

Overview of Spring & Spring Boot

Douglas C. Schmidt

d.schmidt@vanderbilt.edu

www.dre.vanderbilt.edu/~schmidt

Professor of Computer Science

**Institute for Software
Integrated Systems**

**Vanderbilt University
Nashville, Tennessee, USA**



Learning Objectives in this Lesson

- Understand what Spring & Spring Boot are



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.



Spring Data

Provides a consistent approach to data access – relational, non-relational, map-reduce, and beyond.



Spring Cloud

Provides a set of tools for common patterns in distributed systems. Useful for building and deploying microservices.



Spring Cloud Data Flow

Provides an orchestration service for composable data microservice applications on modern runtimes.



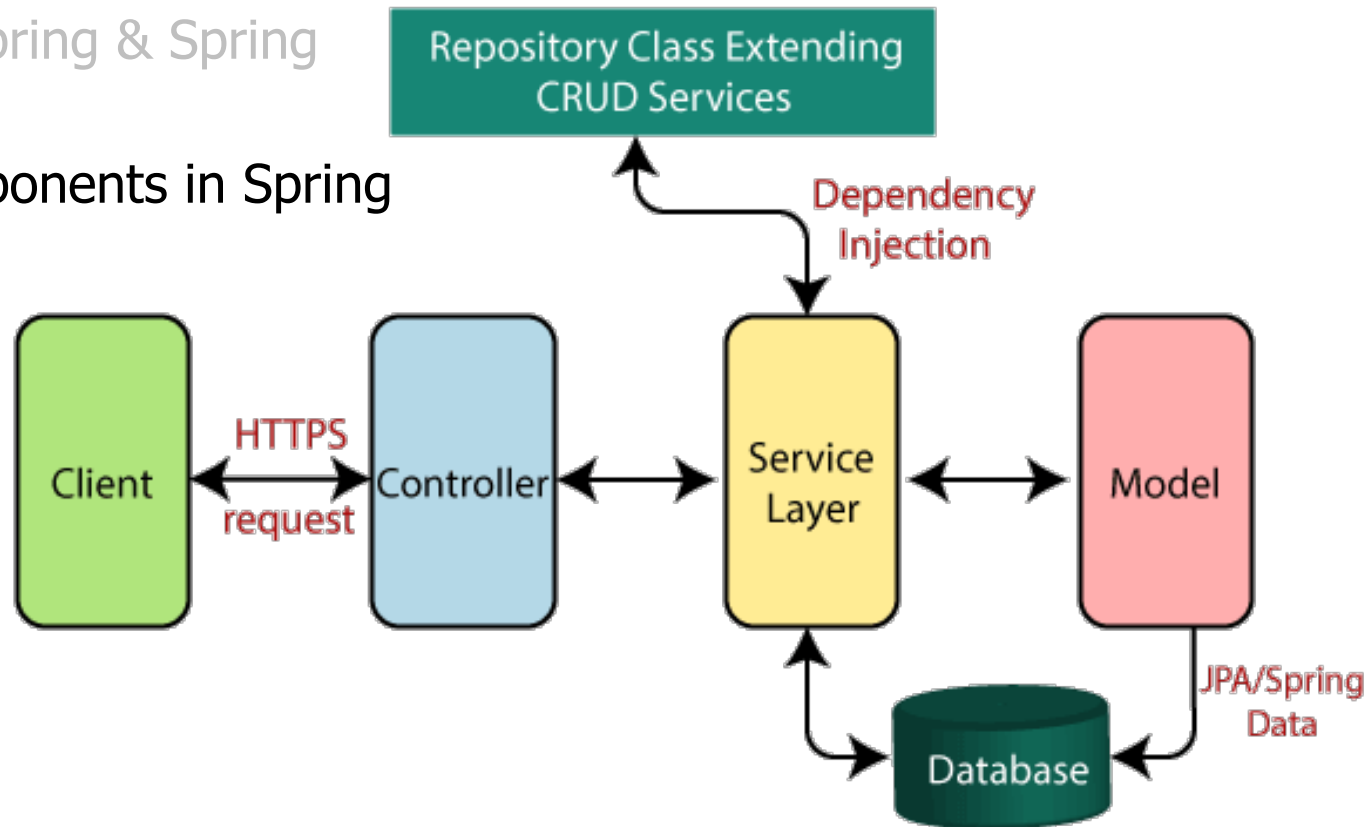
Spring Security

Protects your application with comprehensive and extensible authentication and authorization support.

See spring.io/projects

Learning Objectives in this Lesson

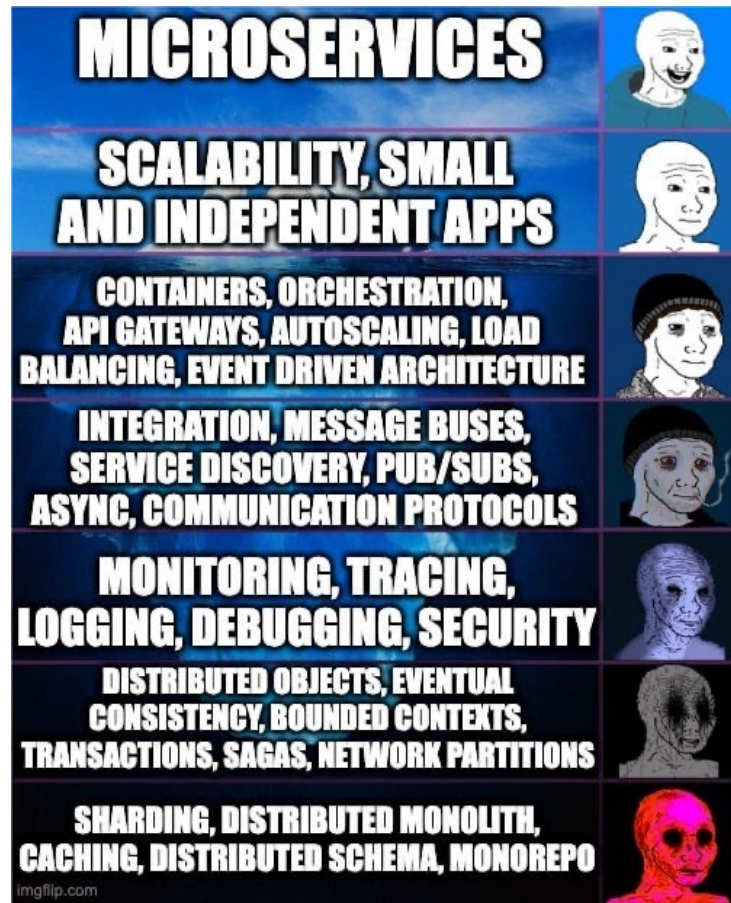
- Understand what Spring & Spring Boot are
- Recognize key components in Spring Boot



See spring.io/projects/spring-boot

Learning Objectives in this Lesson

- Understand what Spring & Spring Boot are
- Recognize key components in Spring Boot
- Be aware of microservice architectures



