

SPRING FRAMEWORK

백재현

목록

- 프레임워크란?
- 스프링 프레임워크란?
- 스프링 프레임워크 역사
- 스프링프레임워크 구조
- 스프링 프레임워크 특징
- 정리
- with SPRING VS only JAVA

“저는 ‘Framework’입니다”

spring

Framework ?

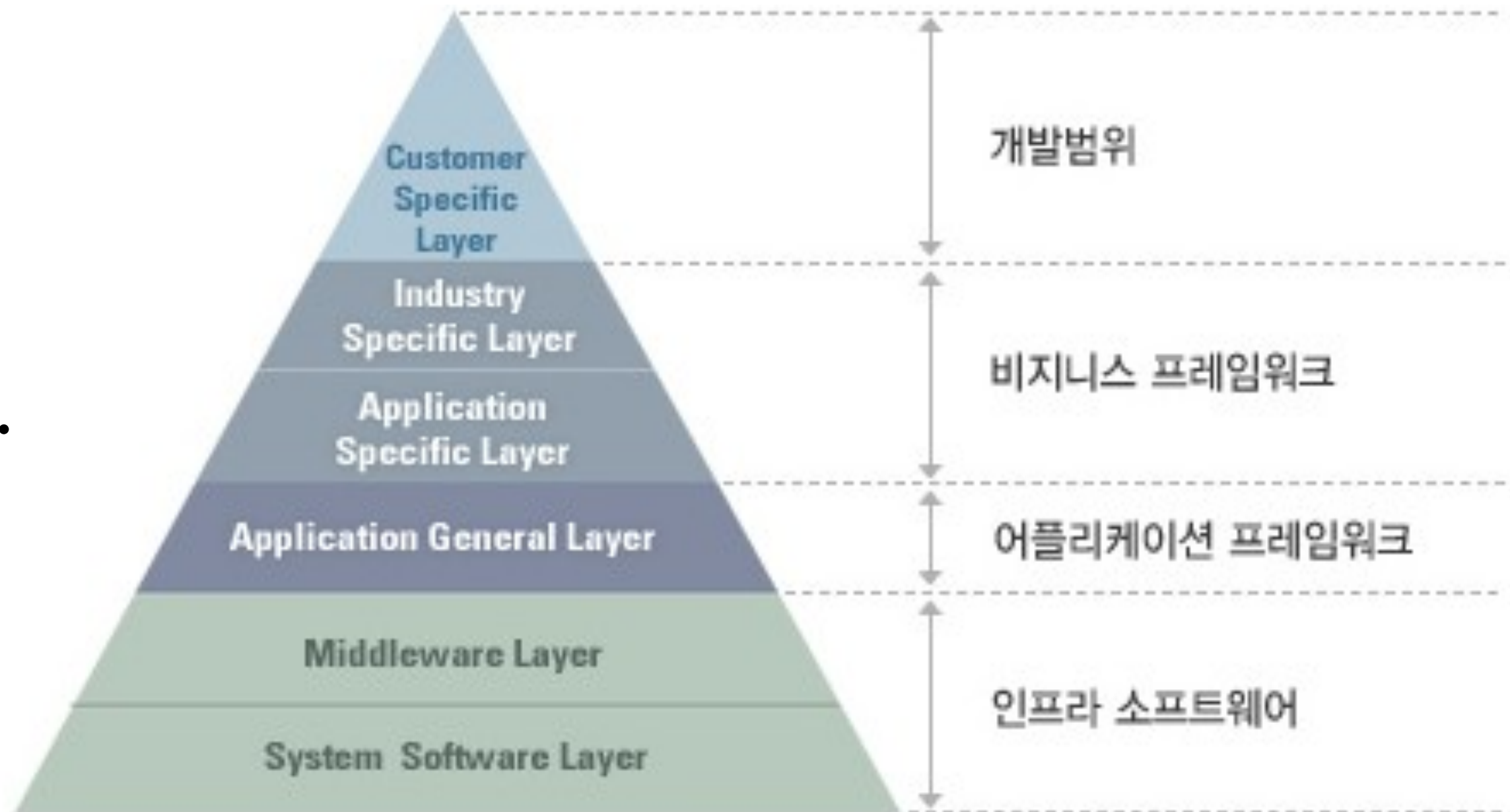
“소프트웨어의 구체적인 부분에 해당하는 설계와 구현을 재사용이 가능하게끔 일련의 협업화된 클래스들을 제공하는 것”

