



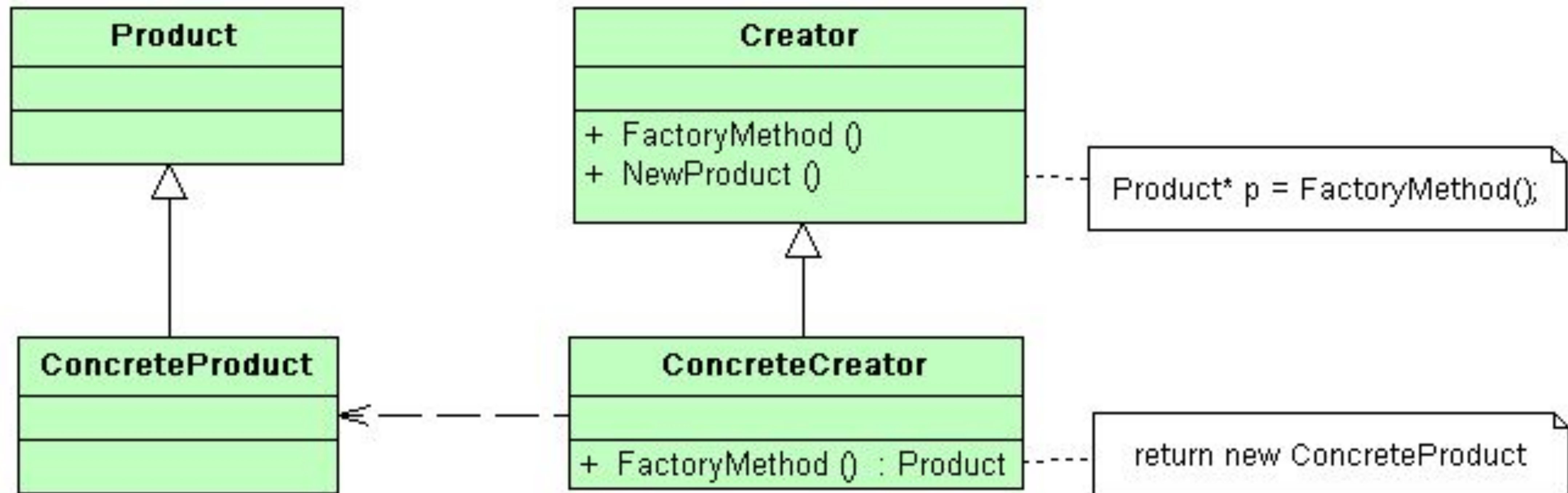
spring

by Pivotal™