Overview of Spring & Spring Boot

Overview of Spring & Spring Boot

- Spring is a dependency-injection framework & an inversion of control container for developing web apps on the Java platform



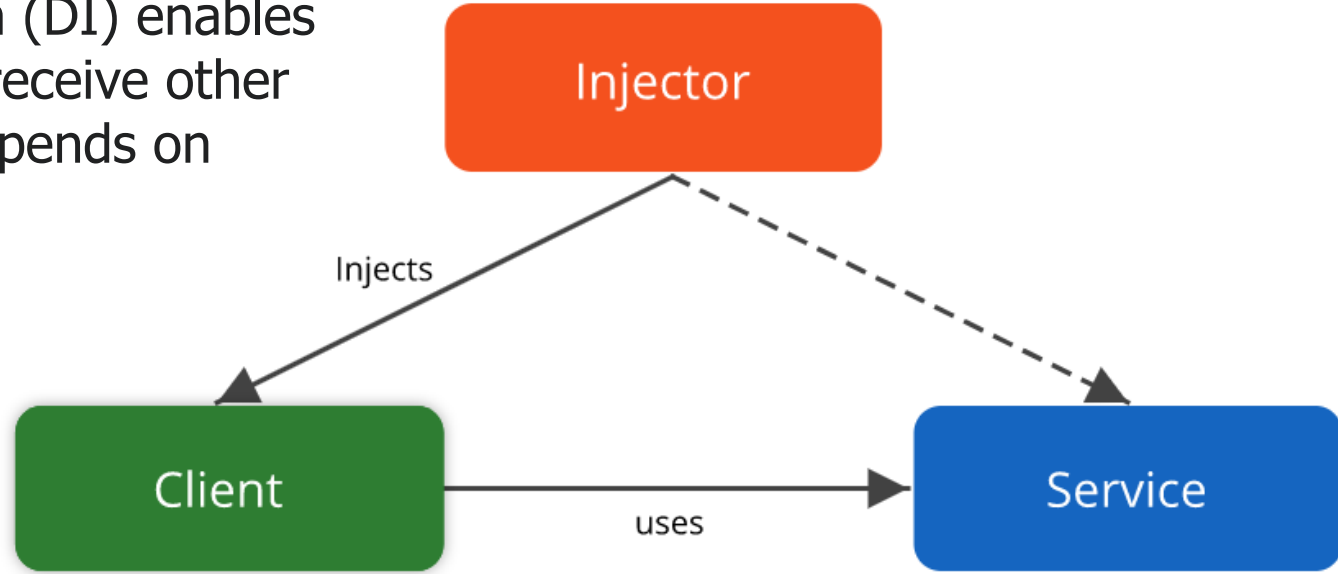
spring

by Pivotal™

See spring.io

Overview of Spring & Spring Boot

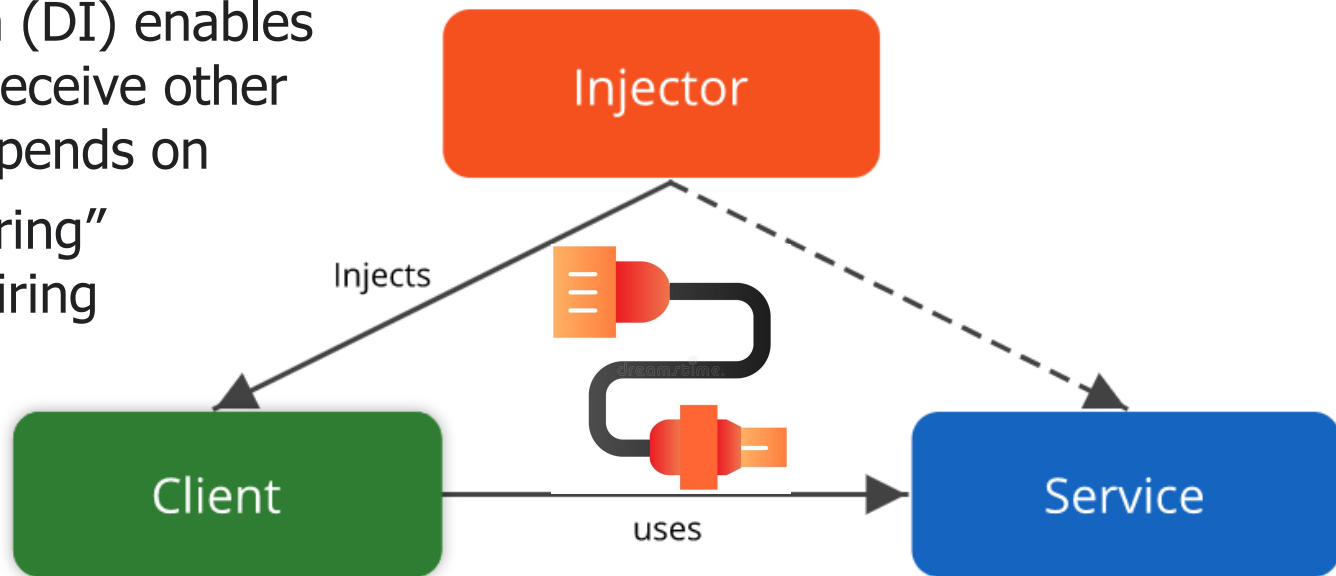
- Dependency injection (DI) enables an object/service to receive other objects/services it depends on



See en.wikipedia.org/wiki/Dependency_injection

Overview of Spring & Spring Boot

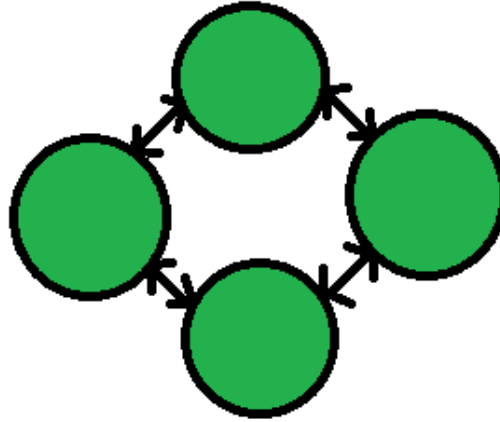
- Dependency injection (DI) enables an object/service to receive other objects/services it depends on
- DI enables “autowiring” in lieu of manual wiring



See www.baeldung.com/spring-autowire

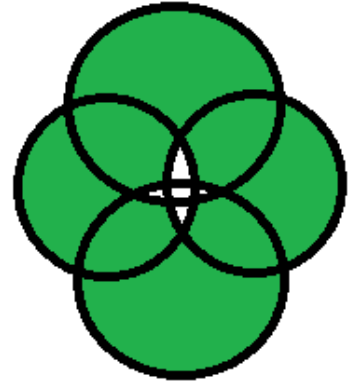
Overview of Spring & Spring Boot

- Dependency injection (DI) enables an object/service to receive other objects/services it depends on
 - DI enables “autowiring” in lieu of manual wiring
- Separates the concerns of constructing objects & using them, leading to loosely coupled programs



Loose coupling:

1. Less Interdependency
2. Less coordination
3. Less information flow



Tight coupling:

1. More Interdependency
2. More coordination
3. More information flow

Overview of Spring & Spring Boot

- With inversion of control (IoC) the framework runs the main execution thread(s)



See en.wikipedia.org/wiki/Inversion_of_control

Overview of Spring & Spring Boot

- With inversion of control (IoC) the framework runs the main execution thread(s)
 - Implements the “Hollywood Principle”

Don't call us, we'll call you



Overview of Spring & Spring Boot

- Spring contains various projects



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.



