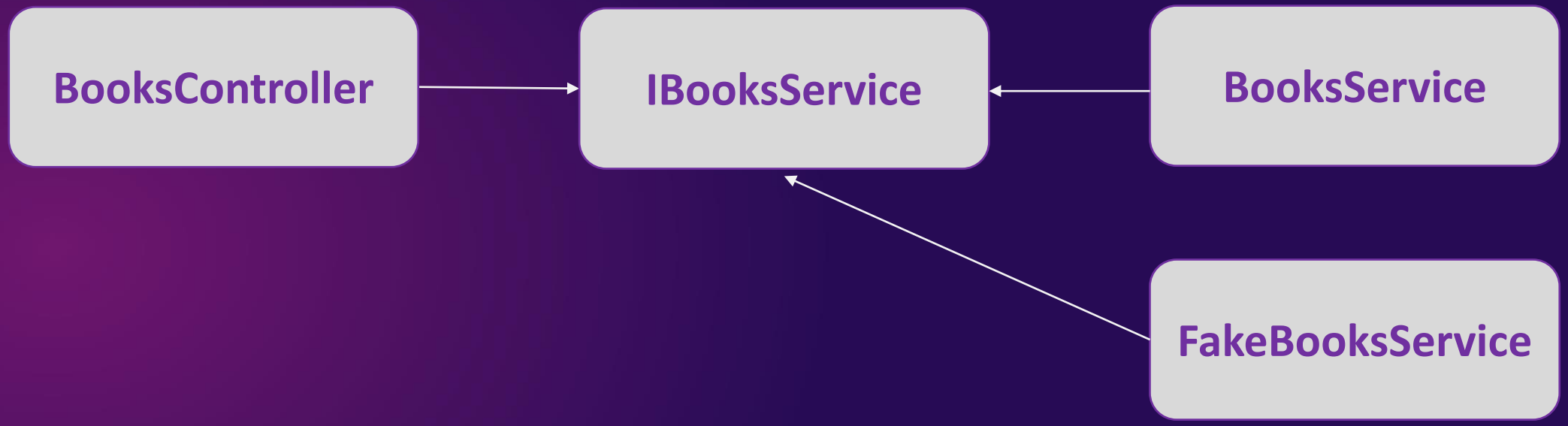
Code Dependencies

Dependency Inversion Principle

1. High-level modules should not depend on low-level modules. Both should depend on abstractions.

2. Abstractions should not depend on details. Details should depend on abstractions.

Dependency Injection



Service Lifetime

Transient	Creates a new instance every time.	When we register a service with the transient lifetime, it will create a new instance every time.
Scoped	Creates just one instance per request and shares it within the request.	Scoped services are a good choice when we want to maintain the state within a request. For example, consider we're calling a database multiple times in the same request. In this case, it is best to use a scoped lifetime as it keeps the same repository object in the memory and reuses it within the same request.
Singleton	Creates just a single instance and shares it across the application	In theory, a singleton service is the most memory-efficient as they are created once and reused everywhere. Also, it is an excellent choice when we want to maintain an application-wide state. Application configuration, logging service, data caching, etc. are some of the scenarios where we can take advantage of singleton services.