랄프 존슨(Ralph Johnson) - 교수

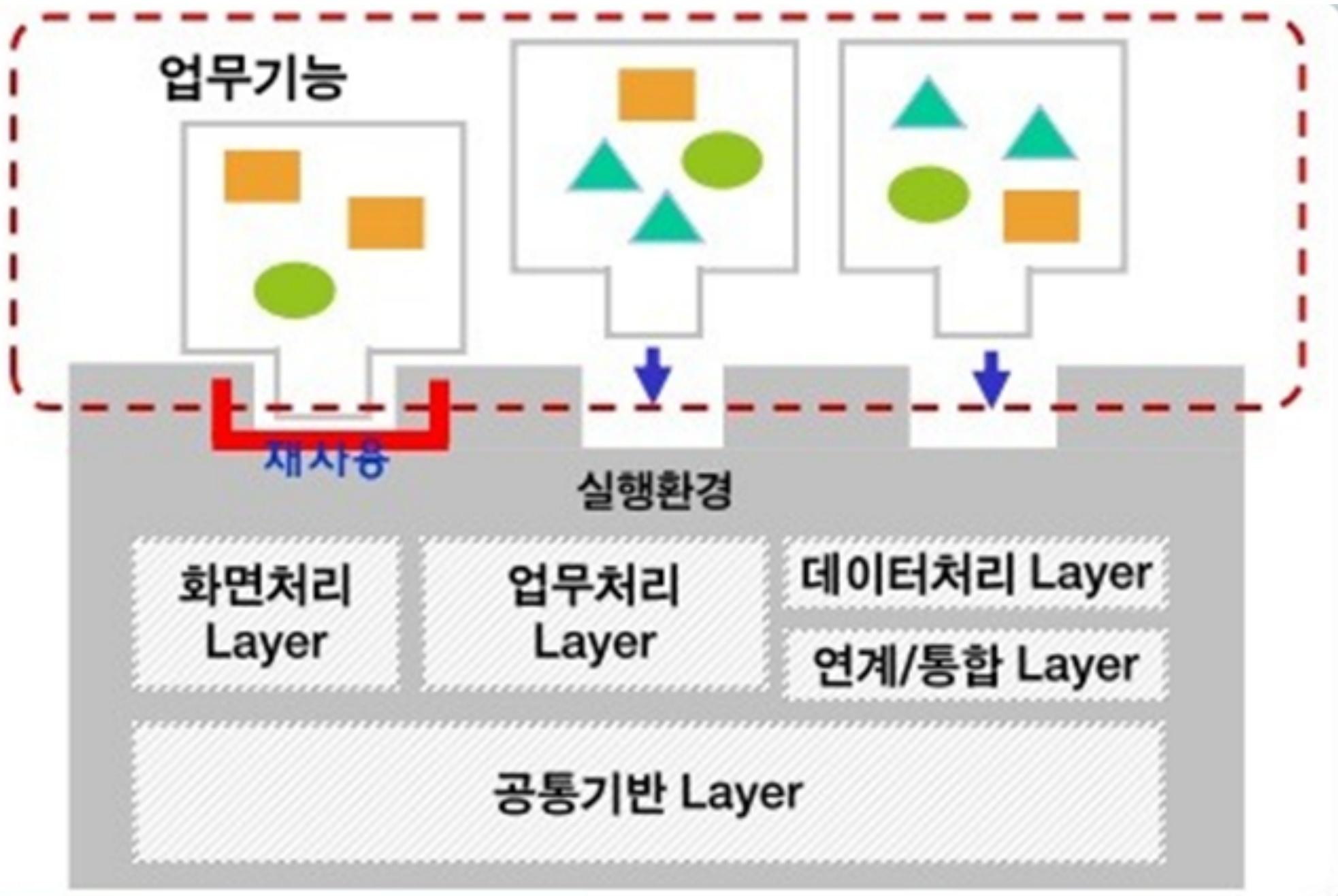
Framework ?

- 사전적 의미 : ‘복잡한 문제를 해결하거나 서술하는데 사용하는 기본 개념 구조’
- 소프트웨어에서의 의미 : 프레임워크는 Black box 형태의 재사용을 통해 적은 공수를 들이면서도 빠른 시간 내에 안정적인 어플리케이션 구성을 가능하게 하는 반제품 형태의 SW 집합을 말한다.
- 왜?
 - 전체 시스템의 통합성, 일관성을 위해
 - 프레임워크가 가이드를 제공, 설계 및 구현 강제
 - 개발자는 비즈니스 로직에 집중 할 수 있는 구조를 갖추
 - 생산성 향상, 안정성 향상
 - 분석, 예상이 가능

Framework ?



Framework ?



framework VS library

특징	프레임워크	라이브러리
유저코드의 작성	프레임워크 클래스를 서브 클래싱 해서 작성	독립적으로 작성
호출흐름	프레임워크코드가 유저코드를 호출	유저코드가 라이브러리를 호출
실행흐름	프레임워크가 제어	유저코드가 제어
객체의 연동	구조프레임워크가 정의	독자적으로 정의

