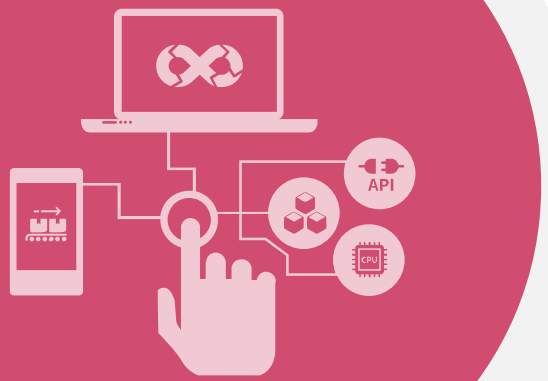
CLOUDZ LABS

Spring Boot Introduction



SK 주식회사

Table of Contents

PART 01 Spring Boot

01 Spring Framework

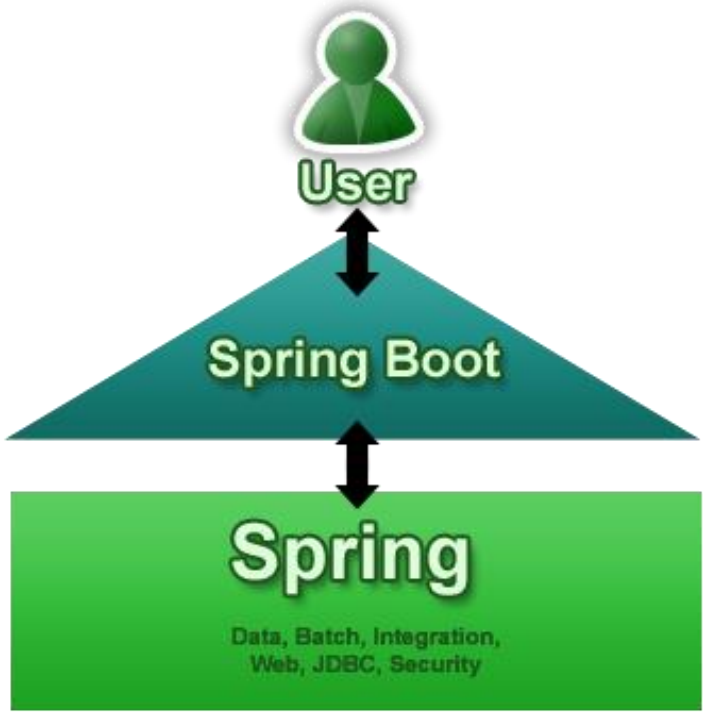
02 Spring Boot

https://spring.io/

The Spring Framework is a Java platform that provides comprehensive infrastructure support for developing Java application.

Spring handles infrastructure so you focus on **your application.**

Spring Boot is designed to
get **you** up and running as **quickly** as possible.



https://projects.spring.io/spring-boot/

Spring Boot makes it easy to create stand-alone, production-grade Spring based Application that

*you can "**just run**".*

SPRING BOOT

Takes an opinionated view of building Spring applications and gets you up and running as quickly as possible.

SPRING FRAMEWORK

Provides core support for dependency injection, transaction management, web apps, data access, messaging and more.

https://spring.io/projects

PART 02 Spring Boot Features

01 Easy to Create

02 Stand Alone

03 Production Grade

Spring Boot makes it **easy to create** stand-alone, production-grade Spring based Application that

you can “**just run**”.

project

Application

Configuration

project

Application

Configuration

project

```
<!-- Spring -->
<dependency>
        <groupId>org.springframework</groupId>
        <artifactId>spring-context</artifactId>
        <version>4.3.10.RELEASE</version>
        <exclusions>
                <exclusion>
                        <groupId>commons-logging</groupId>
                        <artifactId>commons-logging</artifactId>
                </exclusion>
        </exclusions>
</dependency>
<dependency>
        <groupId>org.springframework</groupId>
        <artifactId>spring-webmvc</artifactId>
        <version>4.3.10.RELEASE</version>
</dependency>
<dependency>
    <groupId>org.springframework.data</groupId>
    <artifactId>spring-data-jpa</artifactId>
    <version>1.11.6.RELEASE</version>
</dependency>
<dependency>
        <groupId>org.springframework</groupId>
        <artifactId>spring-aspects</artifactId>
        <version>4.3.10.RELEASE</version>
</dependency>
```

```
<!-- AspectJ -->
<dependency>
        <groupId>org.aspectj</groupId>
        <artifactId>aspectjrt</artifactId>
        <version>1.8.10</version>
</dependency>

<!-- JPA -->
<dependency>
        <groupId>org.hibernate</groupId>
        <artifactId>hibernate-core</artifactId>
        <version>5.0.12.Final</version>
        <exclusions>
                <exclusion>
                        <groupId>org.apache.geronimo.specs</groupId>
                        <artifactId>geronimo-jta_1.1_spec</artifactId>
                </exclusion>
        </exclusions>
</dependency>
<dependency>
        <groupId>org.hibernate</groupId>
        <artifactId>hibernate-entitymanager</artifactId>
        <version>5.0.12.Final</version>
        <exclusions>
                <exclusion>
                        <groupId>org.apache.geronimo.specs</groupId>
        <!-- dbms, logging, test, servlet & etc… -->
```

project

```
<!-- Spring -->
<dependency>
        <groupId>org.springframework</groupId>
        <artifactId>spring-context</artifactId>
        <version>4.3.10.RELEASE</version>
        <exclusions>
                <exclusion>
                        <groupId>commons-logging</groupId>
                        <artifactId>commons-logging</artifactId>
                </exclusion>
        </exclusions>
</dependency>
<dependency>
        <groupId>org.springframework</groupId>
        <artifactId>spring-webmvc</artifactId>
        <version>4.3.10.RELEASE</version>
</dependency>
<dependency>
    <groupId>org.springframework.data</groupId>
    <artifactId>spring-data-jpa</artifactId>
    <version>1.11.6.RELEASE</version>
</dependency>
<dependency>
        <groupId>org.springframework</groupId>
        <artifactId>spring-aspects</artifactId>
        <version>4.3.10.RELEASE</version>
</dependency>
```

```
<!-- AspectJ -->
<dependency>
        <groupId>org.aspectj</groupId>
        <artifactId>aspectjrt</artifactId>
        <version>1.8.10</version>
</dependency>

<!-- JPA -->
<dependency>
        <groupId>org.hibernate</groupId>
        <artifactId>hibernate-core</artifactId>
        <version>5.0.12.Final</version>
        <exclusions>
                <exclusion>
                        <groupId>org.apache.geronimo.specs</groupId>
                        <artifactId>geronimo-jta_1.1_spec</artifactId>
                </exclusion>
        </exclusions>
</dependency>
<dependency>
        <groupId>org.hibernate</groupId>
        <artifactId>hibernate-entitymanager</artifactId>
        <version>5.0.12.Final</version>
        <exclusions>
                <exclusion>
                        <groupId>org.apache.geronimo.specs</groupId>
        <!-- dbms, logging, test, servlet & etc… -->
```

spring-boot-starter-project

```
<parent>
        <groupId>org.springframework.boot</groupId>
        <artifactId>spring-boot-starter-parent</artifactId>
        <version>1.5.6.RELEASE</version>
        <relativePath/> <!-- lookup parent from repository -->
</parent>
<dependencies>
        <dependency>
                <groupId>org.springframework.boot</groupId>
                <artifactId>spring-boot-starter-data-jpa</artifactId>
        </dependency>
        <dependency>
                <groupId>org.springframework.boot</groupId>
                <artifactId>spring-boot-starter-web</artifactId>
        </dependency>
        <dependency>
                <groupId>mysql</groupId>
                <artifactId>mysql-connector-java</artifactId>
                <scope>runtime</scope>
        </dependency>
        <dependency>
                <groupId>org.springframework.boot</groupId>
                <artifactId>spring-boot-starter-test</artifactId>
                <scope>test</scope>
        </dependency>
</dependencies>
```

spring-boot-starter-project

```
<parent>
        <groupId>org.springframework.boot</groupId>
        <artifactId>spring-boot-starter-parent</artifactId>
        <version>1.5.6.RELEASE</version>
        <relativePath/> <!-- lookup parent from repository -->
</parent>
<dependencies>
        <dependency>
                <groupId>org.springframework.boot</groupId>
                <artifactId>spring-boot-starter-data-jpa</artifactId>
        </dependency>
        <dependency>
                <groupId>org.springframework.boot</groupId>
                <artifactId>spring-boot-starter-web</artifactId>
        </dependency>
        <dependency>
                <groupId>mysql</groupId>
                <artifactId>mysql-connector-java</artifactId>
                <scope>runtime</scope>
        </dependency>
        <dependency>
                <groupId>org.springframework.boot</groupId>
                <artifactId>spring-boot-starter-test</artifactId>
                <scope>test</scope>
        </dependency>
</dependencies>
```

