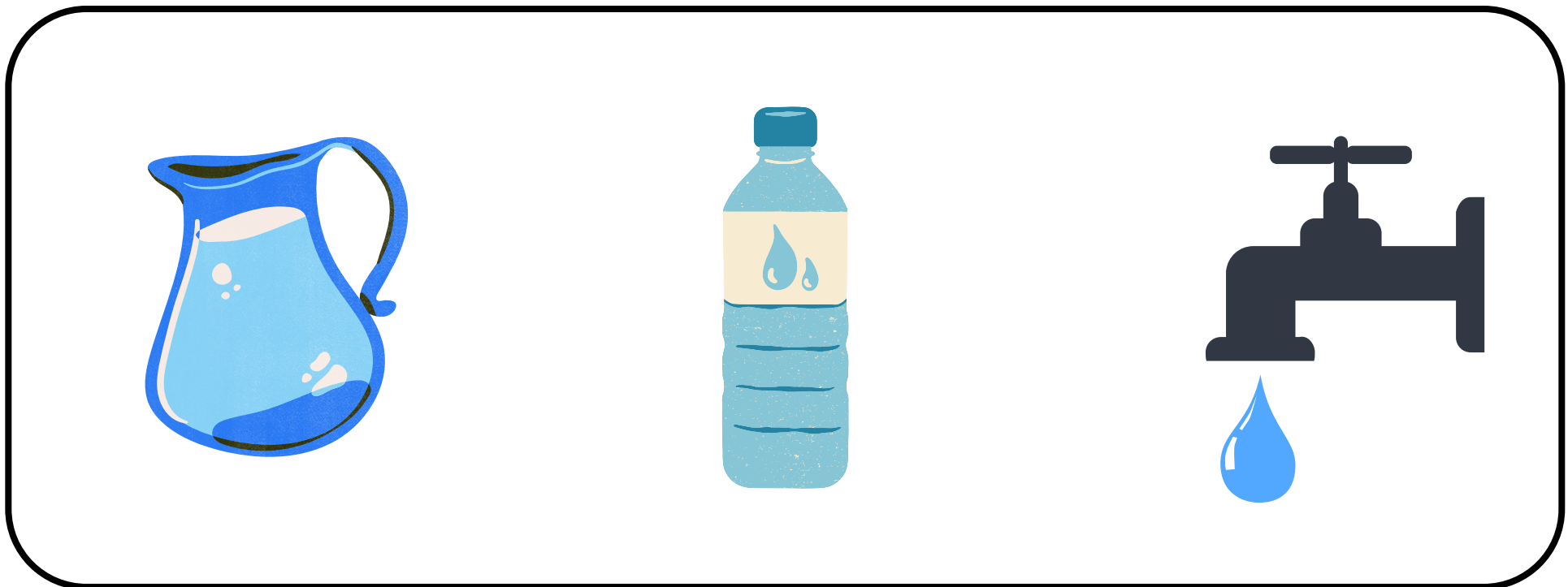


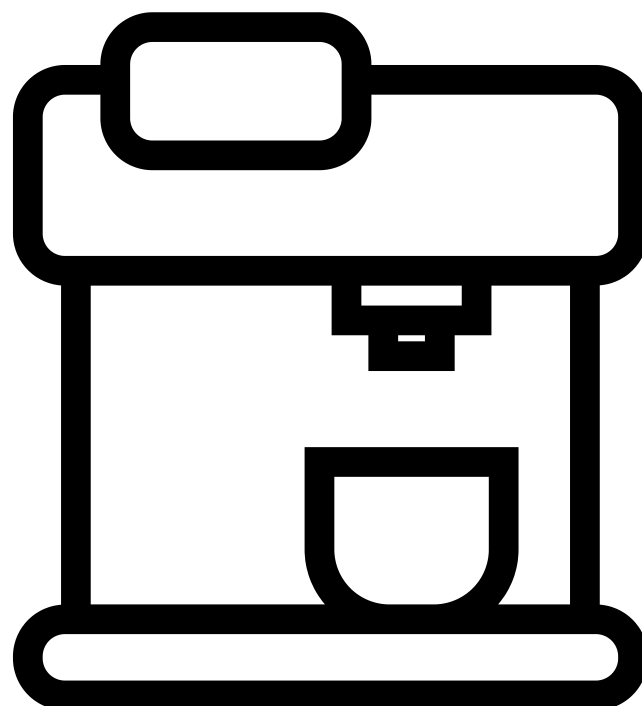
Understand DI & IoC

Analogy

:IWaterProvider

