**

* It will be called after ngOnChanges().
* It will be called only during first change detection.

**ngDoCheck()**

* It will be called for every Change Detection.