spring-boot-starter-project

Application

Configuration

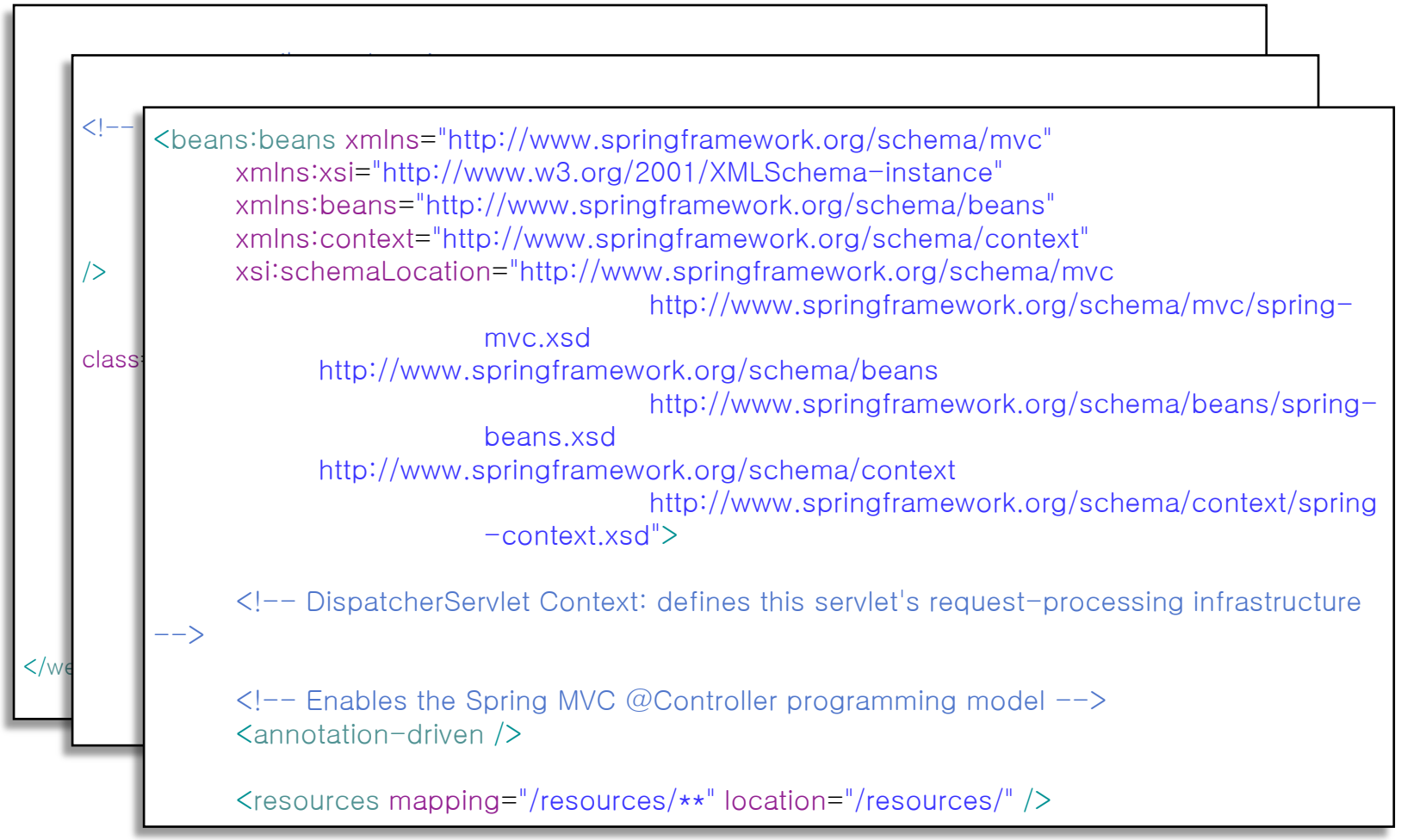
web.xml

```
                <!-- next… —>

        <filter-mapping>
                <filter-name>EncodingFilter</filter-name>
                <url-pattern>/*</url-pattern>
        </filter-mapping>

        <!-- Processes application requests -->
        <servlet>
                <servlet-name>appServlet</servlet-name>
                <servlet-class>org.springframework.web.servlet.DispatcherServlet</servlet-class>
                <init-param>
                        <param-name>contextConfigLocation</param-name>
                        <param-value>/WEB-INF/spring/appServlet/servlet-context.xml</param-value>
                </init-param>
                <load-on-startup>1</load-on-startup>
        </servlet>

        <servlet-mapping>
                <servlet-name>appServlet</servlet-name>
                <url-pattern>/</url-pattern>
        </servlet-mapping>
</web-app>
```

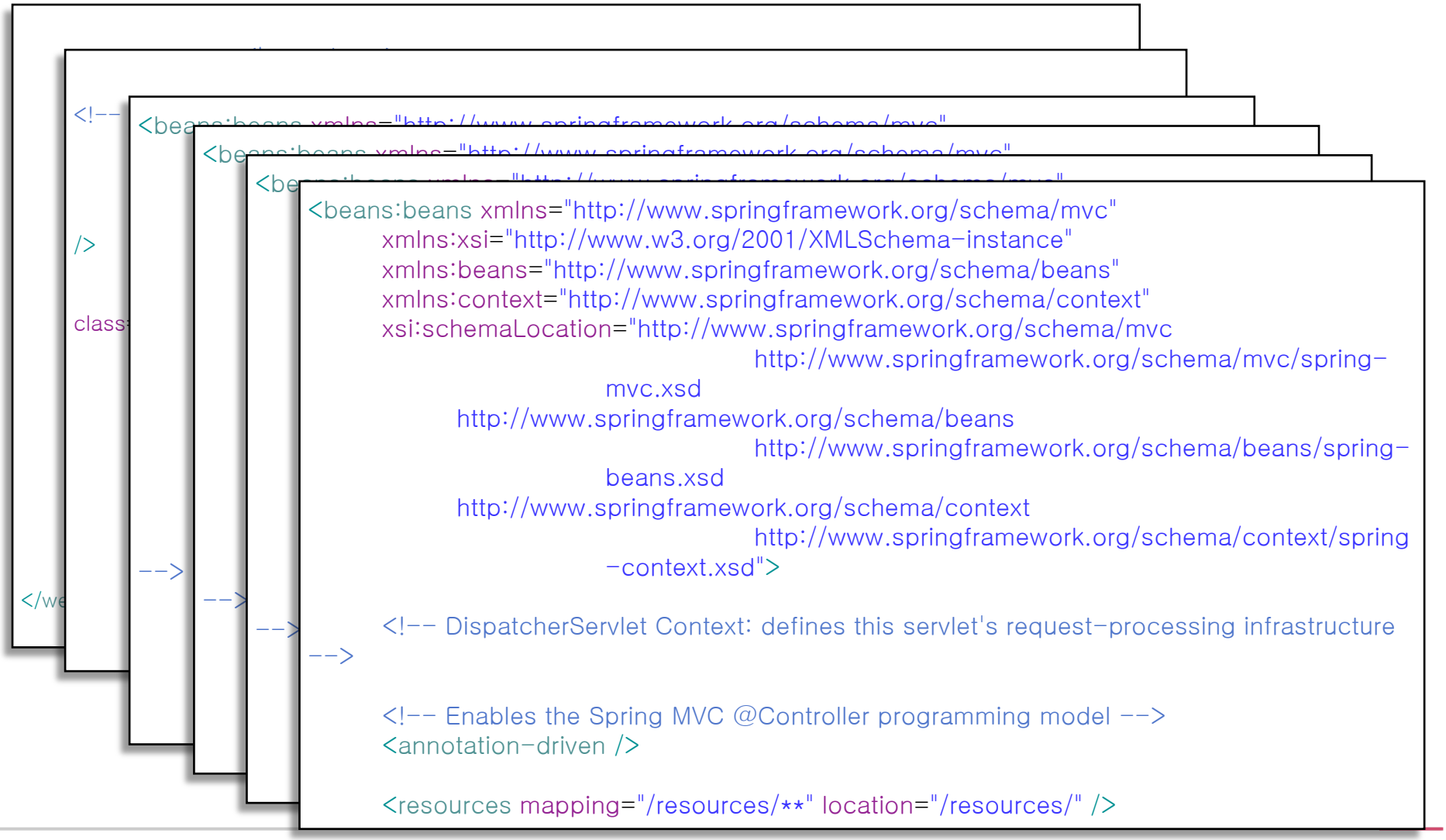
root-context.xml

```
<!-- JPA settings -->
        <bean id="transactionManager" class="org.springframework.orm.jpa.JpaTransactionManager">
                <property name="entityManagerFactory" ref="entityManagerFactory" />
        </bean>
        <bean class="org.springframework.dao.annotation.PersistenceExceptionTranslationPostProcessor"
/>

        <bean id="entityManagerFactory"
class="org.springframework.orm.jpa.LocalContainerEntityManagerFactoryBean">
                <property name="dataSource" ref="dataSource" />
                <property name="packagesToScan" value="com.sample.spring" />
                <property name="jpaVendorAdapter">
                        <bean class="org.springframework.orm.jpa.vendor.HibernateJpaVendorAdapter" />
                </property>
                <property name="jpaProperties">
                        <props>
                                <prop key="hibernate.hbm2ddl.auto">update</prop>
                        </props>
                </property>
        </bean>
```

