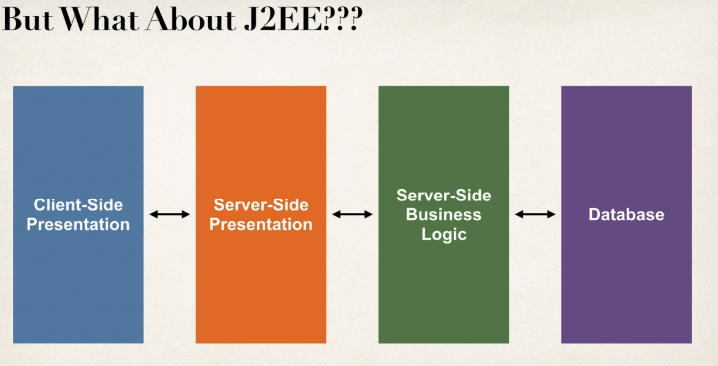
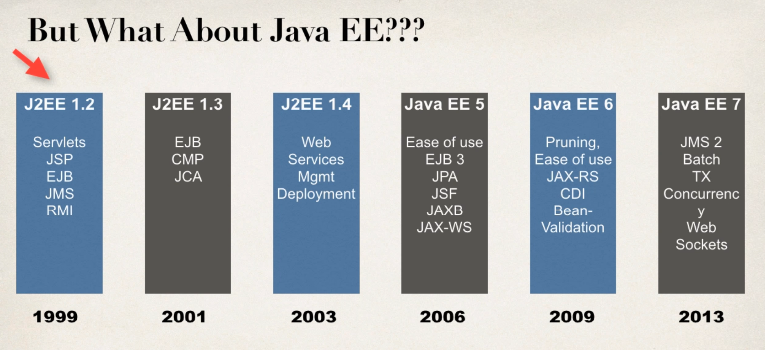
**Spring Framework**

Q. Why Spring?

> Spring is a very popular framework for building Java applications.

> Spring was initially a simpler & lightweight alternative to J2EE i.e. provides a large no. of helper classes… makes things easier.





JSP – Jakarta Server Pages / JavaServer Pages

EJB – Jakarta Enterprise Beans / Enterprise JavaBeans

JMS – Java Message Service

RMI – Remote Method Invocation

CMP – Container Managed Persistence

JCA – Java Connector Architecture

JPA – Java Persistence API

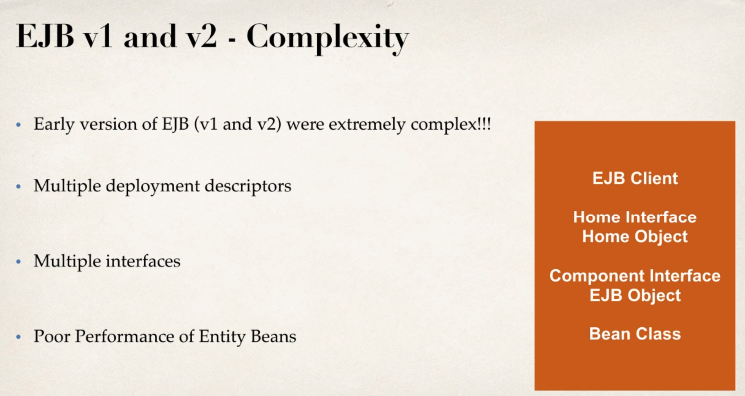
JSF – Java Server Faces

JAXB – Java API for XML Binding

JAX-WS – Java Web Services (SOAP)

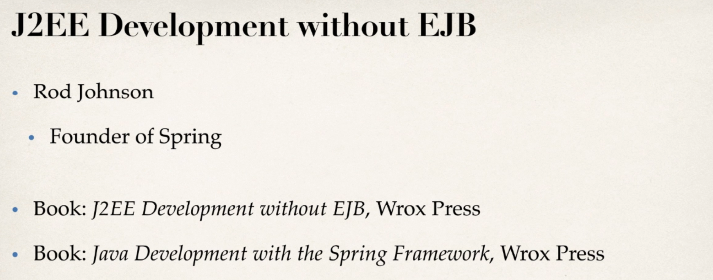
JAX-RS – Java Web Services (REST)

CDI – Context Dependency Injection (IoC)

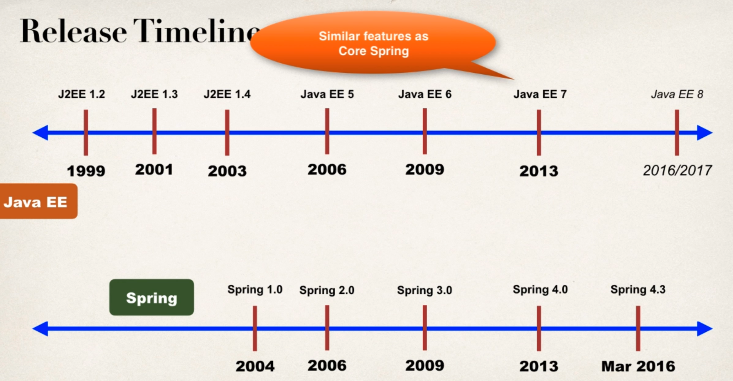


> Entity Beans basically mapping between java class & Database table are just awful slow. Even on one deployment, we actually had to pull our code back out of the production & actually remove the EJB functionality because it slowed everything down.

That’s a lot of developers started to continue to do J2EE but they actually would do it without Enterprise JavaBeans.



