Retour d’expériences : Qualité et co-développement au consortium *ESUP*

Séminaire AMUE "Qualité logicielle et co-développements" du 30/01/2008

Raymond Bourges

ESUP Portail

Licence

- Remarque :
  - Les transparents présentés ici ont été réalisés par :
    - Raymond Bourges (Université de Rennes 1)
  - Certains contenus proviennent des présentations de :
    - Pascal Aubry (Université de Rennes 1)

ESUP Portail

Plan

- Contexte
  - Historique ESUP-Portail
  - Technique
  - Réflexion sur les métiers
- La révélation Spring
- ESUP-Commons
- Bilan
- Avenir

Contexte historique

- Fin 2002 : Début du projet ESUP-Portail
- Début 2003 : Premiers chantiers
  - Choix technologiques importants
    - uPortal
    - CAS
    - Etc.
  - Constitution de groupes de travail
    - Dont un « outils, méthodologie et normes »
- 2005, 2006 : Le besoin de maintenir les développements ET le départ des contractuels !
  - Vers des postes de la fonction publique
  - Vers des SSII

ESUP Portail

Contexte historique - Packaging

- Packaging (un des raisons du succès d'ESUP-Portail je crois)
  - Oblige à formaliser
    - La documentation
    - Les sources
    - Les fichiers de configuration
    - Les procédures d'installation
  - C'est un énorme travail mais se mettre dans cette logique est profitable à terme
    - Maintenabilité
    - Adaptabilité
    - Acceptation
- Quelles formes de package
  - Programme java existants (uPortal, CAS)
  - Programme non java (Horde)
  - Des développements propres (Dossier étudiant, Stockage, etc.)

Contexte historique - Normes

- Les premières documentations de principe :
  - Conventions de codage
  - Règles de nommage des packages
  - Règles de nommage des fichiers de configuration
  - Règles de nommage des livrables ESUP
  - Règles concernant les caractères accentués
  - Fichier de licence ESUP Portail
- Plus tard, les documents sur les outils :
  - Normalisation des build.xml et build.properties de ant
  - Framework MAG
  - Utilisation de subversion
  - Utilisation de sourcesup

ESUP Portail

Contexte historique – Premier bilan

- Des choses bien ☺
  - La « chance » d'être obligés de travailler à distance avec des existants différents !
    - Oblige à communiquer
    - Oblige à paramétrer
    - Mais prend aussi beaucoup de temps…
  - Synergie autour d'un langage commun (java) et d'outils communs (eclipse, ant, svn, sourcesup.cru.fr)
- Des choses moins bien ☹
  - Chaque développement est fait par une personne qui a SA propre façon de faire
    - Difficultés à maintenir et à faire évoluer
  - **Peu de mutualisation car peu de contribution**

ESUP Portail

Contexte technique

- Langage Java
  - Structurant
    - L'expérience acquise est importante
  - Mais tentaculaire
    - Parfois difficile de savoir par où commencer
    - Qu'est-ce qui est important ou pas ?
- SOA, Web Service, Développement en couches
  - OK mais comment fait-on concrètement ?
- Le reste
  - Internationalisation, accessibilité, etc.

- **Beaucoup de questions…**

ESUP Portail

Réflexions sur les métiers et le libre

- Un métier ou des métiers ?
  - Développeur
    - Interface, métier, base de données, …
  - Exploitant
    - Configuration, déploiement, optimisation
  - Graphiste, ergonome, etc.
  - Beaucoup de compétences, de transversalité
- Le libre dans le Sup.
  - Beaucoup de consommateurs mais peu de producteurs
  - Beaucoup d'actions de recensement mais peu d'incitations au développement
  - Beaucoup de développements libres mais peu de diffusion

ESUP Portail

La révélation Spring

- Historique
  - Premiers travaux ESUP fin 2005
  - Formation interne ESUP début 2006
  - Formations à la communauté avec l'aide de l'AMUE fin 2006
- Les principes
  - Simplifie la vie du développeur en gérant le cycle de vie des objets (**moins de lignes de code = moins de bugs**)
  - Conteneur léger (tourne dans un simple Tomcat)
  - Systématise l'utilisation d'interface (introduction **par la pratique** de la notion de développement en couches)
- **Servira de base à ESUP-Commons**

