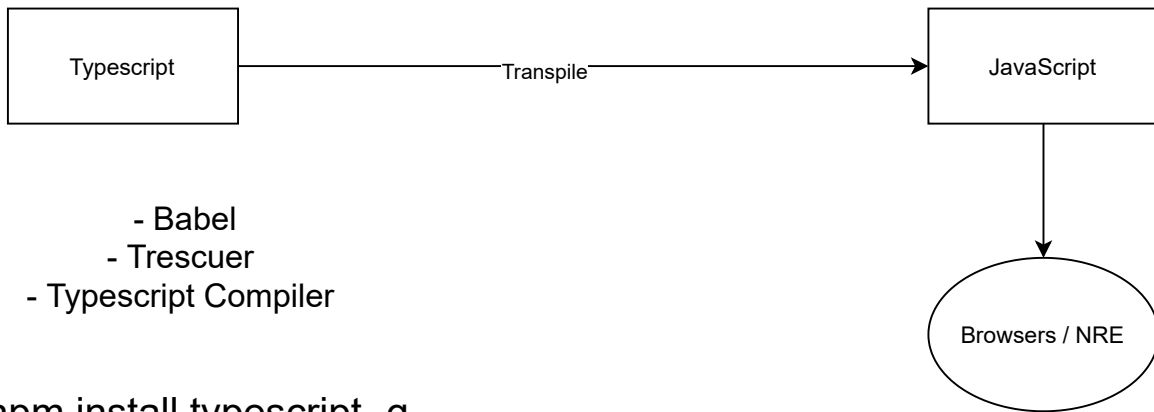


JS Libraries and Frameworks-

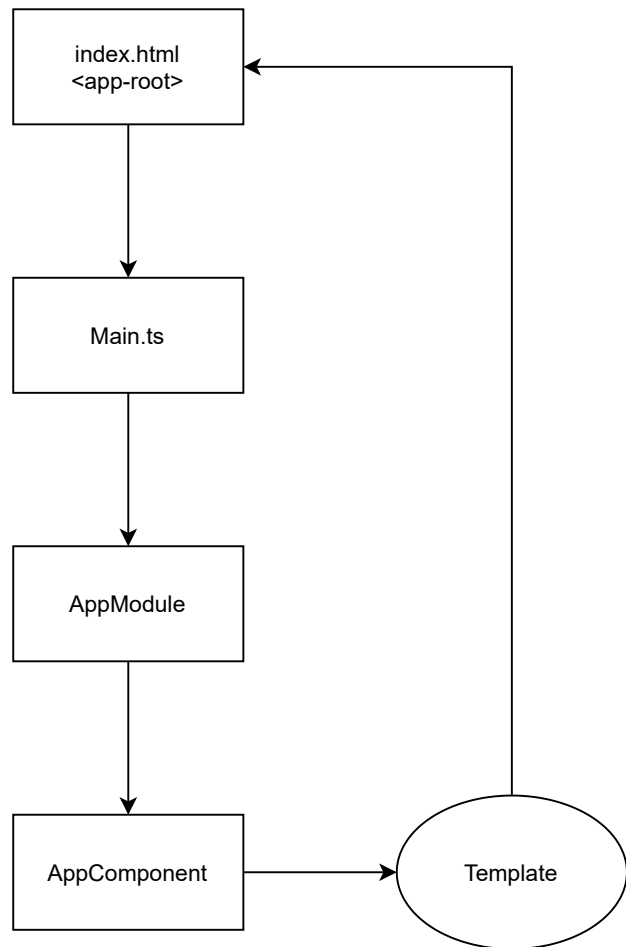
- jQuery : DOM manipulation and ajax
- React - model is changed very frequently. Component based architecture
- KnockoutJS - MVVM- 2 way data binding
- BackboneJS - MVC at client end
- PolymerJS - Custom Components
- EmberJS - Framework, Frequent internal API Changes
- NodeJS - platform; NRE - Server-side script
- ExpressJS - Web app framework for Node App
- VueJS - Library
- Angular : DOM Manipulation, 2 way data binding, SPA, template, MVC, Component etc,

