

10 juin 2008



Paris
JUG

www.parisjug.org

www.parisjug.org



Xebia

aneo
the other solution

valtech

BK Consulting

OceaneConsulting
Prestation Java





10 juin 2008

Les nouveautés Spring « à la source »

Julien Dubois /
Michaël Isvy
SpringSource



www.parisjug.org



Copyright © 2008 ParisJug. Licence CC – Creative Commons 2.0 France – Paternité – Pas d'Utilisation Commerciale – Partage des Conditions Initiales à l'Identique





« For years, we've created great technology. Today we're creating more than ever. [...]»

Please challenge us as we challenge the status quo. »

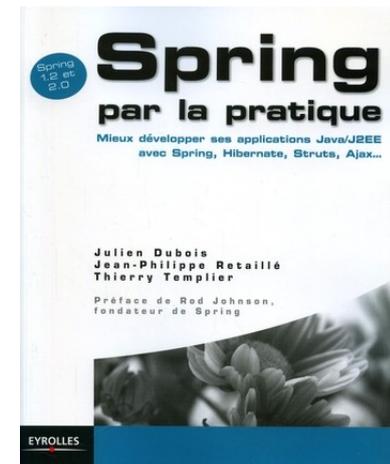
-- Rod Johnson

Intervenant #1



Julien Dubois

- Directeur Régional France de SpringSource
- Co-auteur de « Spring par la pratique » (Eyrolles, 2006)
- 10 ans d'expérience dans le développement, dont les 8 dernières en Java
- Membre de l'OSSGTP



Intervenant #2



• Michaël Isvy

- Consultant senior et responsable du pôle formation chez SpringSource France
- Contributeur sur Spring Security
- Consultant Java / J2EE et formateur depuis 8 ans
 - A formé plus de 500 personnes aux technologies Java
 - Spring, Hibernate, JSF, Maven, Websphere, Tomcat...



Sommaire (1/2)

- **Les projets Spring**
 - Tour d'horizon des projets Spring
 - Les dernières nouveautés
- **Spring Security 2.0**
 - Présentation de Spring Security
 - Mise en place de Spring Security

Sommaire (2/2)

- **SpringSource Application Platform**
 - OS-JEE-Aïe ??
 - Les fonctionnalités
 - Plus que de la glue
 - La polémique
 - L'avenir