〈표 1〉 프레임워크와 라이브러리의 비교

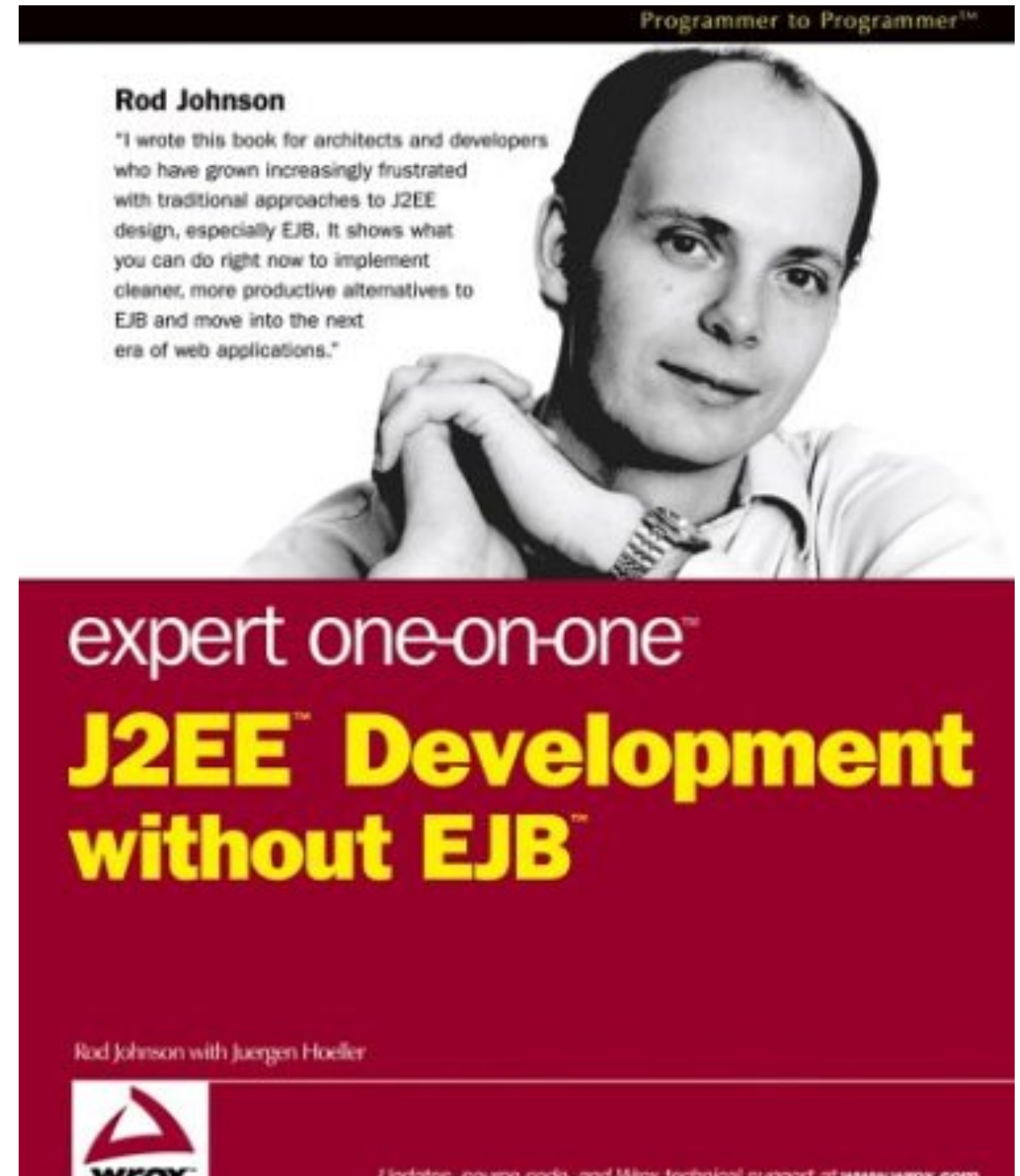
Spring framework

서비스 그룹	서비스 명	오픈소스 명	1.0 (jdk1.5이상)	3.0 (jdk1.7이상)	라이선스	비고
Presentation layer	Core	Spring Framework	2.5.6	4.1.2 3.2.9-배치	Apache 2.0	기능확장
	Ajax Support	Ajax Tags	-	1.5.7	Apache 2.0	
	Internationalization	Spring Framework	2.5.6	4.1.2 3.2.9-배치	Apache 2.0	
	Security	Commons Validator	1.3.1	1.4.0	Apache 2.0	
	UI Adaptor				Apache 2.0	자체개발
Business Logic layer	Process Control	Spring Web Flow	-	2	Apache 2.0	3.0부터 개별적용
	Exception Handling	Spring Framework	2.5.6	4.1.2 3.2.9-배치	Apache 2.0	기능확장
Persistent layer	DataSource	Spring Framework	2.5.6	4.1.2 3.2.9-배치	Apache 2.0	
	Data Access	iBatis SQL Maps	2.3.4	2.3.4	Apache 2.0	기능확장
		MyBatis	-	3.3.0 3.2.7-배치	Apache 2.0	기능확장
		Hibernate	-	4.3.11	LGPL 2.1	

전자정부 프레임워크는 스프링 프레임워크 기반에 모듈을 추가/수정한것

스프링의 역사

- 로드존슨이 2002년 출간한 저서 Expert One-on-One J2EE Design and Development에 선보인 코드가 시초.
- 2003년 6월 최초 아파치 2.0 라이선스로 공개.
- 2004년 3월에 1.0 버전이 릴리즈
- 현재 4.3.8.RELEASE버전 사용중
- 곧 5.x 정식 릴리즈 예정 - 스냅샷버전 제공중