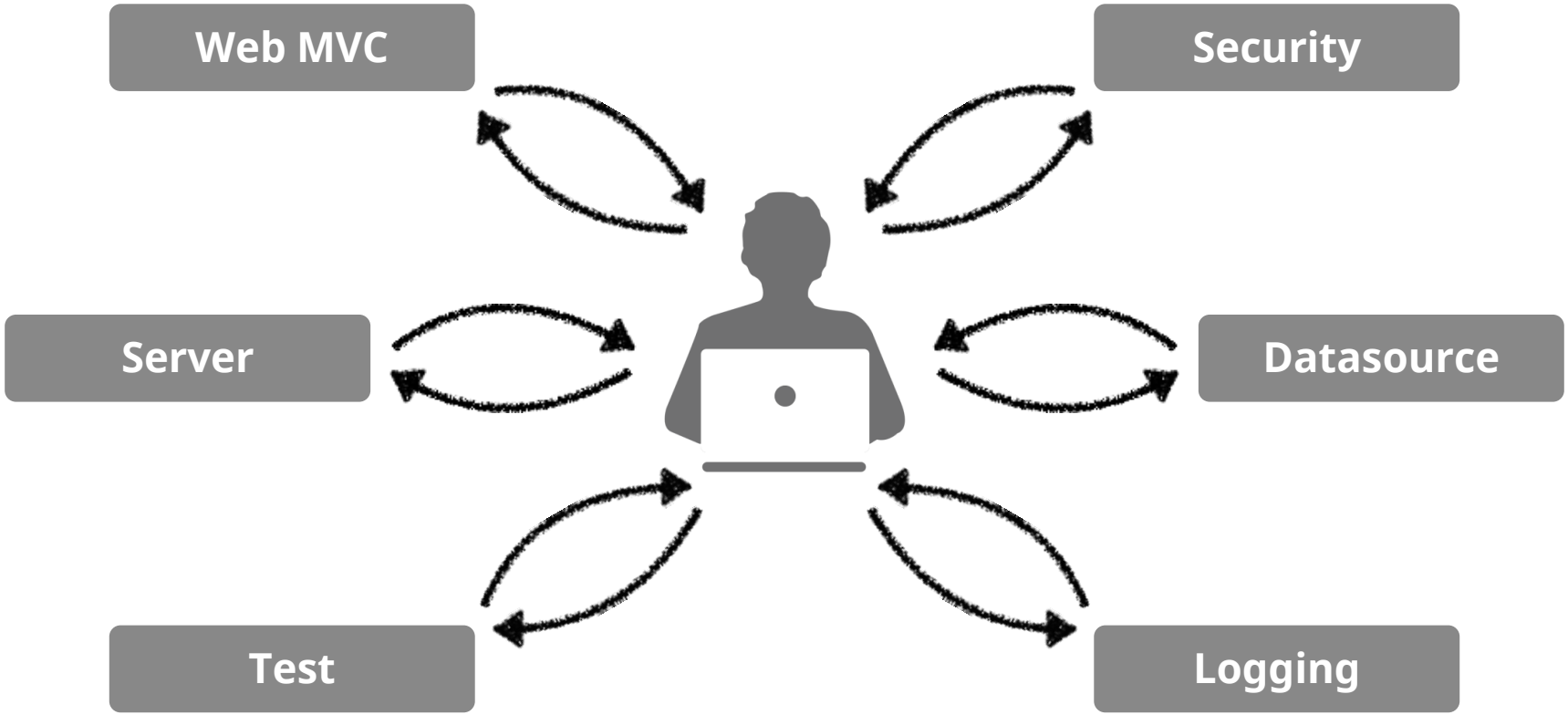
servlet-context.xml



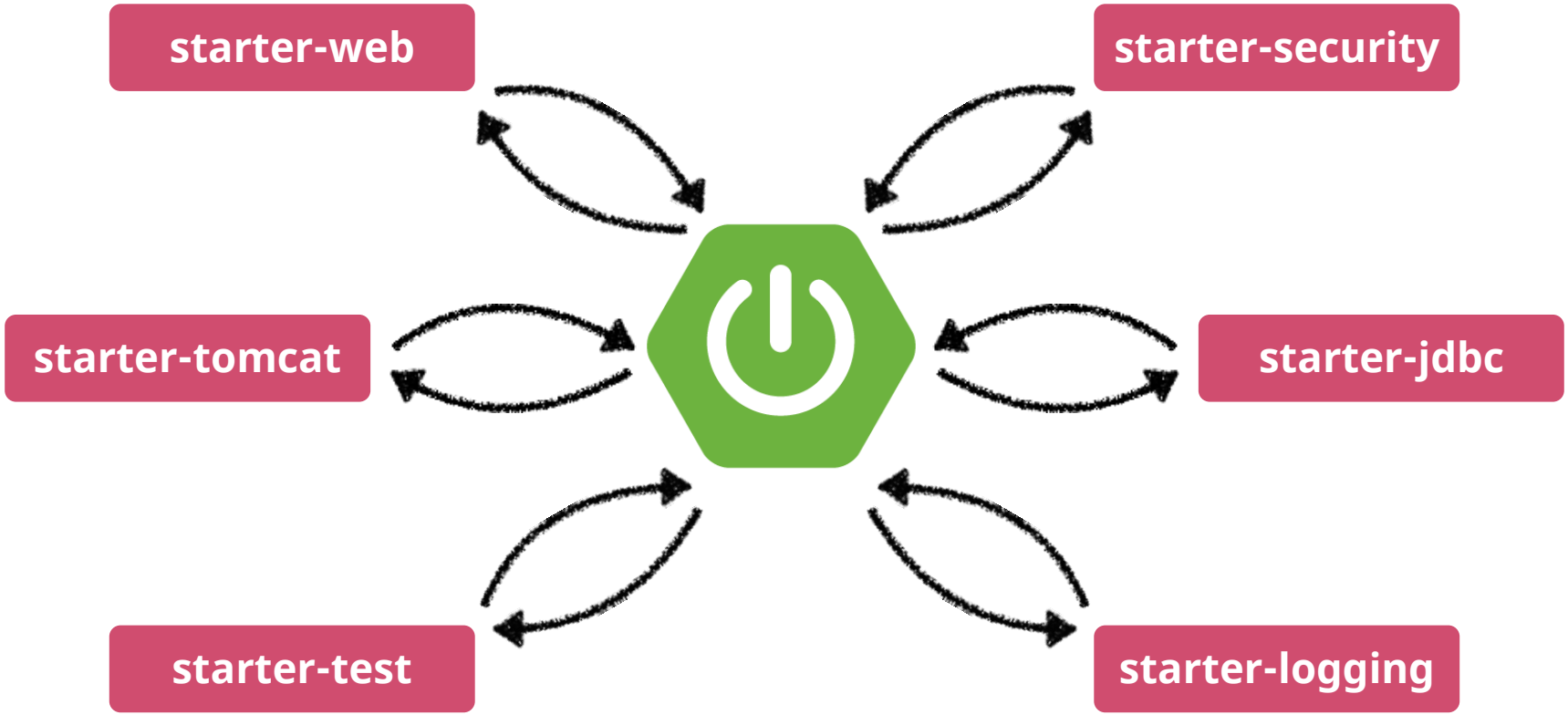
xml....