ESUP Portail

Esup-commons

- Un **« environnement »** pensé **par des développeurs**
  - Pour eux-mêmes
  - Pour les collègues développeurs
  - Pour les exploitants
- C’est aussi une méthodologie
  - Environnement de développement avec une **suite d’outils de production**
  - Chaque élément à **sa place** (code métier, IHM, fichier de configuration)
  - Prévoit **la génération automatique** des packages, de la documentation, l’utilisation d’un gestionnaire de version (sourcesup)

ESUP Portail

Les objectifs de esup-commons

- **Simplifier la maintenance**
- Faciliter l'adaptation aux configurations locales
- Permettre l'évolution des fonctionnalités
- Augmenter la productivité
- **Faciliter la mobilité des développeurs**
- Uniformiser les installations d'applications
- **Faciliter les contributions**
- **Fiabiliser les applications**
- Améliorer l'accessibilité

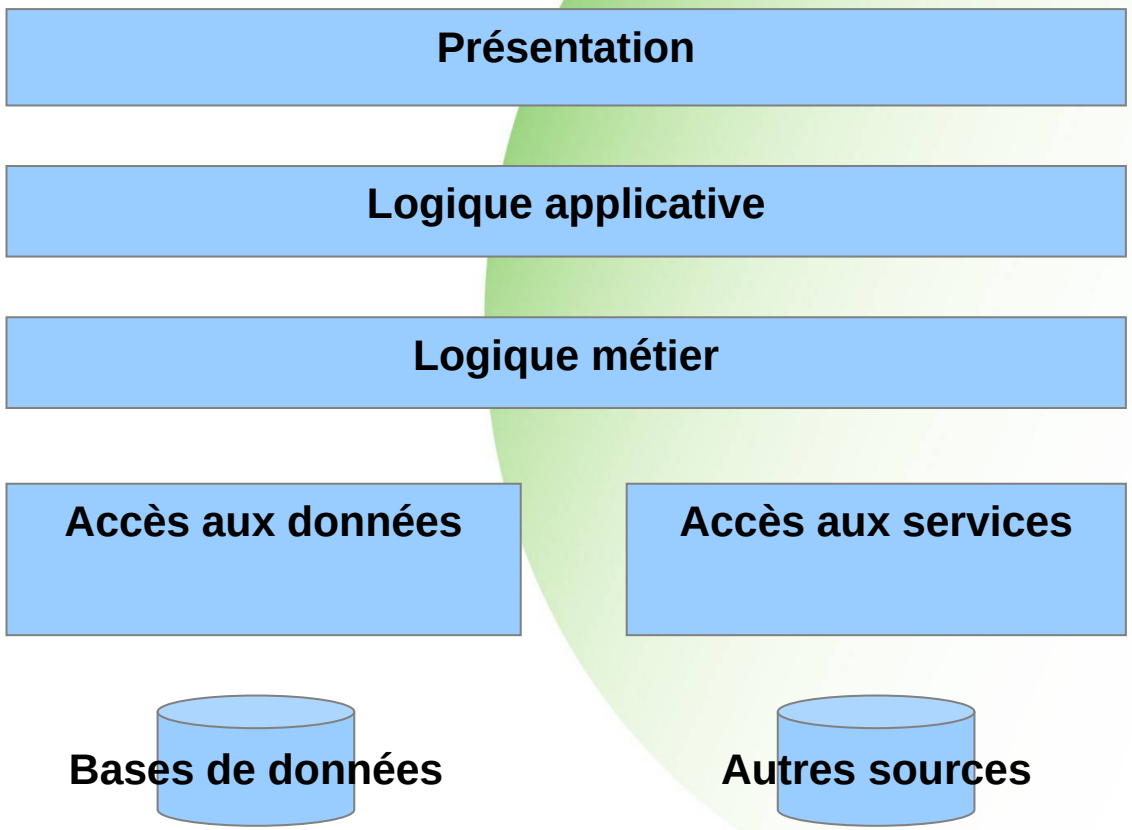
ESUP Portail

Les grands principes

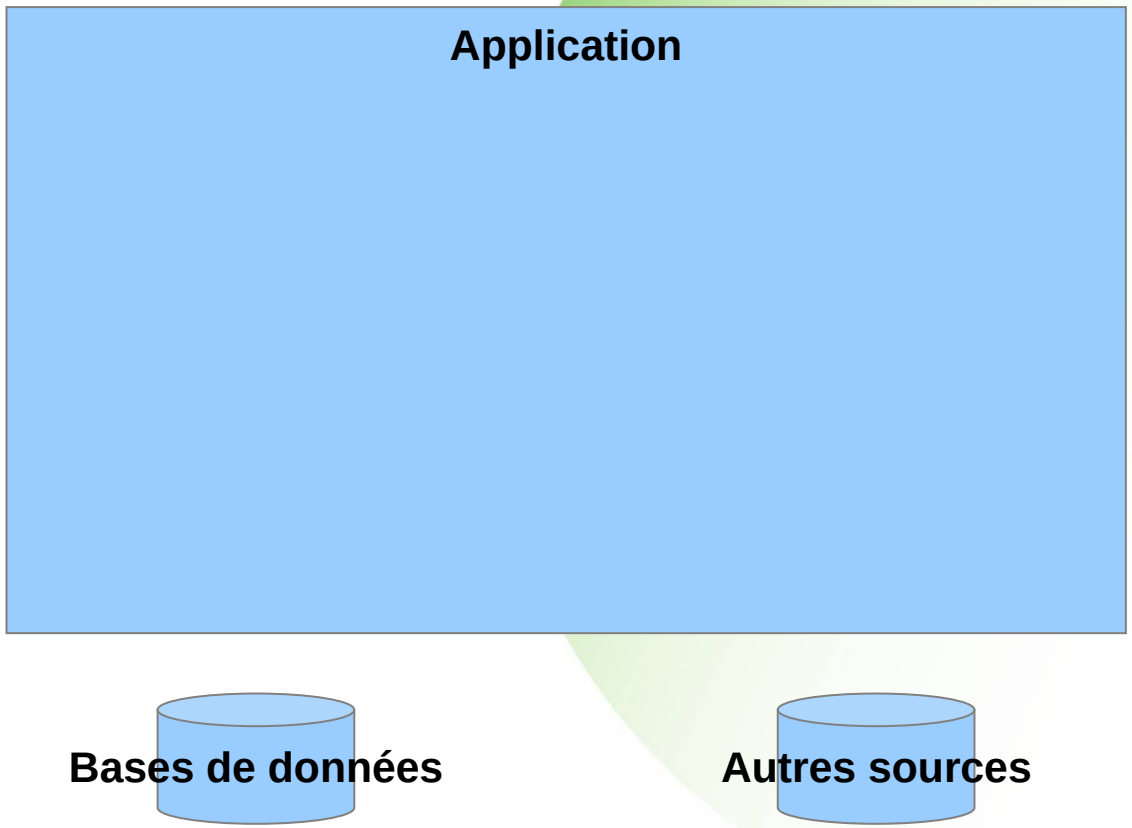
- Utilise dès que possible des outils de haut niveau !
- Sépare bien tes couches !
- Abstrait tes objets !
- Relis ton code deux semaines après !

