Introduction to Spring

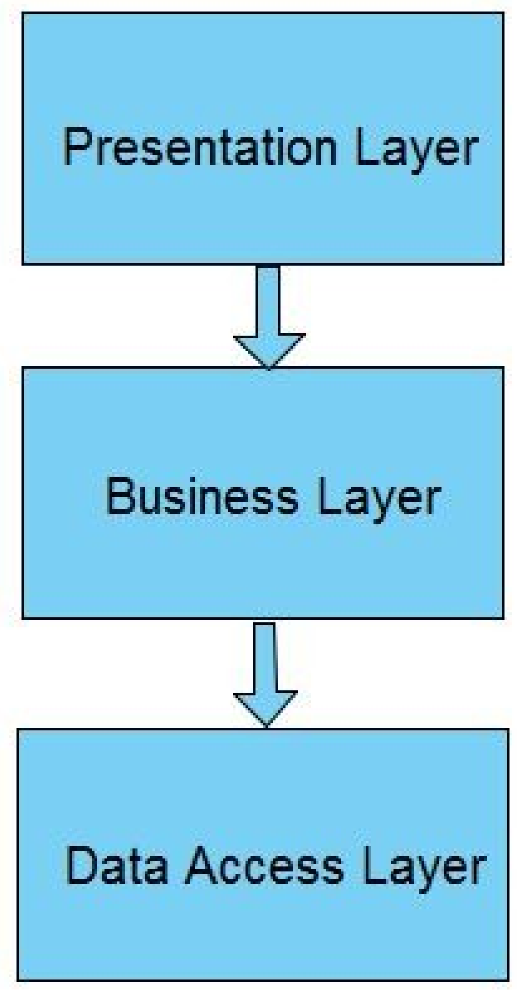
Agenda

- **What's Spring**
- **Inversion of Control and Dependency Injection**
- **Modules of Spring**
- **Demos**

What is Spring ?

- Spring is an open source enterprise framework
- Light-weight, inversion of control container
- POJO-based programing model
- Simplicity
- Productivity

Traditional Enterprise Application



Business Objects are the core

Why container?

- **Managing Business Objects is an important part in an enterprise application, especially large-scale applications**
- **Business Objects are usually Singletons**
- **Without container**
  - Many custom-coded Singleton classes – “Singleton hell”
  - Many coded factories classes
  - Many forms of self-configurations : property files, xml …
  - Applications lack of consistency and maintenance is expensive
- **With container**
  - Business Objects (Application objects) run inside a container, and are managed by the container

Lightweight container

- **Life-cycle management**
  - A container control the lifecycle of application running within it
  - Object creation,
  - Object destruction
- **Object Lookup**
  - Provides a way of obtaining references to managed objects
- **Configuration**
  - Provides a consistent way of configuring objects
- **Dependency Resolution**
  - Manage relationships between managed objects

POJO-Based Programming Model

- **Plain Old Java Objects**
  - Not extend/implement any special class, interface
- **Simple, Fast**
- **Container-dependent**
  - Easy to change containers
  - Reusable
- **Easy to test**
  - Just create an object and test it with JUnit

Inversion of Control

- **General concept of frameworks**
  - Framework vs library?
- **Code written is called by frameworks**
  - Holywood principle - “don't call us, we call you”
- **Plugable and modular architecture**
  - Just write your code/component and plug it in and leave the rest for the container