Application



SpringBootApplication



@SpringBootApplication

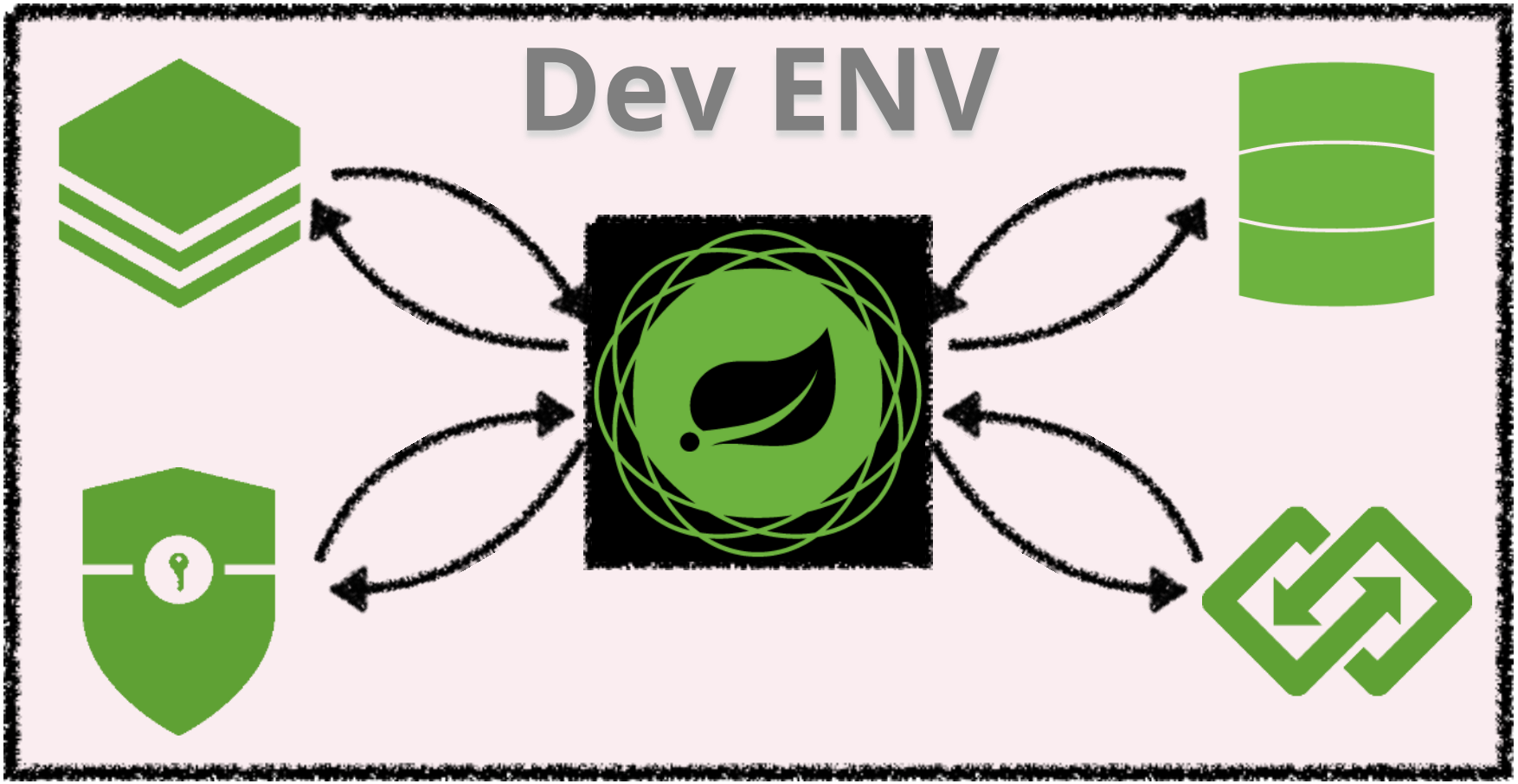
```
package com.sample.boot;

import org.springframework.boot.SpringApplication;
import org.springframework.boot.autoconfigure.SpringBootApplication;

@SpringBootApplication
public class SpringBootSampleApplication {

	public static void main(String[] args) {
		SpringApplication.run(SpringBootSampleApplication.class, args);
	}

}
```

