



JEEVANEEO
CONSEIL



Le Bureau Electronique
Quick Overview

Alexandre Borgoltz
JUG – Poitou Charentes
2009-05-14

Sommaire



- Origines
- Principes
- Architecture
- Eclipse RCP appliqué au BE
- Bonnes pratiques
- Mauvaises pratiques
- Difficultés rencontrées



Les origines – Qualité de service



- Multiplication des écrans
 - Inconfort de l'utilisateur
 - Alt-tab devient fastidieux
 - → moindre productivité

- Multiplication des re-saisies
 - Lenteur (face au client)
 - Risque augmenté d'erreurs
 - Même info demandée plusieurs fois



Les origines - TCO



- Multiplication des outils/technologies
 - complexité/coût d'appropriation
 - Adhérence entre applications qui doivent communiquer sur le poste → « spaghetti »
 - Mixité Clients lourds/légers et maison/progiciel



Les principes : Lanceur universel



Sans BE :

- Hétérogénéité du lancement d'applications
 - Menu Démarrer
 - Raccourcis bureau
 - Navigation internet
 - Favoris du navigateur
 - ...

- A l'utilisateur de savoir (→ formation)



Les principes : Lanceur universel

Avec BE :

- Toutes les applications sont appelées par le BE
 - Automatiquement
 - A la demande (catalogue)

□ Le BE est *le* lanceur universel des applications du poste

→ Ergonomie simplifiée



Les principes : Intégration graphique



Sans BE :

- Nombreuses applications ouvertes
« enchevêtrées »
- Barre des tâches surchargée
- Alt+tab syndrom*
- Nécessité parfois de fermer temporairement des applications pour « faire le ménage »
- Navigation laborieuse



Les principes : Intégration graphique



Avec BE :

- Les applications restent ouvertes
- Le BE ne m'affiche que les applications (et raccourcis) pertinents *pour la posture de travail courante*
- Les applications sont rassemblées dans une seule fenêtre (vs *alt+tab*) et agencées (vs *enchevêtrées*)
- L'utilisateur se concentre sur son métier, pas sur la manipulation de ses outils
- Agencement orienté « vis-à-vis »
- Navigation aisée



Les principes : Partage contextuel



Sans BE :

- Multiples ressaisies
 - Même questions posées plusieurs fois au client
 - Risque accru d'erreurs
- Moindre qualité et productivité de service



Les principes : Partage contextuel



Avec BE :

- Le partage de données entre applications est automatisable. Tel un copier-coller automatique
- Gain de productivité (temps)
- Gain de qualité de service (moins d'attente en ligne, chaque question n'est posée qu'une fois, moins d'erreurs)



Les principes : Individualisation



Sans BE :

- Pas de personnalisation, au-delà éventuellement du poste
- Même poste, même navigation



Les principes : Individualisation



Avec BE :

- L'agencement graphique dans le BE est fonction
 - des droits définis dans le référentiel d'entreprise
 - du profil (poste, métier, entité, lieu géographique...)

- Pour chaque « cas d'utilisation » est définie en central un agencement graphique spécifique, une liste de raccourcis contextuels...

- L'utilisateur ne choisit plus de changer d'outil mais de « cas d'utilisation »

- Agencement graphique, favoris... sont personnalisables par l'utilisateur final individuellement

Les principes : Référentiel des applications



Sans BE :

- Pas de référentiel des applications
- Applications dispersées
- Connaissance humaine nécessaire



Les principes : Référentiel des applications



Avec BE :

- Référentiel unique des applications
- Catégorisation des applications et metadata
- Moteur de recherche « unique » sur le poste
- Administration de ce référentiel en central
- Gestion d'accès centralisée attachée à ce référentiel



Illustration : le BE organise l'espace



Un découpage offrant une agrégation d'applications par espace thématique et par situation de travail

