

# Retour d'expérience

Julien BERNARD  
Crédit Agricole Asset Management



Forum PHP 2005

# Sommaire

1. Crédit Agricole Asset Management
2. Etat des lieux applicatifs
3. Plate-forme de développement
4. Retour d'expérience sur PHP
5. Conclusion

# I. Crédit Agricole S.A.

Mieux connaître  
le Groupe Crédit Agricole

1<sup>er</sup> groupe  
bancaire  
français



21 millions  
de clients

présence dans  
60 pays



134 000  
collaborateurs



34 200  
administrateurs



5,7 millions  
de sociétaires



41 Caisses  
Régionales



7 200  
agences



2 599  
Caisses  
Locales



# I. Crédit Agricole S.A.

- Banque de détail
  - réseau Crédit Agricole
  - réseau LCL
- Services financiers spécialisés
- Gestion d'actifs
- Assurances
- Banque privée
- Banque de financement et d'investissement



# I. Crédit Agricole Asset Management

- Gestion d'OPCVM destinés à la clientèle de particuliers, d'entreprises et d'investisseurs institutionnels et services annexes
- 381,9 Mds d'euros sous gestion (au 30/06/2005)
- 3400 personnes
- 8 centres de gestion
- 400 personnes à l'informatique
- une équipe de 15 personnes dédiée aux développements web



## 2. Etat des lieux applicatifs

- + de 35 sites web Internet/Extranet/Intranet
  - Sites CAAM et filiales
    - sites éditoriaux
    - 15 sites différents - 6 langues
    - informations financières en temps réel
    - forte fréquentation (+500.000 visiteurs par mois)
  - Services aux clients (sécurisé)
    - Espace privée pour la clientèle en direct
    - Espace de ressources pour les réseaux de vente
    - Consultation des comptes en ligne et transactions
  - Services spécialisés (transactionnel critique)
    - Augmentation de capital
    - Centrale d'achats d'OPCVM
  - Intranets
    - Consultation du référentiel produit
    - 5 intranets d'entreprise pour CAAM et ses filiales
    - Déontologie : Gestion des déclarations en ligne
    - 5 intranets métiers : gestion, marketing, pôle info...
    - Annuaire d'entreprise


# 3. Plate-forme de développement

- Description des composants
  - PHP : serveur d'applications principal, utilisé dans 98% des applications
  - APACHE : serveur web leader du marché
  - LINUX : OS performant, efficace et stable.
  - TOMCAT : serveur d'applications secondaire, utilisé à la marge
  - MYSQL : serveur de bases de données principal, utilisé pour tout nouveau développement
  - CVS : serveur de versionning de code source, utilisé au quotidien dans le développement
  - OPENLDAP : serveur LDAP, utilisé pour l'annuaire d'entreprise
  - JONAS : serveur d'EJB, utilisé à la marge pour certains composants purement java
  - SUN SOLARIS/SYBASE : serveur de bases de données secondaire, utilisé de manière historique

PHP 4.3.8 + apc cache 3	Tomcat 4.1.29					
Apache 2.0.48		Mysql 4.1.7	CVS 1.12.9	Openldap 2.1.22	Jonas 4.1.2	Sybase 12.0.7
Linux Red Hat Enterprise Server 3.0						Solaris 5.8
Intel/Compaq DL360 G4						Sun Fire v440

# 3. Plate-forme de développement

- Les outils de développement
  - Eclipse
    - Environnement de développement multi-langages
    - Evolutif, nombreux plugins
  - Caameleon : Framework SITS
    - mutualisation des composants
    - Structuration des sites
    - Gestion du contenu éditorial, des news, des documents et du multilinguisme
    - Gestion de l'authentification (plugin LDAP)
    - Gestions des profils et des accès, des sessions et de la navigation
    - Monitoring, statistiques, versionning, livraisons packagées...
- Contexte projets
  - Relation MOE/MOA en flux tendu, priorité au changement
    - Mantis : Workflow déstresseur de la relation MOE/MOA
  - Homogénéité des développements
    - Normes de développement standardisées
    - Forte communication, points d'équipe réguliers autour de thèmes techniques
  - Passage de connaissances en permanence
    - Gestion du turn-over facilitée
    - Pas de spécialisation des compétences
  - Réactivité forte aux demandes, pragmatisme et simplicité des solutions



