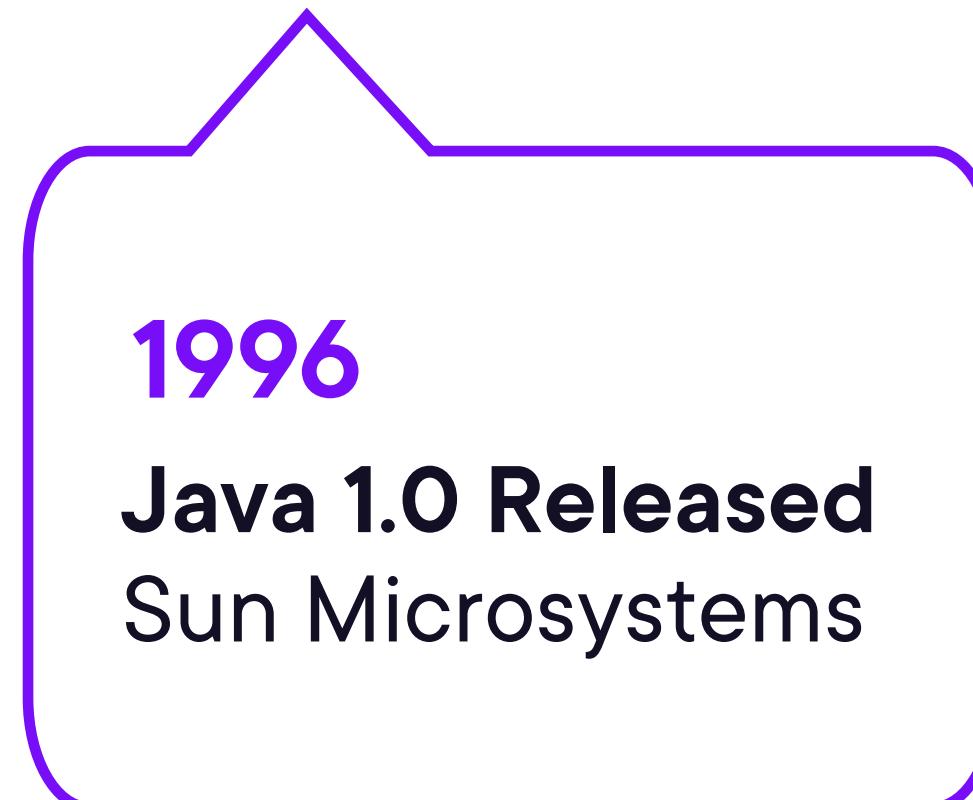


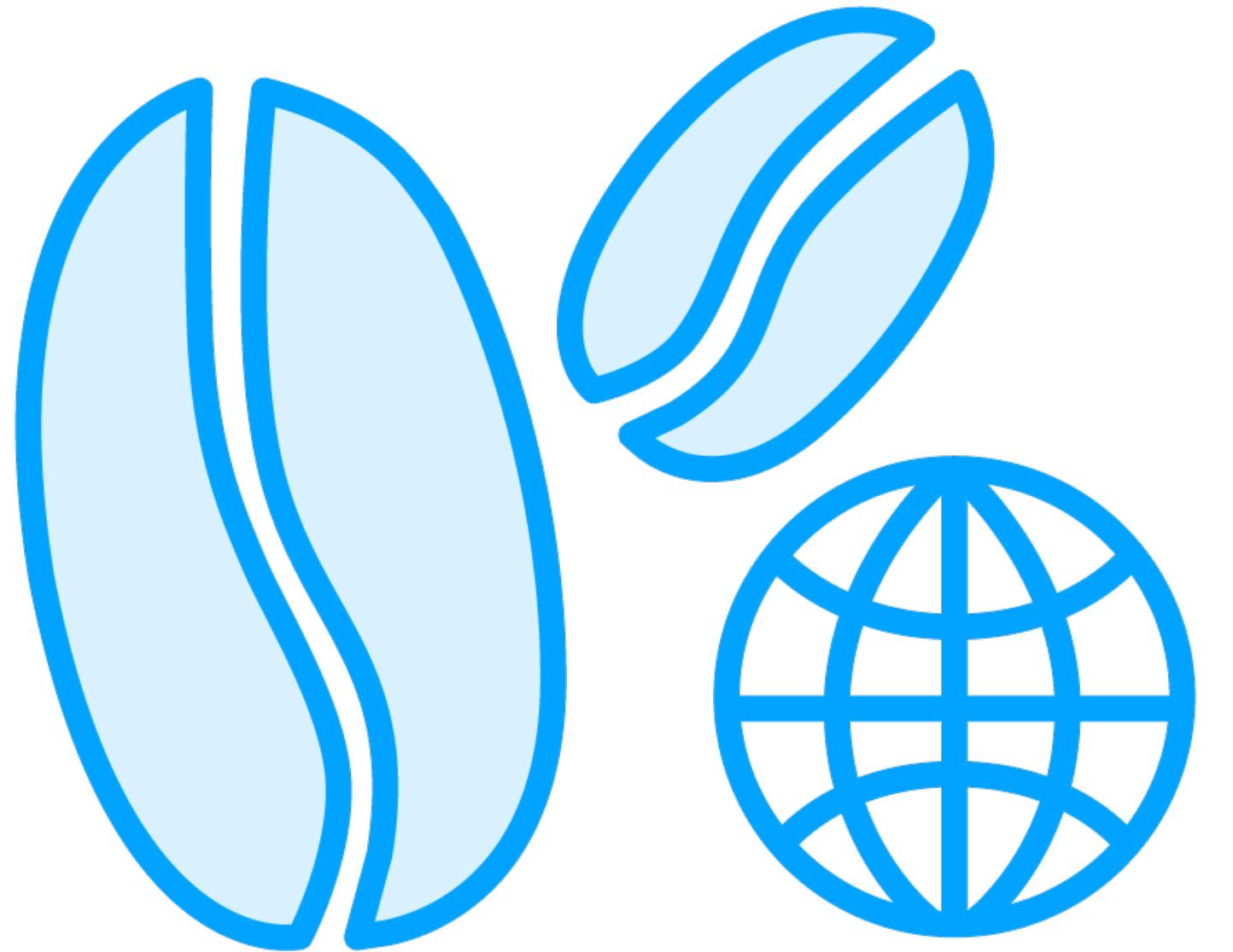
A Brief History of Java

Stuff like
Cobol
Fortran
C/C++



Now

What Is Spring?