Un catalogue pour accéder à l'ensemble du Système d'information par catégories

Des raccourcis correspondants à la situation en cours

Applications hébergées façon portlet

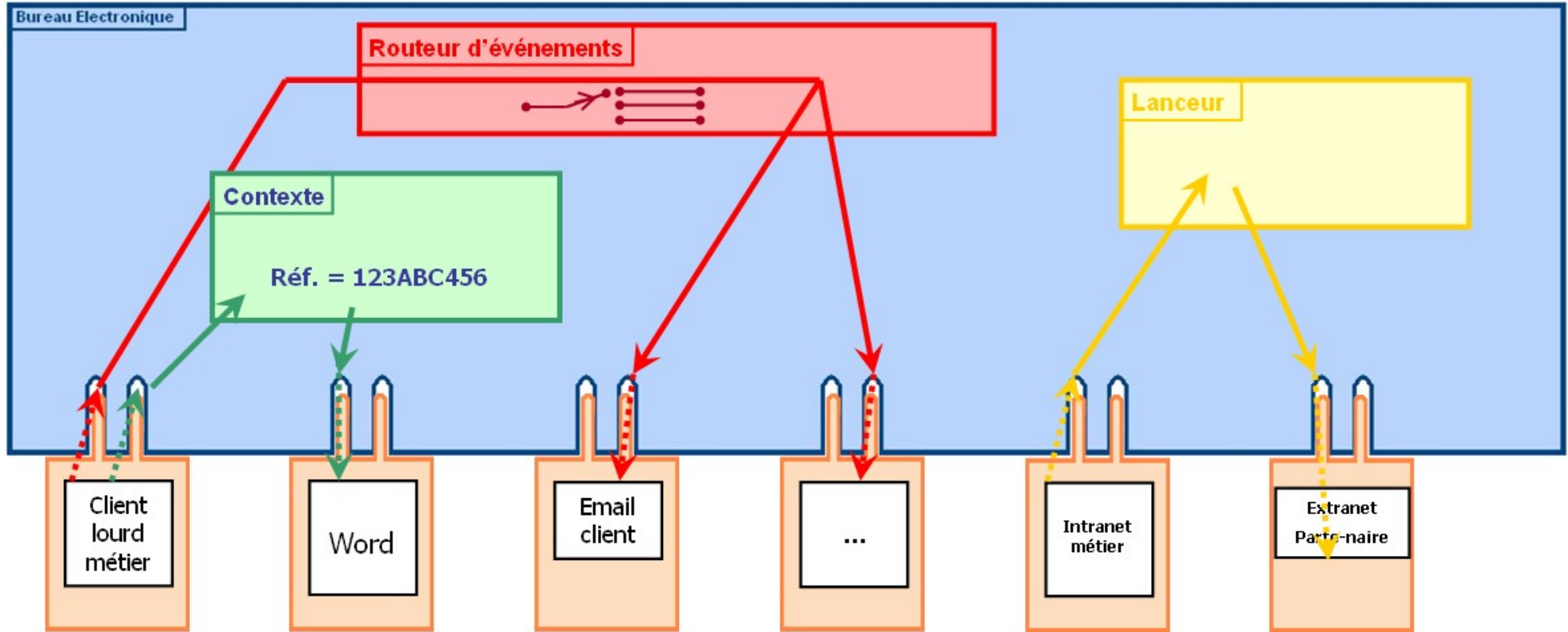
The screenshot displays a web application interface with several components:

- Navigation Bar:** Contains icons for 'Home', 'Demo', 'JEEVANEO', and 'Perso'. A red dashed box highlights this bar.
- Left Panel (Catalogue des Ressources):** A tree view showing a search for 'JUG' and a list of resources under 'Maj.' including 'Applications de démonstration', 'Apache Con', 'Clientèle', 'Eclipse Con', 'JUG', 'Demos', 'Demos Delphi', and 'JavaOne'. A blue dashed box highlights this panel.
- Form1:** A registration form with fields for 'Civilité', 'Prénom', 'Nom', 'Date de naissance', 'Profession', 'Adresse', 'Téléphone', 'Portable', 'Code Postal', 'Ville', and 'E-Mail'. A yellow dashed box highlights the 'Favoris' section below it, listing 'Demos' and 'Demos Delphi'.
- Right Panel (Google Search):** A Google search page for 'Alexandre Borgoltz' with search buttons and links. A green dashed box highlights this panel.