[www.mantisbt.org](http://www.mantisbt.org)



# 3. Plate-forme de développement

- Normes de développements standardisées : phpBeau
  - Plugin pour Eclipse
  - Batch quotidien pour établir les statistiques et noter les progressions
  - Basé sur les “PHP Coding Standards”

```
function getLanguage( $option = '' ) {  
    $language =& ContextFactory::singleton( 'contextObject', 'lan  
    ContextFactory::doRetrieve( "language" );  
    $val = $language->getToSave();  
  
    // recup profil par defaut  
    $bdd_fw = ServiceLoader::singleton( 'connectbdd', 'fw', true  
    $res_language = $bdd_fw->requete( 'GET_DEFAULT_LANGUAGE', arr  
  
    if ( $res_language ) {  
        $language->setMe( $res_language[0]['ID_LANGUAGE'] );  
        ContextFactory::doSave( "language" );  
        return ( $res_language[0]['ID_LANGUAGE'] );  
    }  
  
    return ( false );  
}  
  
function setLanguage( $id_language ) {  
    $language =& ContextFactory::singleton( 'contextObject', 'lan
```

Disponible sur : <http://sourceforge.net/projects/phpbeau>

# 4. Retour d'expérience sur PHP

- Le langage PHP en lui-même
  - Simple
    - Typage faible
    - Syntaxe simple et peu contraignante
  - Performant
    - Rapidité d'exécution
    - Montée en charge aisée
  - Complet :
    - Interconnexion XML, Ftp, Java, Mail, Shell, \*SQL, ...
    - Environnements de développement aboutis : Eclipse, Zend Studio, PHPEdit
    - Bibliothèques de frameworks et de composants prêts à l'emploi : PEAR
  - Efficace
    - Fonctions natives très complètes
    - Implémentation intuitive
  - Standard
    - Interconnexions selon les standards du marché
- L'environnement PHP
  - Les solutions opensource/php prêtes à l'emploi sont nombreuses et très souvent de qualité
  - Les espaces d'échange sur Internet assurent un support quasiment incollable
  - Les développeurs sont faciles à trouver, efficaces rapidement et bon marché

# 4. Retour d'expérience sur PHP

- L'adhésion en entreprise
  - systématiquement dans une approche bottom-up
  - le meilleur effet de levier reste le coût faible de la solution
    - peu coûteux à mettre en oeuvre/maintenir/faire évoluer.
    - prise de risque très faible car quasiment aucun investissement
  - nécessité de prouver avant tout
    - être chaque jour le commercial de PHP
    - être pragmatique et professionnel
  - il faut encore démystifier les discours “faciles”
    - *“l'opensource supprime les éditeurs, il n'y a donc plus personne pour assumer les problèmes”*
    - *“PHP, c'est bien pour faire des sites simples”*
    - *“si Mr X quitte la société, la plate-forme ne tiendra pas plus de deux mois”*
  - devient un formidable outil pour l'entreprise après adhésion
    - les décideurs prennent conscience de la souplesse que permet la technologie
    - bien positionné, PHP devient le “couteau-suisse” de l'entreprise

# 5. Conclusion

- La bonne démarche
  - Faire du PHP proprement et professionnellement
  - Pour faire du web, PHP apparaît comme étant la solution idéale
  - Utiliser des solutions éprouvées
  - Chercher à bénéficier des composants opensource existants
  - De manière générale, pas de PHP à tout prix
  - Rester Crédible
  - Evoluer en permanence
  - Etre à l'écoute
- Les évolutions en cours sur la plate-forme
  - Tests unitaires
  - Conception orienté objet plus poussé (PHP5 + design patterns)
  - Mise à disposition de CAAMELEON sous licence libre

# Des questions?

[julien.bernard@ca-sits.com](mailto:julien.bernard@ca-sits.com)