Spring Data

Provides a consistent approach to data access – relational, non-relational, map-reduce, and beyond.



Spring Cloud

Provides a set of tools for common patterns in distributed systems. Useful for building and deploying microservices.



Spring Cloud Data Flow

Provides an orchestration service for composable data microservice applications on modern runtimes.



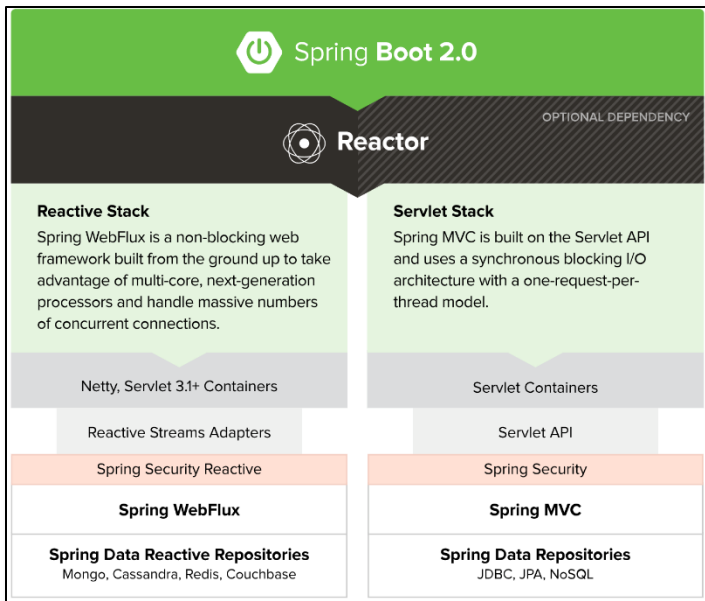
Spring Security

Protects your application with comprehensive and extensible authentication and authorization support.

See spring.io/projects

Overview of Spring & Spring Boot

- Spring contains various projects
- We focus on Spring Boot 2.0



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.



Spring Data

Provides a consistent approach to data access – relational, non-relational, map-reduce, and beyond.



Spring Cloud

Provides a set of tools for common patterns in distributed systems. Useful for building and deploying microservices.



Spring Cloud Data Flow

Provides an orchestration service for composable data microservice applications on modern runtimes.



Spring Security

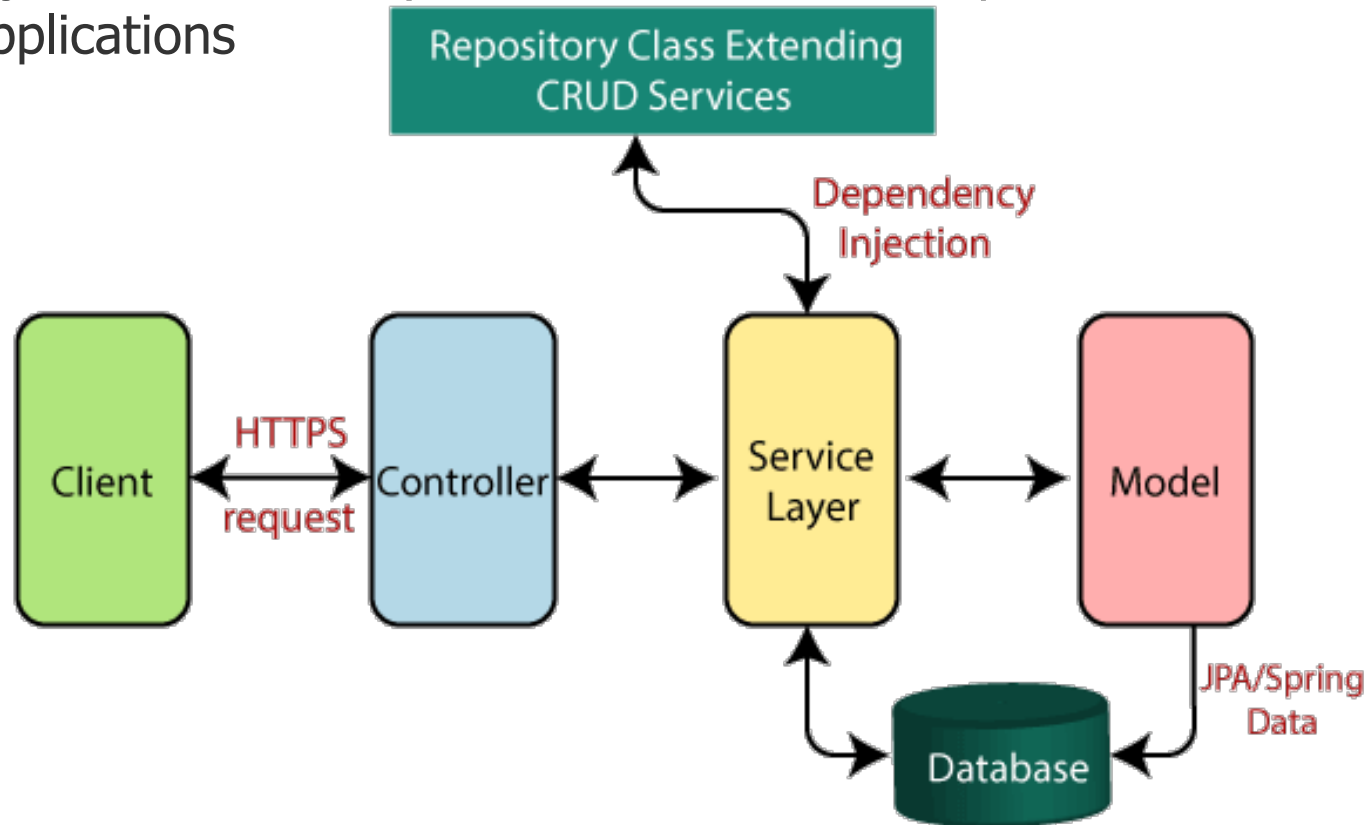
Protects your application with comprehensive and extensible authentication and authorization support.