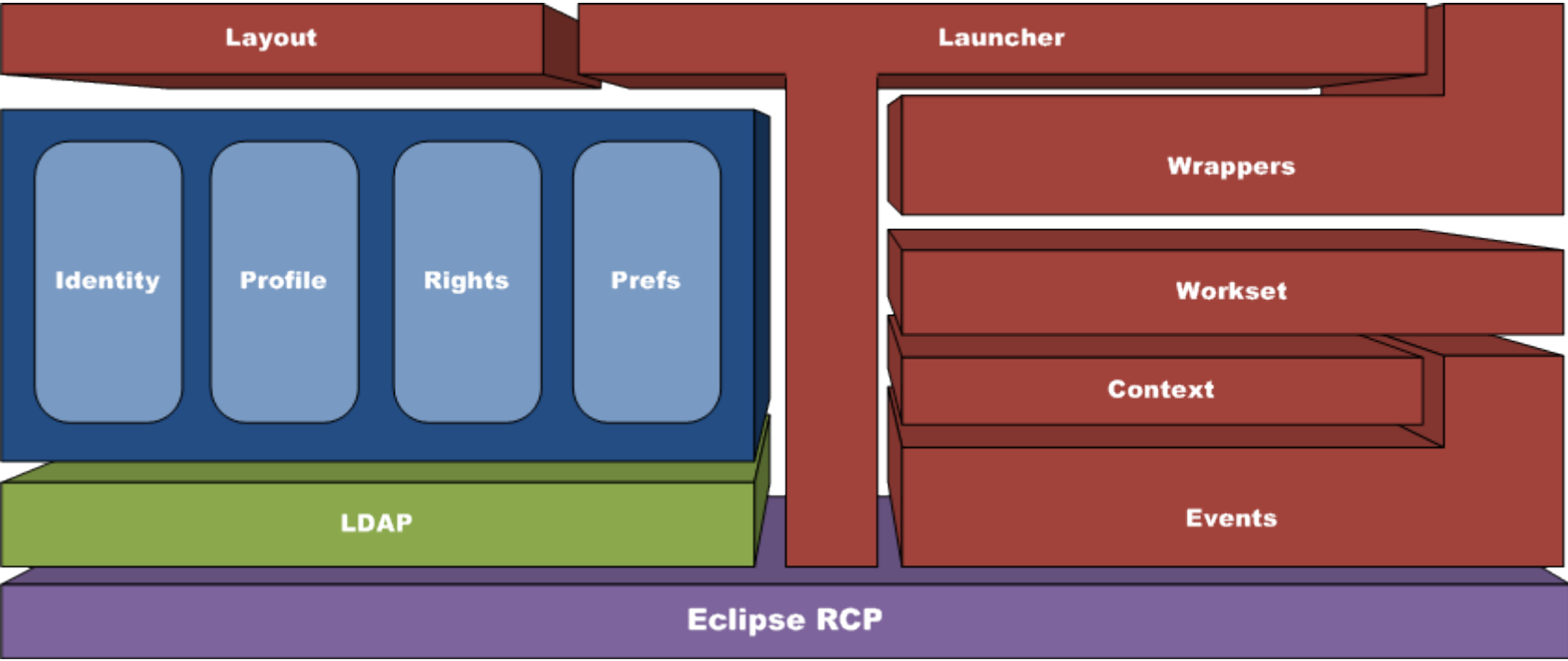
Interactions applicatives : Illustration