ESUP Portail

L’idéal standard

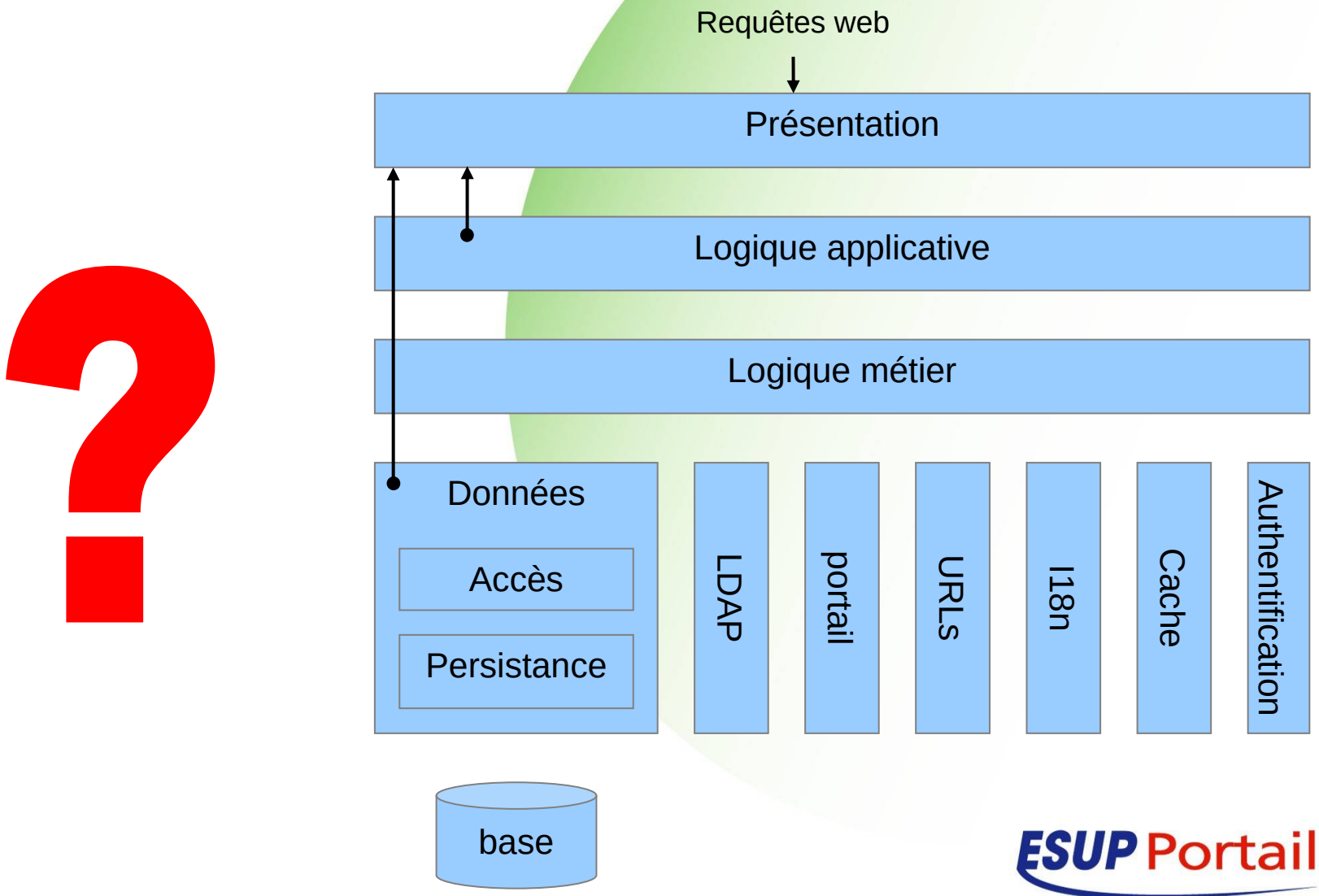


Dans la pratique ;-)



