Dependency Injection

https://angular.dev/guide/di

Dependency Injection

Kas tai?

- Dizaino šablonas (design pattern), leidžiantis lengviau dalintis priklausomybėmis.
- Automatizuoja priklausančių klasių kūrimą, bei leidžia keisti jas skirtinguose lygmenyse.
- Aprašomas `@Injectable` arba `@InjectionToken` dekoratoriumi.
- Importuojamas konstruktoriuje, arba naudojant `inject()` pagalbinę f-ją.

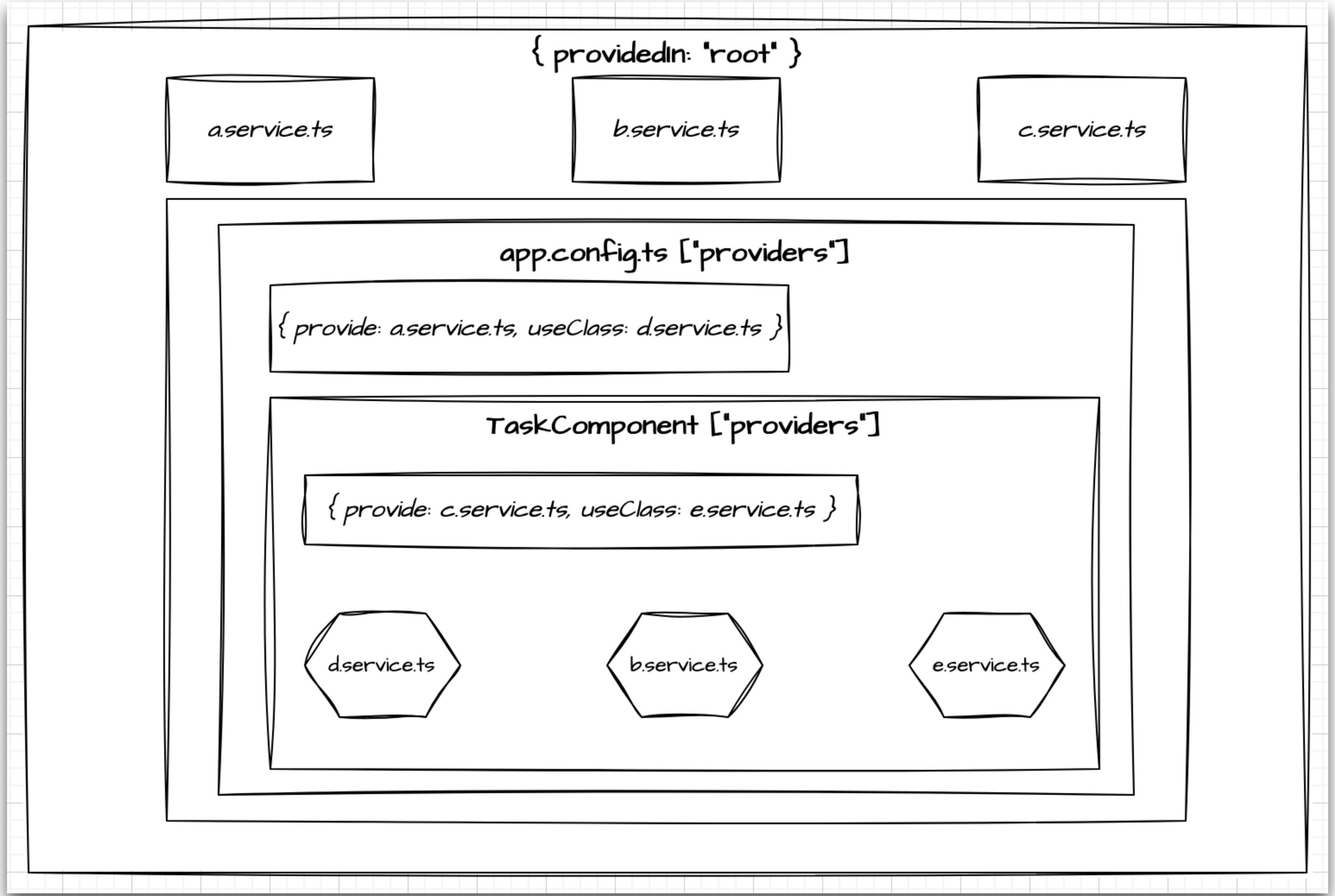
```
@Injectable({
  providedIn: 'root'
})
export class ExampleService {
  constructor() { }
}
```

```
export class MainComponent {
  #service = inject(ExampleService);
}
```

Dependency Injection

Kodėl to reikia?

- Padeda:
  - Atskirti priklausomybes nuo komponentų.
  - Pakeisti priklausomybes testuojant.
  - Keisti priklausomybių logiką skirtinguose aplikacijos sluoksniuose.


{ providedIn: "root" }
a.service.ts
b.service.ts
c.service.ts
app.config.ts ["providers"]
{ provide: a.service.ts, useClass: d.service.ts }
TaskComponent ["providers"]
{ provide: c.service.ts, useClass: e.service.ts }
d.service.ts
b.service.ts
e.service.ts

Užduotis #1 - Perkelti užduotis į service.

1. **Pridėti tasks parametrą**, kuris būtų prieinamas iš išorės (eksponuotų užduočių sąrašą).

2. **Sukurti addTask metodą**, kuris leistų pridėti naują užduotį.

    - Užduotys turėtų automatiškai gauti unikalų identifikatorių (id) jas pridedant.

3. **Sukurti getTaskById metodą**, kuris pagal pateiktą id grąžintų konkrečią užduotį.