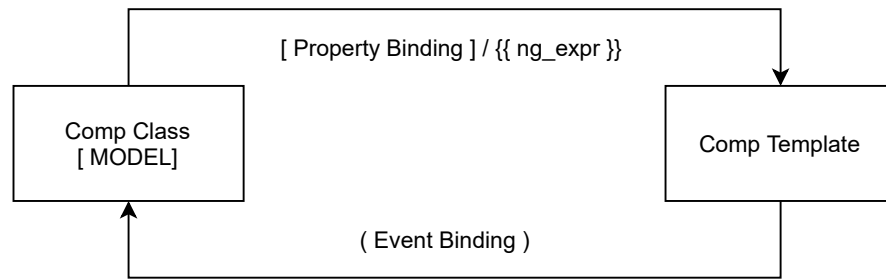


- Babel
- Trescuer
- Typescript Compiler

> npm install typescript -g

Angular Bootstrap Process

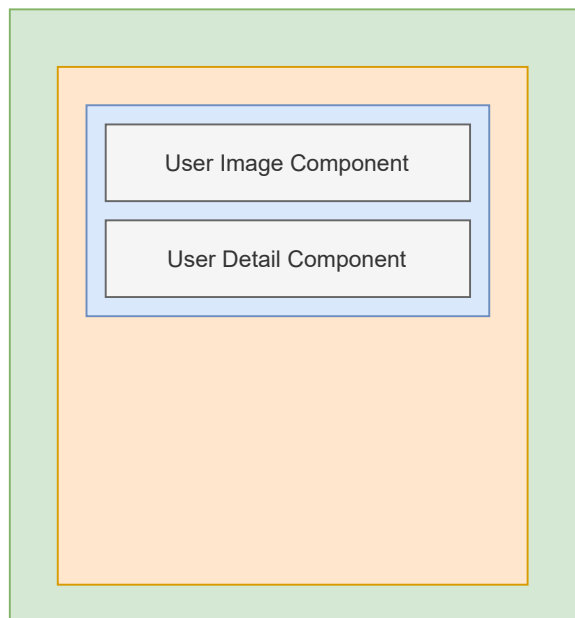
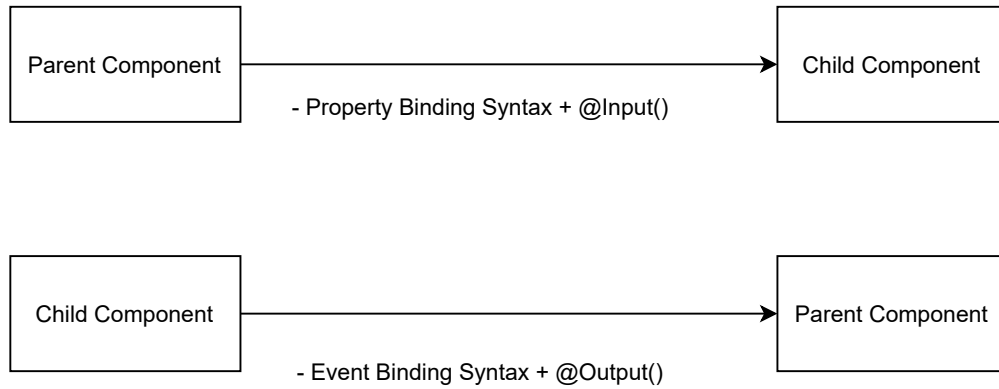