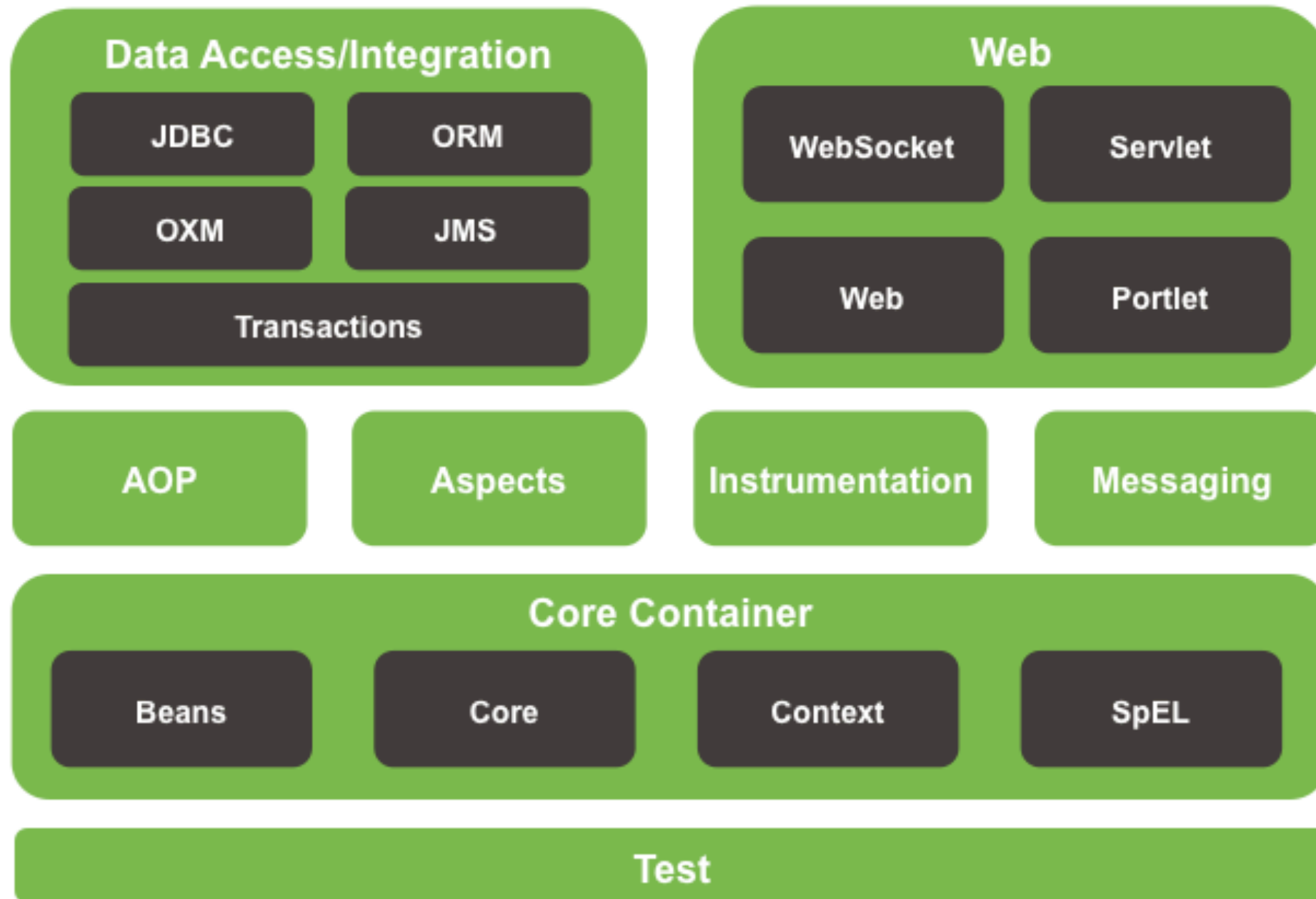


SPRING FRAMEWORK

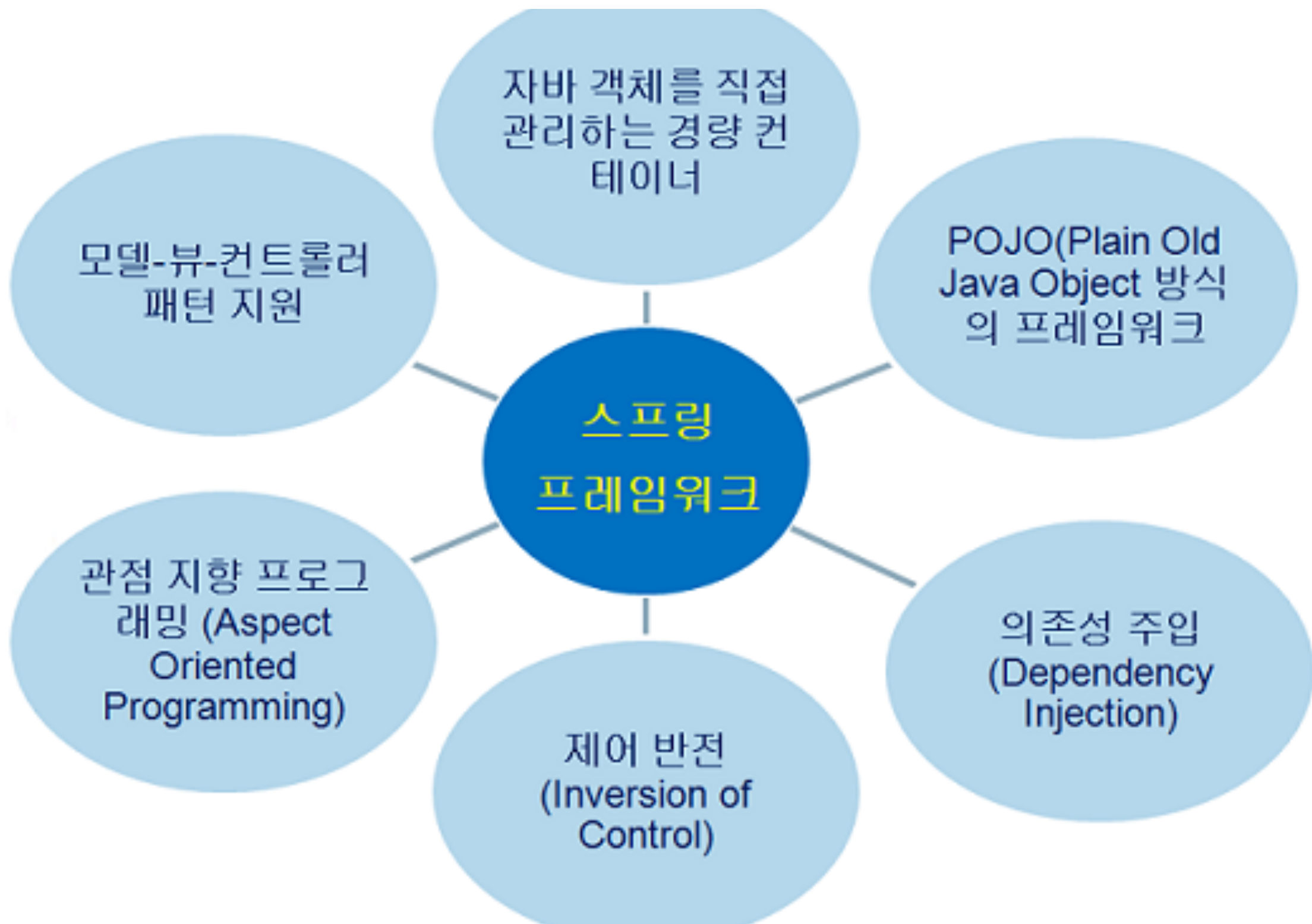
- 자바(JAVA) 플랫폼을 위한 오픈소스(Open Source) 애플리케이션 프레임워크(Framework)
- 자바 엔터프라이즈 개발을 편하게 해주는 경량급 애플리케이션 프레임워크
- 자바 개발을 위한 프레임워크로 종속 객체를 생성해주고, 조립해주는 도구
- 자바로 된 프레임워크로 자바SE로 된 자바 객체(POJO)를 자바EE에 의존적이지 않게 연결해주는 역할.



