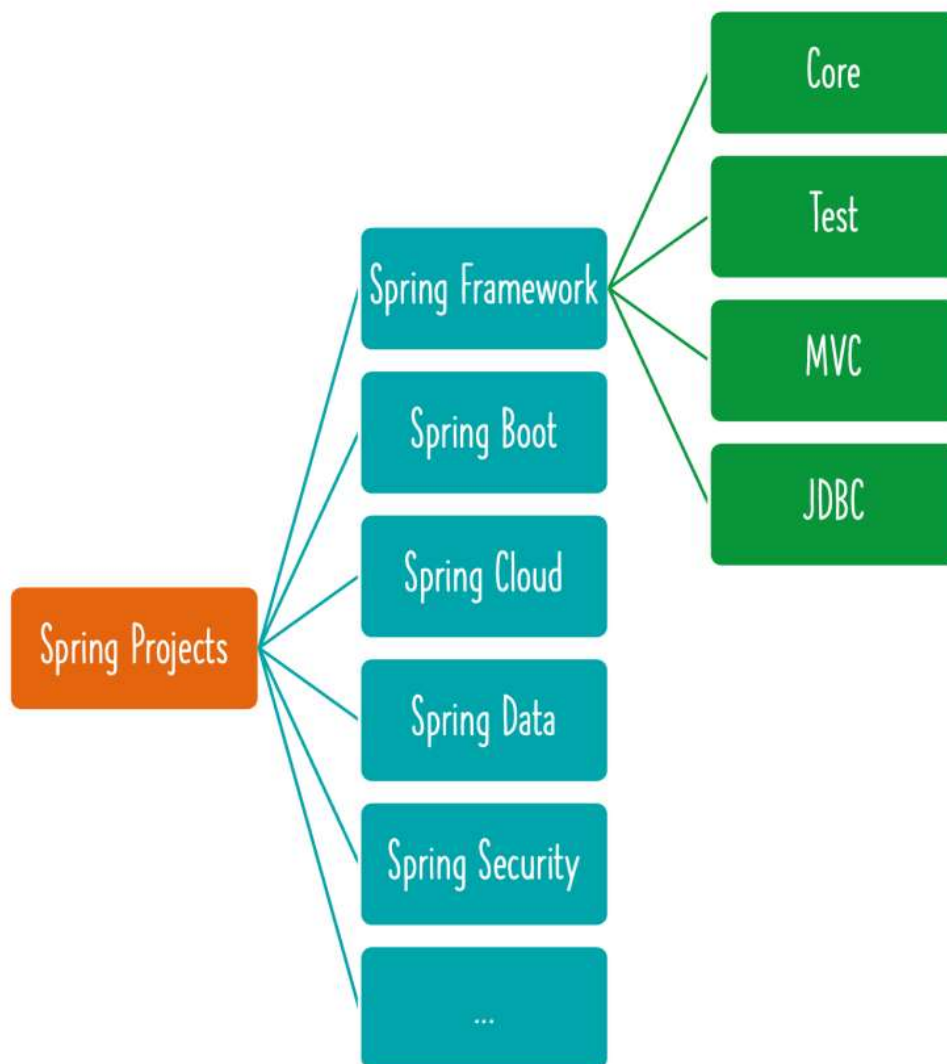


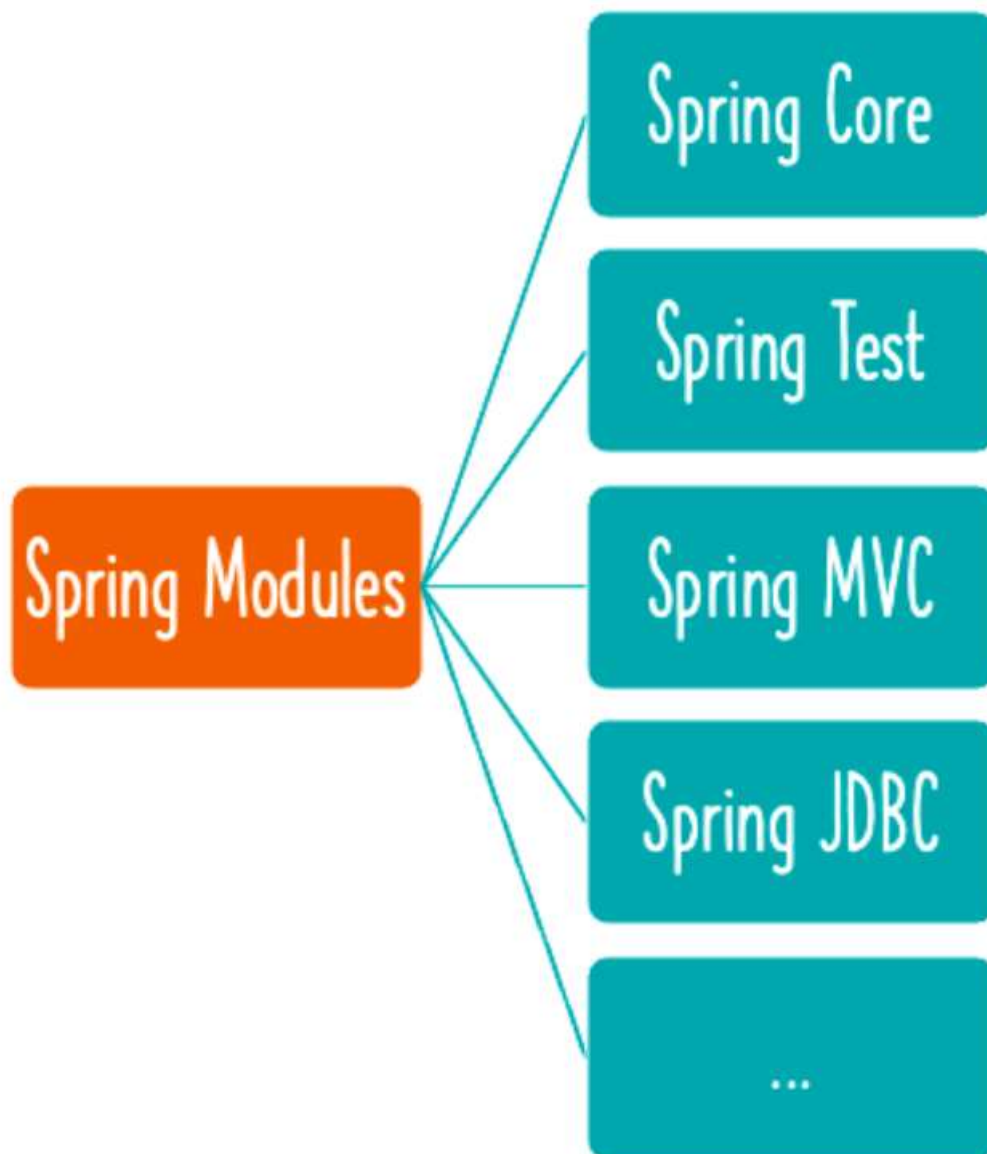
EXPLORING SPRING BIG PICTURE – FRAMEWORK, MODULES AND PROJECTS

1. Spring Core: IOC Container, Dependency Injection, Auto Wiring, ...
 - a. These are the fundamental building blocks to:
 - i. Building web applications
 - ii. Creating REST API
 - iii. Implementing authentication and authorization
 - iv. Talking to a database
 - v. Integrating with other systems
 - vi. Writing great unit tests
2. Let's now get a Spring Big Picture:
 - a. Spring Framework
 - b. Spring Modules
 - c. Spring Projects



FRAMEWORK AND MODULES

1. Spring Framework contains multiple Spring Modules:
 - a. Fundamental Features: Core (IOC Container, Dependency Injection, Auto Wiring, ...)
 - b. Web: Spring MVC etc (Web applications, REST API)
 - c. Web Reactive: Spring WebFlux etc
 - d. Data Access: JDBC, JPA etc
 - e. Integration: JMS etc
 - f. Testing: Mock Objects, Spring MVC Test etc
2. Why is Spring Framework divided into Modules?
 - a. Each application can choose modules they want to make use of.
 - b. They do not need to make use of everything in Spring framework!



SPRING PROJECTS

1. Hierarchy: Spring Projects → Spring Framework → Spring Modules
2. Why is Spring Eco system popular?
 - a. Loose Coupling: Spring manages creation and wiring of beans and dependencies.
 - i. Makes it easy to build loosely coupled applications.
 - ii. Make writing unit tests easy! (Spring Unit Testing)
 - b. Reduced Boilerplate Code: Focus on Business Logic
 - i. Example: No need for exception handling in each method!
 1. All Checked Exceptions are converted to Runtime or Unchecked Exceptions
 - c. Architectural Flexibility: Spring Modules and Projects
 - i. You can pick and choose which ones to use (You DON'T need to use all of them!)
 - d. Evolution with Time: Microservices and Cloud
 - e. Spring Boot, Spring Cloud etc!