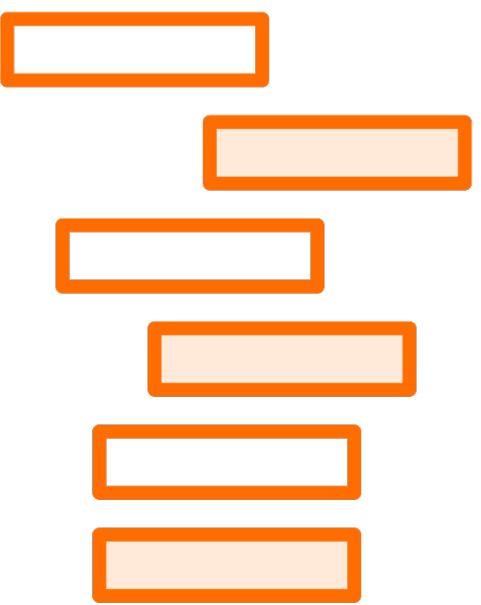
Java

Kotlin
JRuby
Groovy
BeanShell

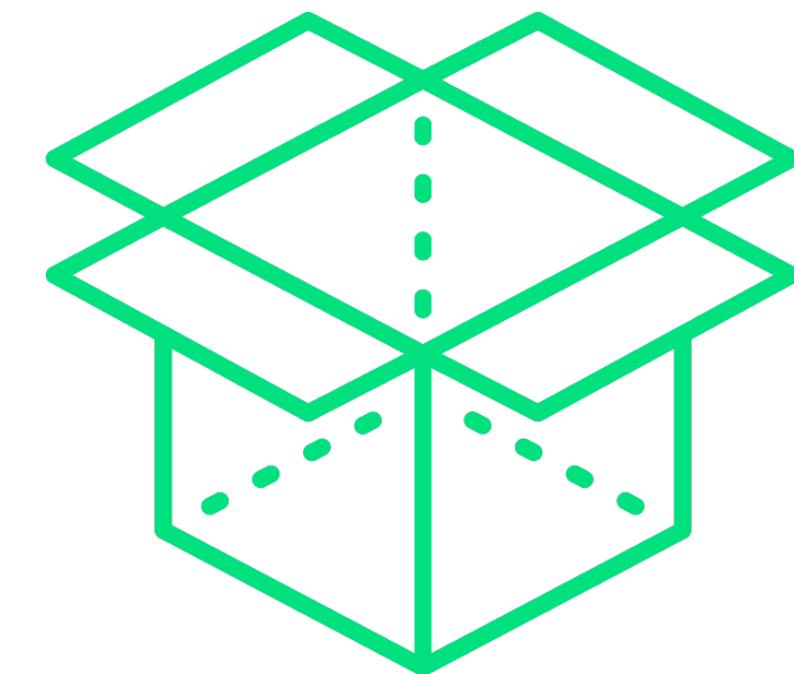
What Is Spring?