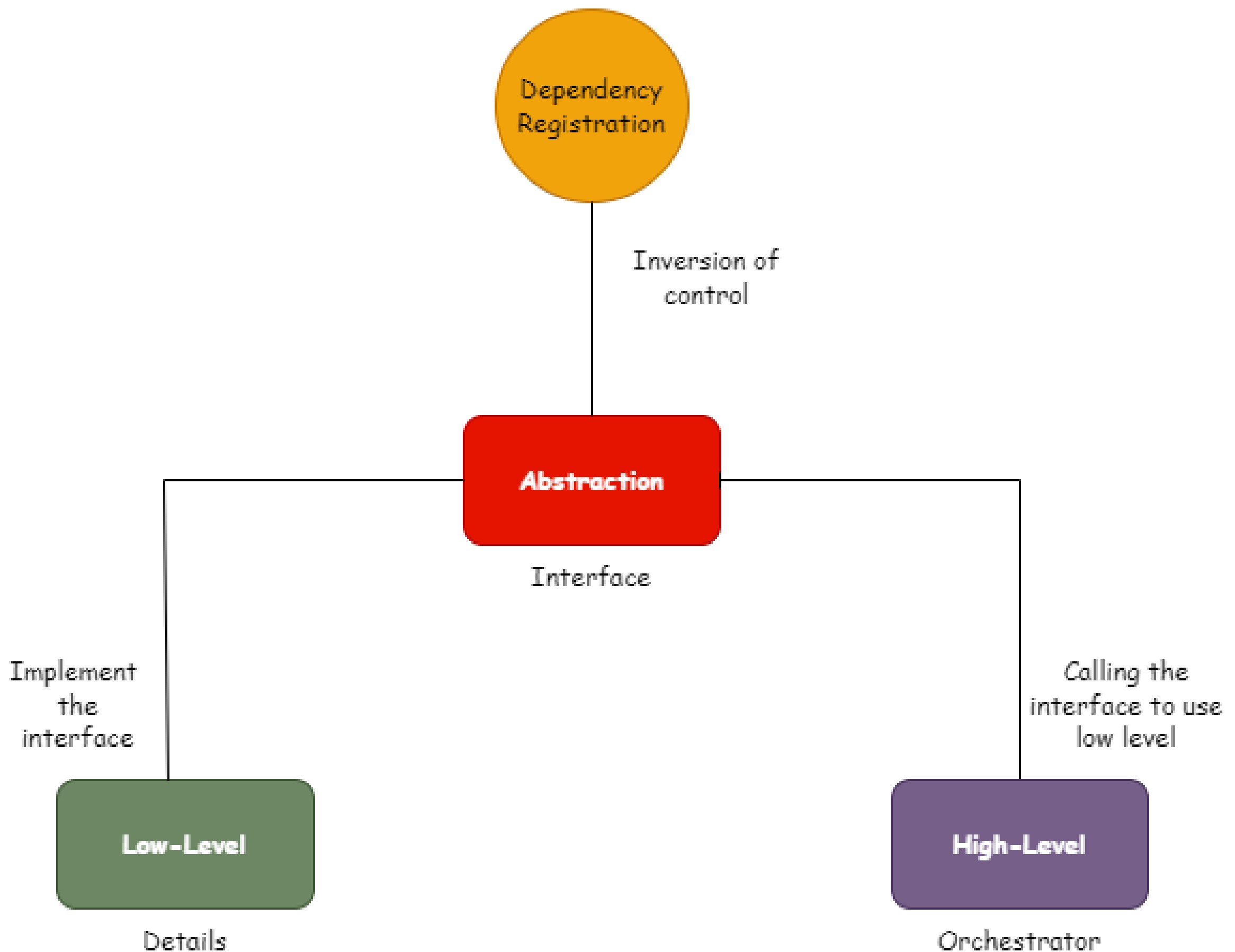


A coffee machine needs water to boil,
but it doesn't matter who supplies the
water