spring-boot-starter-project

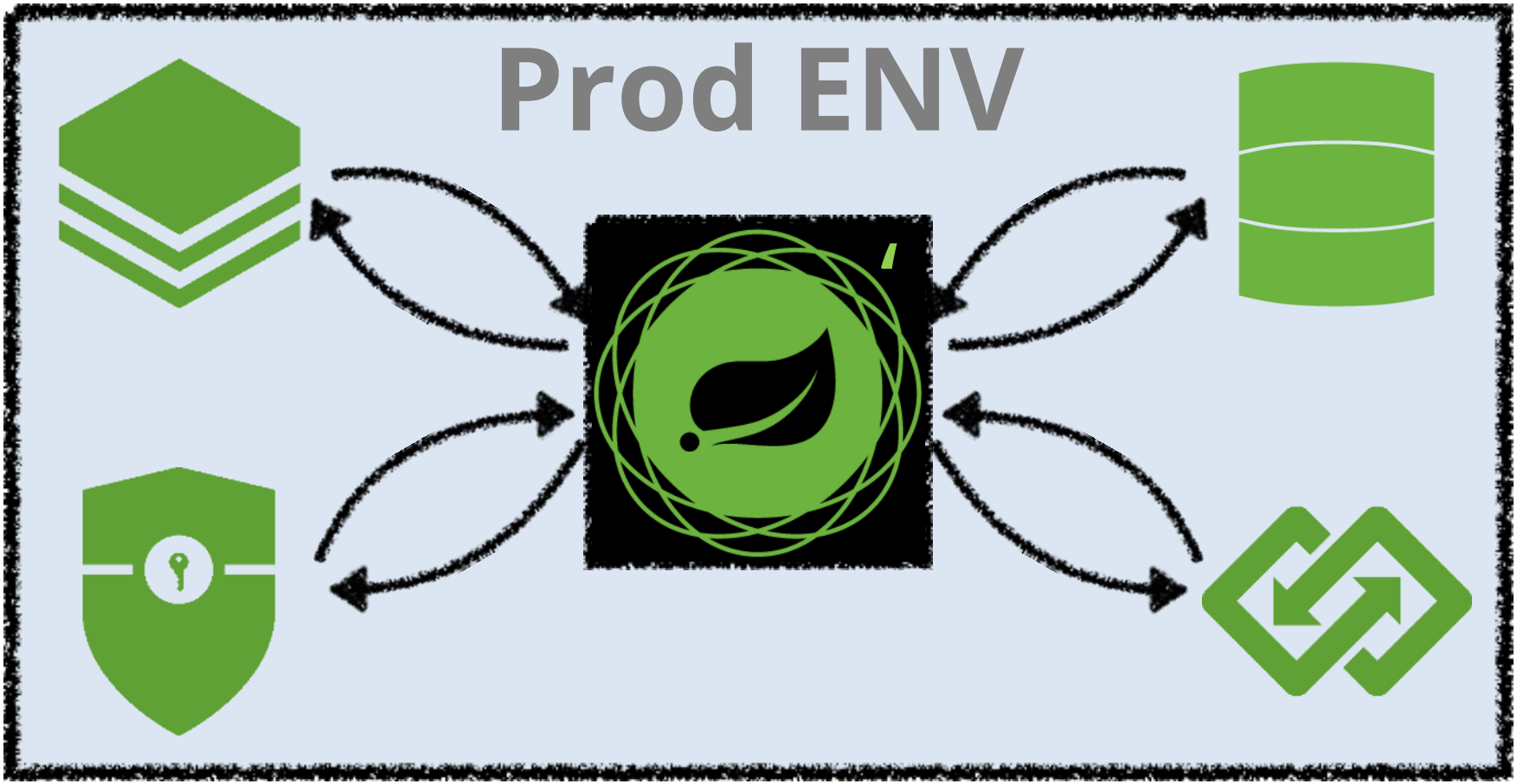
@SpringBootApplication

Configuration

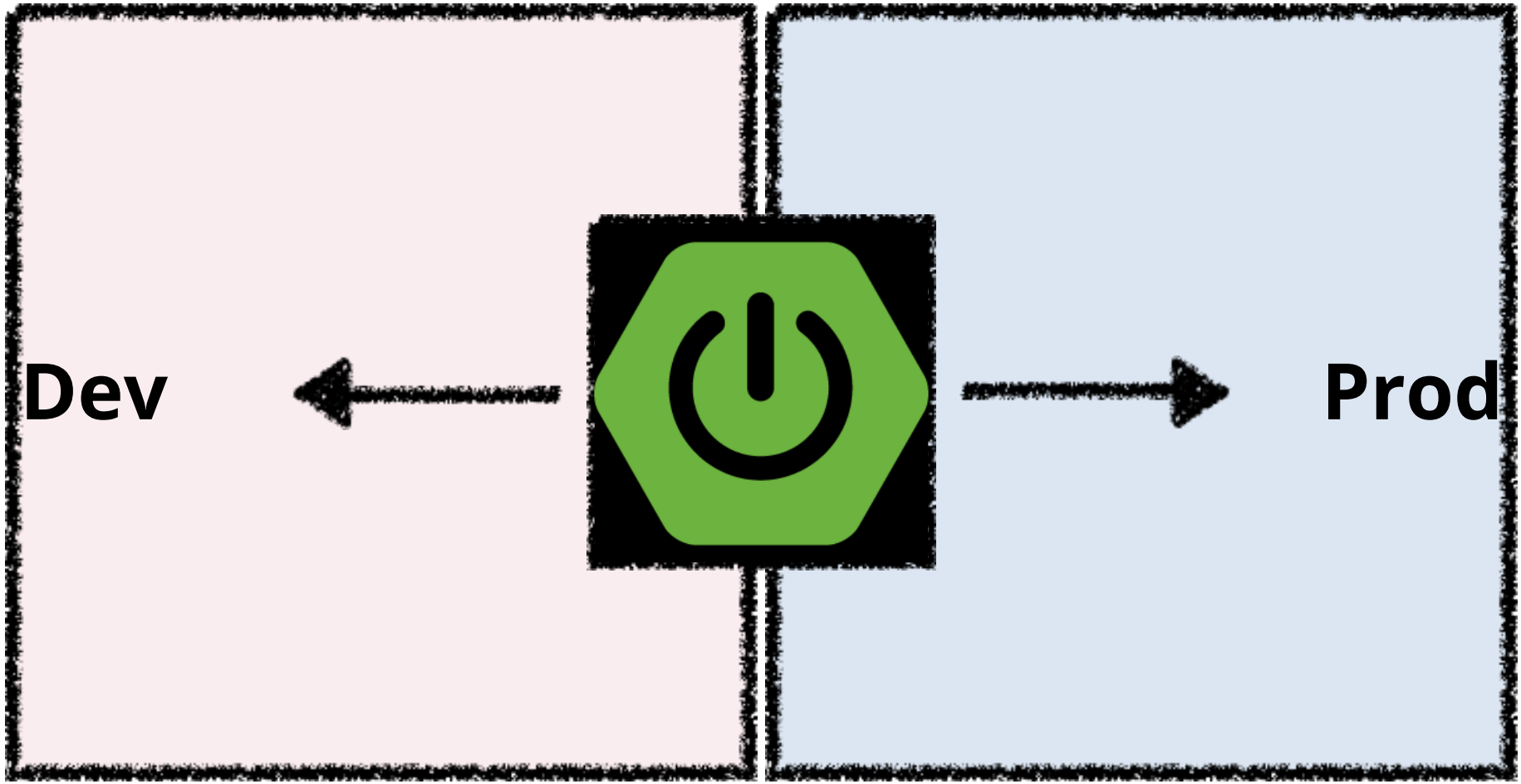
Configuration



Configuration



application-{profile}.yml



application-{profile}.yml

```yaml
# Common
spring:
  jpa:
    hibernate:
      ddl-auto: update
---
# Localhost
spring:
  profiles: default
  datasource:
    url: jdbc:mysql://localhost:3306/testdb
    username: root
    password:
    driver-class-name: com.mysql.jdbc.Driver
---