See www.baeldung.com/new-spring-boot-2

Key Components in Spring Boot

Key Components in Spring Boot

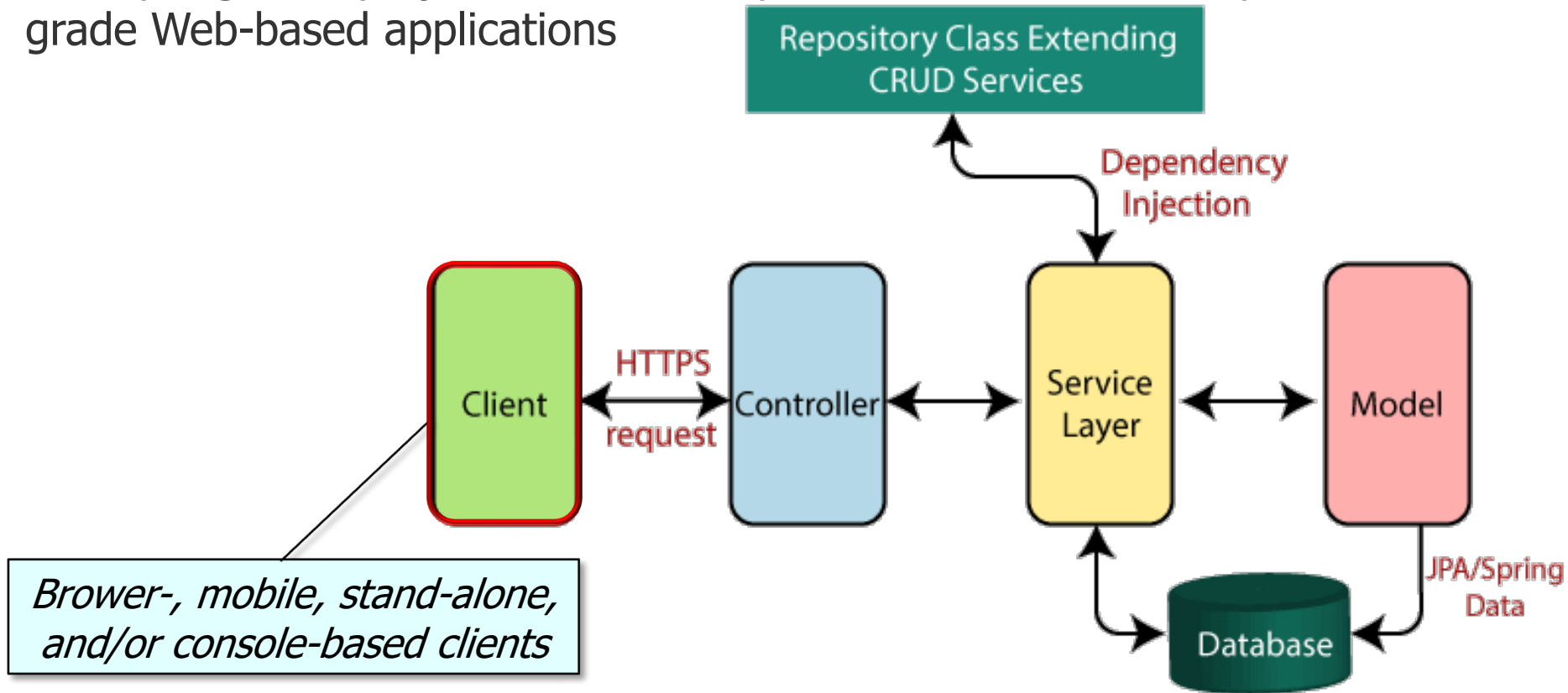
- The Spring Boot project makes it easy to create stand-alone, production-grade Web-based applications



See spring.io/projects/spring-boot

Key Components in Spring Boot

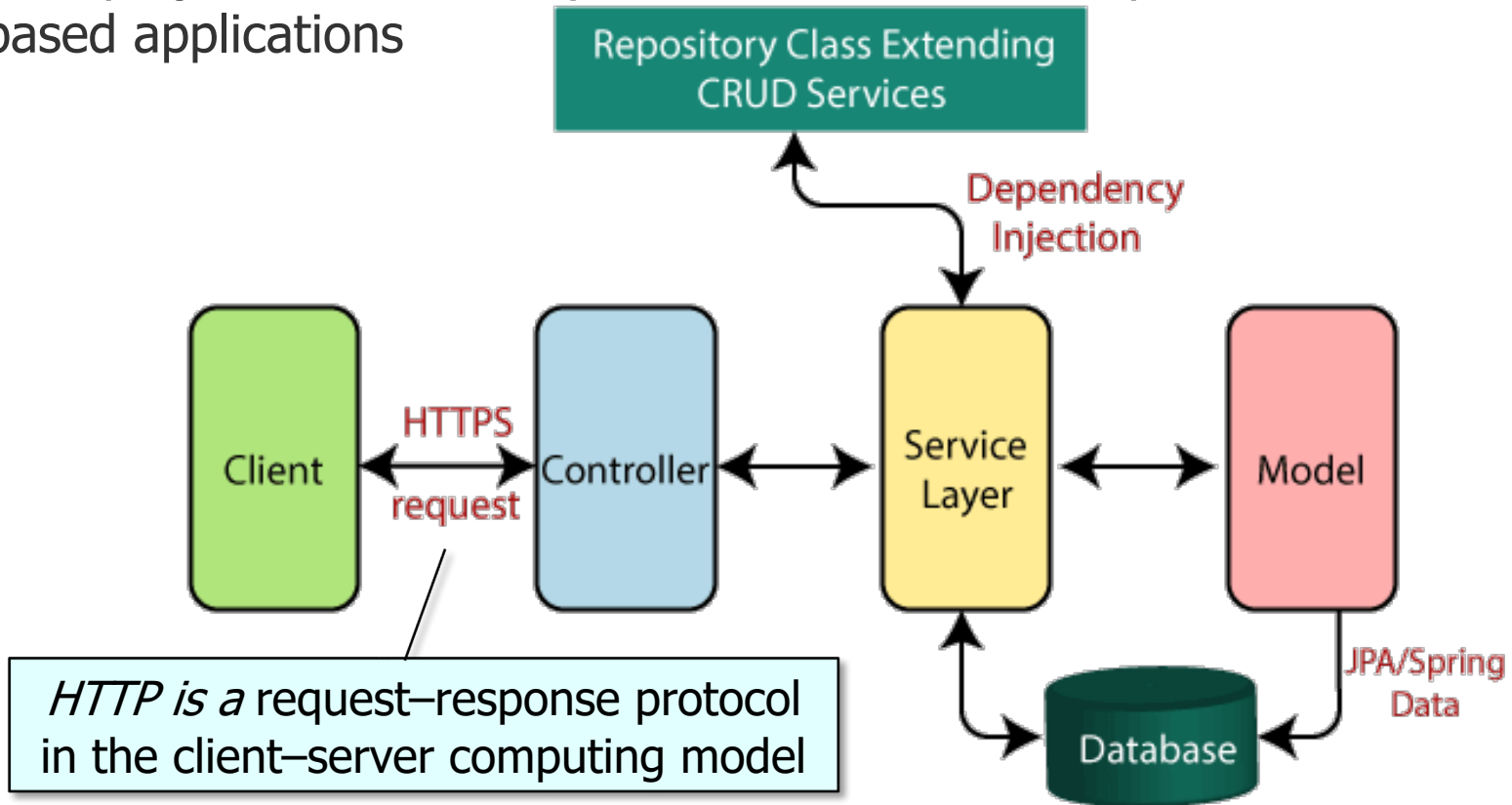
- The Spring Boot project makes it easy to create stand-alone, production-grade Web-based applications