@romain-od

How it's works



Abstraction



```
public interface IWaterProvider
{
    public Water GetWater();
}
```



@romain-od

Low-Level



```
public class Carafe : IWaterProvider
{
    public Water GetWater()
    {
        //Implement logic of a Carafe
        //let's rocks you imaganation
        return new Water();
    }
}
```



@romain-od

High-Level



```
public class CoffeeMachine(IWaterProvider waterProvider)
{
    public Tank WaterTank { get; set; }
    public void FillTank()
    {
        WaterTank = new Tank
        {
            Content = waterProvider.GetWater()
        };
    }
}
```



@romain-od

Dependence Registration



```
services.AddSingleton<IWaterProvider, Carafe>();
```



```
services.AddScoped<IWaterProvider, Carafe>();
```



```
services.AddTransient<IWaterProvider, Carafe>();
```

Which one choose ?

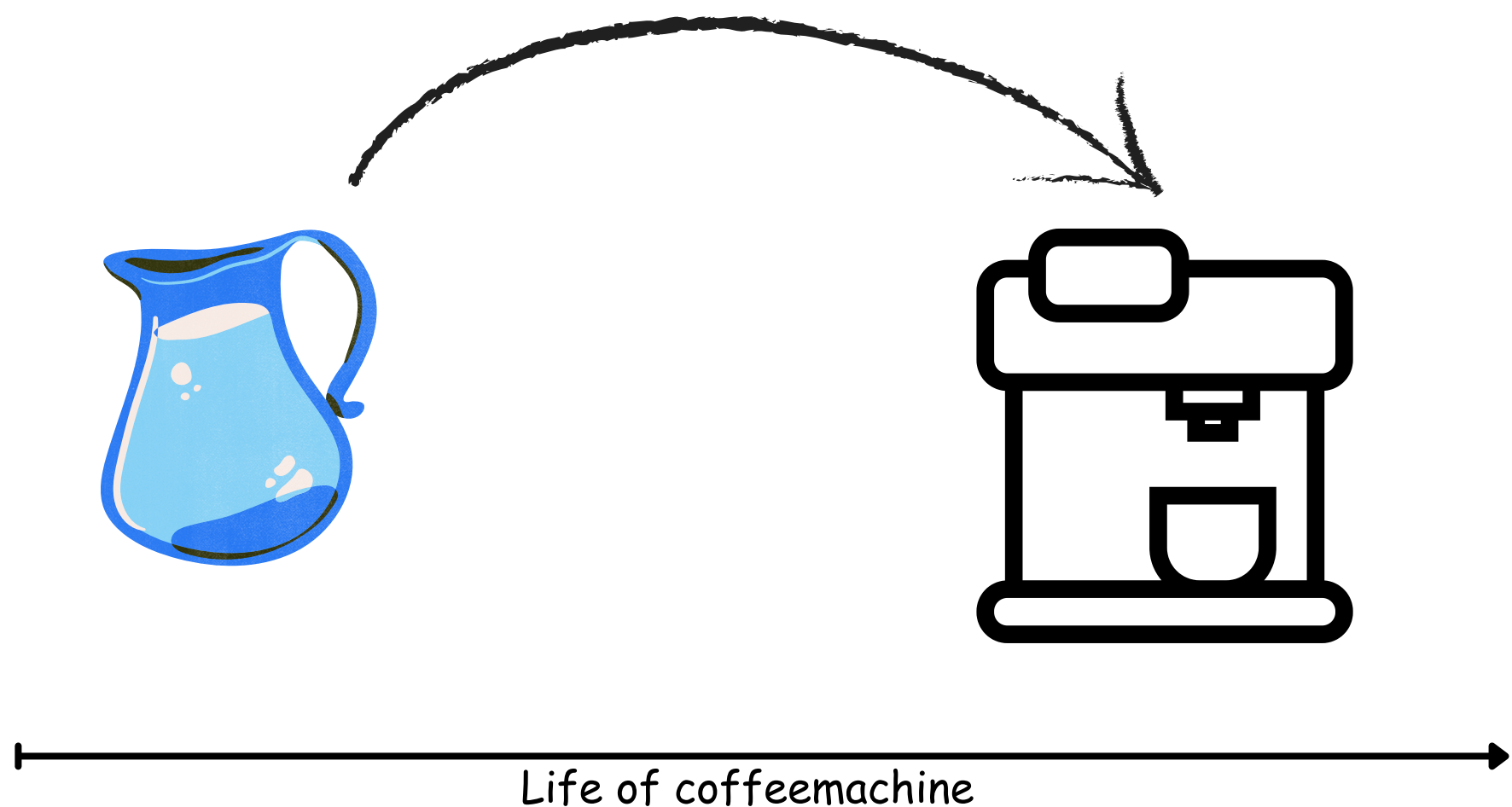


@romain-od

Singleton

A single instance of the dependency is created and shared across the entire application.



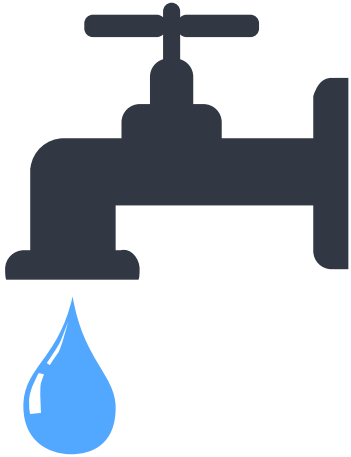
The singleton approach is lifetime, it means every day you use the coffee machine you use a carafe to fill the water tank.



Scoped

A new instance of the dependency is created once per scope. A scope can be thought of as a unit of work.

The scoped approach, it's like today I committed to use the carafe, tomorrow I will use a bottle.

Monday	Tuesday	Wednesday
		



@romain-od

Transient

A new instance of the dependency is created each time it is requested

With transient approach, we take the first provider we found to fill the water tank each time we use the coffee machine.

8:00	8:45	10:00	12:35
			