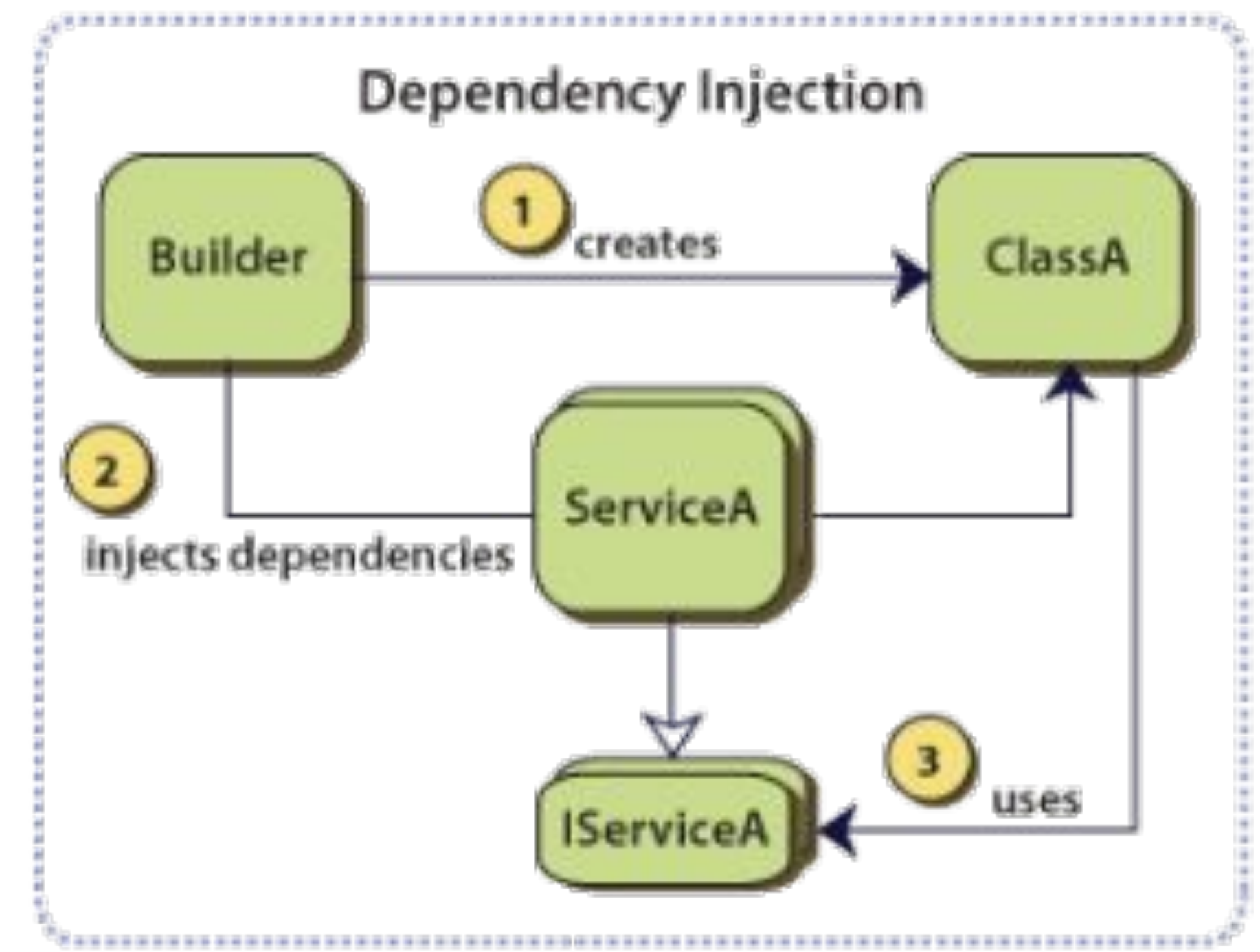
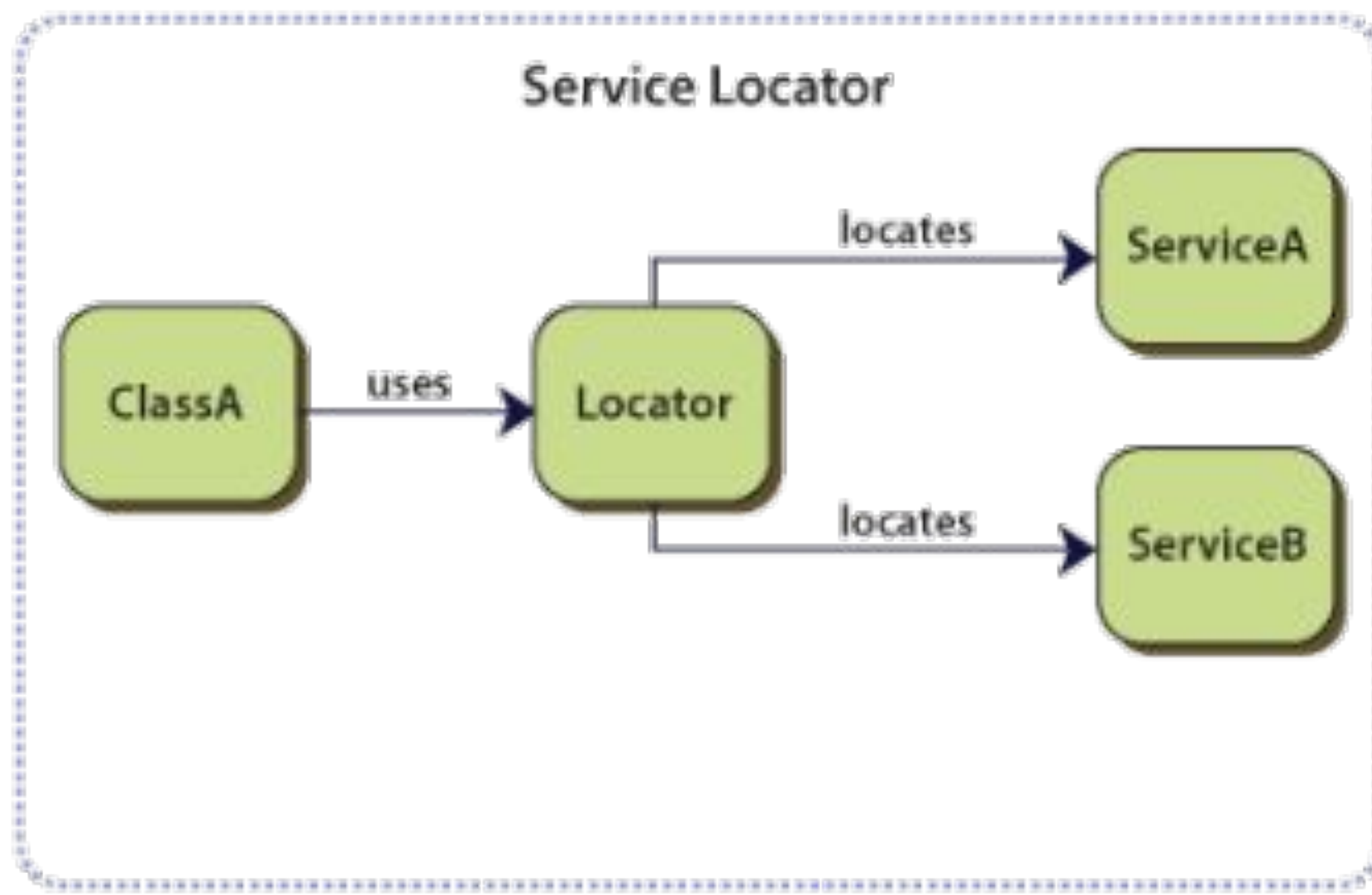
Plan