# Production
spring:
  profiles: prod
  datasource:
    url: jdbc:mysql://${DB_SERVICE_HOST}:${DB_SERVICE_PORT}/${MYSQL_DATABASE}
    username: ${MYSQL_USER}
    password: ${MYSQL_PASSWORD}
    driver-class-name: com.mysql.jdbc.Driver
```

spring-boot-starter-project

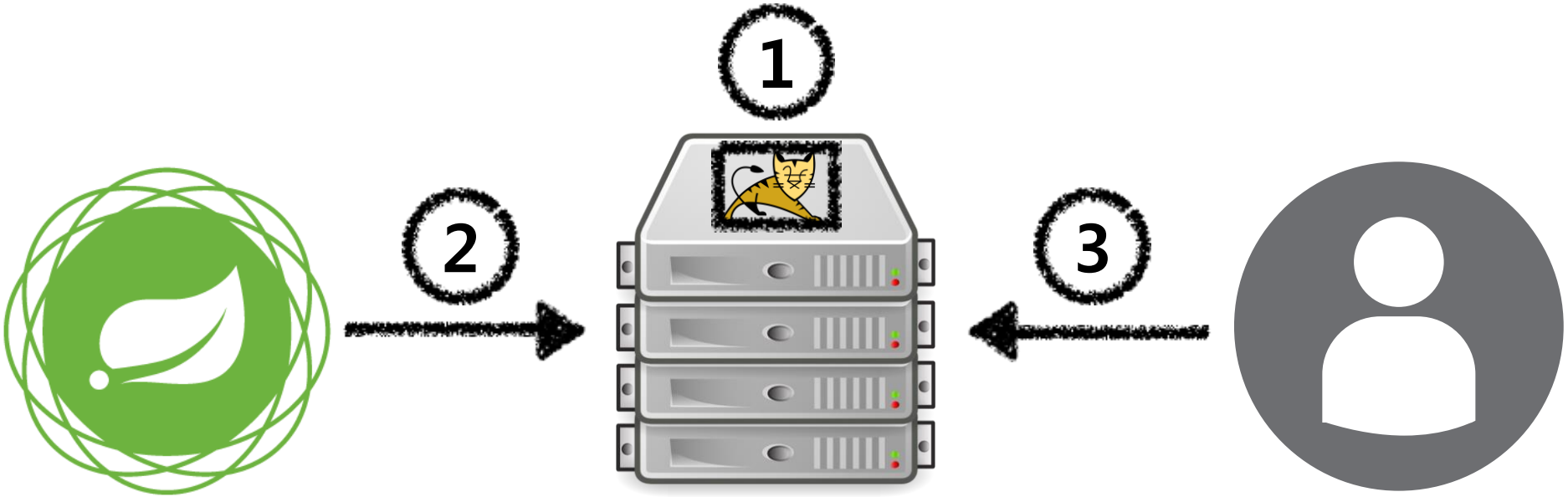
@SpringBootApplication

application-{profile}.yml

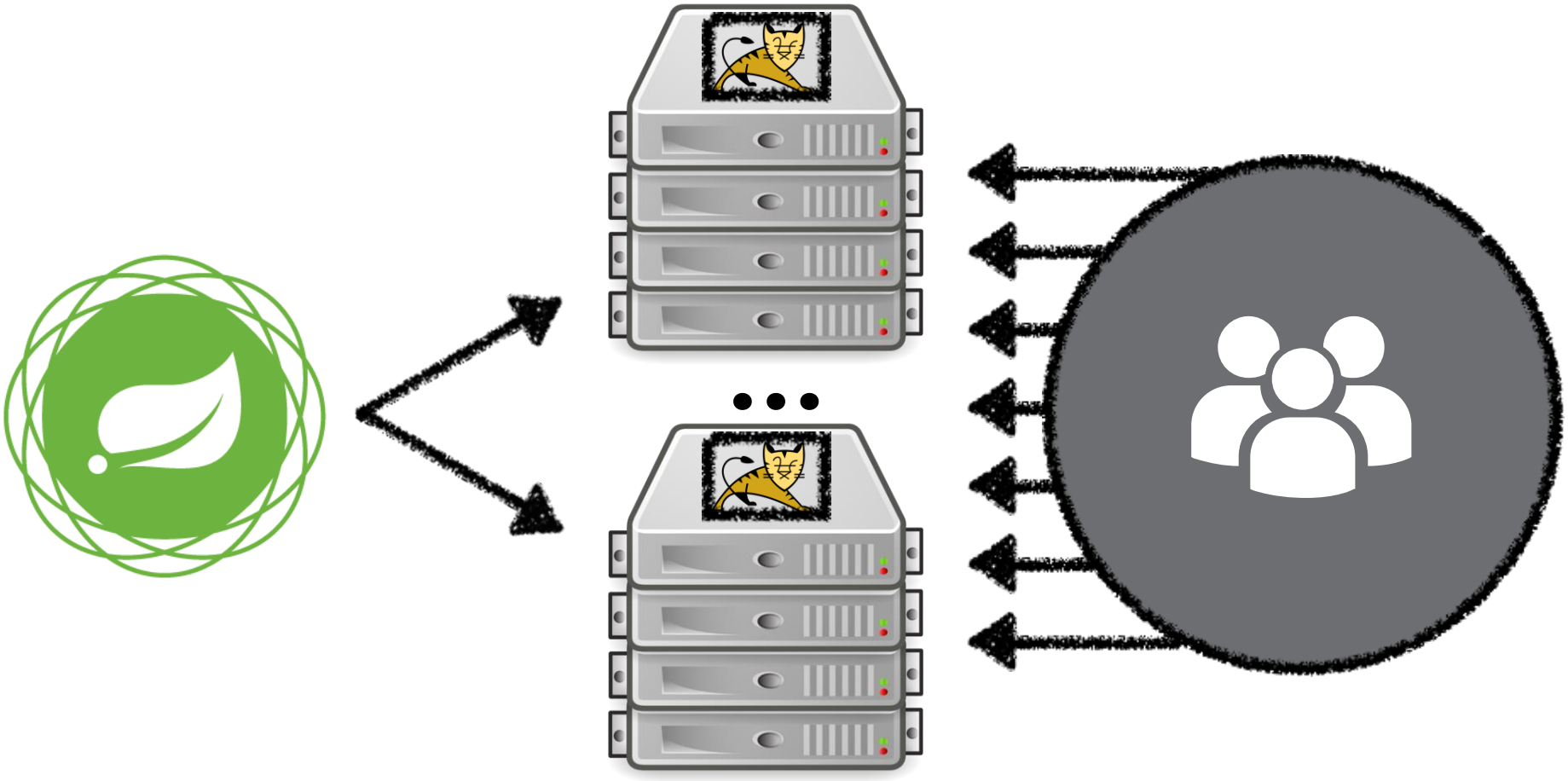
Spring Boot makes it easy to create **stand-alone**, production-grade Spring based Application that