What about reference ?

Can we use Scoped service in Singleton?
Can we use Singleton service in Transient?
Can we use Transient service in Scoped?
Can we use Singleton service in Scoped?
Can we use Scoped service in Transient?
...



Captive Dependency

Captive Dependency is a dependency with an incorrectly configured lifetime. A service should never depend on a service that has a shorter lifetime than its own.

	Transient	Scoped	Singleton
Transient	✓	✓	✓
Scoped		✓	✓
Singleton	✗	✗	✓



Can we use a different provider to fill the Water Tank and rinse the filter ?

.NET 8 introduce keyed services

Keyed services managing Dependency Injection (DI) services by associating them with keys for registration and retrieval.

We use methods `AddKeyedSingleton`, `AddKeyedScoped`, or `AddKeyedTransient` to register a service with a specific key.

We use the `[FromKeyedServices]` attribute and specify the corresponding key to access a registered service



Keyed services



```
services.AddKeyedSingleton<IWaterProvider, Faucet>("Faucet");  
services.AddKeyedSingleton<IWaterProvider, Carafe>("Carafe");
```



```
public class Filter([FromKeyedServices("Faucet")] IWaterProvider waterProvider)  
{  
    public void Rinse()  
    {  
        Water liquid = waterProvider.GetWater();  
        //Use liquid to Rinse the filter  
    }  
}
```



```
public class WaterTank([FromKeyedServices("Carafe")] IWaterProvider waterProvider)  
{  
    public Water Content { get; set; }  
    public void FillTank()  
    {  
        Content = waterProvider.GetWater();  
    }  
}
```





Thanks for reading
Share your thoughts
Each one Teach one

Follow me for more sharing