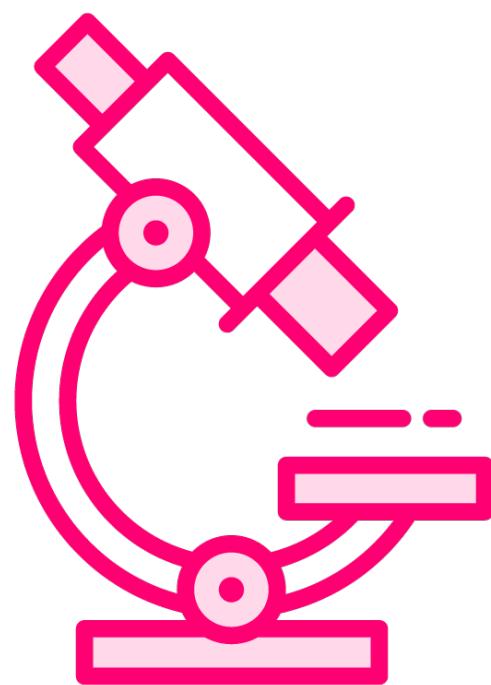
Fast



Integrates



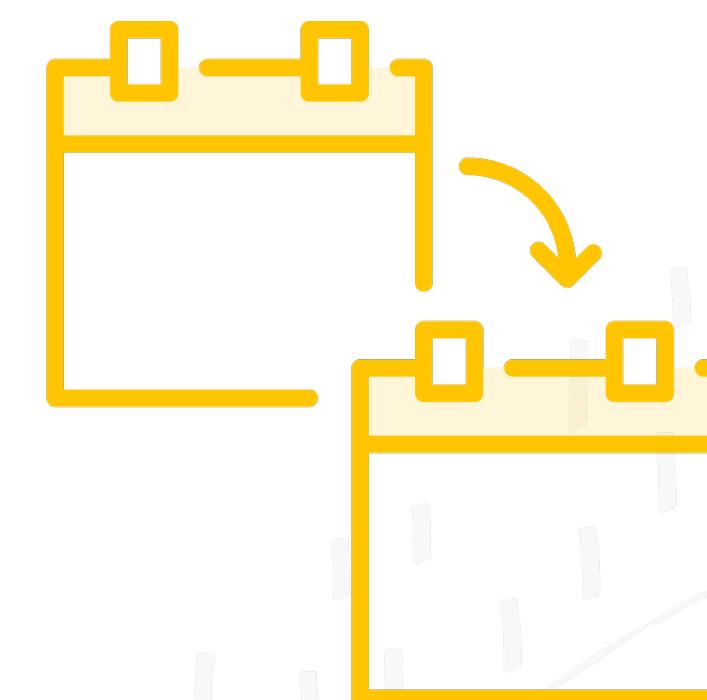
Open Source



Micro Service



Serverless



Event Driven

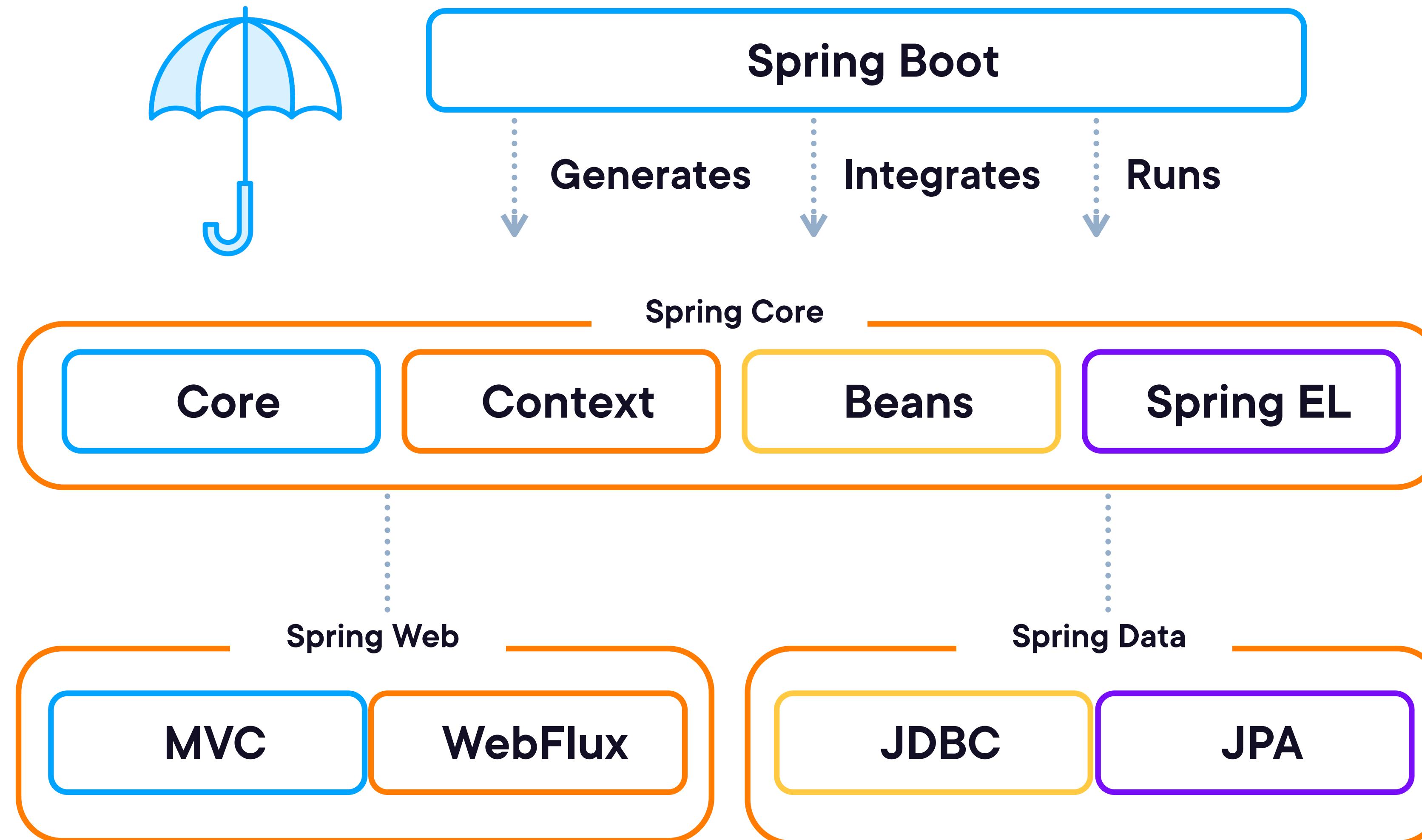
What Is Spring?