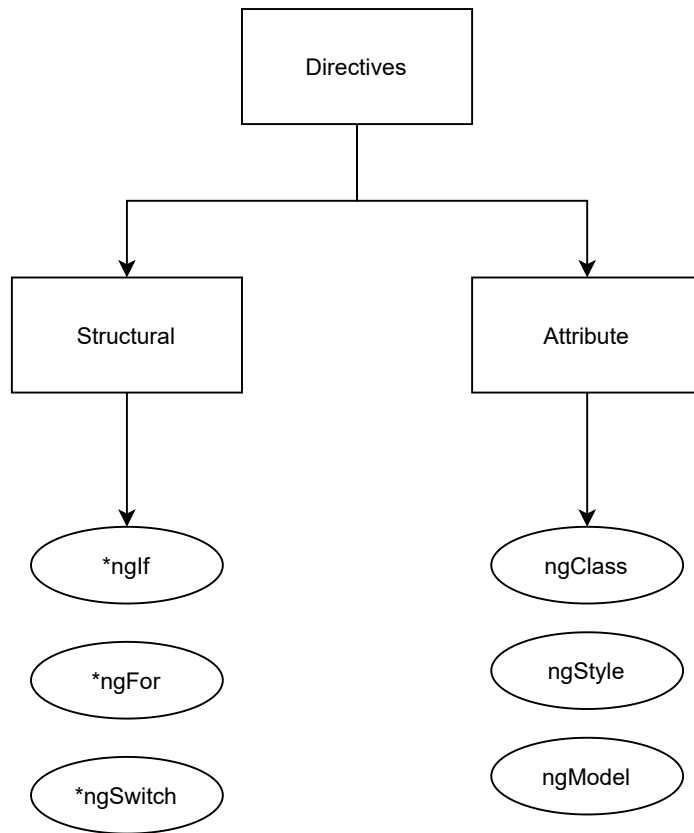


2 way data binding

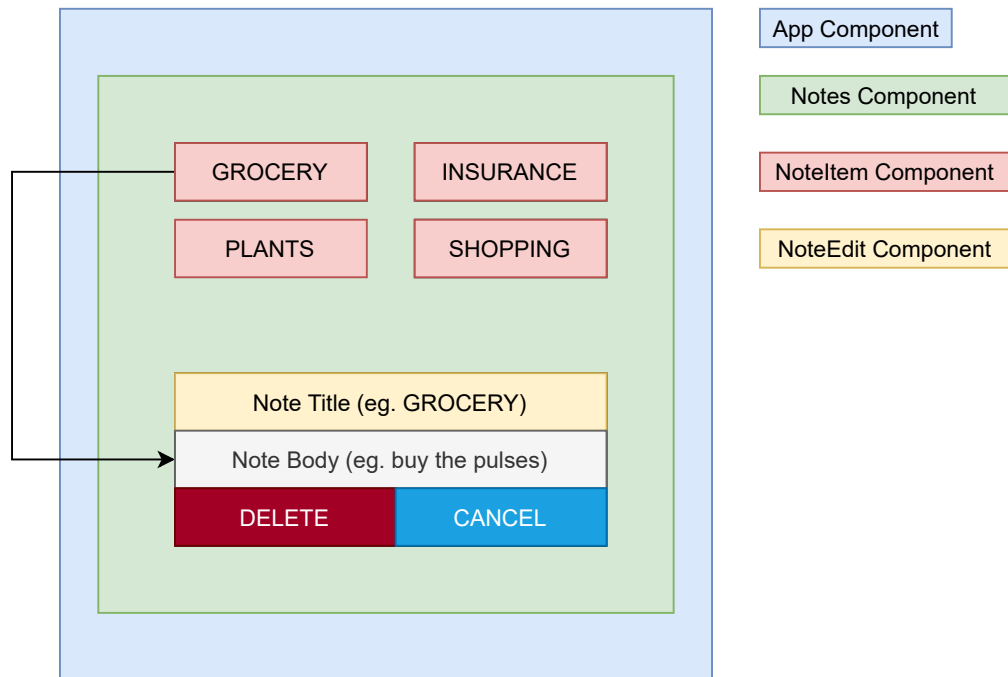
- Clubbing the [PB] + (EB)
- Banana in the Box - [(ngModel)]

Component Communication





Notes App



The Notes App - Instructions

- App should contain the notes collections as follows :

```
[  
  { id : 1, title : "grocery", body : "to buy the pulses"},  
  { id : 2, title : "shopping", body : "to buy the jeans"},  
  { id : 3, title : "plants", body : "to pot the plants"},  
  { id : 3, title : "insurance", body : "to renew car insurance"},  
]
```

- Should create three components mentioned in picture :