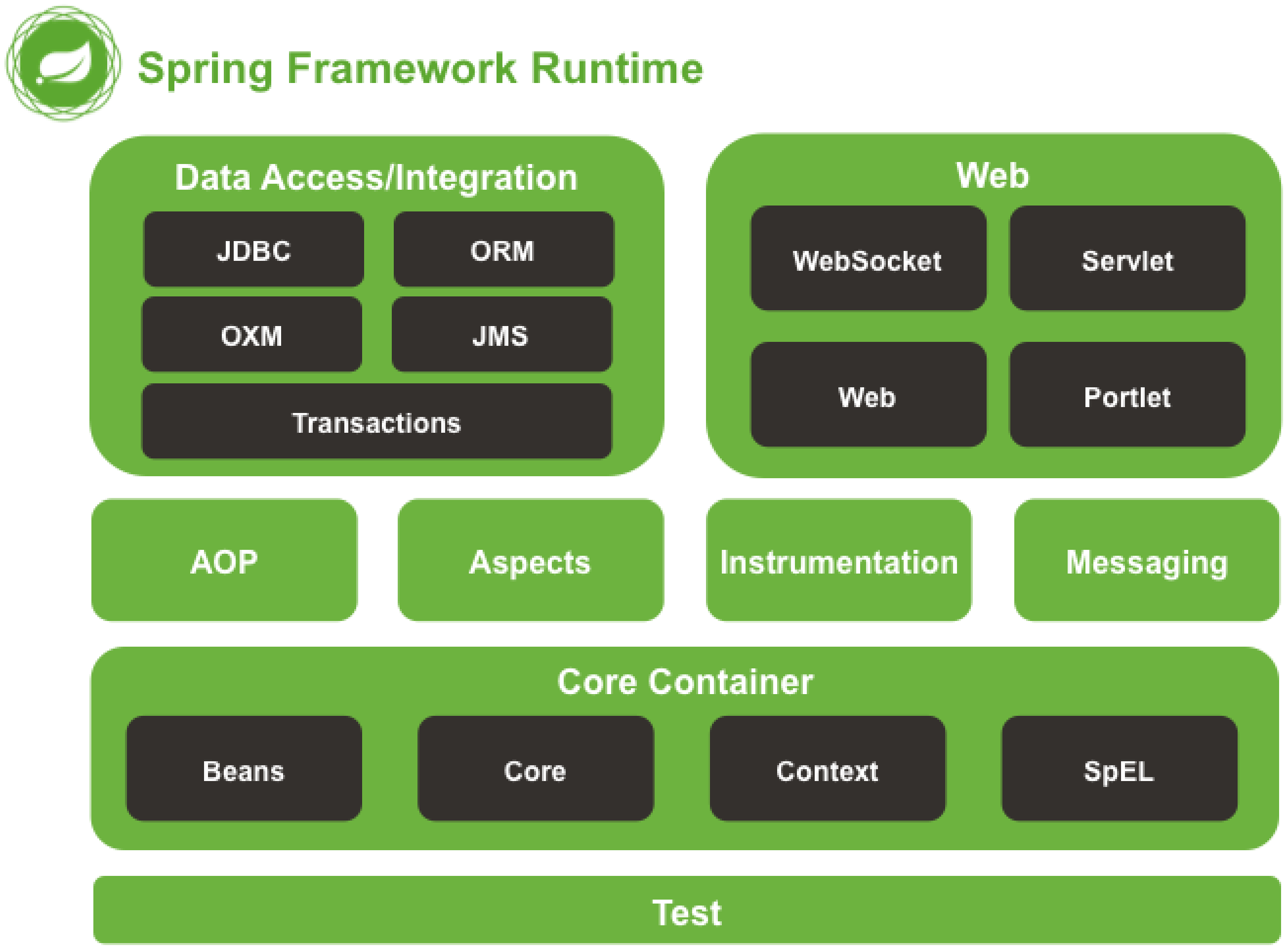
Inversion of Control in Action

Dependency Injection

- A type of Inversion of Control
- Automatically resolve dependencies
- Eliminate code for looking up dependencies (objects)

Dependency Injection in Action

Spring Framework Modules



Most important parts of Spring

- **Dependency Injection**
- **Aspect-Oriented Programming**
  - Handles cross-cutting concerns
- **Data Access**
  - ORM
  - Transaction management
- **Web**
  - MVC
  - REST

Dependency Injection

- **MaiVT**
-

Demo