Spring Framework

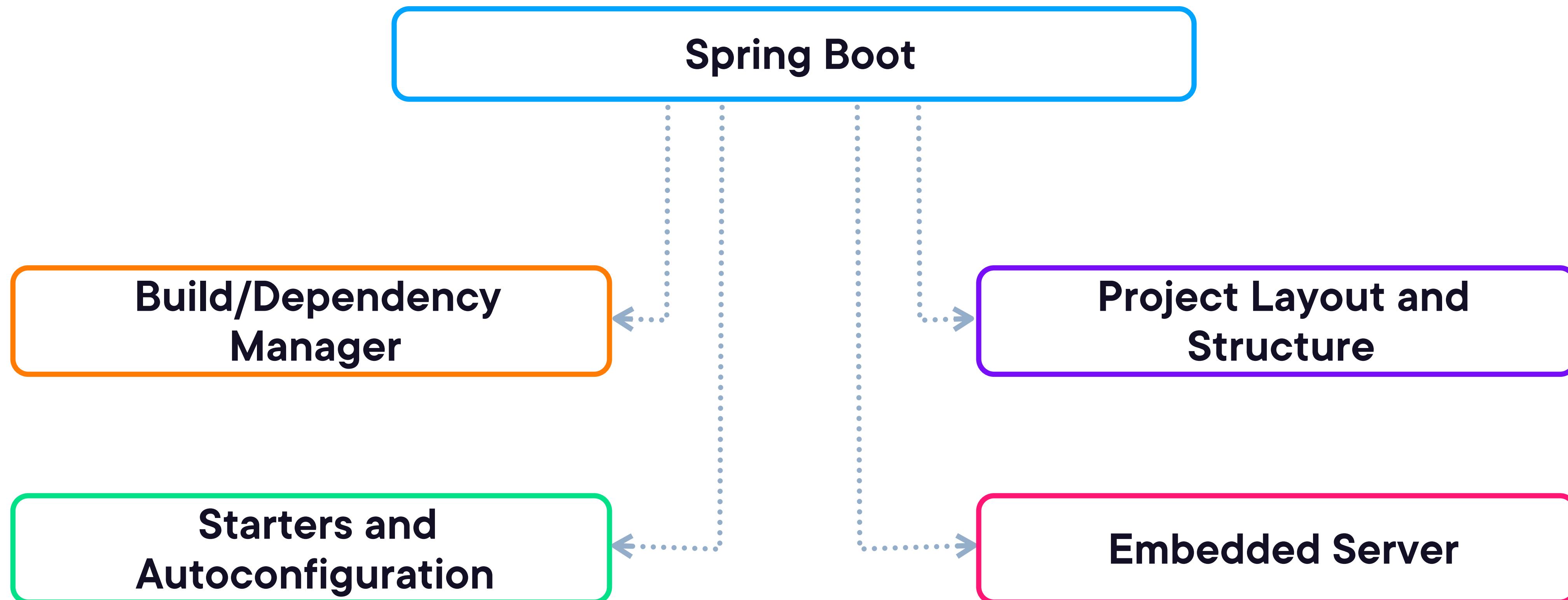
Spring MVC
Spring Data

Spring Boot

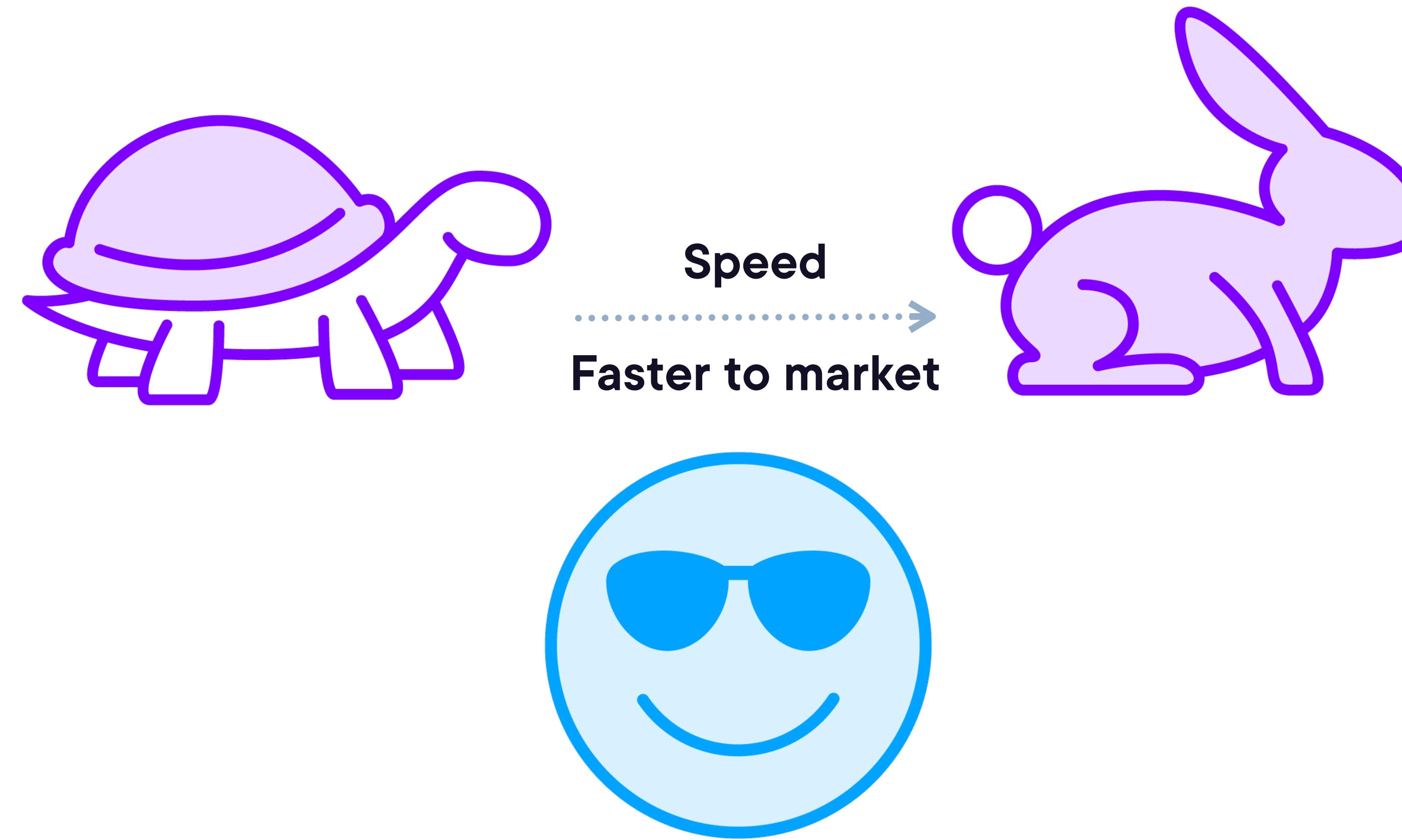
Where Does Spring Boot Fit in the Spring Ecosystem?



Where Does Spring Boot Fit in the Spring Ecosystem?



Why Was Spring Boot Built?



Ruby on Rails - Python / Django - JavaScript / Node

What's New in Spring Boot 3?

Java 17

- Switch statements, text blocks, etc

Java EE moved to Jakarta EE

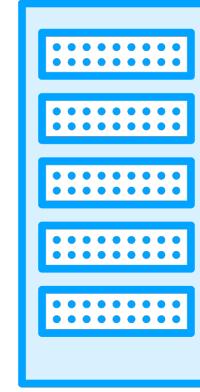
Observability

Native Executables

Does Spring Boot Work with Modern Cloud-based Applications?

**If you can run Java, you can
run Spring and Spring Boot**

Does Spring Boot Work with Modern Cloud-based Applications?



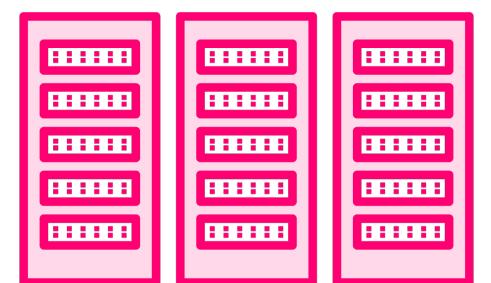
Physical servers



Virtual servers



Cloud computing or serverless



On/off premise

What Alternatives or Other Frameworks Are Comparable?

