

# Spring: what is it and why do I use it?

developing web applications using free open source software

Contact: [derek@derekfrost.co.uk](mailto:derek@derekfrost.co.uk)

# What is Spring? (briefly)

- Spring is an application development framework for Java (a version is also available for .NET)
- It contains prewritten code to do the tedious bits, leaving you free to work on high value code
- You can pick and mix its components and, if needs be, integrate other frameworks to build a complete application
- Used for data processing and database oriented web applications (Banks use it including BACS)
- It's free and open source, licensed under Apache 2

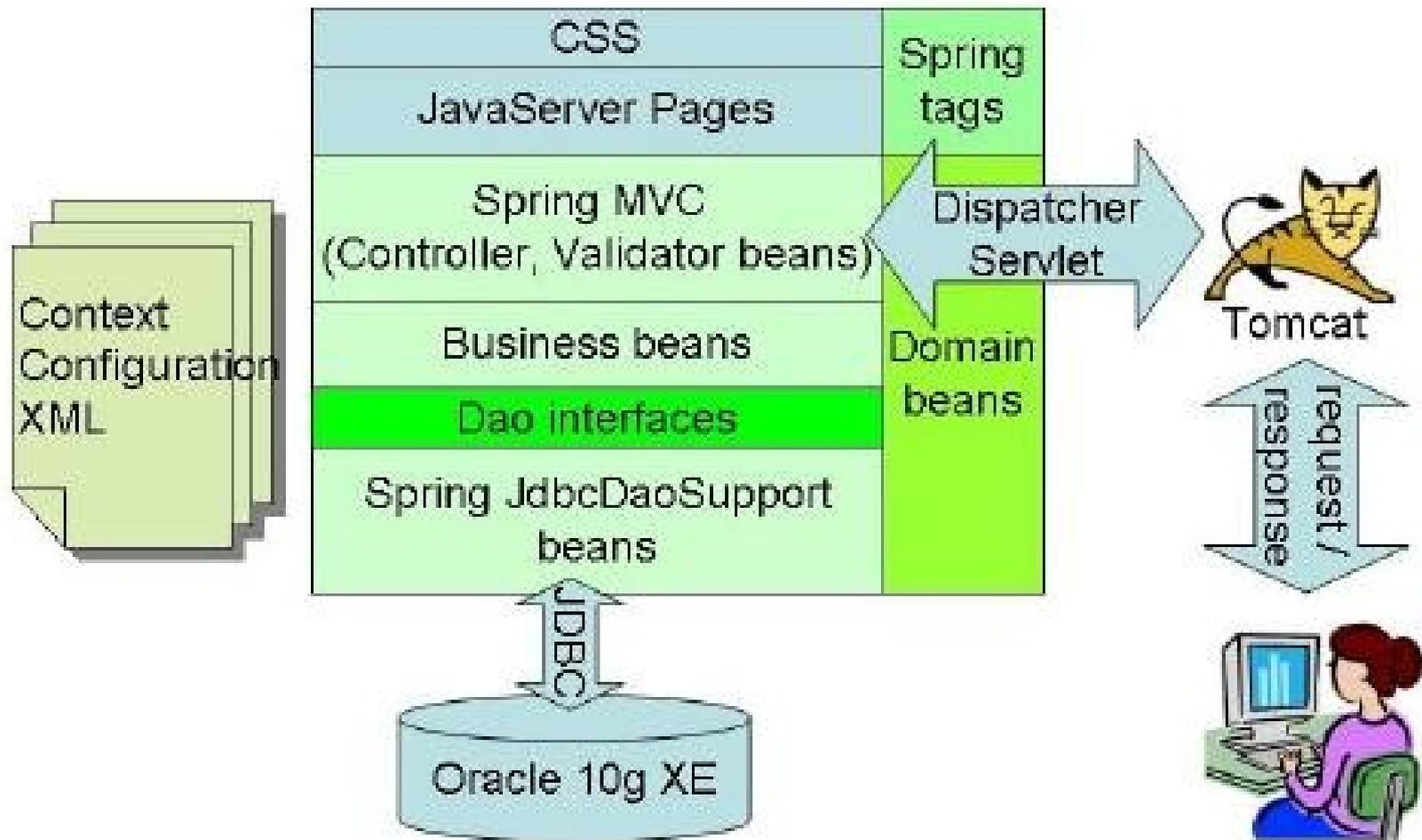
# Why do I use it? - Java (JVM)

- Bewildering choice of technologies (PHP, .NET, Java, Ruby, Groovy etc) – where do I start?
- Where do I want my software to run (platform) and what do I want it to do (framework)?
- Chose Java (JVM) because it does everything I need to do on all of the platforms I need to do it on (for free!)
- Also easier to install on Windows than .NET, while Linux support allows a wider choice of hosting options
- However I needed to avoid bloated slow frameworks and any special hosting requirements
- I wanted to kick Java EE into the long grass if possible

# Why do I use it? - Spring Framework

- Avoids the need for bloated slow frameworks (even the Application server), it's lightweight, fast and everything you need is built into the application
- Kicks Java EE into the long grass by allowing you to develop scalable applications without it
- Provides a powerful but highly portable data processing container that can be “rewired” to suit a range of hosting options (using IOC, annotation, AOP, declarative)
- Produce applications with simple and flexible hosting requirements (choose cheap / scalable / reliable hosting)
- Add emerging UI technologies to existing business logic (MVC, JSON, SOAP, CORBA, GWT, iOS/Android?...)

# A Spring Web Application



# Why do I use it? - Tomcat & MySQL

- Affordable Java hosting uses Tomcat and MySQL
- Tomcat is simpler to deploy and uses fewer resources than a full application server (1/3 memory, 1/4 start time)
- Spring allows any required libraries and frameworks to be added directly to your project and provides a managed container to host them (life cycle management etc.)
- The single .WAR file produced can be deployed to a default Tomcat configuration, simplifying installation
- MySQL proved more reliable than other databases I tried and was better supported by third party tools
- A full application can be installed on Windows in under 10 minutes with no restart needed (one click installer?)

# Questions?

Thank you for listening

Contact: [derek@derekfrost.co.uk](mailto:derek@derekfrost.co.uk)

Some useful web sites:

<http://www.springsource.com/developer/spring>

<http://tomcat.apache.org/>

<http://www.oracle.com/technetwork/java/javase/downloads/index.html>

<http://www.mysql.com/>