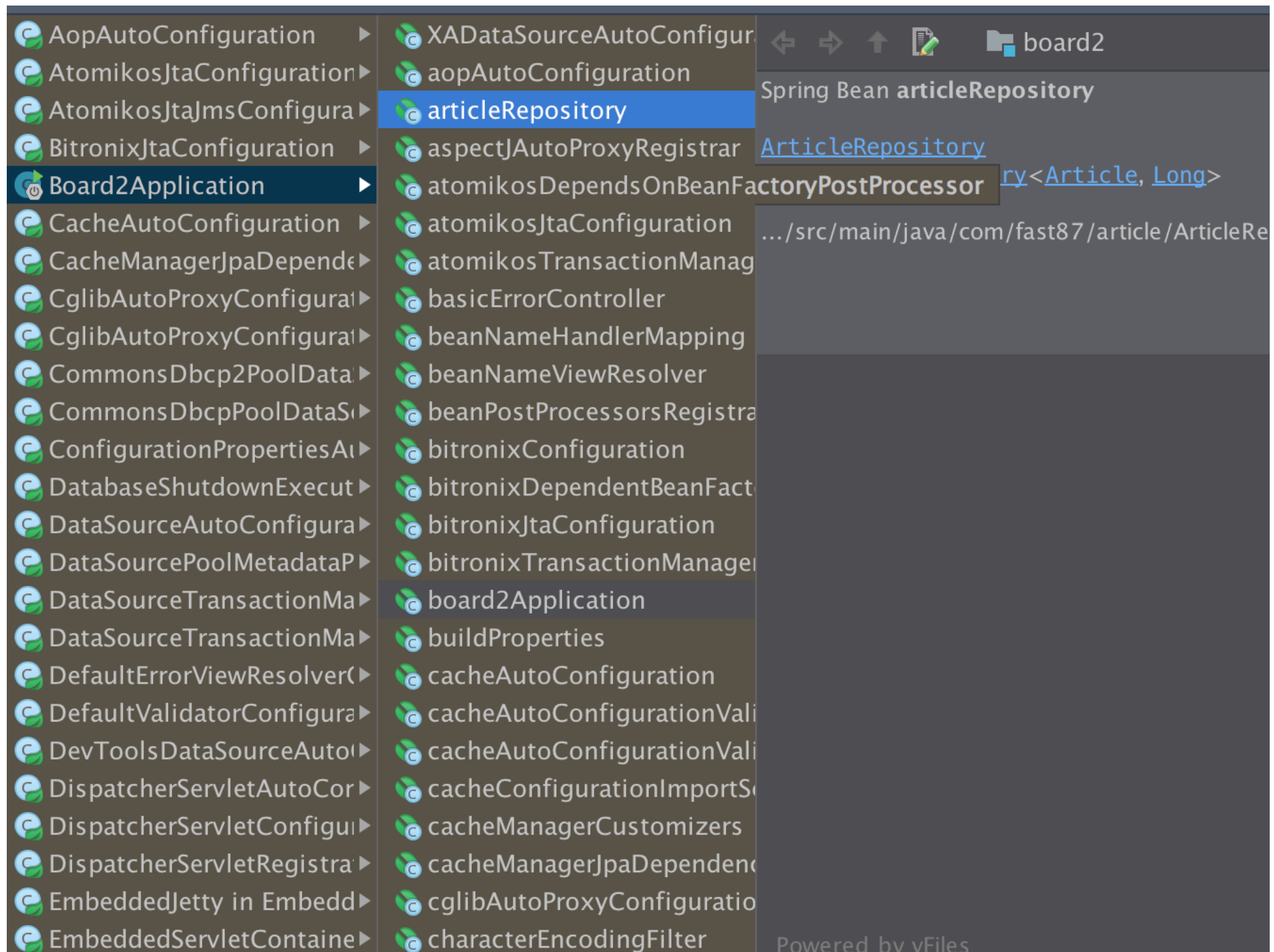
Spring Framework Runtime



스프링 프레임워크 구조



스프링 프레임워크의 특징



“java 객체의 생성, 소멸과 같은 라이프 사이클을 관리하는 경량 컨테이너”