Java Alternatives

Straight Java and/or JEE

Micronaut

Quarkus

Vert.x

VS

Non Java Alternatives

Javascript and NodeJS

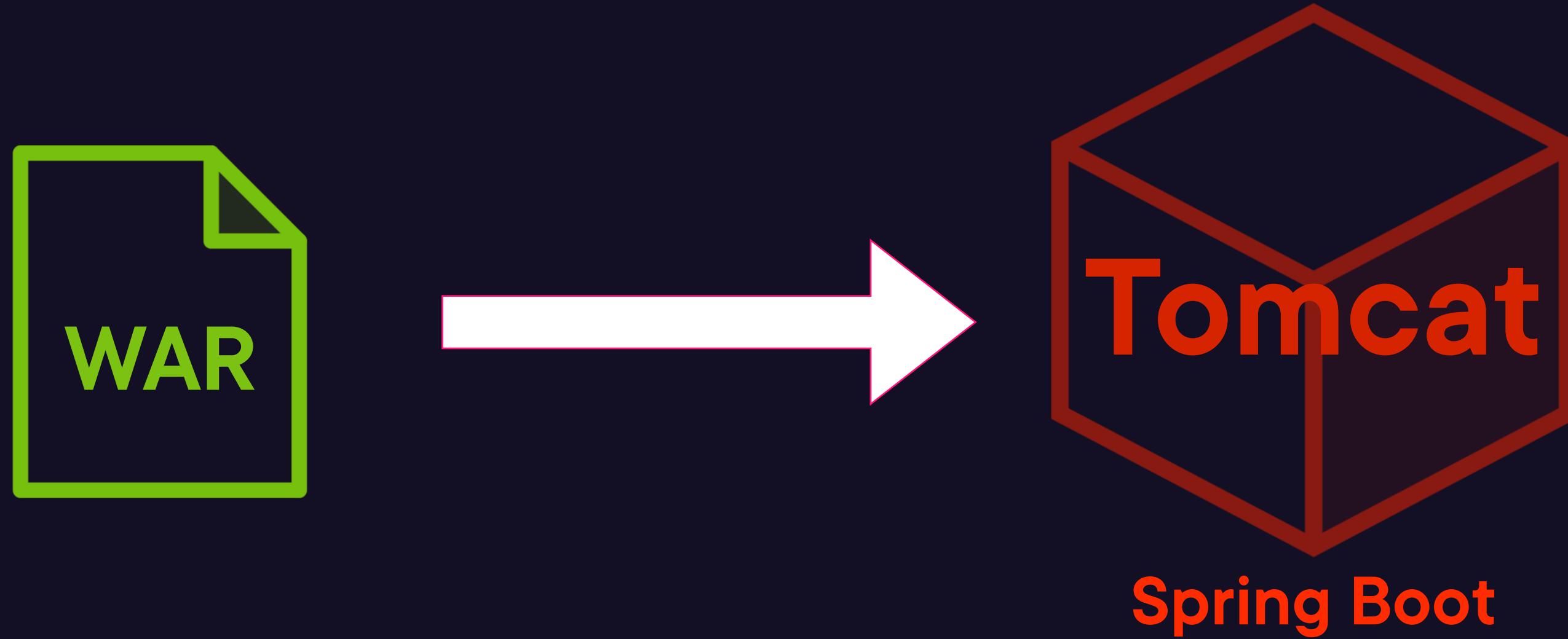
Ruby on Rails

Python and Django

PHP - Cake/Laravel/Symfony

Microsoft C# and .net

Just a Java App, Nothing More

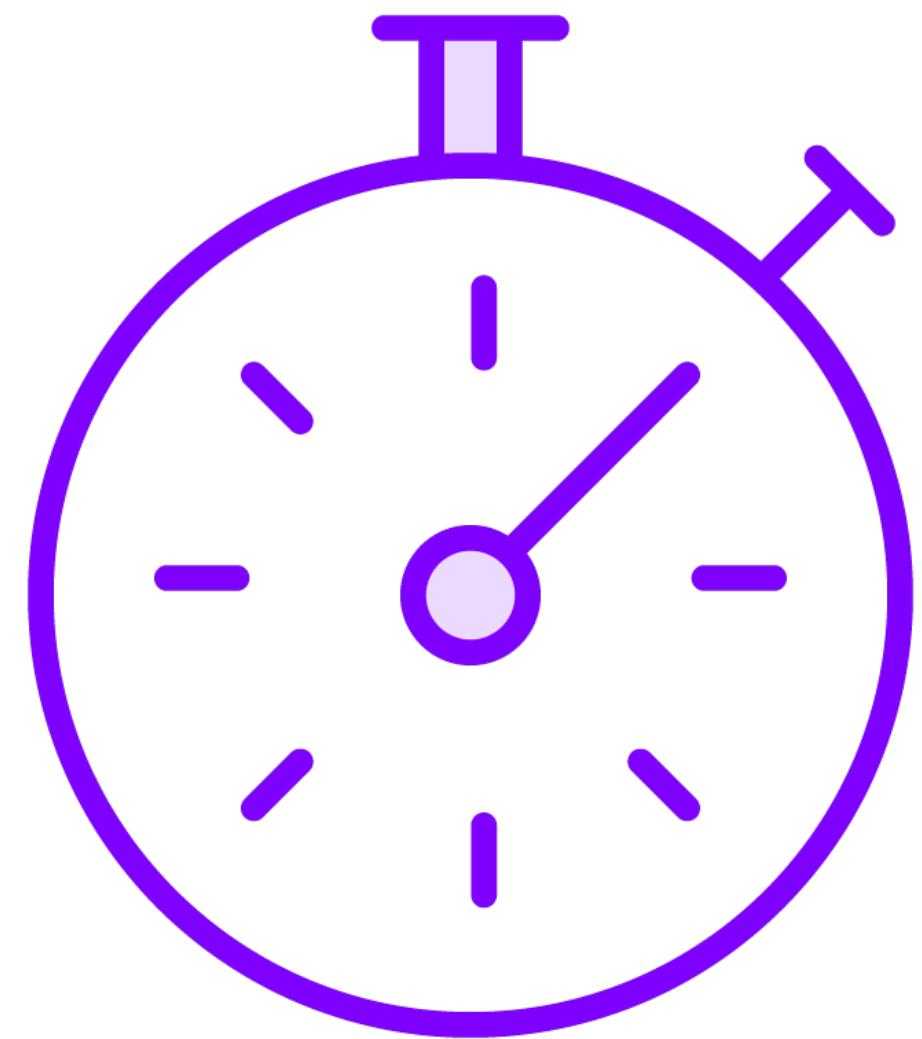


```
public static void main(String[] args) {  
    SpringApplication.run(MyApplication.class, args);  
}
```

Structure and Dependency Management

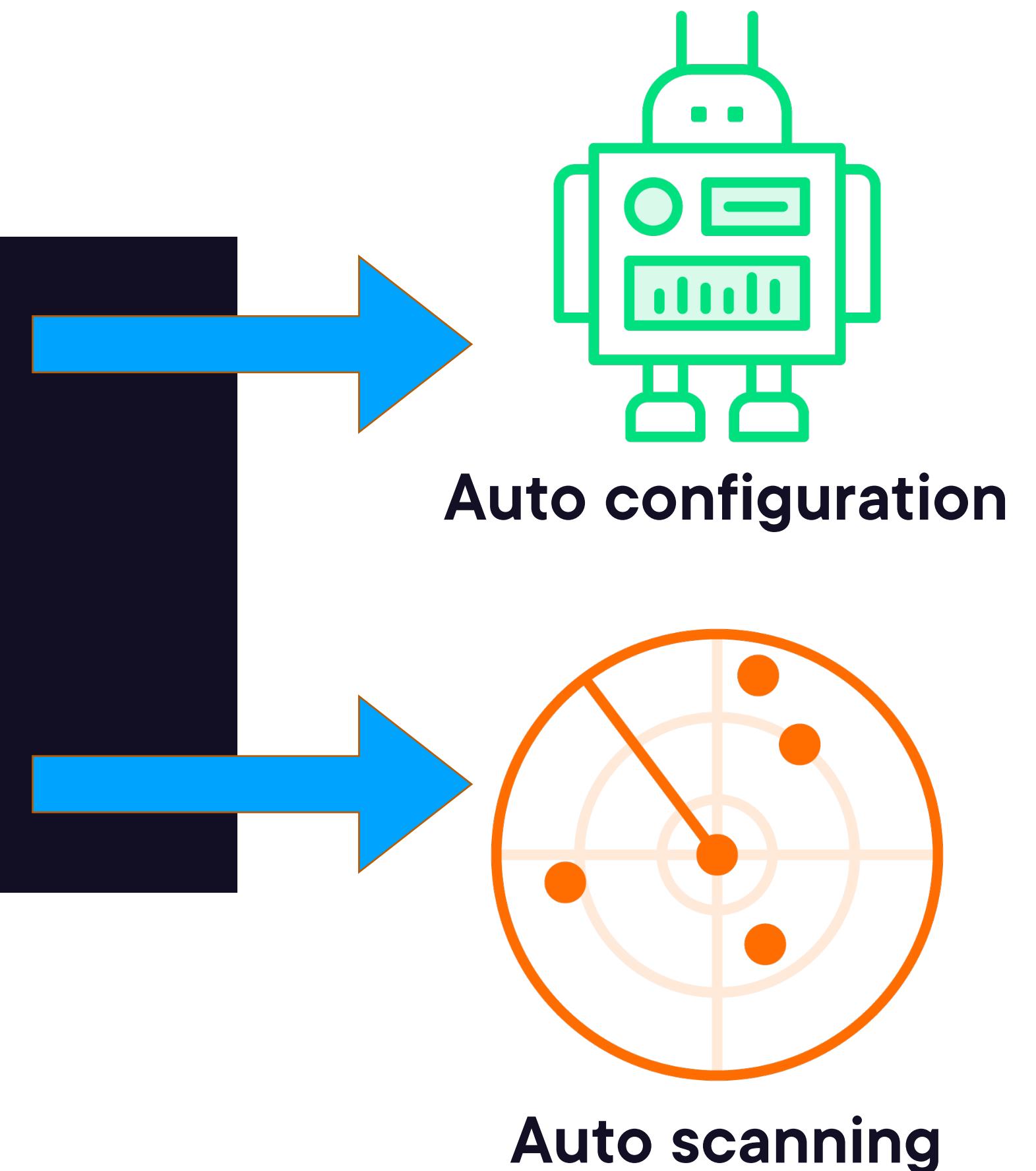
```
<parent>
  <groupId>org.springframework.boot</groupId>
  <artifactId>spring-boot-starter-parent</artifactId>
  <version>3.1.2</version>
  <relativePath/>
</parent>
...
<dependency>
  <groupId>org.springframework.boot</groupId>
  <artifactId>spring-boot-starter-data-jpa</artifactId>
</dependency>
<dependency>
  <groupId>org.springframework.boot</groupId>
  <artifactId>spring-boot-starter-web</artifactId>
</dependency>
```