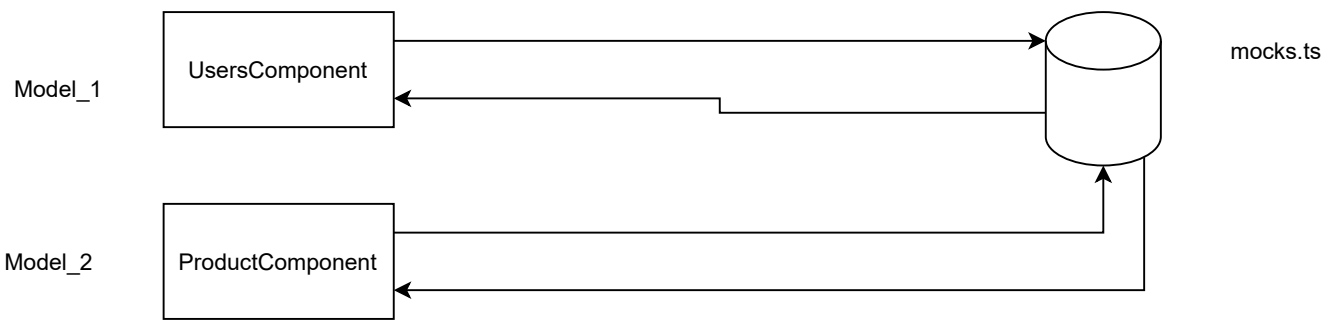
- Notes : Container component, will contains NoteItem and NoteEdit Component
- NoteItem : displays individual item present in notes collection
- NoteEdit : display the note body and enables to delete the note from the collection

- Should display NoteEdit component on click of any of the NoteItem

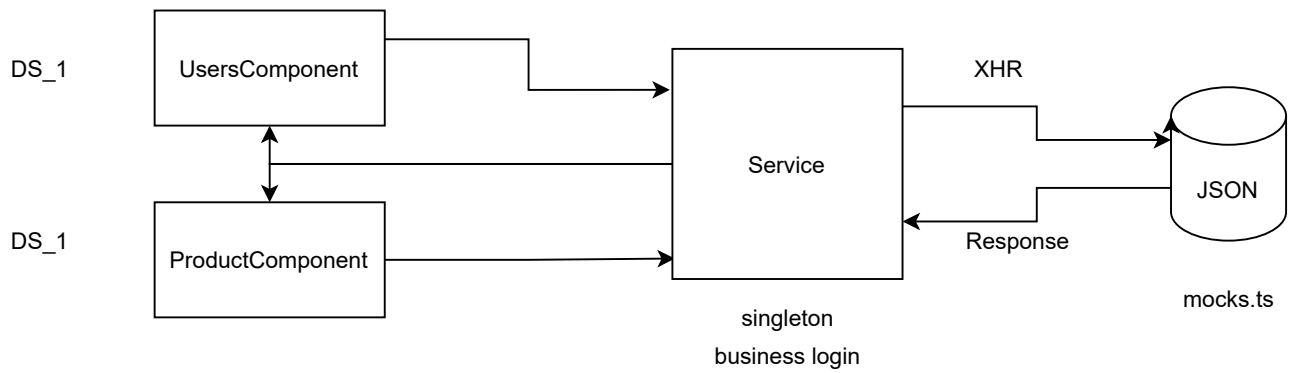
- Should use @Input / @Output for component communication.

Form & Form element - State and Classes

- ngValid | ngInvalid
- ngPristine | ngDirty
- ngTouched | ngUntouched

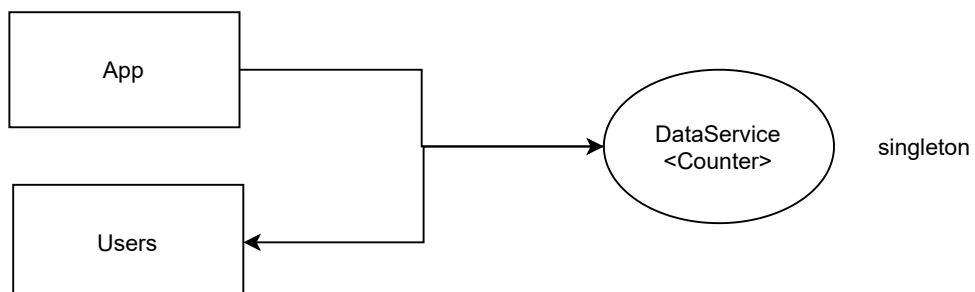
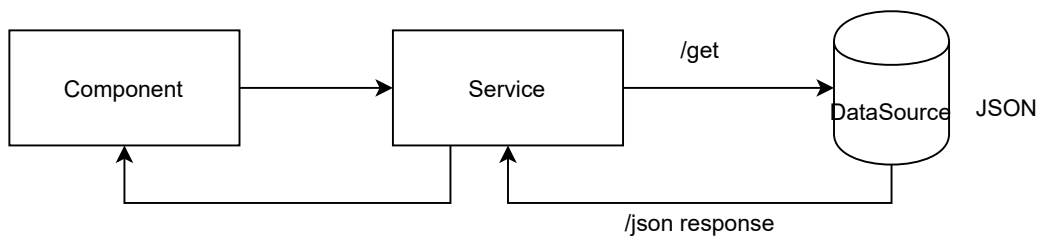