ESUP Portail

L’architecture de esup-commons



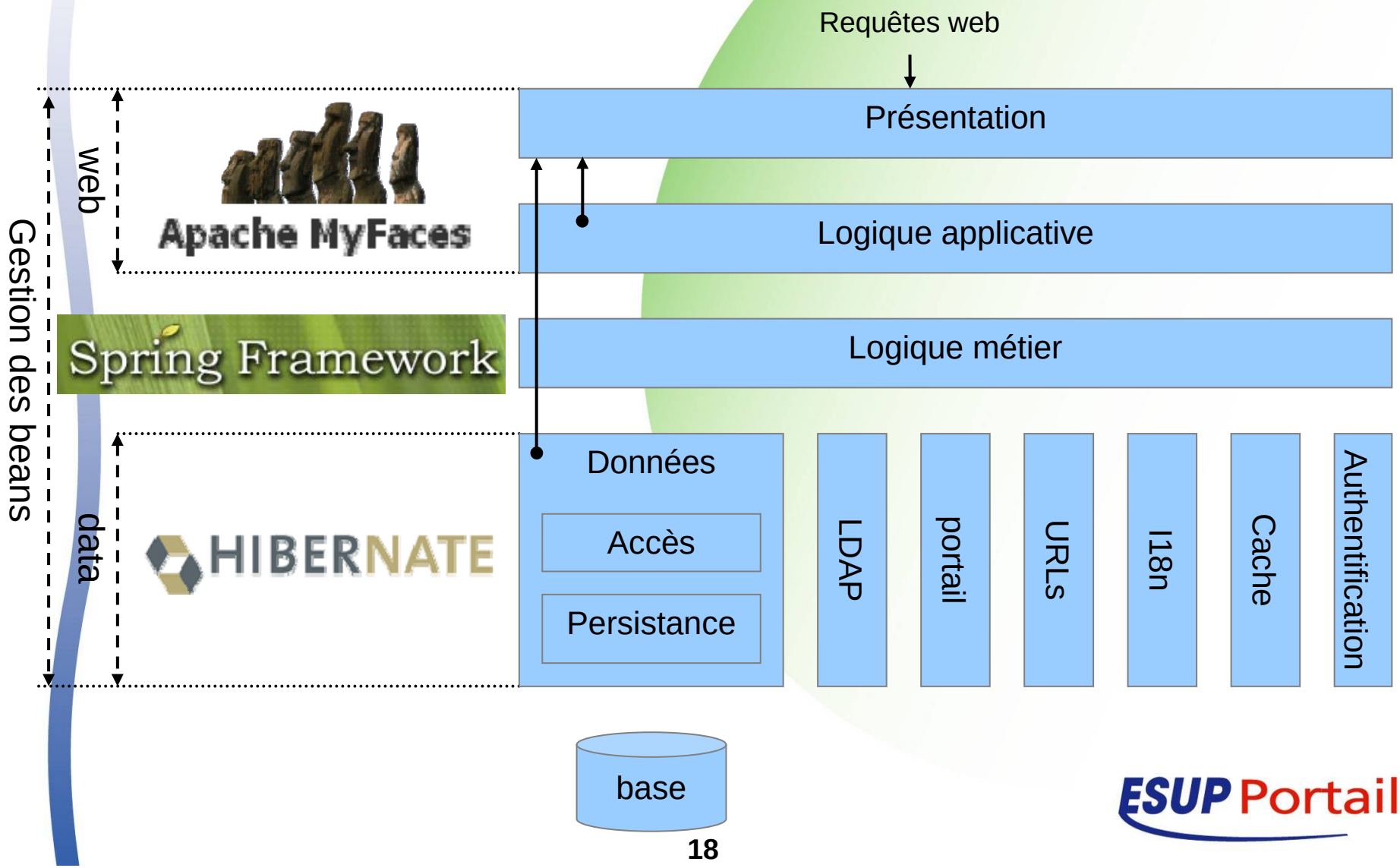
Portlet/servlet : même combat

- Nous ne développons pas des portlets ou des servlets, nous **développons des applications**

- Il est très intéressant de pouvoir faire tourner une application tantôt en servlet, tantôt en portlet
  - Environnement de Développement plus simple en servlet
    - NB : On est encore ici dans une logique de **donner du confort au développeur**
  - Permet de séparer les problèmes inhérents aux portlets
    - Publication, déploiement, interaction avec le portail
  - Diffusion en quick-start

ESUP Portail

L’architecture de esup-commons



ESUP Portail

Les « plus » de esup-commons

- Gestion des transactions
- Gestion des versions
  - **Cohérence** entre la version développée et la base de données
- Gestion des exceptions
  - Pour avoir des messages clairs et donc une **correction rapide** des problèmes
- Internationalisation native
  - Et donc une **externalisation** des messages de l'application
- Envoi de courriers électroniques
- Commandes en ligne
- Web services