Opinionated Auto-configuration



Follow or customize?

```
@SpringBootApplication  
public class DemoApplication {  
    ...  
}
```



Spring Boot's Emphasis on Time to Market

The screenshot shows the Spring Initializr web application interface. At the top, there is a green header bar with the text "Meet the Spring team this August at SpringOne." Below the header, the Spring logo and "initializr" are displayed. The main interface is divided into several sections:

- Project**: Options for "Gradle - Groovy", "Gradle - Kotlin", and "Maven".
- Language**: Options for "Java" (selected), "Kotlin", and "Groovy".
- Spring Boot**: Options for "3.2.0 (SNAPSHOT)", "3.2.0 (M1)", "3.1.3 (SNAPSHOT)" (selected), "3.1.2", "3.0.10 (SNAPSHOT)", "3.0.9", "2.7.15 (SNAPSHOT)", and "2.7.14".
- Project Metadata**: Fields for "Group" (com.example), "Artifact" (demo), "Name" (demo), "Description" (Demo project for Spring Boot), "Package name" (com.example.demo), and "Packaging" (Jar selected, War). There are also dropdowns for Java version (20 selected) and Spring Boot version (1 selected).
- Dependencies**: A section titled "Dependencies" with a "ADD DEPENDENCIES..." button.
- Spring Web**: A section titled "Spring Web" with a "WEB" badge. It describes building web applications using Spring MVC and Apache Tomcat.
- Spring Data JPA**: A section titled "Spring Data JPA" with a "SQL" badge. It describes persisting data in SQL stores using Spring Data and Hibernate.

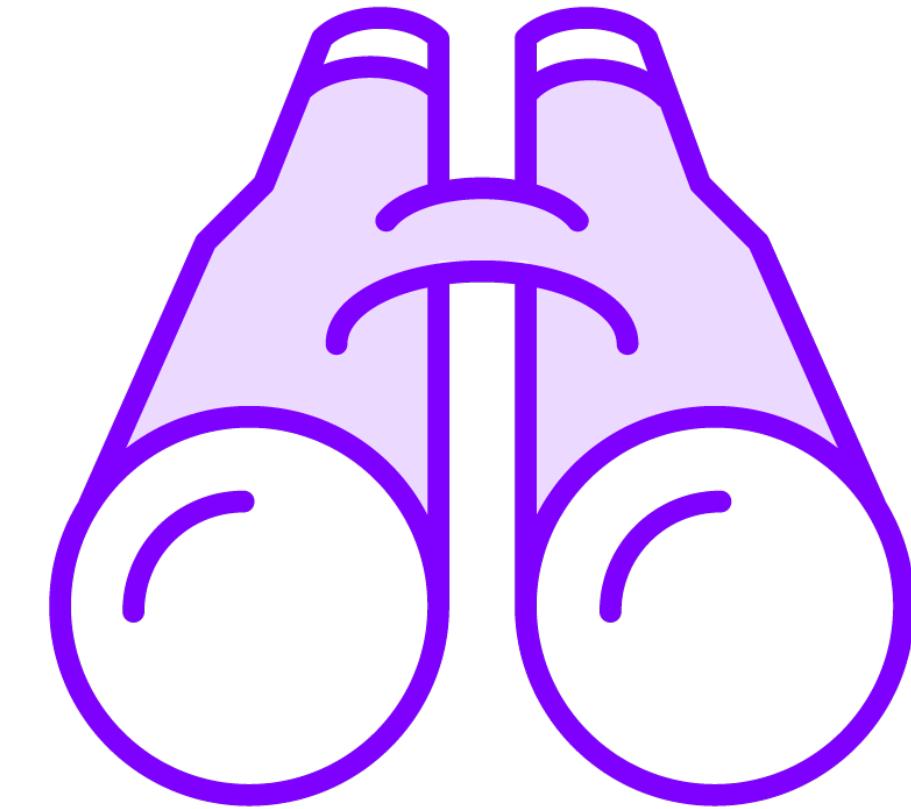
In the center of the page is a large orange box containing the URL <https://start.spring.io/>.

At the bottom of the page, there are social sharing icons for GitHub and Twitter, and three buttons: "GENERATE" (with keyboard shortcut $\text{⌘} + \text{J}$), "EXPLORE" (with keyboard shortcut $\text{CTRL} + \text{SPACE}$), and "SHARE...".

Monitoring, Health Checks, and Metrics

spring-boot-starter-actuator

Spring Boot 3 Observability



@Observed

Actuator

A manufacturing term that refers to a mechanical device for moving or controlling something. Actuators can generate a large amount of motion from a small change