Les projets Spring

- **Tour d'horizon des projets Spring**
- **Les dernières nouveautés**

Tour d'horizon des projets

The Spring Portfolio

Spring Rich
Client

Spring WebFlow

Spring MVC

Spring Dynamic
Modules

Spring Web
Services

Spring Security

Spring
Framework

Spring IDE

Spring Batch

Spring .NET

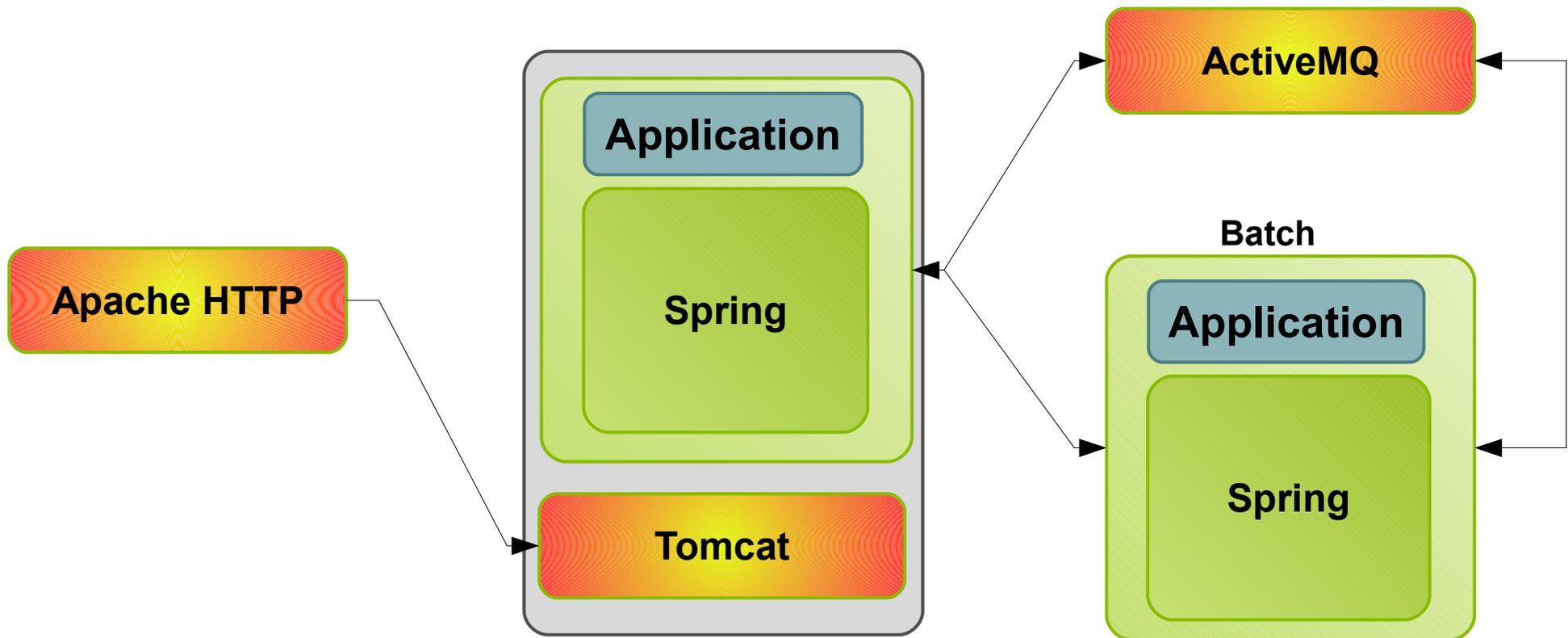
AspectJ

Spring
Integration

Spring LDAP

Tour d'horizon des projets

Les projets Spring vont clairement bénéficier du rachat de Covalent par SpringSource



Les dernières nouveautés

• Beaucoup de nouveautés!!

- Spring 2.5
- Spring.NET 1.1
- Spring Security 2.0
- Spring Web Flow 2.0
- Spring Batch 1.0
- Spring Web Services 1.5
- Spring Integration
- Spring Dynamic Modules 1.0
- Spring IDE 2.0
- AspectJ 1.6



Les dernières nouveautés

- **S'il ne faut en retenir qu'une :**
 - La configuration par annotations

```
<bean id="monBean" class="org.parisjug.Bean">  
  <property name="beanInjecte" ref="beanInjecte" />  
</bean>
```

+

```
private BeanInjecte beanInjecte;  
  
public void setBeanInjecte(BeanInjecte beanInjecte) {  
    this.beanInjecte = beanInjecte;  
}
```

=

```
@Autowired  
private BeanInjecte beanInjecte;
```

Spring Security 2.0

- **Présentation de Spring Security**
- **Mise en place de Spring Security**

Spring Security

- Anciennement appelé Acegi Security
- L'un des projets les plus populaires du portfolio Spring
- **Contributeurs principaux :**
 - Ben Alex (Australie)
 - Luke Taylor (UK)

Spring Security

- **Beaucoup de contributeurs occasionnels**
 - **Employés SpringSource ou externes**
 - **Déclaration de bugs dans JIRA**
 - **Contribution au forum**
 - **Contribution au code**



Toute contribution externe est bienvenue !

si vous êtes intéressés :

michael.isvy@springsource.com

Spring Security et la norme JEE

- **Quelques avantages de Spring Security :**
 - **Portabilité immédiate sur tous les serveurs d'applications**
 - **Gestion des cookies d'authentification**
 - **Possibilité de bloquer 2 authentifications simultanées**
 - **...**

Spring Security et la norme JEE

- **Spring Security n'est pas une extension de la norme**
 - **Mais la syntaxe Java utilisée par Spring Security est compatible avec la norme**
 - **ex: `request.isUserInRole(...)`**

Spring Security

ACEGI Security 1.0

Déjà beaucoup plus que la norme J2EE

Configuration complexe (souvent réservée aux experts)



Spring Security 2.0

Configuration repensée (namespace xml)

Beaucoup de nouvelles fonctionnalités (support OpenId, syntaxe AspectJ...)