ESUP Portail

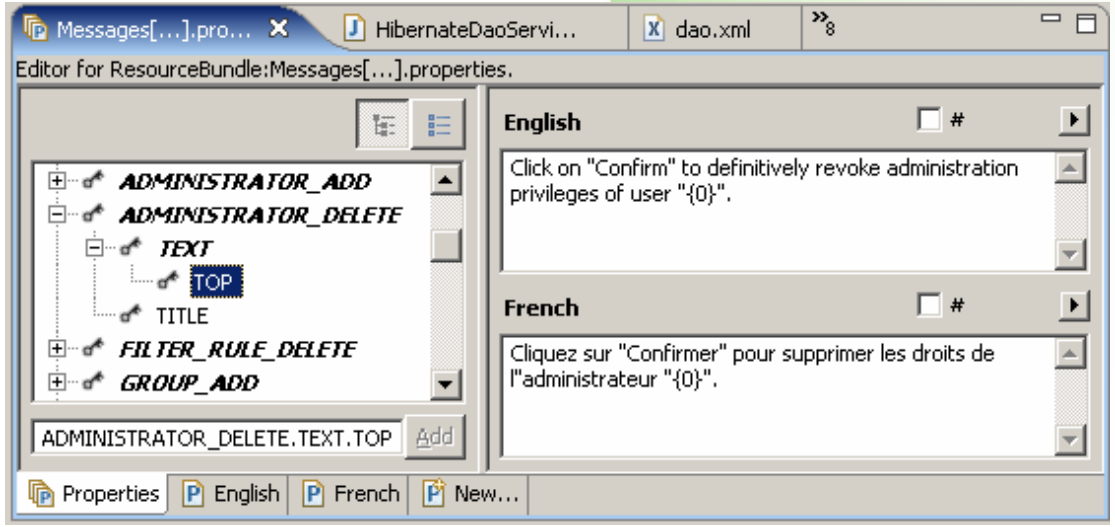
Les « plus » de esup-commons

- Gestion de l'authentification
  - CAS, LDAP, etc.
- Accès au portail
- Accès à LDAP
- Pagination des données
- Taglib dynamique
  - Aucun formatage **en dur** dans les pages
- Gestion des URLs directes (*deep linking*)
- Gestion de cache
  - Encore un **nid à bugs** !

ESUP Portail

Un exemple d'un outil de production

- RBE pour l'édition des fichiers de messages
  - Multilingue
  - Gestion de l'encodage des accents
- ESUP-Commons fournit la mécanique pour lire les fichiers de messages **ET** l'outil d'édition de ces messages
  - On n'est plus dans le « il faudrait faire »
  - Comme les outils sont là cela devient naturel pour le développeur