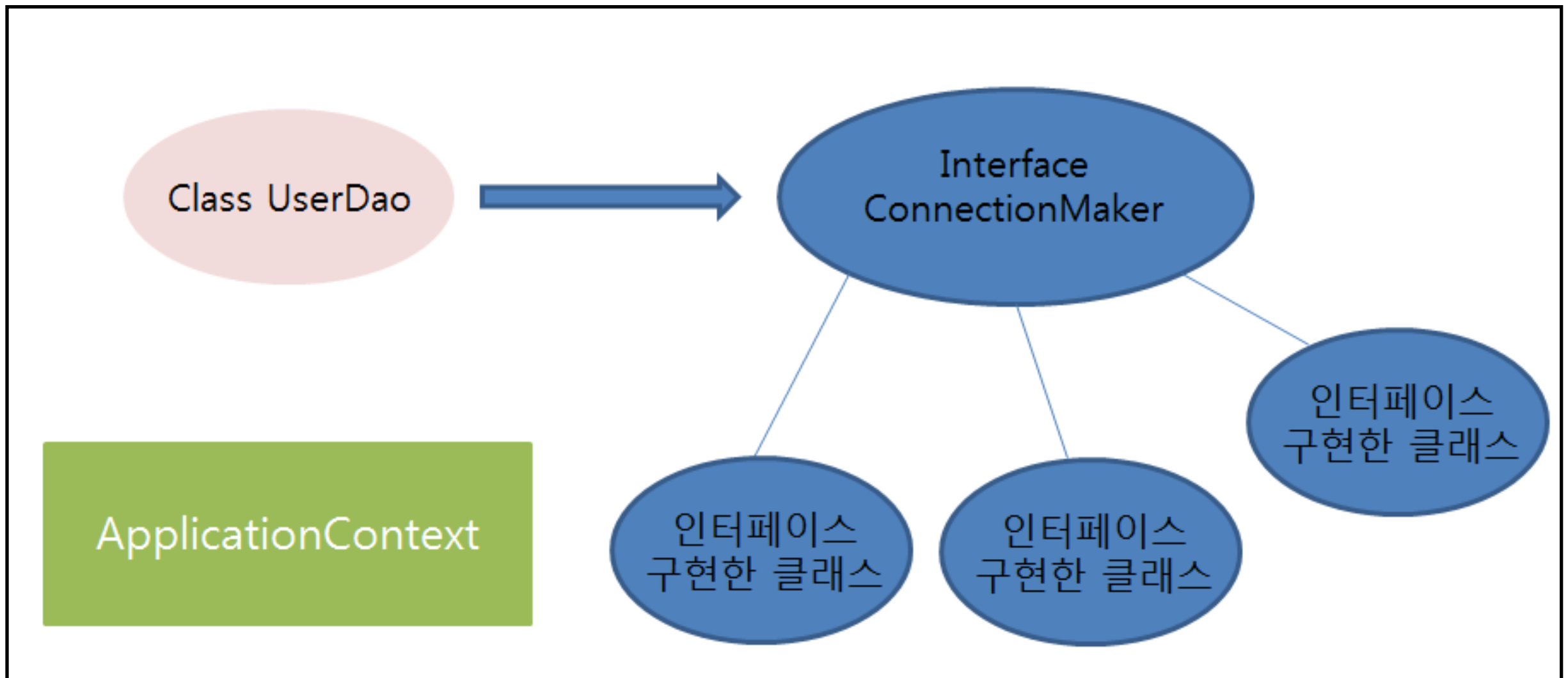
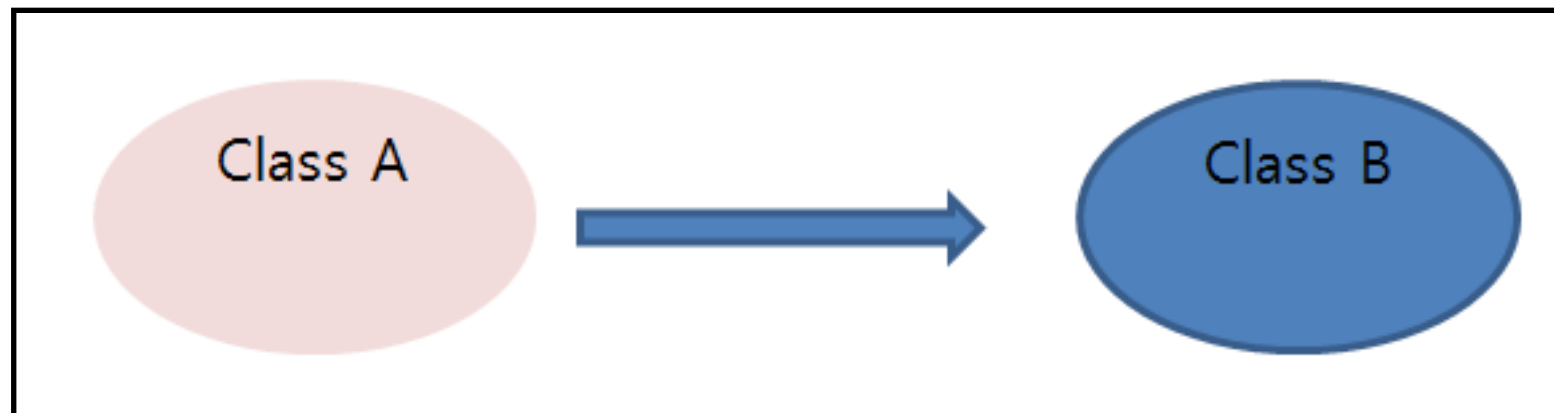
POJO