No more dead fairies

Configuration min. Acegi

```
<bean id="filterChainProxy" class="org.acegisecurity.util.FilterChainProxy">
  <property name="filterInvocationDefinitionSource">
    <value><![CDATA[
      CONVERT_URL_TO_LOWERCASE_BEFORE_COMPARISON
      PATTERN_TYPE_APACHE_ANT

      /**=httpSessionContextIntegrationFilter,logoutFilter,authenticationProcessingFilter,basicProcessingFilter,securityContextHolderAwareRequestFilter,rememberMeProcessingFilter,anonymousProcessingFilter,switchUserProcessingFilter,exceptionTranslationFilter,filterInvocationInterceptor
    ]]></value>
  </property>
</bean>

<bean id="authenticationManager" class="org.acegisecurity.providers.ProviderManager">
  <property name="providers">
    <list>
      <ref local="daoAuthenticationProvider"/>
      <ref local="anonymousAuthenticationProvider"/>
      <ref local="rememberMeAuthenticationProvider"/>
    </list>
  </property>
</bean>

<bean id="jdbcDaoImpl" class="org.acegisecurity.userdetails.jdbc.JdbcDaoImpl">
  <property name="dataSource"><ref bean="dataSource"/></property>
</bean>

<bean id="passwordEncoder" class="org.acegisecurity.providers.encoding.Md5PasswordEncoder"/>
```



Conf. min. Spring Security

```
<beans ...>
  <security:http auto-config="true">

    <security:intercept-url pattern="/secure/**" access="ROLE_ADMIN" />
    <security:intercept-url pattern="/view*" access="ROLE_ADMIN,ROLE_MEMBER" />
    <security:intercept-url pattern="/**" access="IS_AUTHENTICATED_ANONYMOUSLY" />
  </security:http>

  <security:authentication-provider>
    <security:user-service properties="/WEB-INF/users.properties" />
    <security:password-encoder hash="md5" />
  </security:authentication-provider>
</beans>
```

Sommaire – Spring Security

- Présentation de Spring Security
- Mise en place de Spring Security

web.xml

```
<filter>
  <filter-name>springSecurityFilterChain</filter-name>
  <filter-class>
    org.springframework.web.filter.DelegatingFilterProxy
  </filter-class>
</filter>

<filter-mapping>
  <filter-name>springSecurityFilterChain</filter-name>
  <url-pattern>*.do</url-pattern>
</filter-mapping>
```

Configuration Spring

```
<beans ...>
  <security:http auto-config='true'>

  <security:intercept-url pattern="/secure/**" access="ROLE_ADMIN" />
  <security:intercept-url pattern="/view*" access="ROLE_ADMIN,ROLE_MEMBER" />
  <security:intercept-url pattern="/**"
    access="IS_AUTHENTICATED_ANONYMOUSLY" />
  </security:http>

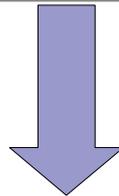
  <security:authentication-provider>
    <security:user-service properties="/WEB-INF/users.properties" />
    <security:password-encoder hash="md5" />
  </security:authentication-provider>
</beans>
```

Pages JSP

- **Gestion de la sécurité dans les pages JSP**
 - **Spring Security fournit une taglib dédiée**

TOP-SECRET INFORMATION

Click [HERE](/admin/deleteAll) to delete all records.



```
<%@ taglib prefix="security"  
    uri="http://www.springframework.org/security/tags" %>
```

...

```
<security:authorize ifAnyGranted="ROLE_ADMIN">  
    TOP-SECRET INFORMATION  
    Click HERE to delete all records.  
</security:authorize>
```

Classes et méthodes

- **Sécurisation basée sur la programmation par Aspects**
- **Configuration Spring**
 - **S'applique à tous les beans Spring**

```
<security:global-method-security>  
  <security:protect-pointcut  
    expression="execution(* com.springsource..*Service.*(..))"  
    access="ROLE_USER" />  
</security:global-method-security>
```

**configuration également
possible par annotations !**

S2AP

- OS-JEE-Aïe ??
- Les fonctionnalités
- Plus que de la glue
- La polémique
- L'avenir

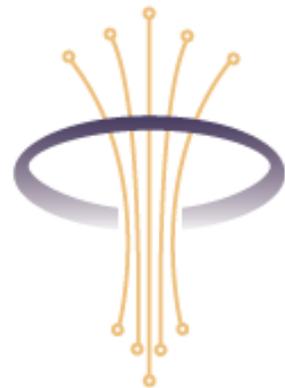
OS-JEE-Aïe ??