See [en.wikipedia.org/wiki/Client_\(computing\)](https://en.wikipedia.org/wiki/Client_(computing))

Key Components in Spring Boot

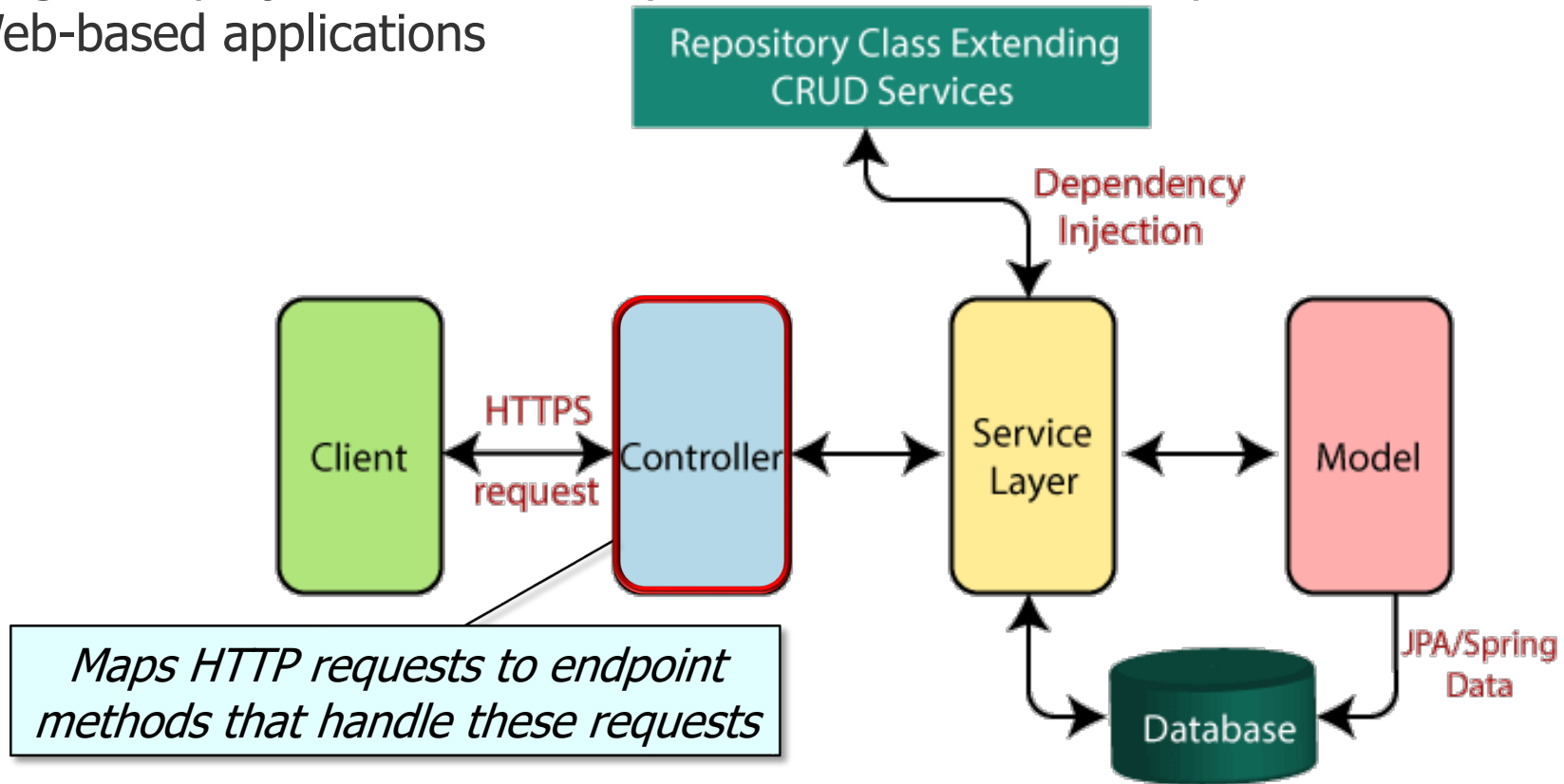
- The Spring Boot project makes it easy to create stand-alone, production-grade Web-based applications