- Un mot d'ordre : couplage lâche



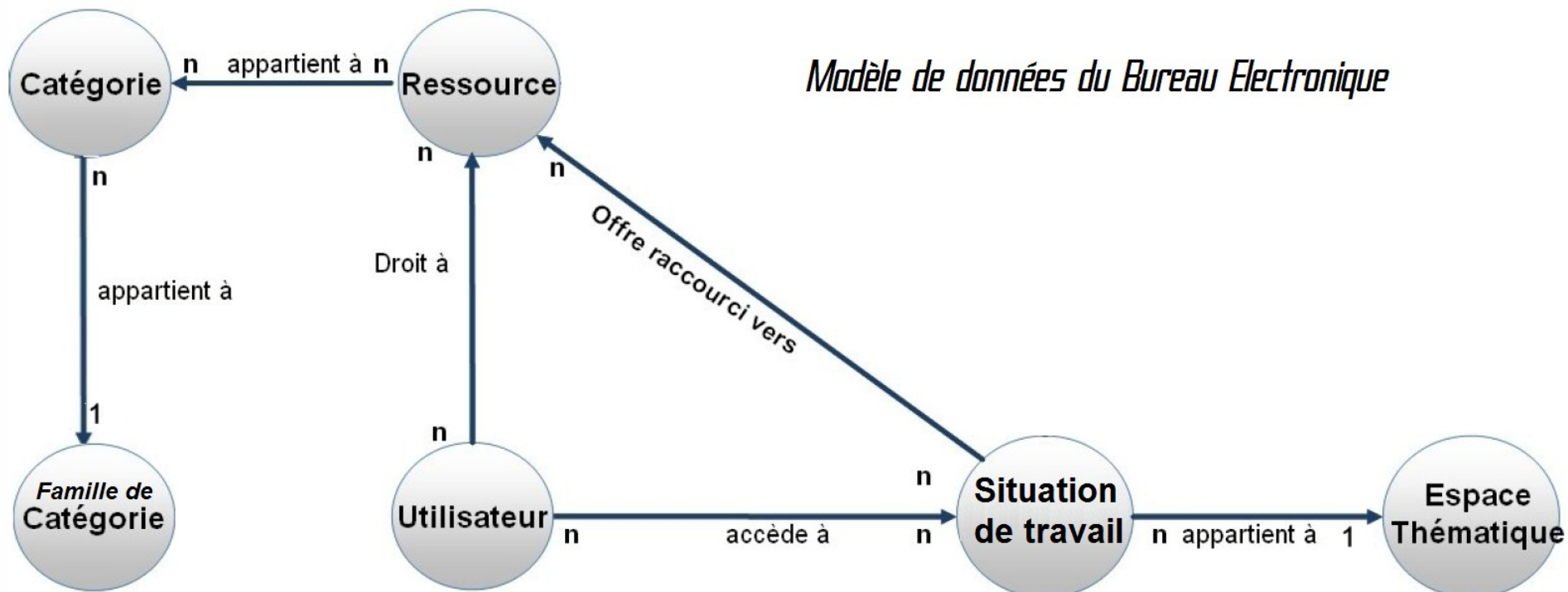
BE Eclipse - composants



BE Eclipse – modèle de données



Modèle de données du Bureau Electronique



Eclipse RCP et BE – 1/5



- ❑ Chaque application hébergée l'est dans une **Vue**
- ❑ Une situation de travail est représentée par une **Perspective** et des vues utilitaires (favoris, raccourcis contextuels...)
- ❑ Le déploiement centralisé est assuré par le mécanisme natif d'**update management (update site)**



Eclipse RCP et BE - 2/5



□ Modularité, gestion des versions, réutilisabilité...
sont assurés par l'architecture en **plugins** et
l'utilisation massive d'**extensions**:

- Le BE est une **application RCP**
- Les contrats sont définis par des **points d'extensions** ou des **API publiées**
- Chaque wrapper est un (ensemble de) **plugins**



Eclipse RCP et BE - 2/3



- ❑ La gestion d'**activities** permet de gérer simplement les droits sur les perspectives et les vues
- ❑ La personnalisation explicite s'appuie sur le mécanisme natif de gestion des **preferences**



Eclipse RCP et BE - 4/5



- ❑ L'interface graphique utilise les composants génériques mais *OS-native* **SWT/jFace**
- ❑ L'intégration native d'**activeX** permet sans code natif supplémentaire (code client java uniquement) d'intégrer Office, IE, progiciels etc.



Eclipse RCP et BE - 5/5



- ❑ Le composant natif **SWT browser** permet d'héberger des pages web avec un moindre effort

- ❑ Les **fragments** natifs et leur gestion de **librairies natives** assurent *l'indépendance à la plateforme* tout en assurant une intégration maximale (JNI), naturelle et aisée avec l'OS (vs implémentations pures java, parfois plus lentes et/ou limitées)



BE Eclipse - Démo!



- Ergonomie du BE
 - Vis-à-vis
 - Onglets
 - Minimize-maximize
 - Moteur de recherche
 - Partage de contexte