- **OSGi est une spécification qui définit :**
 - Des modules en Java (pas des Jars, des vrais modules!)
 - Un cycle de vie clair pour ces modules
 - Un annuaire de services
- **OSGi est utilisé :**
 - Dans des IDE (Eclipse)
 - Des systèmes embarqués (BMW Infotainment)
 - Des téléphones mobiles (Motorola, Nokia)
 - Des serveurs d'applications



OS-JEE-Aïe ??

- **Cette spécification est gérée par l'OSGi Alliance**
 - Le JCP n'est pas la seule source de spécification Java au monde :-)
 - SpringSource est membre de cette alliance (au même titre que IBM, Oracle, Redhat ou Sun...)



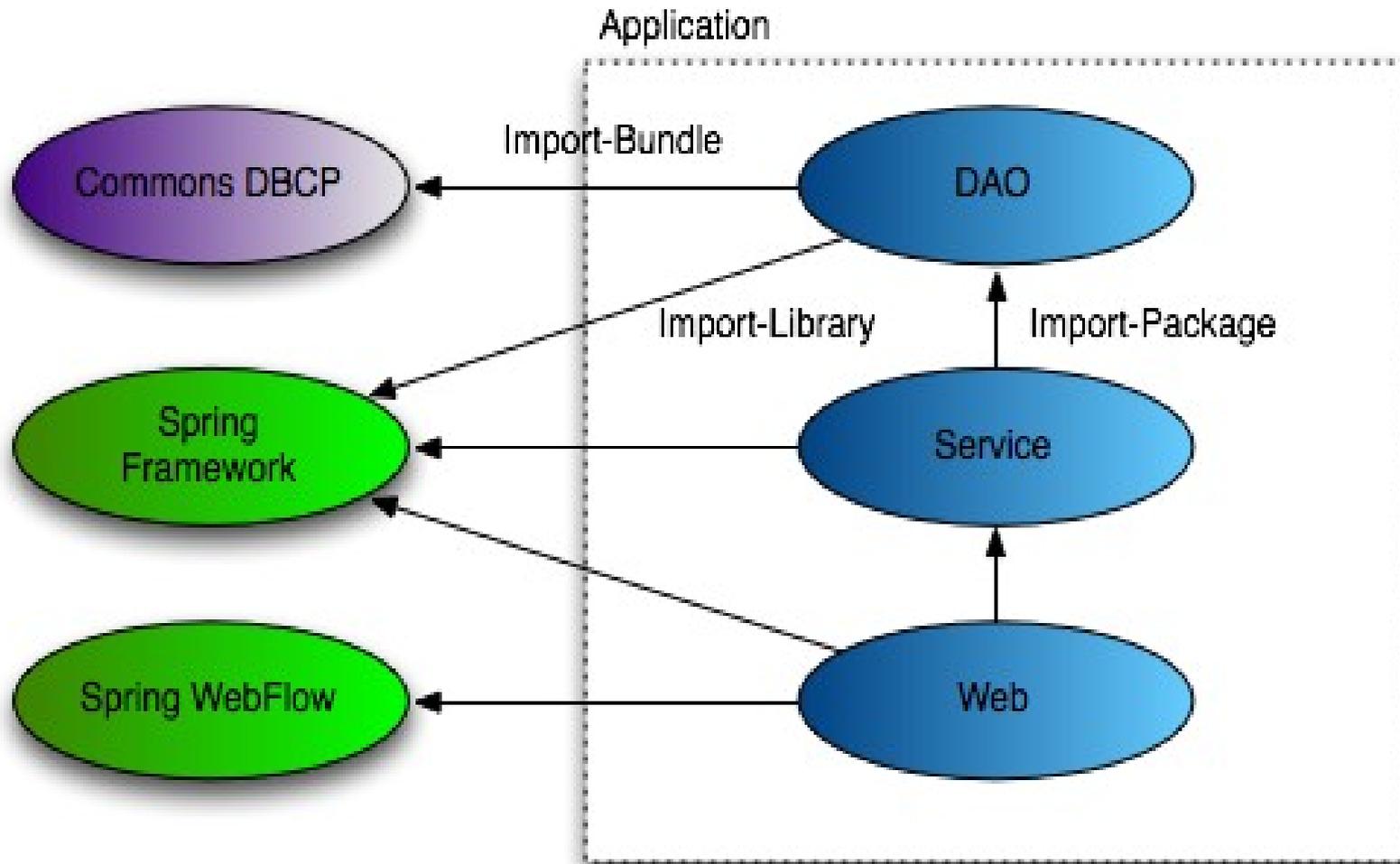
OSGi™
Alliance
Member

S2AP : Les fonctionnalités



S2AP : Les fonctionnalités

- Un exemple :



S2AP : Les fonctionnalités

• Pour les développeurs

- Une vraie architecture par module : ne sont publiques que les APIs... publiques.