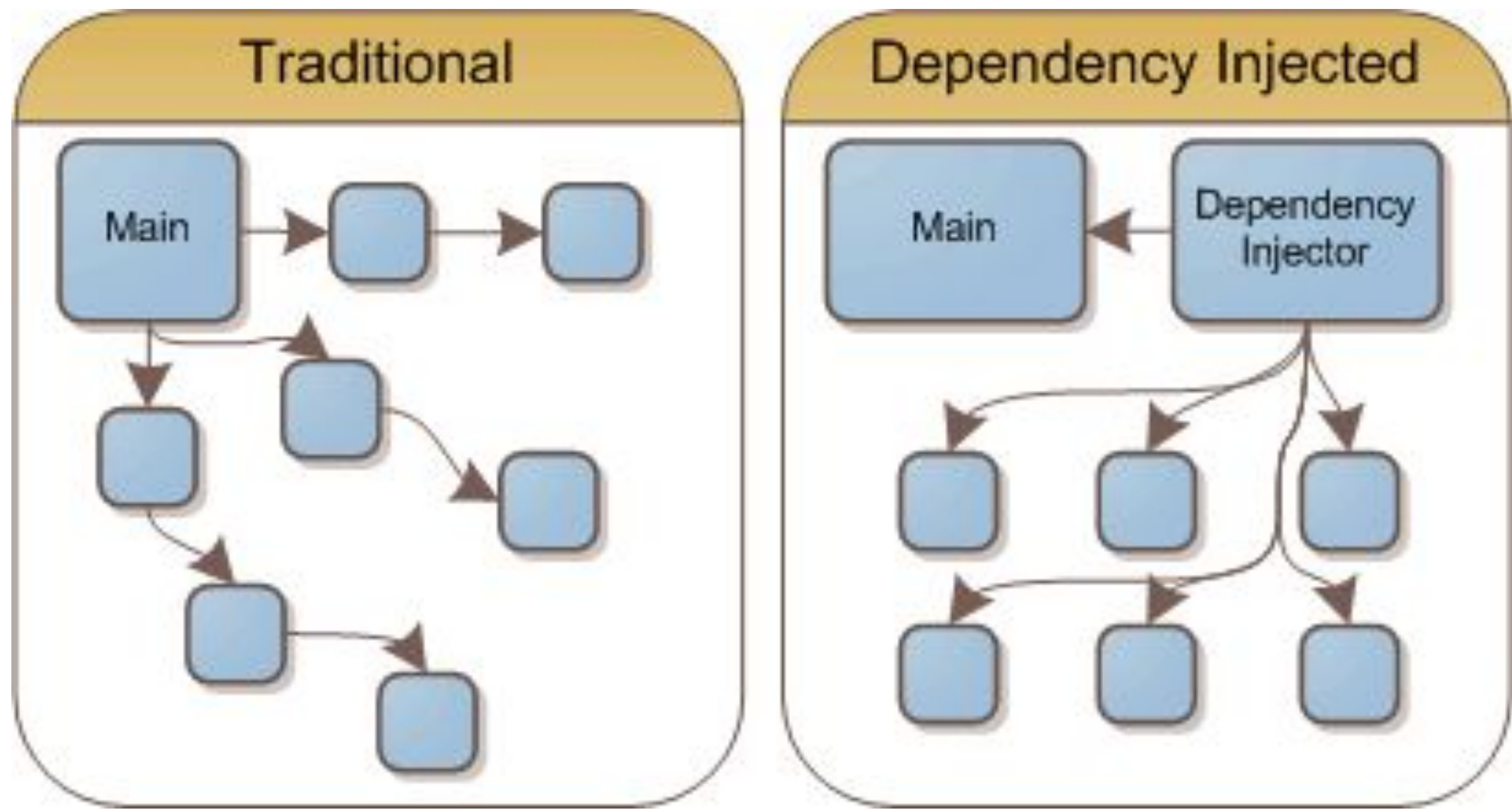
- Factory Method
- Service Locator
- Dependency Injection (Spring Framework)
- Summary