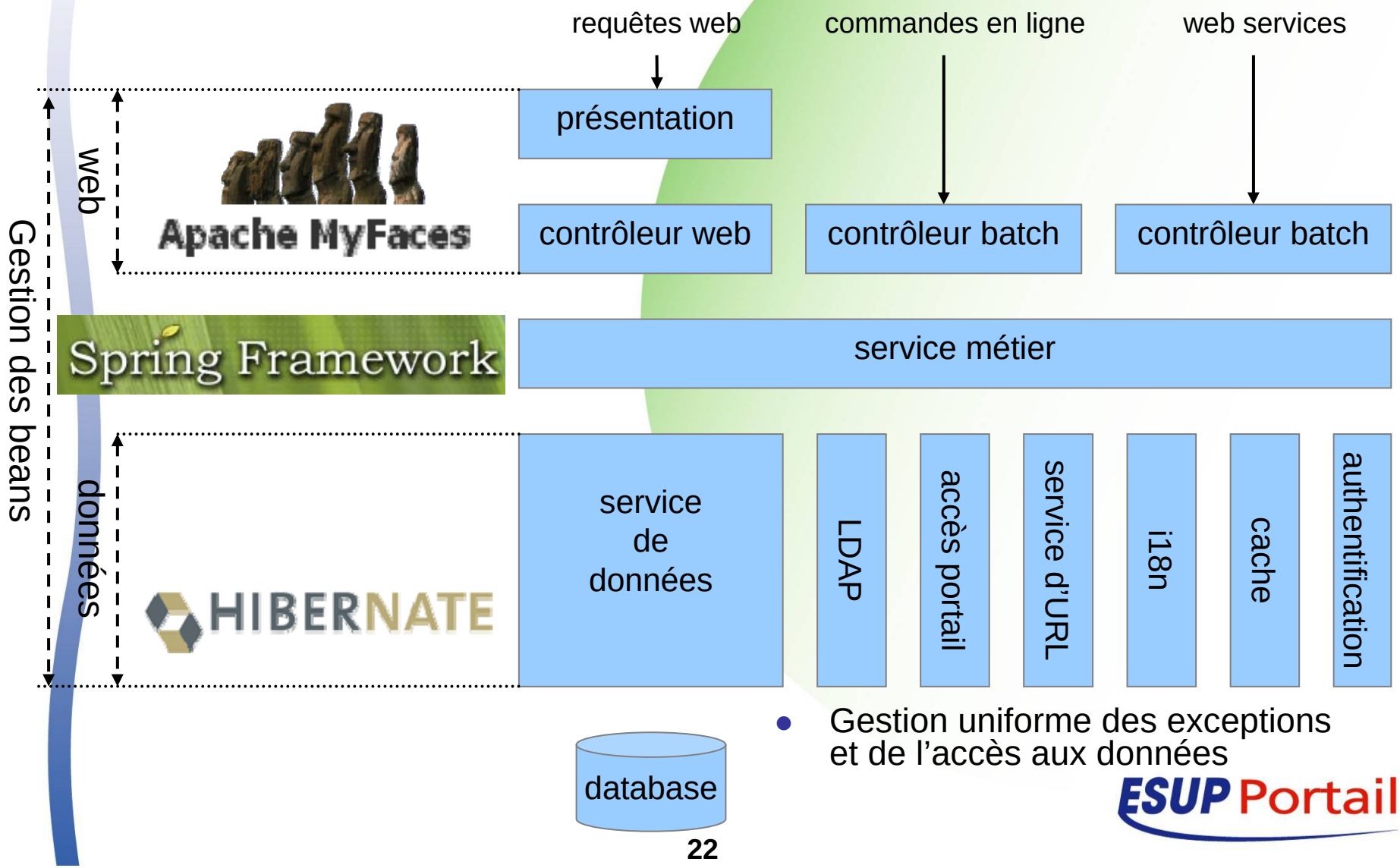


Mais aussi :

- CheckStyle
- Subclipse
- Etc.

ESUP Portail

Web services



- Gestion uniforme des exceptions et de l'accès aux données

ESUP Portail

Bilan

- Plus de 80 Personnes formées à ESUP-Commons
  - Un support de formation de plus de 150 pages
- Présentation JRES de Pascal Aubry
  - http://2007.jres.org/planning/paperd5a2.html?pid=97
- Utilisé pour les développements
  - ESUP-Portail
  - ORI-OAI
  - Rennes 1
  - Ailleurs
- En perpétuelle évolution
  - Besoin de suivre les technologies
  - Besoin de maintenir l'effort de formation

Avenir

- Encore du travail et des questions…
  - Faut-il définir un activité de « mainteneur d'environnement de développement » comme ESUP-Commons ?
  - Ouverture à l'international
  - Intégrer plus fortement d'autres outils
    - TPTP (analyse l'utilisation et les performances du code)
    - FindBugs (permet de détecter les nids à bugs)
    - JUnit (intégrer dans la doc de formation la méthodologie à suivre pour les tests –bouchons de test, simulation de sessions Web–)
  - Préparer demain
    - Lien avec des outils de gestion de roadmap
    - OSGI (plugins logiciels)
    - Utilisation des annotations JAVA
    - Passage à Seam

ESUP Portail

Conclusion

- Pour moi la qualité passe par l'humain
  - C'est souvent vu comme une contrainte
- Il faut donc proposer des outils qui simplifient la vie des développeurs
  - La qualité devient (plus) naturelle
  - On a gagné quand ils ne peuvent plus se passer des outils
- Offrir ces outils demande un gros travail…
  - A des personnes dont ce n'est pas l'activité première
  - Alors même que ces outils sont stratégiques pour la communauté

ESUP Portail