Monitoring, Health Checks, and Metrics

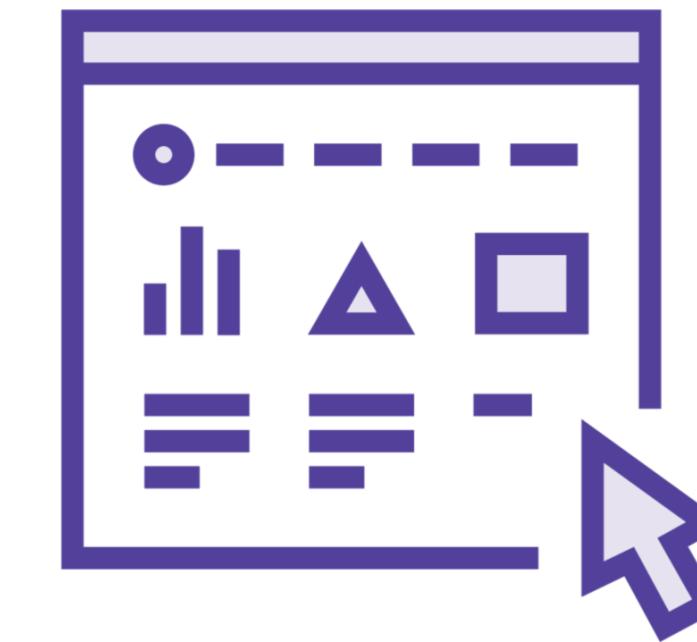
spring-boot-starter-actuator



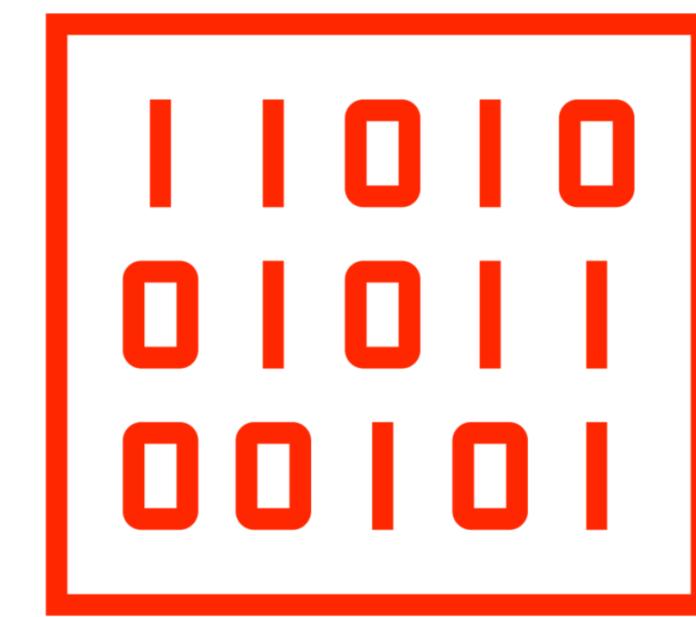
Health Checks



Process Info



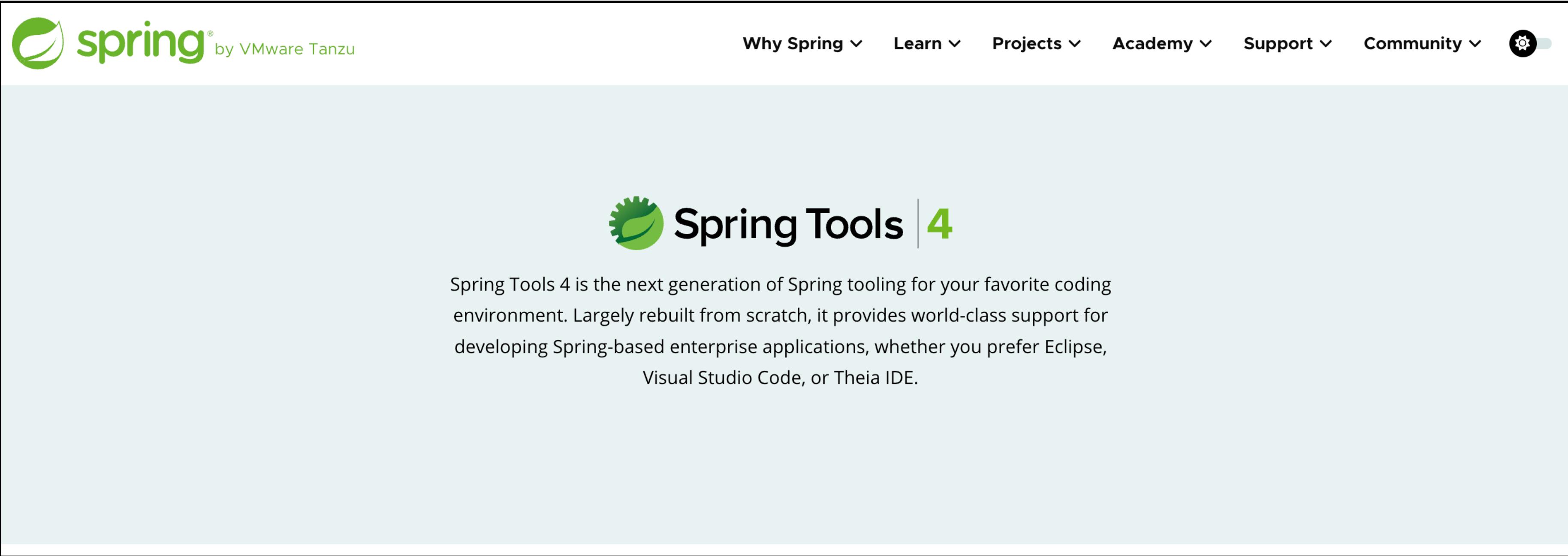
Monitor Console



Custom

spring-boot-admin-starter-server

Tools to Be Successful with Spring Boot

A screenshot of the official Spring website. At the top left is the green Spring logo with the word "spring" and "by VMware Tanzu". To the right are navigation links: "Why Spring", "Learn", "Projects", "Academy", "Support", "Community", and a gear icon. Below the header, there's a large light blue banner. On the left side of the banner is the Spring Tools 4 logo, which consists of a green circular icon with a leaf-like shape next to the text "Spring Tools | 4". To the right of the logo, the text reads: "Spring Tools 4 is the next generation of Spring tooling for your favorite coding environment. Largely rebuilt from scratch, it provides world-class support for developing Spring-based enterprise applications, whether you prefer Eclipse, Visual Studio Code, or Theia IDE."

The screenshot shows the Spring website's homepage with a focus on the "Tools" section. The Spring logo is at the top left, followed by a navigation bar with links like "Why Spring", "Learn", "Projects", "Academy", "Support", "Community", and a settings gear icon. Below the header is a large light blue banner. On the left of the banner is the "Spring Tools 4" logo, which includes a green circular icon with a leaf-like shape and the text "Spring Tools | 4". To the right of the logo, there is a descriptive paragraph about the tooling: "Spring Tools 4 is the next generation of Spring tooling for your favorite coding environment. Largely rebuilt from scratch, it provides world-class support for developing Spring-based enterprise applications, whether you prefer Eclipse, Visual Studio Code, or Theia IDE."

<https://spring.io/tools>

```
// pom.xml (Maven builds)
```

```
<dependency>
  <groupId>org.springframework.boot</groupId>
  <artifactId>spring-boot-devtools</artifactId>
  <scope>runtime</scope>
  <optional>true</optional>
</dependency>
```

```
// build.gradle (Gradle builds)
```

```
dependencies {
  developmentOnly("org.springframework.boot:
    spring-boot-devtools")
}
```

- **Automatic restart**
- **Live reload**
- **Property defaults**
- **Global settings**
- **Remote applications**

Ways to Create a Spring Boot App

Spring Initializr

The screenshot shows the Spring Initializr web application. It has sections for Project (Gradle - Groovy selected), Language (Java selected), Dependencies (No dependency selected), and Project Metadata (Group: com.example, Artifact: demo, Name: demo, Description: Demo project for Spring Boot, Package name: com.example.demo, Packaging: Jar selected). Buttons at the bottom include GENERATE, EXPLORE, and SHARE... .