Factory Method



SL and DI





Inversion of Control

- IoC - design principle
- The principle states:
 - A. High-level modules should not depend on low-level modules. Both should depend on abstractions
 - B. Abstractions should not depend on details. Details should depend on abstractions

Summary: FM vs SL

A factory creates objects for you, when requested. Service locator returns objects that may already exist.

- Factory: is a place where objects are created
- Service: is something that can do something for you as a service
- Service locator: is something that can find something that can perform a service

Summary: SL vs DI

- Dependency injection tends to be hard to understand and hard to debug
- With Service Locator every “user” of a service has a dependency on the Service Locator
- Using dependency injection, dependencies can be more clear
- Dependency injection might make testing easier, but a Service Locator can also be testable **if it is correctly designed**

Wrong:

```
public Foo() {  
    this.bar = new Bar();  
}
```

Wrong:

```
public Foo() {  
    this.bar = ServiceLocator.Resolve<Bar>();  
}
```

Wrong:

```
public Foo(ServiceLocator locator) {  
    this.bar = locator.Resolve<Bar>();  
}
```

Right:

```
public Foo(Bar bar) {  
    this.bar = bar;  
}
```

Only the latter makes the dependency on `Bar` explicit.

References

- <https://steveschols.wordpress.com/2012/05/14/dependency-injection-vs-service-locator/>
- <https://martinfowler.com/articles/injection.html>