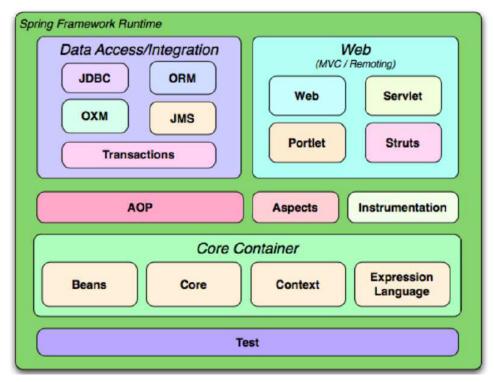
Memento Spring Framework

Une vingtaine de modules, 6 briques...

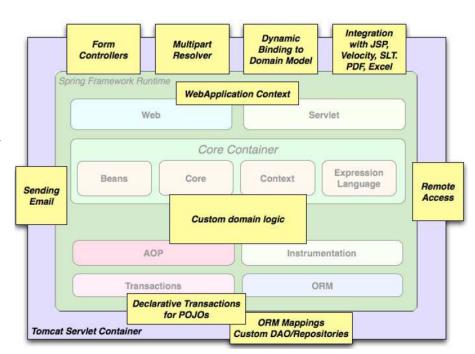
- Core Container (BeanFactory...)
- AOP : Aspect Oriented Programmation (fonctions transverses)
- Data Access / Data Integration (JDBC, accès aux données, gestion connexions et gestion des erreurs)
- Instrumentation
- Test
- Web



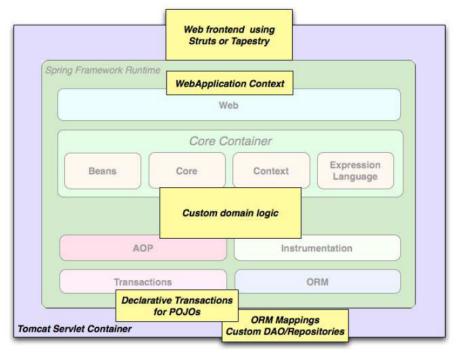
Vue d'ensemble du Framework Spring

L'inversion de Contrôle (IoC) : «Ne nous appelez pas, nous vous appellerons!»

- -- Container configurable
- Injection des dépendances
- Simplification et reconfiguration externe



Usages des éléments du Framework Spring



Usage du Framework Spring avec Hibernate

Conteneur léger de Spring : — Gestion des objets applicatifs (Beans) et de leur cycle de vie — Contexte d'application — Annotations

- Objects / Classes importantes :
 - Bean (POJO : Plain Old Java Object), géré par le conteneur
 - BeanFactory (accès conteneur, peut être créée)
 - ApplicationContext (fabrique et registre Bean's), étend BeanFactory, contexte
 - méthodes fabriques de beans
 - support messages et internationalisation
 - support chargement fichiers locaux / distants
 - génération évènements pour les beans
 - hiérarchisation informations disponibles pour les beans

Fichier de Configuration (XML):

```
— Accès via contexte
```

— BeanFactory ctx =
 new ClassPathXmlApplicationContext("spring.xml");

```
<?xml version="1.0" encoding="UTF-8"?>
<beans xmlns="http://www.springframework.org/schema/beans"
    xmlns:xsi="http://www.w3.org/2001/XMLSchema-instance"
    xmlns:p="http://www.springframework.org/schema/p"
    xmlns:aop="http://www.springframework.org/schema/aop"
    xmlns:lang="http://www.springframework.org/schema/lang"
    xmlns:context="http://www.springframework.org/schema/context"
    xsi:schemaLocation="http://www.springframework.org/schema/beans
    http://www.springframework.org/schema/beans.xsd">
```

Quelques exemples de configuration :

</beans>

```
<context:annotation-config />
<context:component-scan base-package="package.impl.beans" />
```

```
<bean class="package.to.class.ParentTraceur" />
<bean name="logAd" class="package.aop.Traceur" />
<aop:config>
    <aop:pointcut id="sondesPointCut"</pre>
        expression="execution(int package.impl.*.getValeur())" />
    <aop:advisor advice-ref="logAd" pointcut-ref="sondesPointCut" />
</aop:config>
<aop:aspectj-autoproxy />
<bean id="traceurAspect" class="package.aop.TraceurAspect" />
<bean name="messageSource"</pre>
        class="org.springframework.context.support.ResourceBundleMessageSource">
    cproperty name="basename" value="messages" />
</bean>
<bean id="beanEmetteur"</pre>
        class="package.impl.EmetteurAlerteConsoleSansRepetition"
        scope="prototype" />
<!-- scope="prototype||singleton" -->
<bean id="beanSonde" class="package.impl.SondeCompteurFichier">
    cproperty name="repertoire" value="c:/temp" />
</bean>
<bean id="beanSondeUsage" class="package.impl.SondeUsageDisque">
    cproperty name="disque" value="c:" />
</bean>
<bean id="beanSondeDetect" class="package.impl.SondeDetectionFichier">
    <constructor-arg value="c:/temp/test.txt" />
    <constructor-arg value="true" />
</bean>
<bean id="surveillanceDefaut"</pre>
        class="package.spring.SystemeDeSurveillance"
        abstract="true">
    cproperty name="frequence" value="5" />
    cproperty name="seuil" value="20" />
</bean>
<bean id="beanSondeTemp" class="package.spring.impl.SondeFactory"</pre>
    factory-method="creerSondeTemp" />
<bean id="beanSondeDisque1" class="package.spring.impl.SondeUsageMultiDisques">
    cproperty name="disques">
        <array>
            <value>C:</value>
            <value>E:</value>
```

```
</array>
       <!-- same for list, set, map, props -->
    </bean>
<bean id="beanSondeDisque2" class="package.spring.impl.SondeUsageMultiDisques">
    cproperty name="disques" value="C:,E:" />
   <!-- only values (not references !) -->
</bean>
<!-- <bean id="beanSurveillance" parent="surveillanceDefaut" >
    cproperty name="seuil" value="1" />
    cproperty name="sonde" ref="beanSondeDisque2" />
    cproperty name="emetteur" ref="beanEmetteur" />
</bean> -->
<!-- <bean id="beanSurveillance"
   parent="surveillanceDefaut"
   p:seuil="1"
   p:sonde-ref="beanSondeDisque2"
   p:emetteur-ref="beanEmetteur" /> -->
<bean id="beanSurveillance"</pre>
   parent="surveillanceDefaut"
   p:seuil="1"
   init-method="demarrer"
   destroy-method="arreter" />
    <!-- autowire="byName" -->
<alias name="beanSondeDisque2" alias="sonde" />
<alias name="beanEmetteur" alias="emetteur" />
<lang:groovy id="sondeGroovy"</pre>
       script-source="testSonde.groovy" refresh-check-delay="5">
    <lang:property name="message" value="Passage dans Groovy" />
</lang:groovy>
<bean id="beanSurveillanceMultiple"</pre>
       class="package.spring.SystemeDeSurveillanceMultiple">
    cproperty name="surveillances">
       st>
            <ref bean="beanSurveillance" />
            <bean parent="surveillanceDefaut">
                cproperty name="sonde" ref="sondeGroovy" />
                cproperty name="emetteur" ref="beanEmetteur" />
           </bean>
       </list>
    </bean>
```