See en.wikipedia.org/wiki/Hypertext_Transfer_Protocol

Key Components in Spring Boot

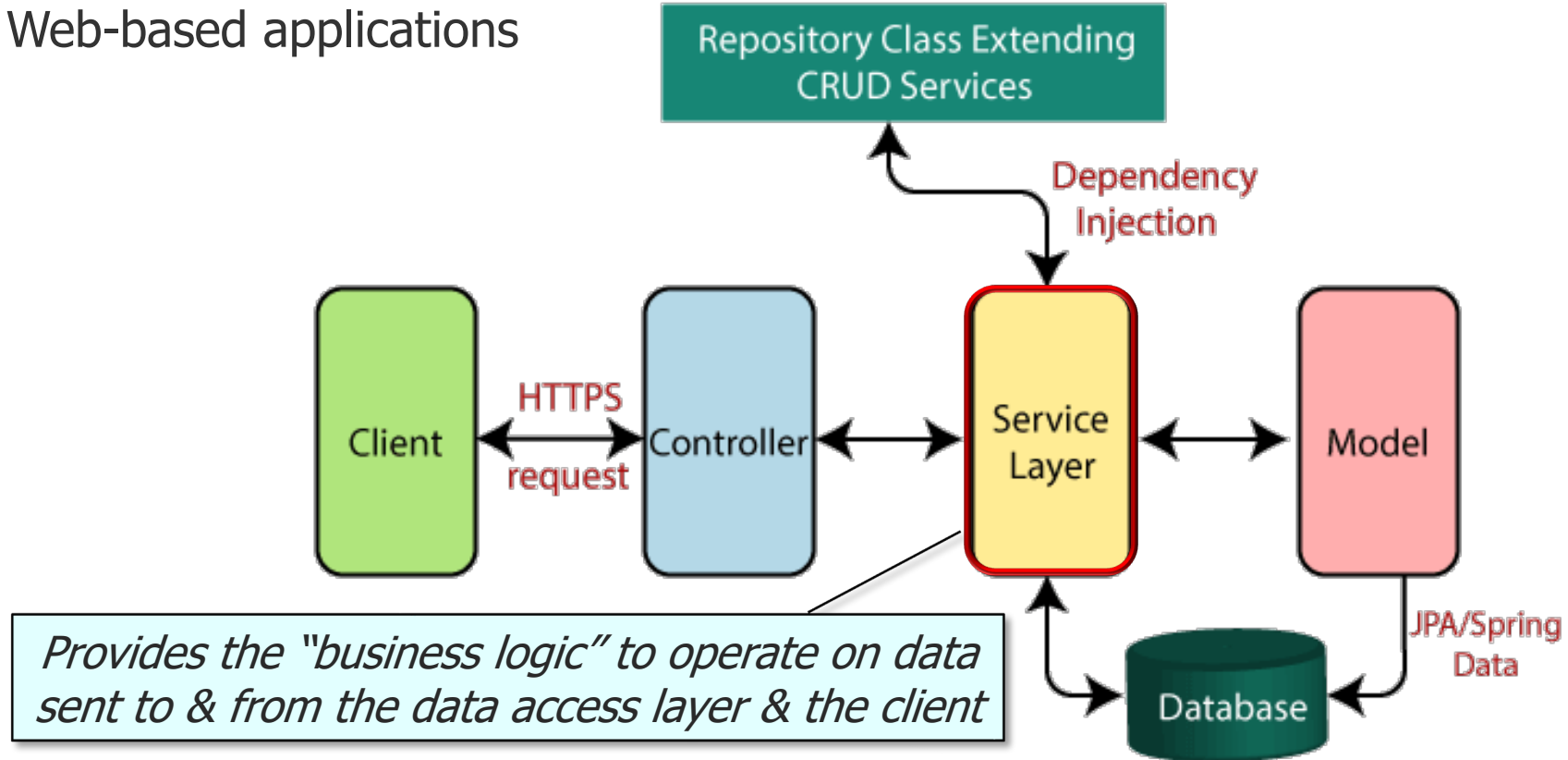
- The Spring Boot project makes it easy to create stand-alone, production-grade Web-based applications



See spring.io/guides/gs/rest-service

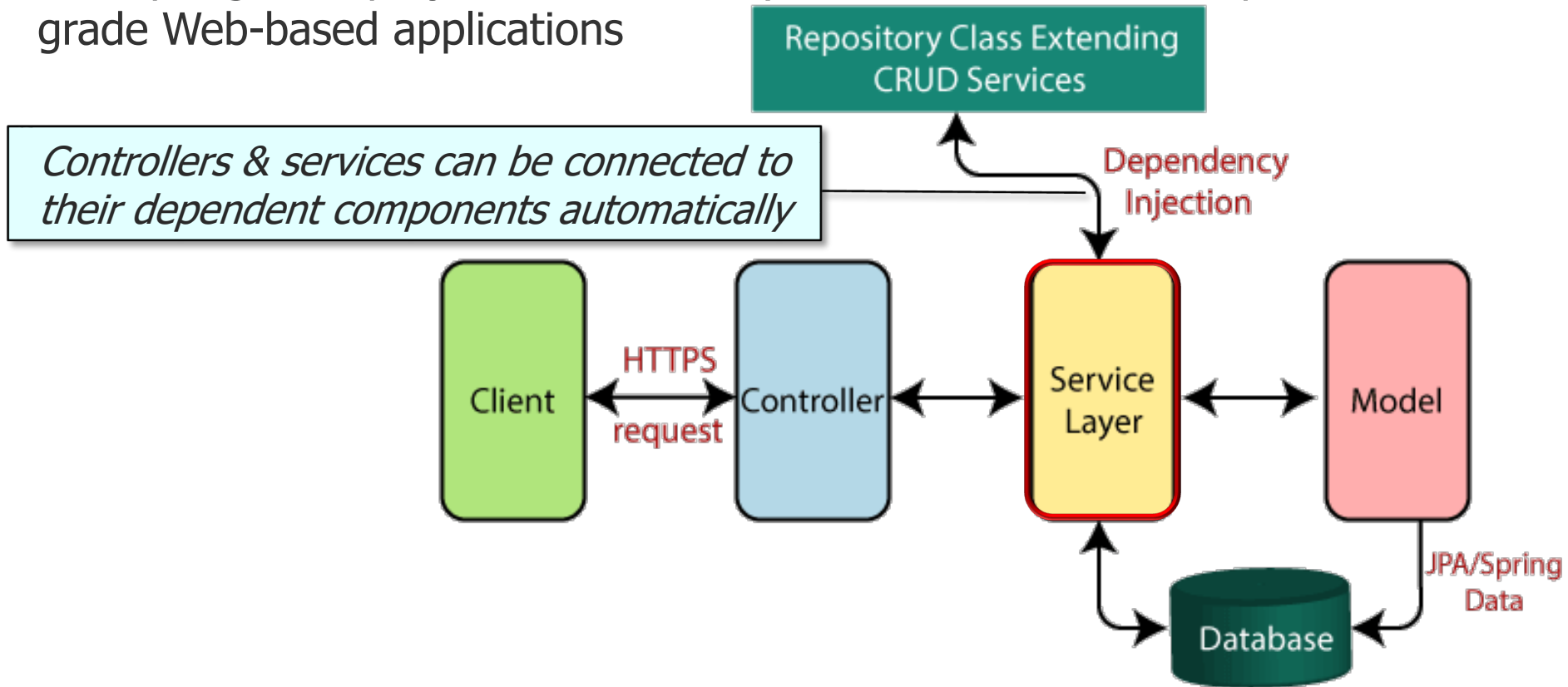
Key Components in Spring Boot

- The Spring Boot project makes it easy to create stand-alone, production-grade Web-based applications