Spring Tools 4

The screenshot shows the Spring Tools 4 website. It highlights that it's the next generation of Spring tooling for Eclipse, Visual Studio Code, or Theia IDE. Below is a section for Visual Studio Code with a screenshot of the VS Code interface showing Java code. A button at the bottom says SPRING TOOLS 4 VSCode Marketplace.

Overview of Spring Boot Build Tools

Project layout and structure

Manage project dependencies

Package project for deployment

Overview of Spring Boot Build Tools

Maven

<https://maven.apache.org/>

Gradle

<https://gradle.org/>

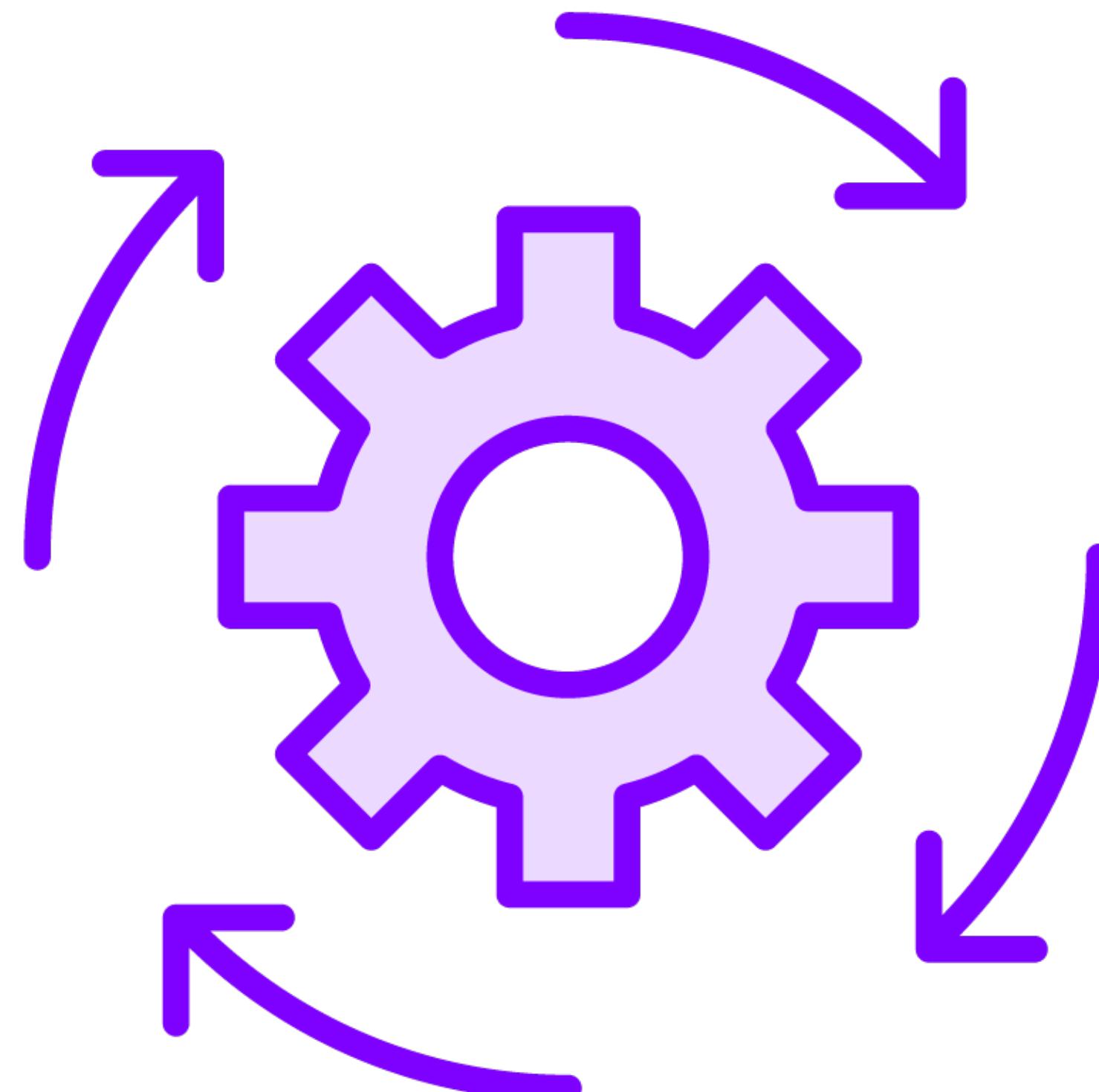
Maven Commands

```
> mvn clean  
  
> mvn compile  
  
> mvn test  
  
> mvn package  
  
> mvn install  
  
> mvn spring-boot:run  
  
> mvn spring-boot:build-image  
  
> mvn spring-boot:repackage  
  
> mvn dependency:tree  
  
> mvn clean install
```

Gradle Commands

```
> ./gradlew clean  
  
> ./gradlew build  
  
> ./gradlew test  
  
> ./gradlew install  
  
> ./gradlew bootRun  
  
> ./gradlew bootJar  
  
> ./gradlew bootWar  
  
> ./gradlew tasks  
  
> ./gradlew dependencies  
  
> ./gradlew clean build
```

Customizing the Build



- Project dependency management**
- Spring Boot upgrades**
- Build plugins**
- Environment configuration**

Starting and Stopping Spring Boot Apps

“mvn package” or “./gradlew build”

`java -jar <jar_file>.jar`

Demo



Spring Initializr

Project in IntelliJ

Launch application

Auto Configuration in Action

Auto-configuration is a very useful and powerful feature of Spring Boot, which takes a “convention-over-configuration” approach.

How it Works

Process

Finds JARs on the classpath
and auto-configures bean

Auto-configures

Data source for Hibernate or
DispatcherServlet for Spring
MVC

Automatic Configuration



Beans

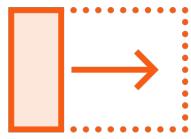
Add your own beans, like a datasource bean



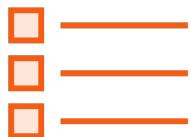
Database Support

Default embedded database support backs away

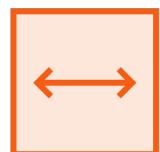
Auto-Configuration Insights



Start application with --debug switch



Add a simple property to application.properties



Use the Spring Boot Actuator

Annotations

@SpringBootApplication Annotation

@SpringBootConfiguration

Replaces @Configuration and annotates a class as a configuration

@EnableAutoConfiguration

Tells Spring Boot to configure beans

@ComponentScan

Tells Spring Boot to scan current package and subpackages