- Plain Old Java Object
- 자바개발자가 마음대로 정의 할수 있는 객체
- Object를 간단히 슈퍼클래스로 둔 보통의 평범의 그자체 순수 자바클래스

POJO

- 클래스 상속을 강제하지 않는다.
- 인터페이스 구현을 강제하지 않는다.
- 어노테이션 사용을 강제하지 않는다.

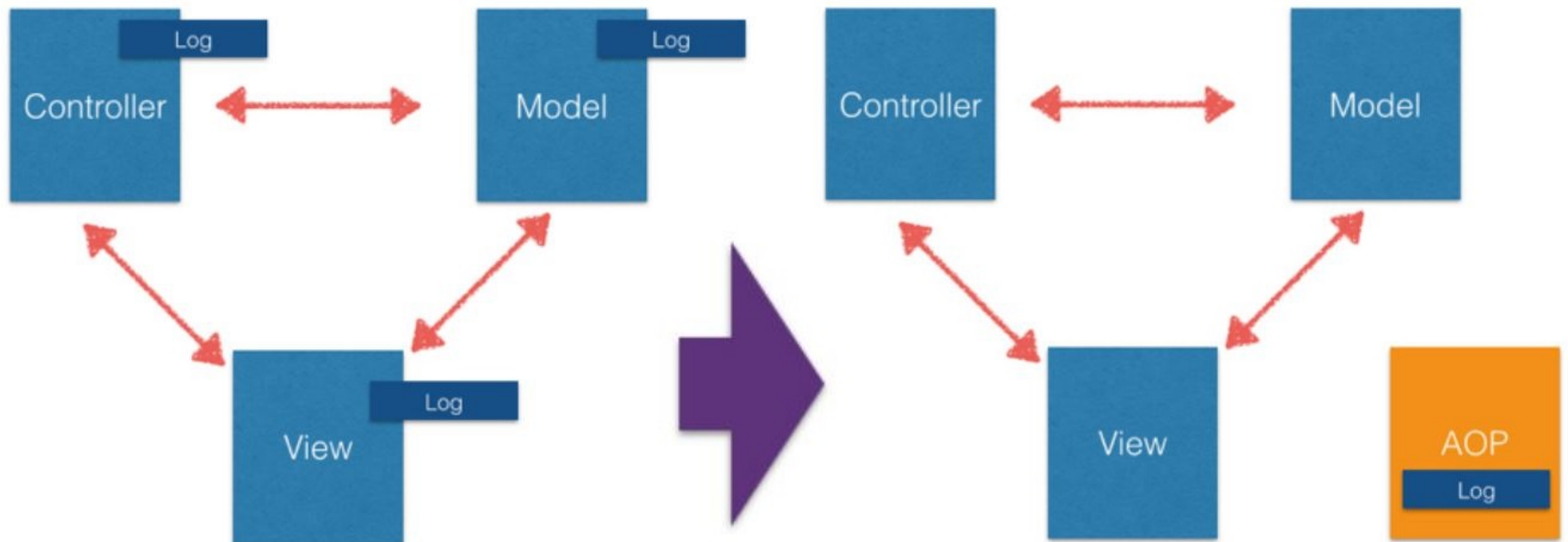
DI/IoC



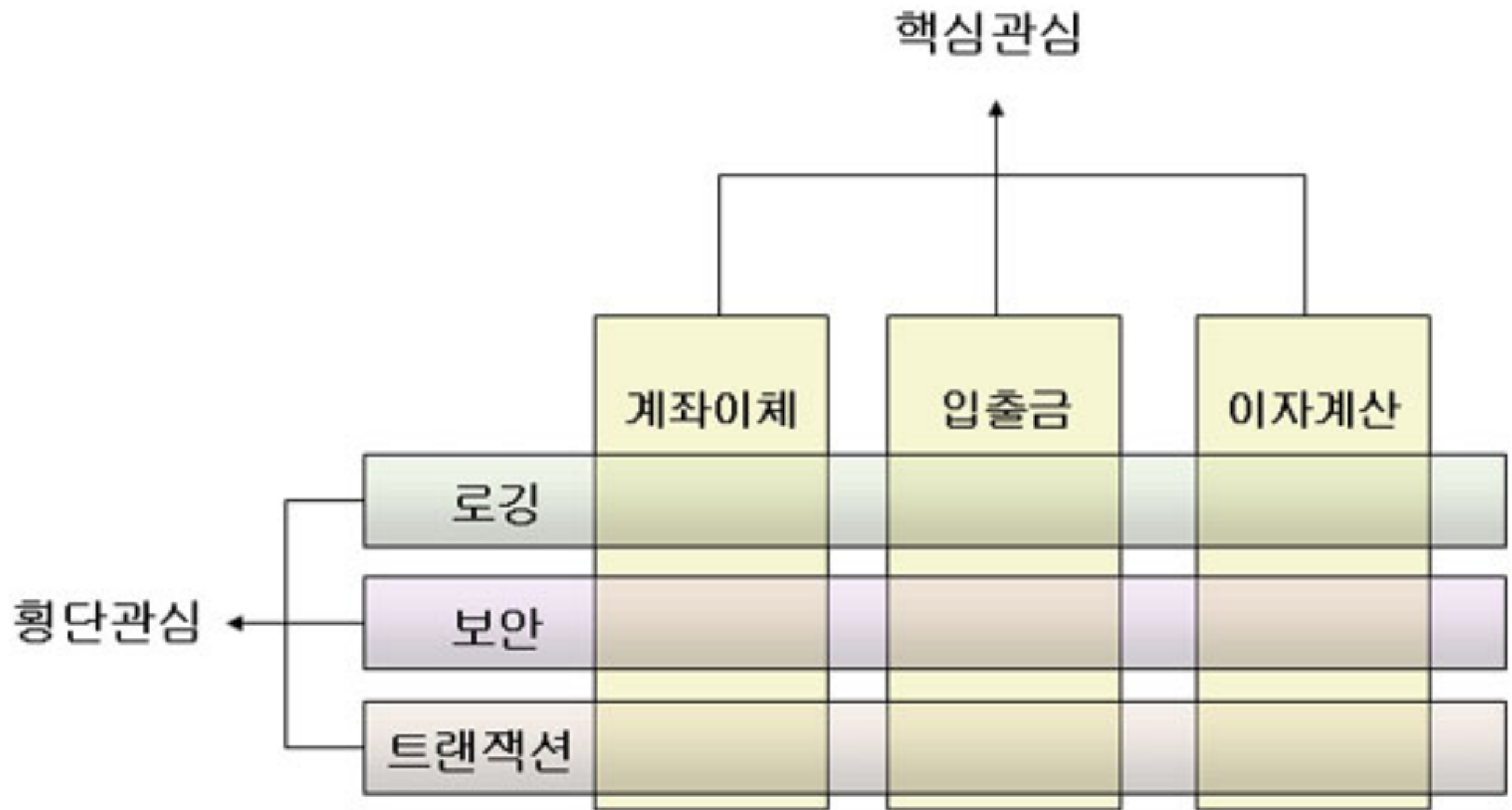
AOP

- Aspect Oriented Programming
- 관점 지향 프로그래밍 을 의미한다.
- 횡단 관심사의 분리

AOP



AOP



정리1

- 경량 컨테이너로서 자바 객체를 직접 관리.
 - 각각의 객체 생성, 소멸과 같은 라이프 사이클을 관리하며 스프링으로부터 필요한 객체를 얻어올 수 있다.
- 스프링은 POJO(Plain Old Java Object) 방식의 프레임워크.
 - 일반적인 J2EE 프레임워크에 비해 구현을 위해 특정한 인터페이스를 구현하거나 상속을 받을 필요가 없어 기존에 존재하는 라이브러리 등을 지원하기에 용이하고 객체가 가볍다.
- 스프링은 제어 반전(IOC : Inversion of Control)을 지원.
 - 컨트롤의 제어권이 사용자가 아니라 프레임워크에 있어서 필요에 따라 스프링에서 사용자의 코드를 호출한다.

정리2

- 스프링은 의존성 주입(DI : Dependency Injection)을 지원