data service log service auth service

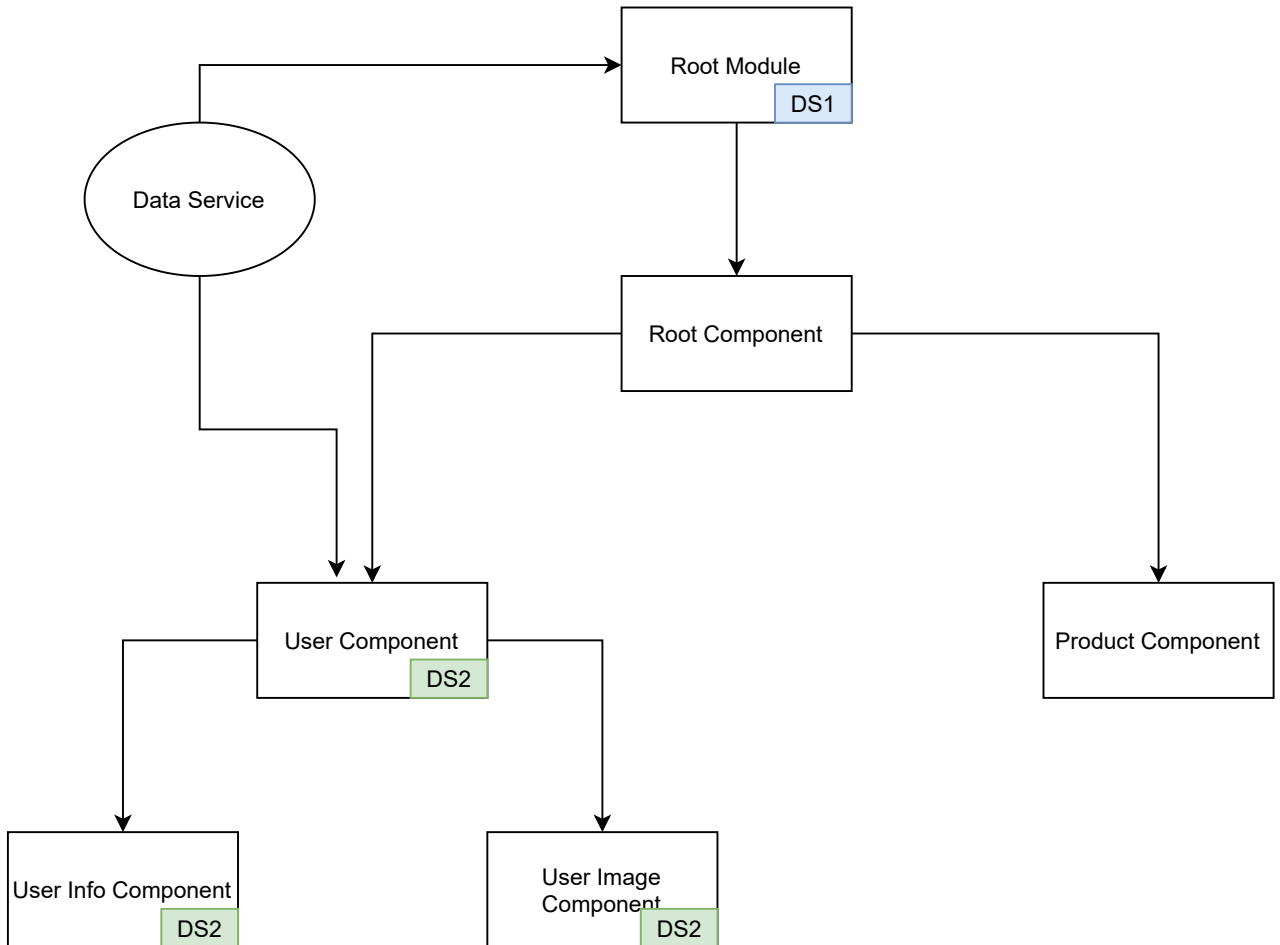