Key Components in Spring Boot

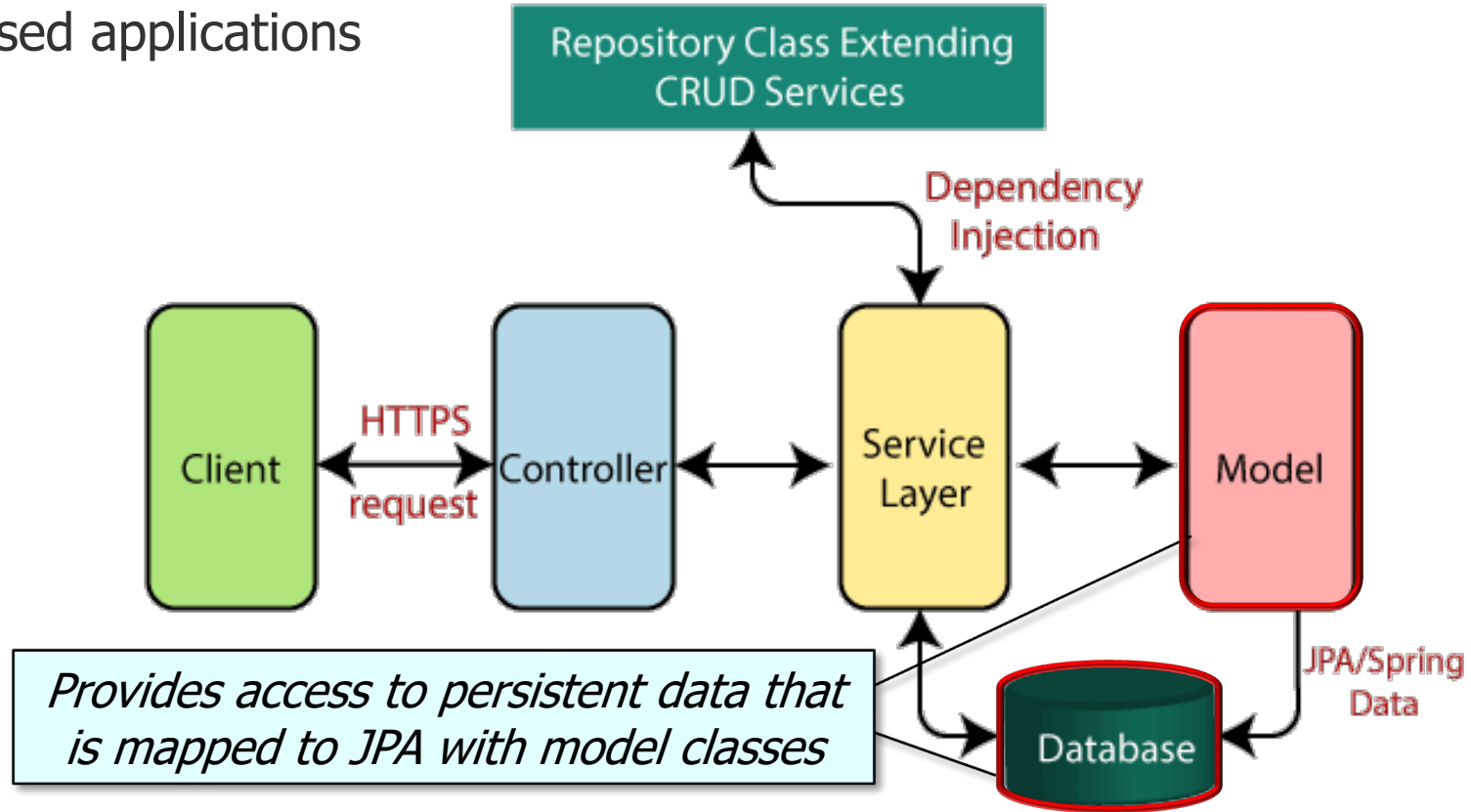
- The Spring Boot project makes it easy to create stand-alone, production-grade Web-based applications



See en.wikipedia.org/wiki/Dependency_injection

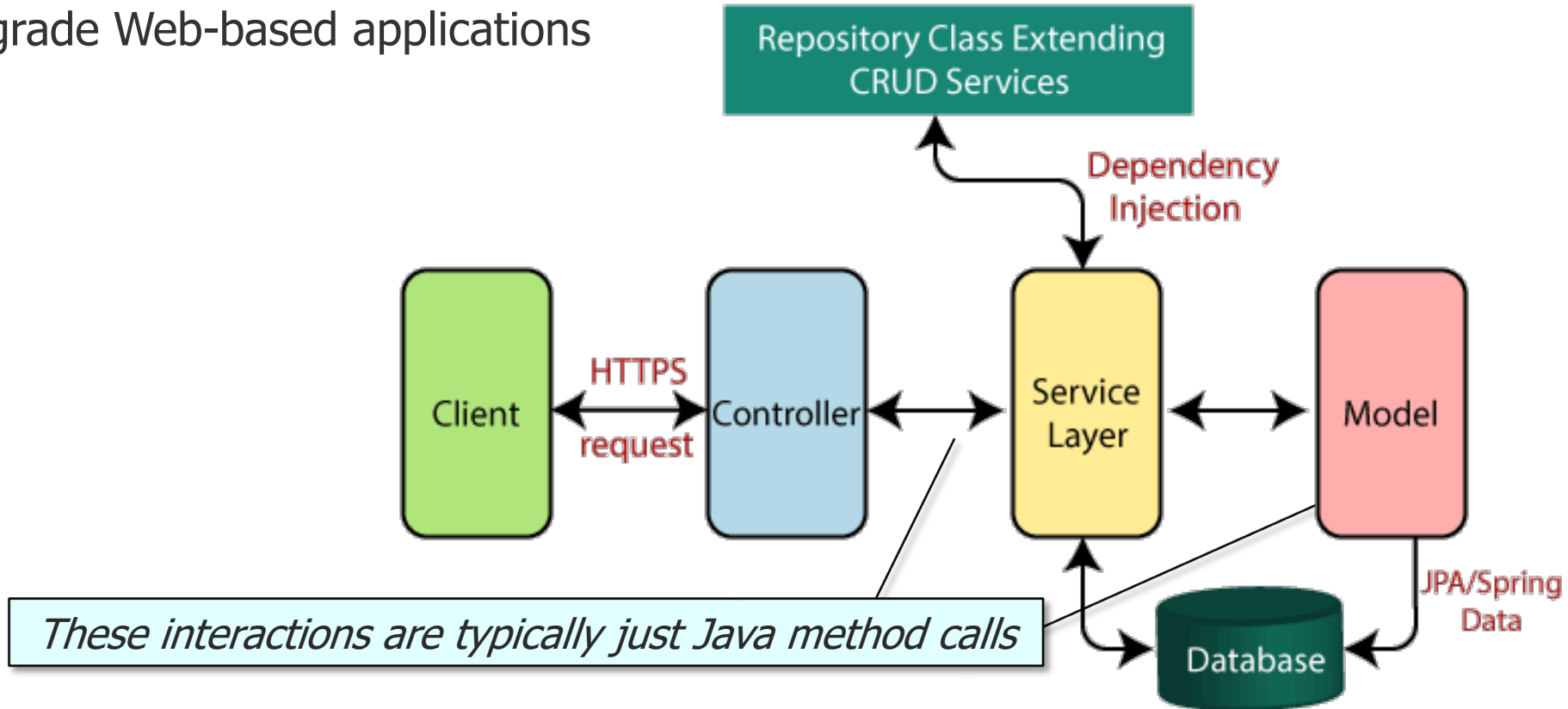
Key Components in Spring Boot

- The Spring Boot project makes it easy to create stand-alone, production-grade Web-based applications



Key Components in Spring Boot

- The Spring Boot project makes it easy to create stand-alone, production-grade Web-based applications



Overview of Microservices

Overview of Microservices

- Microservices are an architectural pattern that arranges an application as a collection of loosely coupled, fine-grained services, communicating via lightweight protocols