Release Timeline of J2EE & Spring

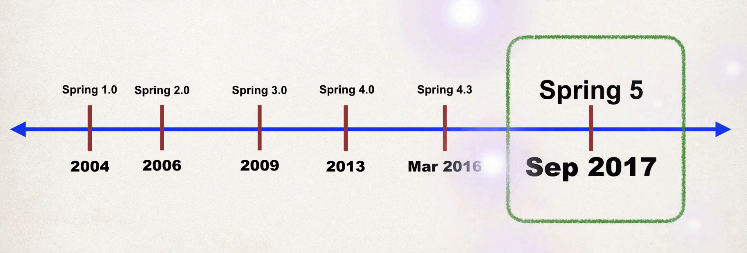


EJB 3.1 in Java EE 6 was much easier to use i.e. in J2EE 6, added CDI Context Dependency Injection IoC.

So from **Java EE 7**, you can do everything as **Spring** can do but the only problem though is that they were just a little bit too late & unfortunately, EJB just kind of has a bad name & also, Spring has huge momentum, huge market share.

> Spring is a lightweight framework, simple, easy to use & a lot of developers like it.

What’s New in Spring 5?



> Updated minimum requirements for Java 8 or higher.

> Deprecated legacy integration for: Tiles, Velocity, Portlet, Guava etc.

> Upgraded Spring MVC to use new versions of Servlet API 4.0

> Added new reactive programming framework: Spring WebFlux

**Spring Framework Overview**

> Spring official Website: - www. spring.io

**> Goals of spring**

a) Lightweight development with Java POJOs (Plain – Old – Java – Objects))

[Make it much simpler to build, as compared to the heavyweight EJBs from the early versions of J2EE]

b) Dependency injection to promote loose coupling

[So instead of hard wiring your objects together, you simply specify the wiring via a configuration file or annotations.]

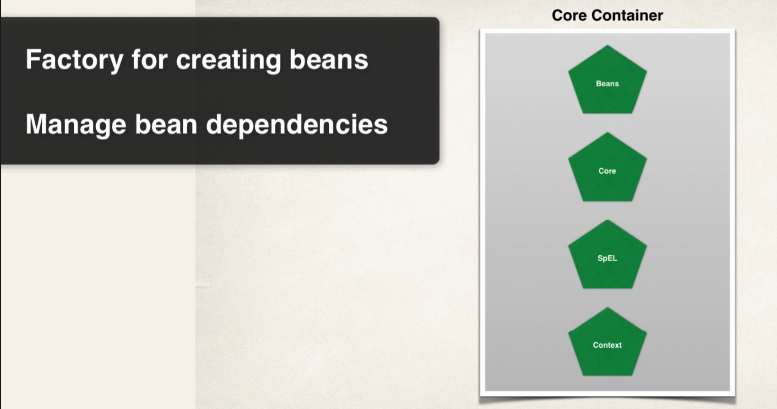
c) Declarative programming with Aspect – Oriented – Programming (AOP)

[Basically allow you to add some application wide services to your given objects]

d) Minimize boilerplate Java code.

[In early days of J2EE, there was a lot of code that you had to write, so the folks at Spring created a collection of helper classes to make it easier.]

**Spring Core**



> So the core container is like heart of Spring framework.

> It basically manages how beans are created.

> It has a bean factory for creating the beans.

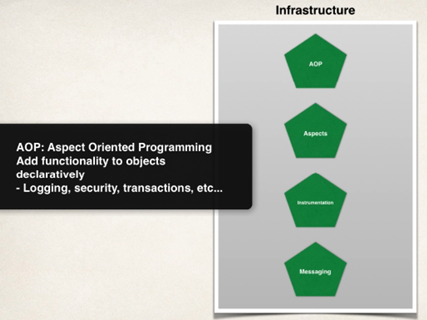
> It basically can read configuration files for setting properties in dependencies.

> Also the Context here is really the spring container that holds the Beans in memory.

> Spring Expression Language (SpEL) is a language we can use within the configuration files to refer to other beans.

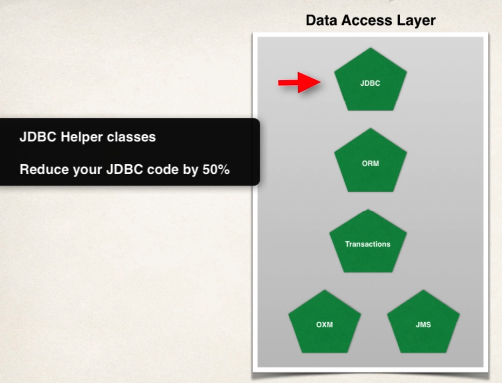
**Spring AOP**

**- AOP, Aspects, Instrumentation, Messaging**



> **Spring AOP** allows you to create application wide services, like logging, security, transactions, instrumentation & then you can apply these services to your objects in a declarative fashion, so no need to modify your code to have support for this. You simply add a configuration in the configuration file or an annotation and that service will be applied to your application.

**Spring Data Access Layer**



> **Spring Data access layer** is for communicating with the database, either a relational database or a NoSQL database and also making use of a message queue.

> Basically, Spring provides some helper classes to make it much easier to access a database, using JDBC & by using these Spring JDBC classes, you can actually reduce your source code by over 50%

> **ORM** (Object to Relational Mapping) is probably the most popular section of this module. Basically, it allows you to hook into Hibernate, or hook into JPAs.