- Profiling
 - Editer une situation de travail

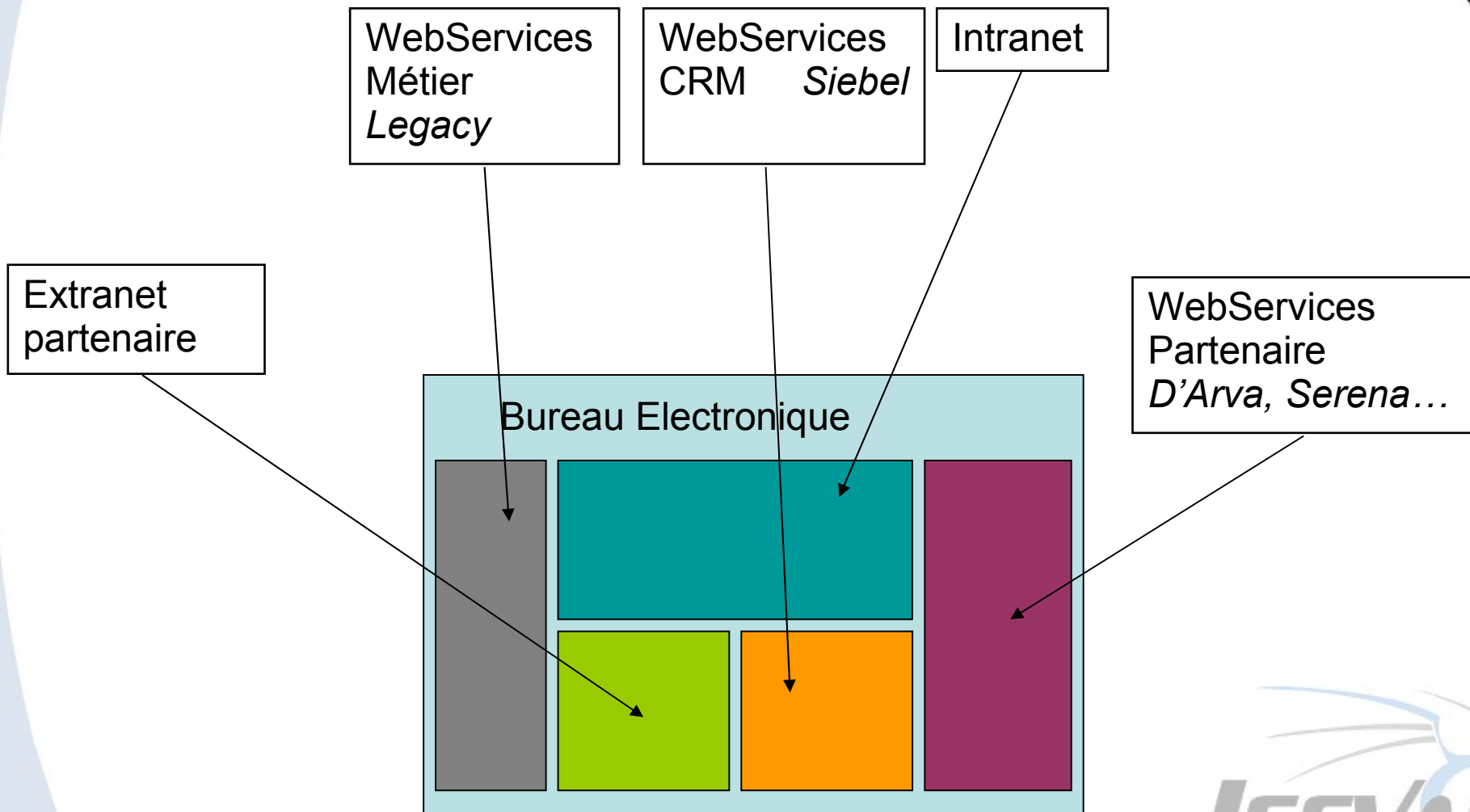
- Gestion des droits
 - Ajouter/enlever des droits en live



BE – Next step



❑ Socle de développement orienté SOA



Bonnes pratiques 1/2



- Indépendance à la plate-forme et à la technologie
- Intégration non intrusive, « inconsciente »
- Le BE : pas une brique critique
- Indépendance au BE des applications hébergées
- Autonomie des applications en dehors du BE



Bonnes pratiques 2/2



- Indépendances des SI entre eux
- Indépendance par rapport à la prise de décision de l'enchaînement
- Déploiement/Mise à jour centralisés
- Administration centralisée, délégable
- S'appuie sur un référentiel des ressources de l'entreprise, central



Mauvaises pratiques



- Ce que le Bureau Electronique NE DOIT PAS être:
 - EAI côté client : toujours privilégier l'intégration par le serveur
 - BPM
 - BE est un conteneur et un lanceur
 - La logique d'orchestration des processus doit être centralisée (donc pas sur le poste)
 - Il doit la favoriser en facilitant l'intégration
 - Porteur de logique métier
 - Ce sont les applications hébergées (et les SIs correspondants) qui *seules* portent la logique métier
 - Risque d'éparpillement de la logique métier → Maintenance lourde et coûteuse



Retour d'expérience 1/2



- Compétences de l'équipe de mise en place :
 - Très orienté eclipse, donc Java
 - MAIS l'ouverture vers les technologies native est aisée
 - → ne pas sous estimer le besoin de compétences « natives » (C/C++, win32, Delphi ...)

- Le composant SWT browser montre rapidement ses limites dès lors que les pages hébergées sont complexes (activeX, applets...)



Retour d'expérience 2/2



- Très peu de bugs sur le BE lui-même
- Excellent accueil par l'utilisateur final : « on a l'impression d'avoir toujours travaillé avec » !!
- Facilitateur de l'intégration des projets suivants sur le poste de travail



Questions?



Réponses!

