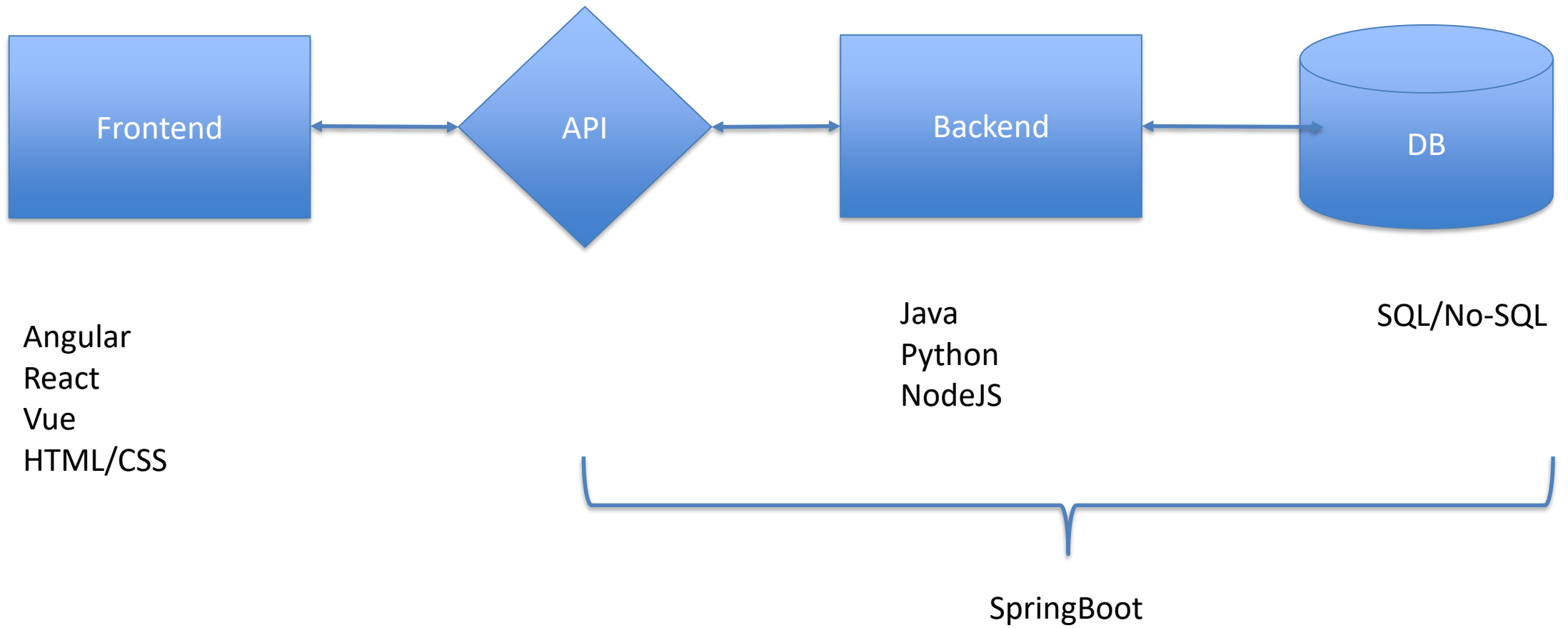




Spring Boot Overview

Frontend vs Backend

- Frontend:
 - - User Interface (HTML, CSS, JS, React, Angular, etc.)
 - - Handles client-side interactions
- Backend:
 - - Spring Boot application logic
 - - Handles requests, business rules, and database operations

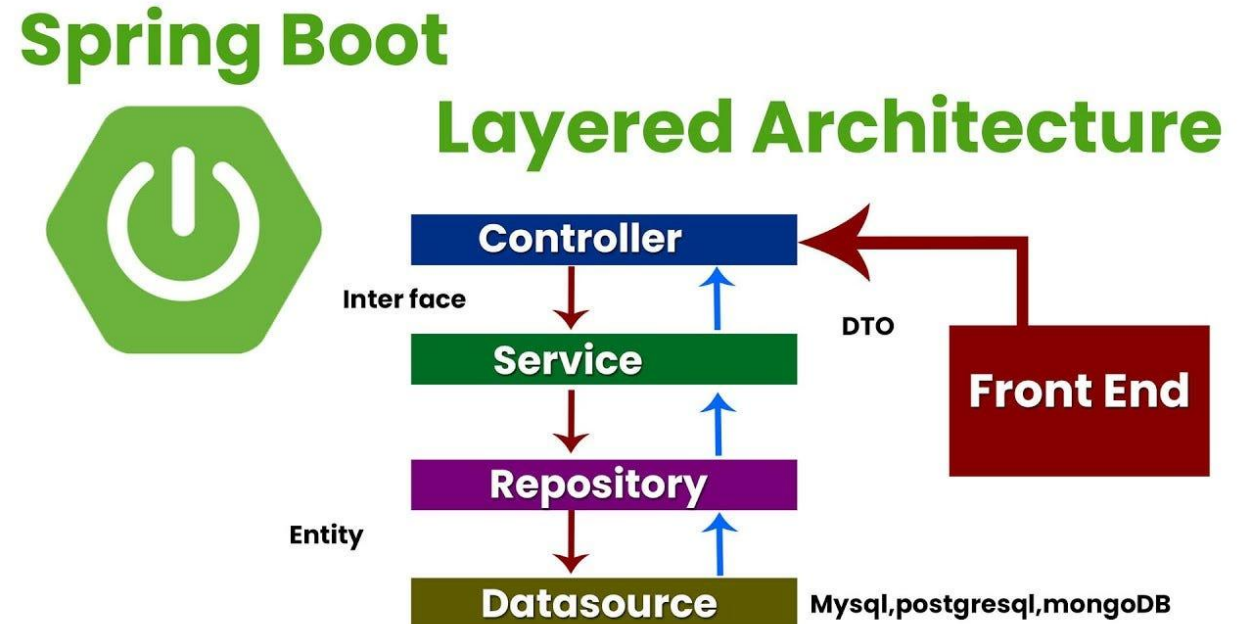


Packaging in Spring Boot

- 1. JAR (default)
 - - Self-contained, includes embedded server
 - - Run with: `java -jar myapp.jar`
- 2. WAR
 - - For deploying to external application servers
 - - Traditional enterprise usage

Spring Boot Layered Architecture

- Controller:
 - Handles HTTP requests/responses
- DTO (Data Transfer Object):
 - Transfers data between layers
- Service:
 - Contains business logic, uses repositories
- Entity:
 - Maps to DB tables (JPA @Entity)
- Repository:
 - Data access layer, CRUD operations



Common Spring Boot Annotations

- `@Entity` - Marks class as JPA entity
- `@Id` - Primary key
- `@Query` - Custom queries in repository
- `@RestController` - REST API controller
- `@RequestMapping` - Maps URL paths
- `@CrossOrigin` - Enables CORS
- `@Autowired` - Dependency injection
- `@Service` - Business logic class
- `@Transactional` - DB transaction management
- `@AllArgsConstructor` / `@NoArgsConstructor` - Lombok constructors

Database Connectivity

- `spring.jpa.hibernate.ddl-auto=update`
 - Schema handling (update, create, validate)
- `spring.datasource.url=jdbc:mysql://localhost:3306/db`
 - Database location
- `spring.datasource.username & password`
 - Credentials
- `spring.datasource.driver-class-name=com.mysql.cj.jdbc.Driver`
 - JDBC driver for MySQL
- `spring.jpa.show-sql=true`
 - Logs SQL queries