- Un développement plus rapide : on ne recharge que le module sur lequel on travaille, et pas toute l'application
- Sans les inconvénients d'OSGi :
 - Faire une application OSGi est vraiment compliqué
 - Le code ne dépend pas des APIs OSGi (ou même de S2AP), nous restons toujours sur un modèle de programmation « POJO »

S2AP : Les fonctionnalités

• Pour la production

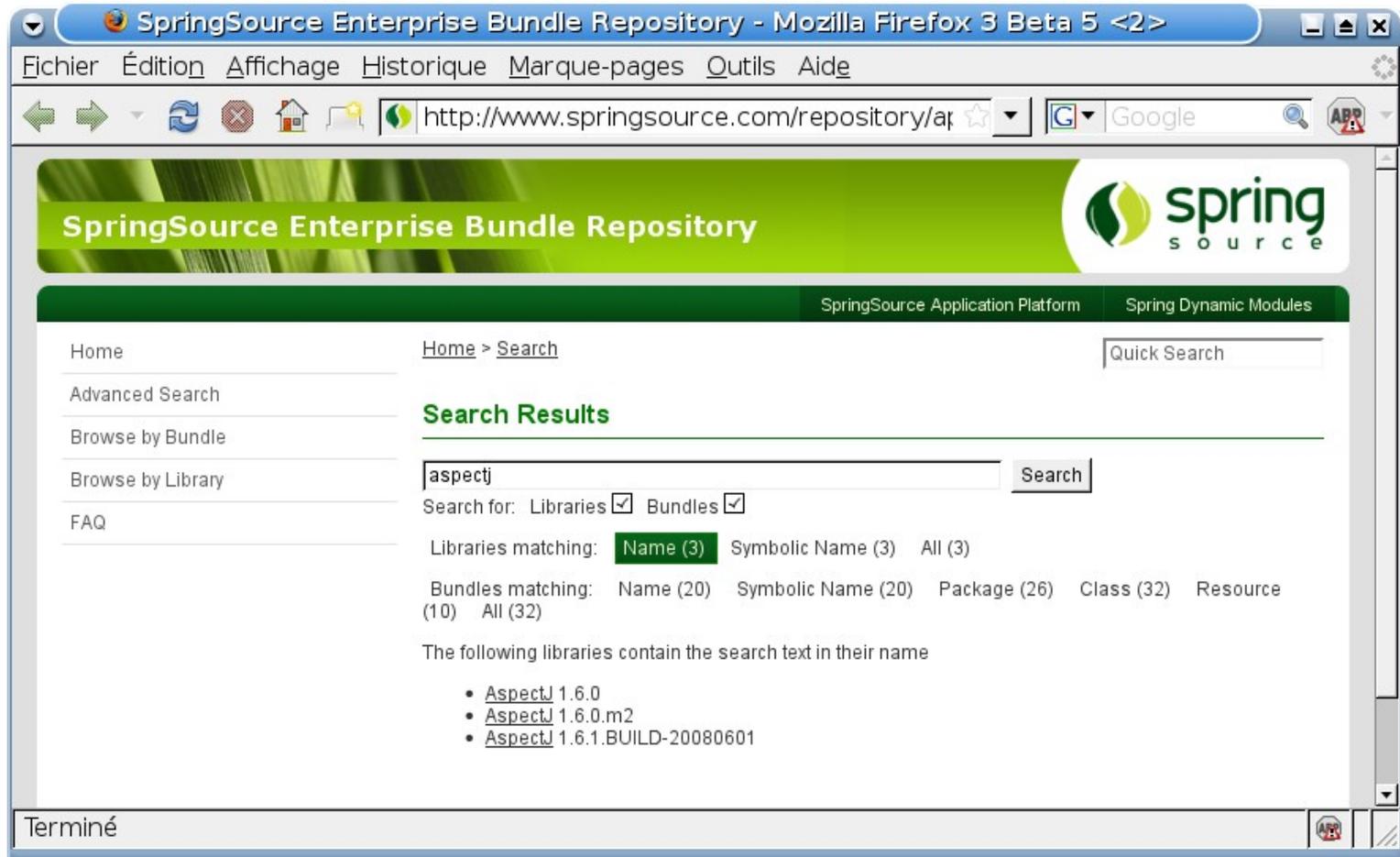
- Mises à jour à chaud sur des serveurs ou des applications en production
- Différentes versions du même module qui tournent en parallèle dans une même application
- Meilleure utilisation des ressources : Spring-DM ne charge que les modules utilisés par le serveur ou les applications
- ... autant de choses impossibles à faire avec un serveur d'applications d'ancienne génération!