 - 각각의 계층이나 서비스들 간에 의존성이 존재할 경우 프레임워크가 서로 연결시켜준다.
- 스프링은 관점 지향 프로그래밍(AOP : Aspect-Oriented Programming)을 지원
 - 트랜잭션이나 로깅, 보안과 같이 여러 모듈에서 공통적으로 사용하는 기능의 경우 해당 기능을 분리하여 관리할 수 있다.
- 스프링은 영속성과 관련된 다양한 서비스를 지원
 - iBatis나 Hibernate 등 이미 완성도가 높은 데이터베이스 처리 라이브러리와 연결할 수 있는 인터페이스를 제공한다.
- 스프링은 확장성이 높음.
 - 스프링 프레임워크에 통합하기 위해 간단하게 기존 라이브러리를 감싸는 정도로 스프링에서 사용이 가능하기 때문에 수많은 라이브러리가 이미 스프링에서 지원되고 있고 스프링에서 사용되는 라이브러리를 별도로 분리하기도 용이하다.

스프링을 사용하는 이유

- 많이 사용한다.
- 쉽다.(이건 뻥이다. 하지만 EJB 등 이전 시대의 개발 방식보다 쉽다.)
- 코드 품질을 보장한다.
- 유지보수성이 좋다.
- 개발의 편의성 - 협업에 유리하다.
- 오픈소스
- 자바를 기반으로 한 프레임워크 중에서 스프링만큼 잘 설계되고 확장가능한 프레임워크가 없다.

with SPRING VS *only* JAVA

With Spring Framework

- spring initializr 선택
- Freemarker, web 선택
- <https://github.com/jojonari/withspring.git>