you can “**just run**”.

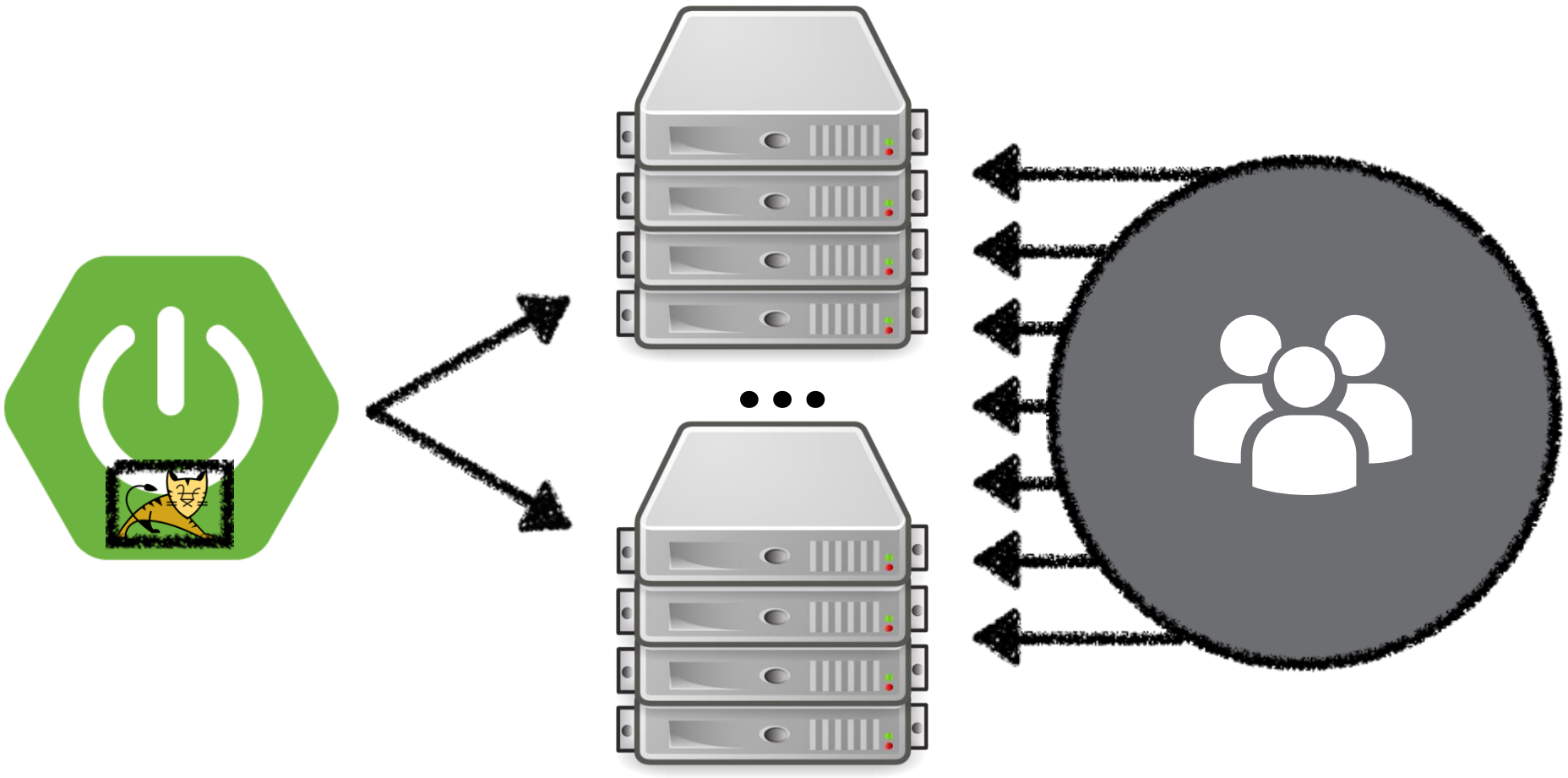
WAS Setting



n times WAS Setting



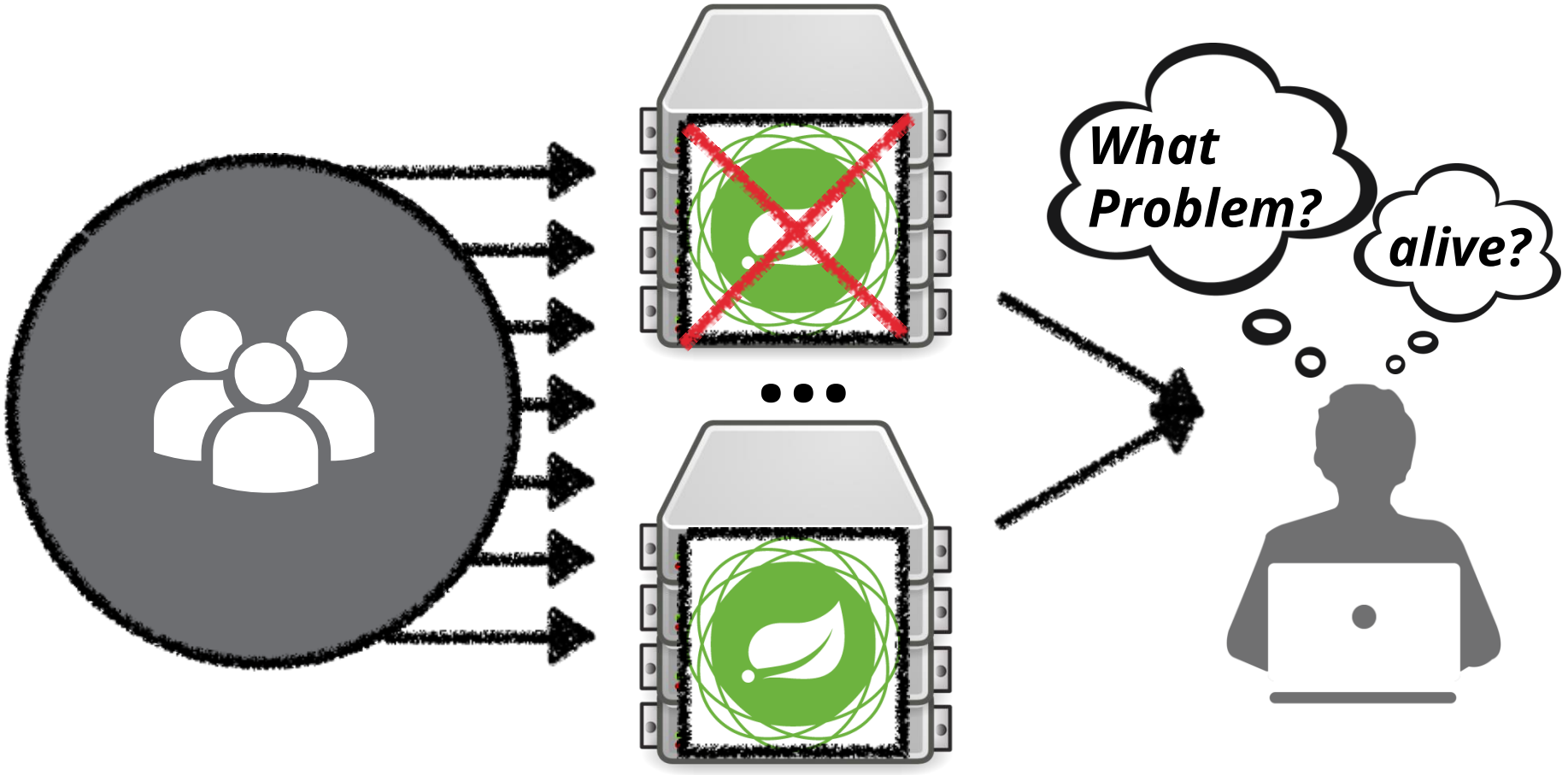
1 times WAS Setting



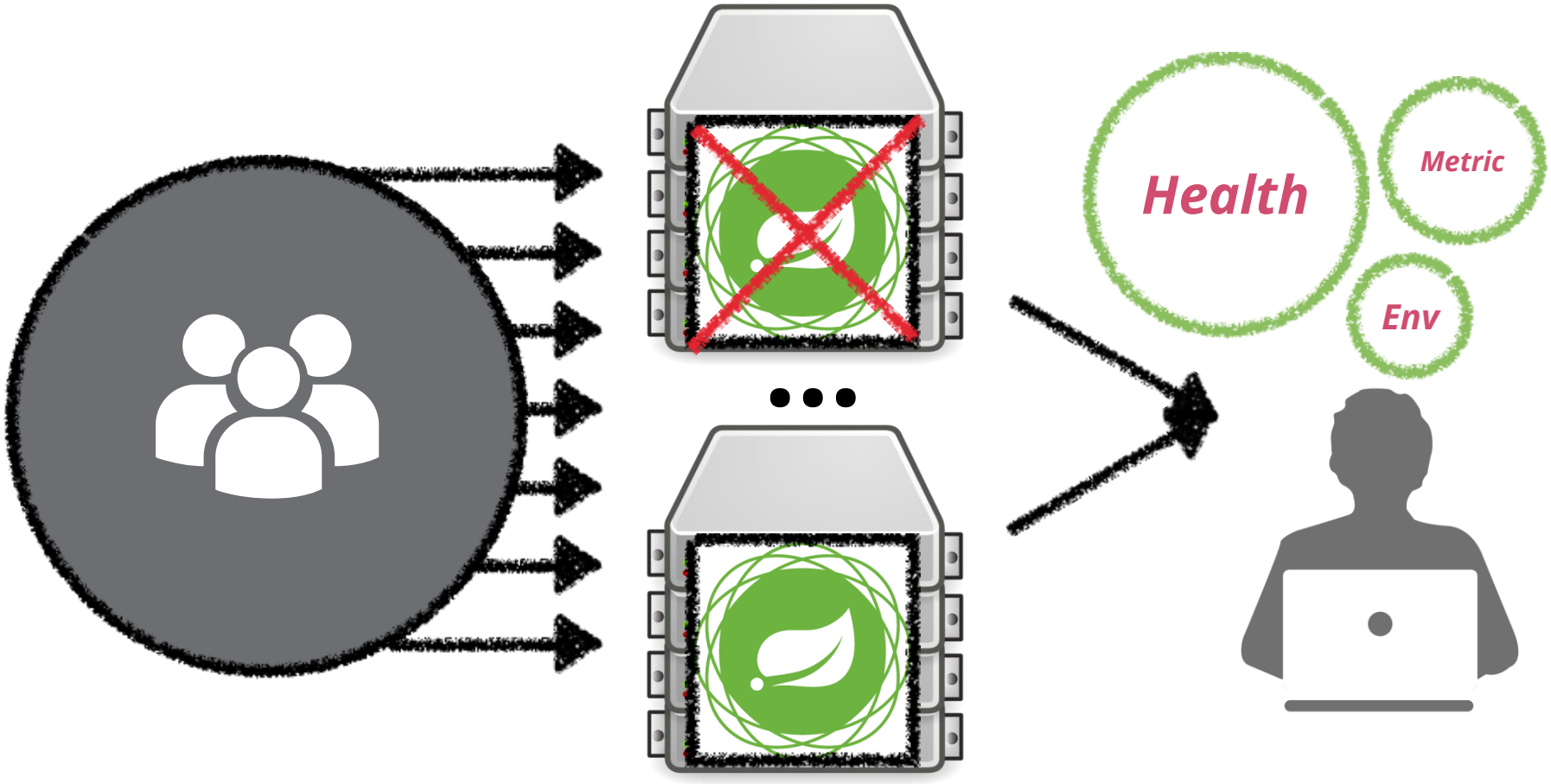
Spring Boot makes it easy to create stand-alone, **production-grade** Spring based Application that

you can “**just run**”.