S2AP : Plus que de la glue

- **S2AP ce n'est pas que Spring + Tomcat + OSGi**
 - Le load-time weaving fonctionne en environnement OSGi
 - Le scanning du classpath fonctionne en environnement OSGi
 - Le classloader context de la thread fonctionne en environnement OSGi
 - ... on vous avait prévenu, OSGi c'est vraiment compliqué!

S2AP : Plus que de la glue

- **SpringSource Enterprise Bundle Repository**
 - + de 300 librairies validées et OSGi-fiées



S2AP : La polémique

- **S2AP ne respecte pas les standards!**
 - Nous faisons partie de la JSR 316, laquelle spécifie JEE 6
 - Antonio peut témoigner!
 - Nous visons les profils A et B de la future spec
 - Le JCP n'est pas la seule source de bonnes specs
 - Nous suivons également l'OSGi Alliance
 - Nous considérons que certaines spécifications du profil C sont trop lourdes ou n'apportent pas de valeur ajoutée

S2AP : La polémique

- **S2AP utilise la GPL v3!**
 - S2AP est un nouveau développement, tous les autres produits Spring qui étaient sous licence Apache le resteront
 - La GPL v3 nous garantit qu'il n'y aura pas de serveur d'application commercial basé sur S2AP, sans que rien ne soit redonné à la communauté

**FREE AS IN
FREEDOM**
RICHARD STALLMAN'S
CRUSADE FOR FREE SOFTWARE



S2AP : L'avenir

- **Une version finale pour bientôt**
 - La version actuelle est la beta 6
 - La v.1.0 devrait être rapidement disponible
- **De nouvelles fonctionnalités**
 - Un profil Spring Batch!
- **Du tooling, du monitoring**
 - Amélioration du plugin Eclipse, et intégration dans SpringSource Tool Suite
 - Monitoring avec SpringSource Application Management Suite

Résumé / Conclusion

- **Le framework Spring**

- Un projet Open Source exceptionnellement populaire
- Entouré d'un portefeuille de projets complémentaires : sécurité, remoting, batch, Web...

- **SpringSource Application Platform**

- Révolutionne un marché stagnant
- Bientôt prêt pour la production

Bibliographie / liens

- **SpringSource France**
<http://www.springsource.fr>
- **Spring Security**
<http://static.springframework.org/spring-security/site/index.html>
- **SpringSource Application Platform**
<http://www.springsource.com/products/suite/applicationplatform>
- **SpringSource Enterprise Bundle Repository**
<http://www.springsource.com/repository/>



Questions / Réponses

www.parisjug.org



Copyright © 2008 ParisJug. Licence CC – Creative Commons 2.0 France – Paternité – Pas d'Utilisation Commerciale – Partage des Conditions Initiales à l'Identique



Sponsors



Merci de votre attention!



www.parisjug.org



Xebia

aneo
the other solution

valtech

BK Consulting

OceaneConsulting
Prestation Java



Licence



Paternité-Pas d'Utilisation Commerciale-Partage des Conditions Initiales à l'Identique
2.0 France

<http://creativecommons.org/licenses/by-nc-sa/2.0/fr/>