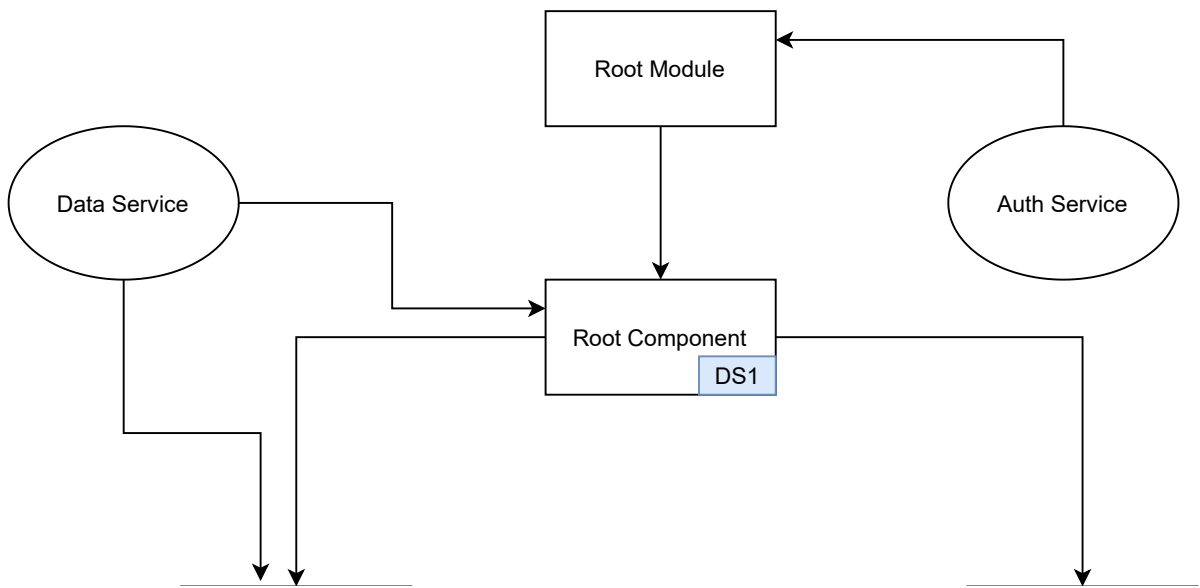
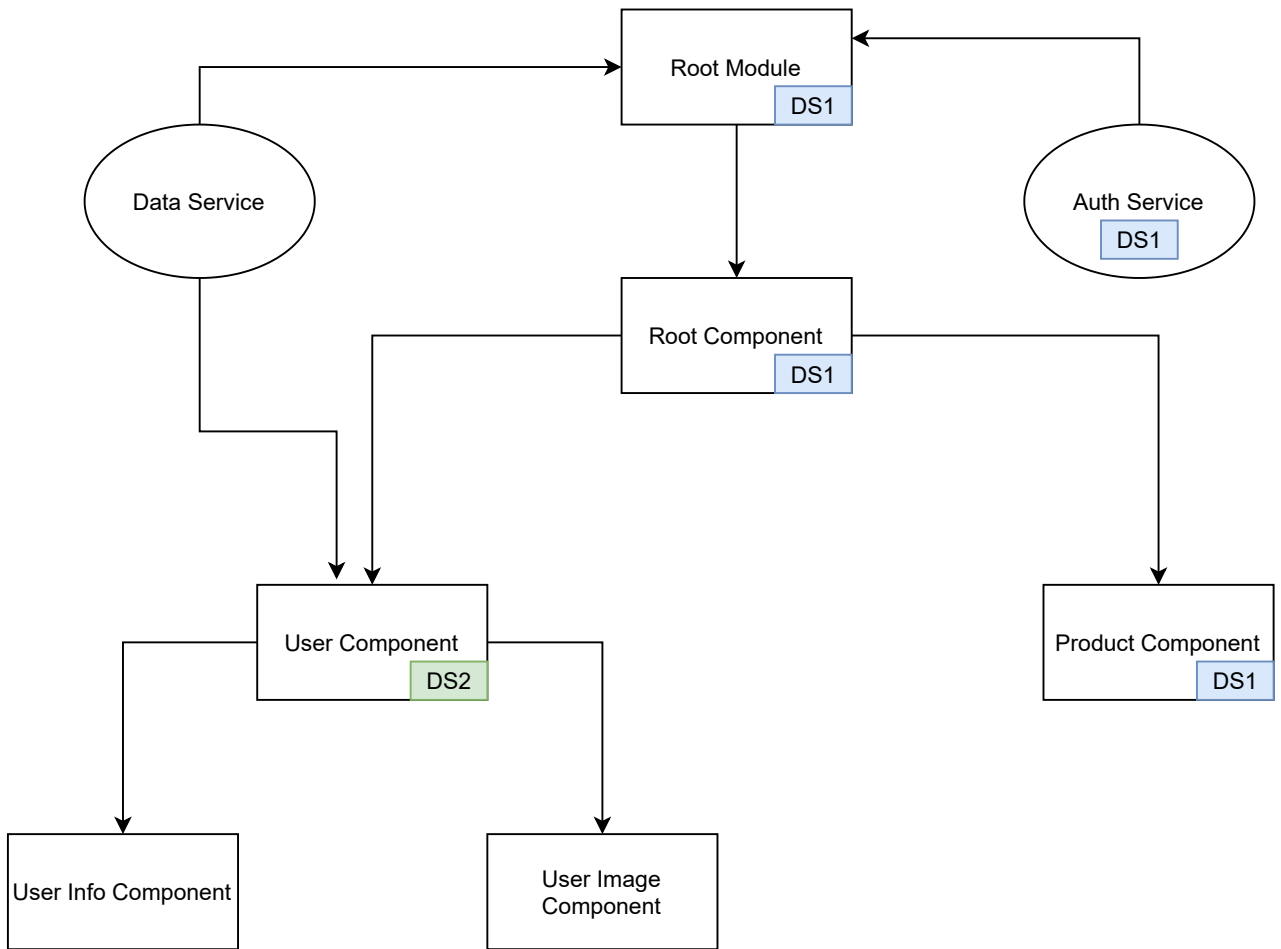
Service Principles