See en.wikipedia.org/wiki/Microservices

Overview of Microservices

- A microservice architecture decomposes an app into small, independent services that can be developed & deployed separately

Microservices



Overview of Microservices

- A microservice architecture decomposes an app into small, independent services that can be developed & deployed separately
- In contrast, a monolithic architecture builds a single, cohesive app with all functionality tightly integrated into a single codebase

Monolithic

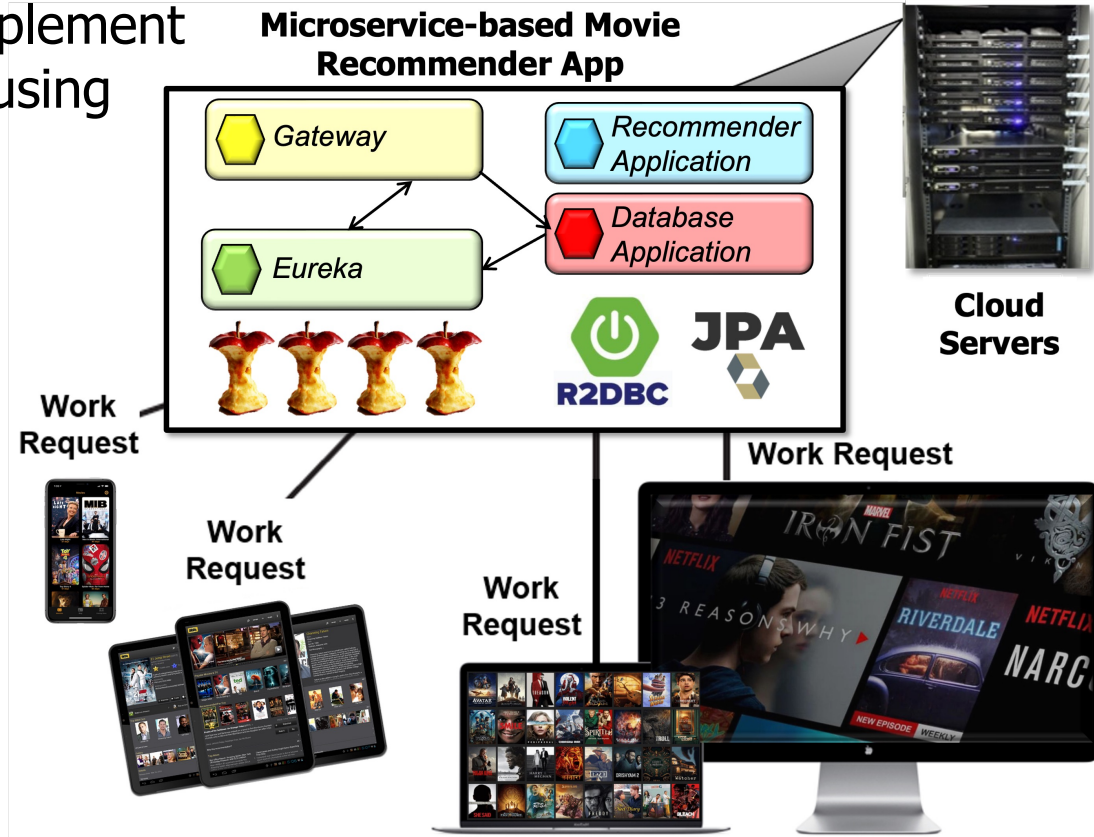


Microservices



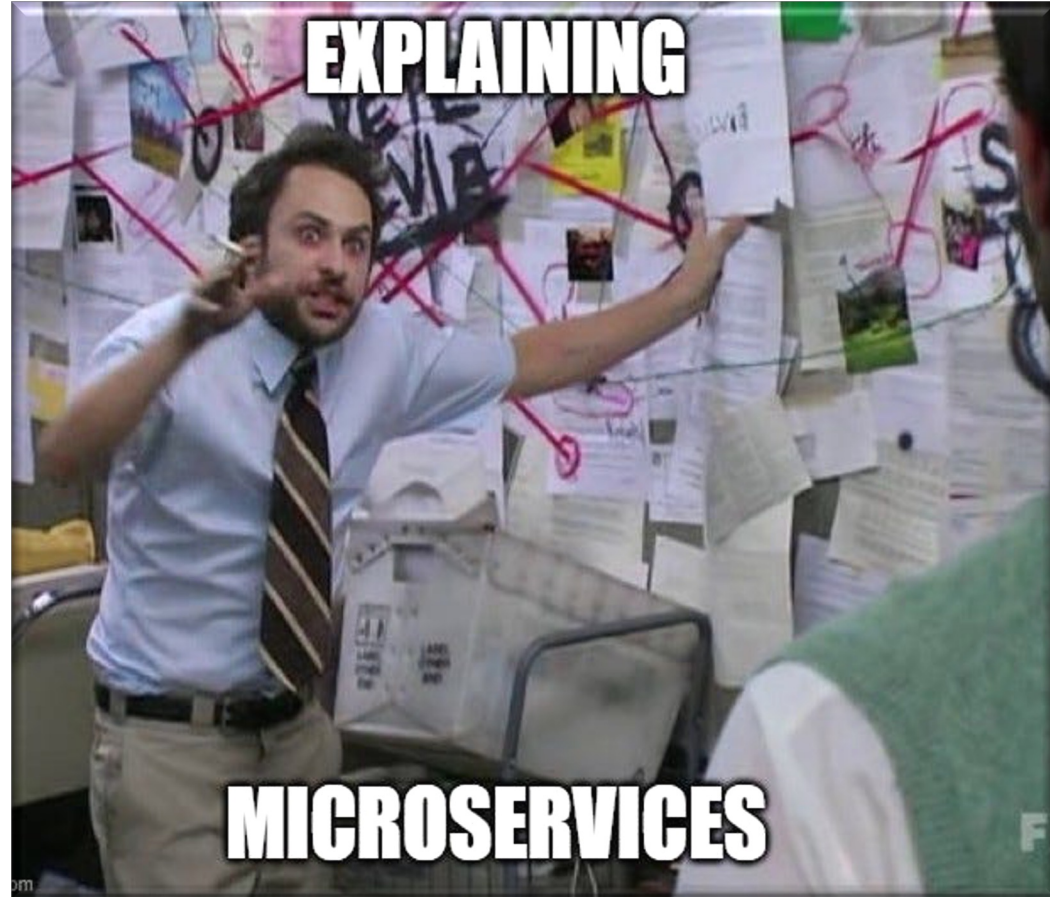
Overview of Microservices

- We'll apply microservices to implement a movie recommendation app using Spring WebMVC & WebFlux



Overview of Microservices

- There are pros & cons of apply a microservices architecture



See medium.com/@PurpleGreenLemon/was-microservices-a-bad-idea-5e52edee1cff

End of Overview of Spring & Spring Boot