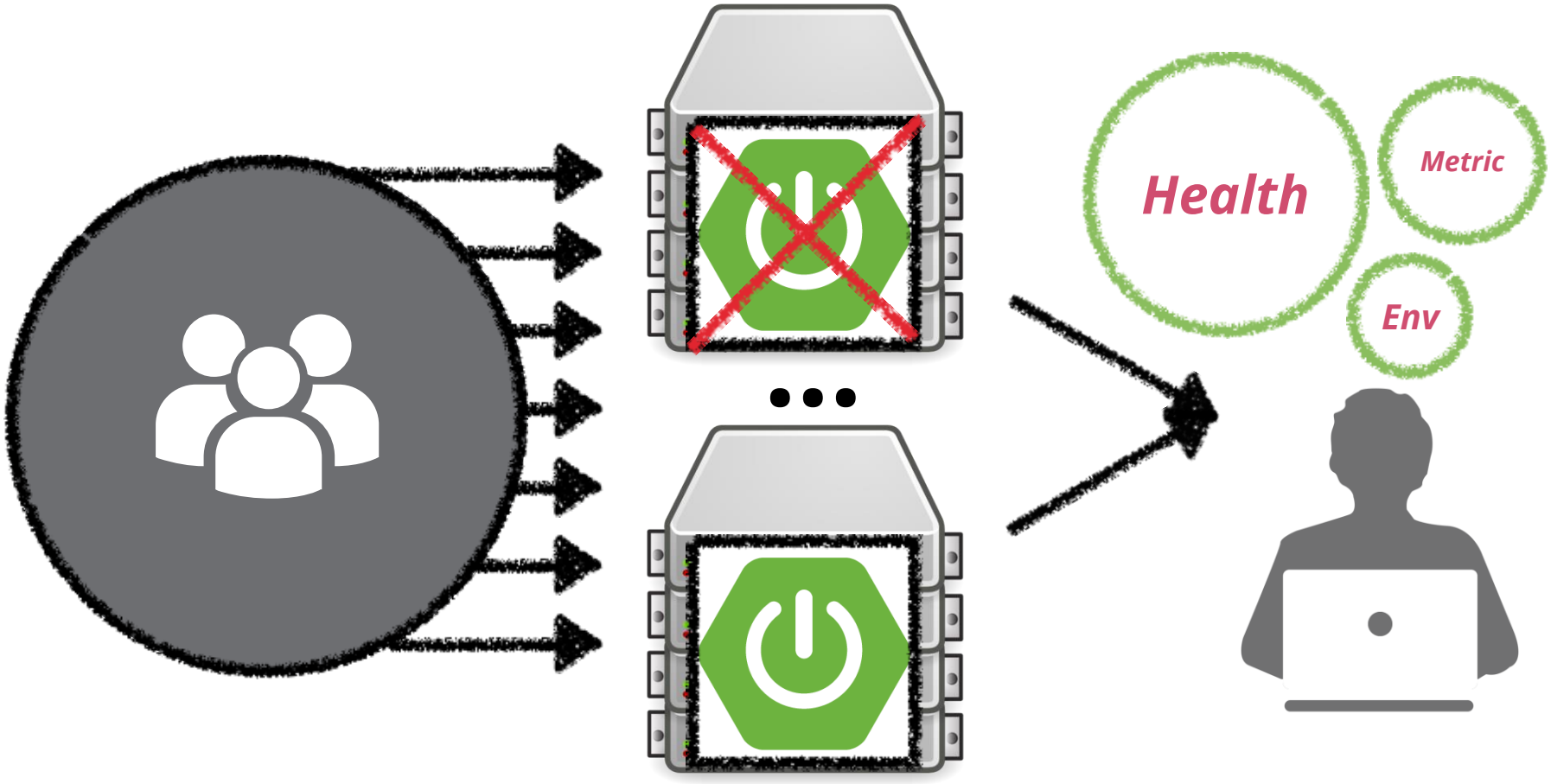
a number of additional features

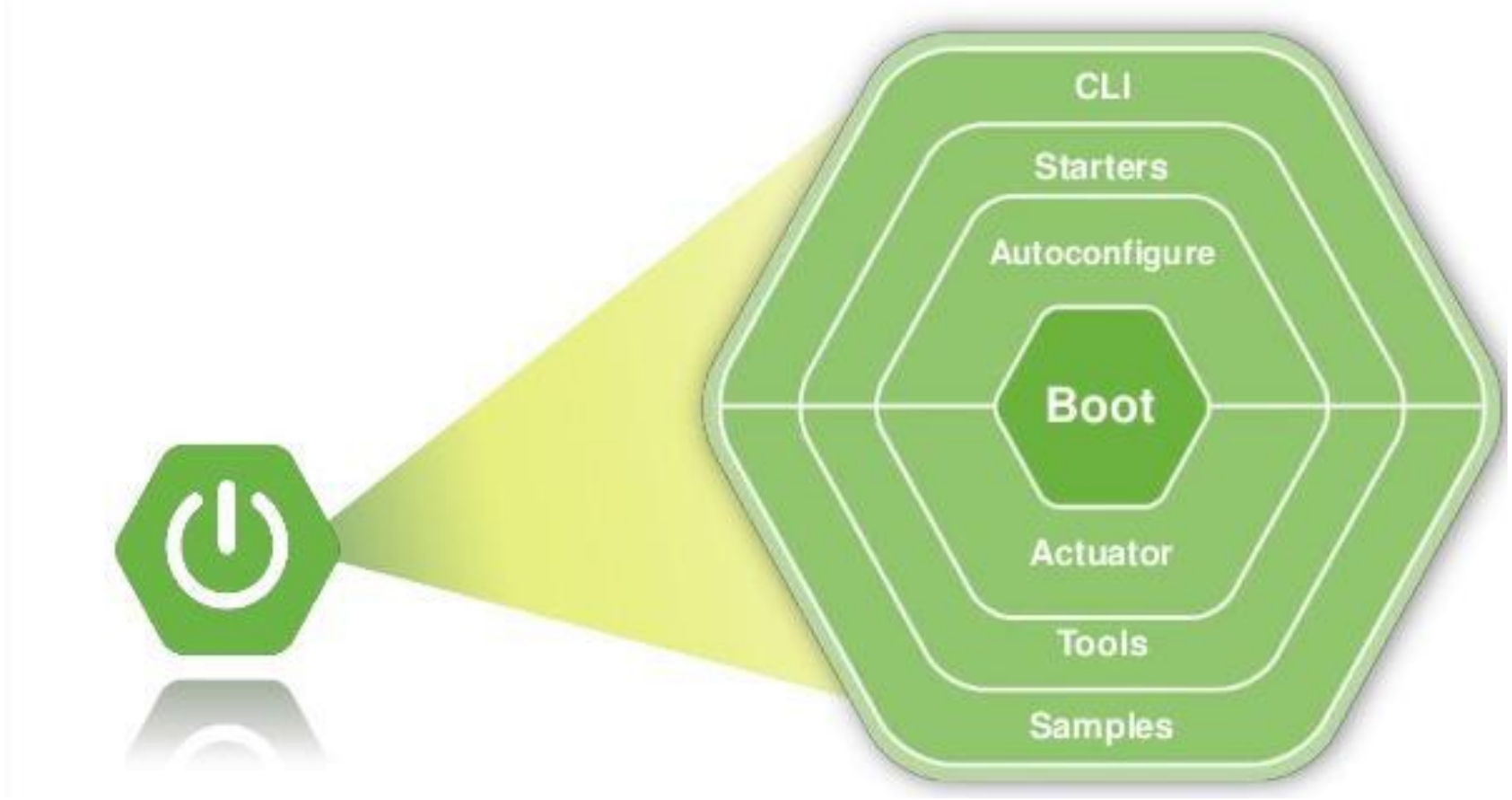


Production-ready features



spring-boot-starter-actuator

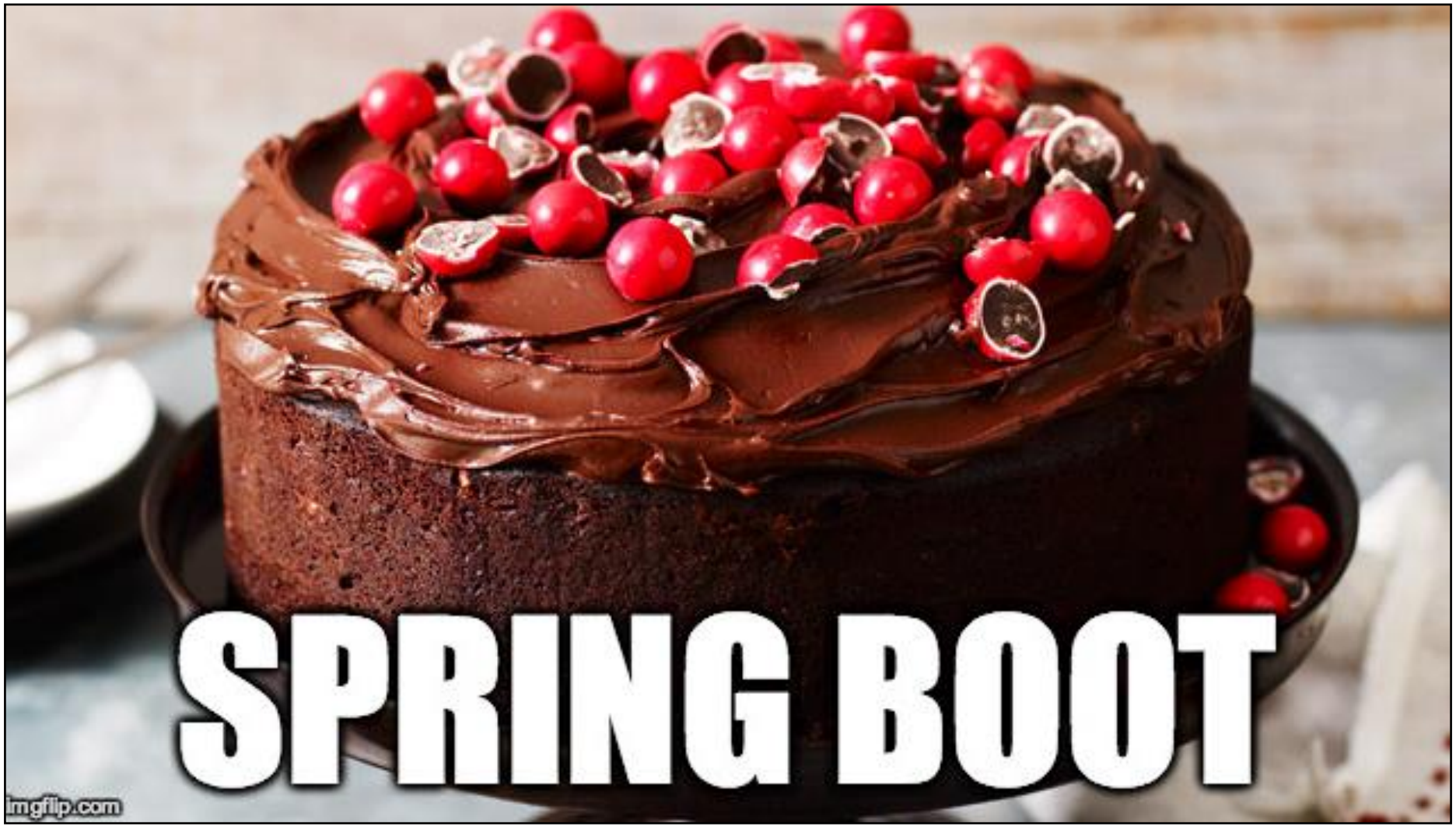




PART 03 Spring VS Spring Boot

01 Spring VS Spring Boot

02 Finally



you can “**just run**”.

Anywhere

감사합니다