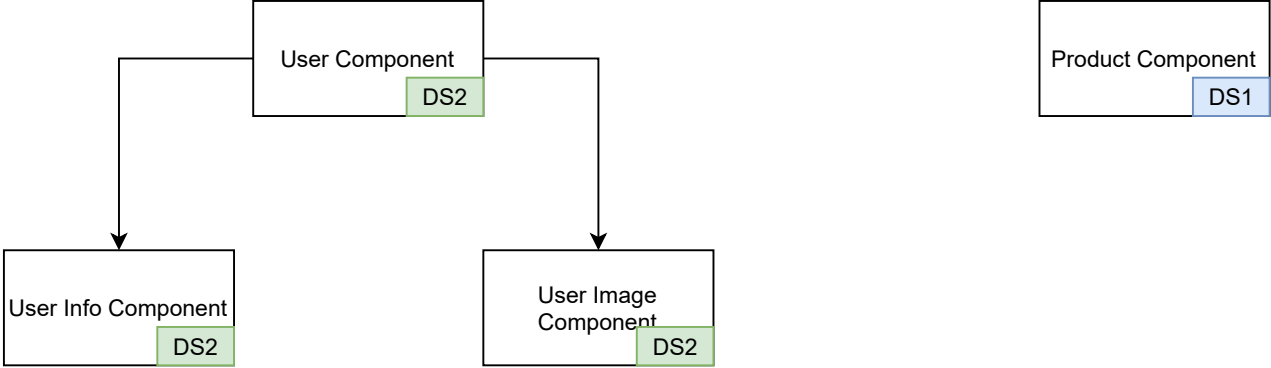
- Single Responsibility
- Dependency Injection



Multiple Service Registration







Handling JS Async -

- Callback Functions

- Promises

- Async ... await

- Observables

Interceptors

