BabboukWeb

Présentation du 30/10/2008/IREMIA

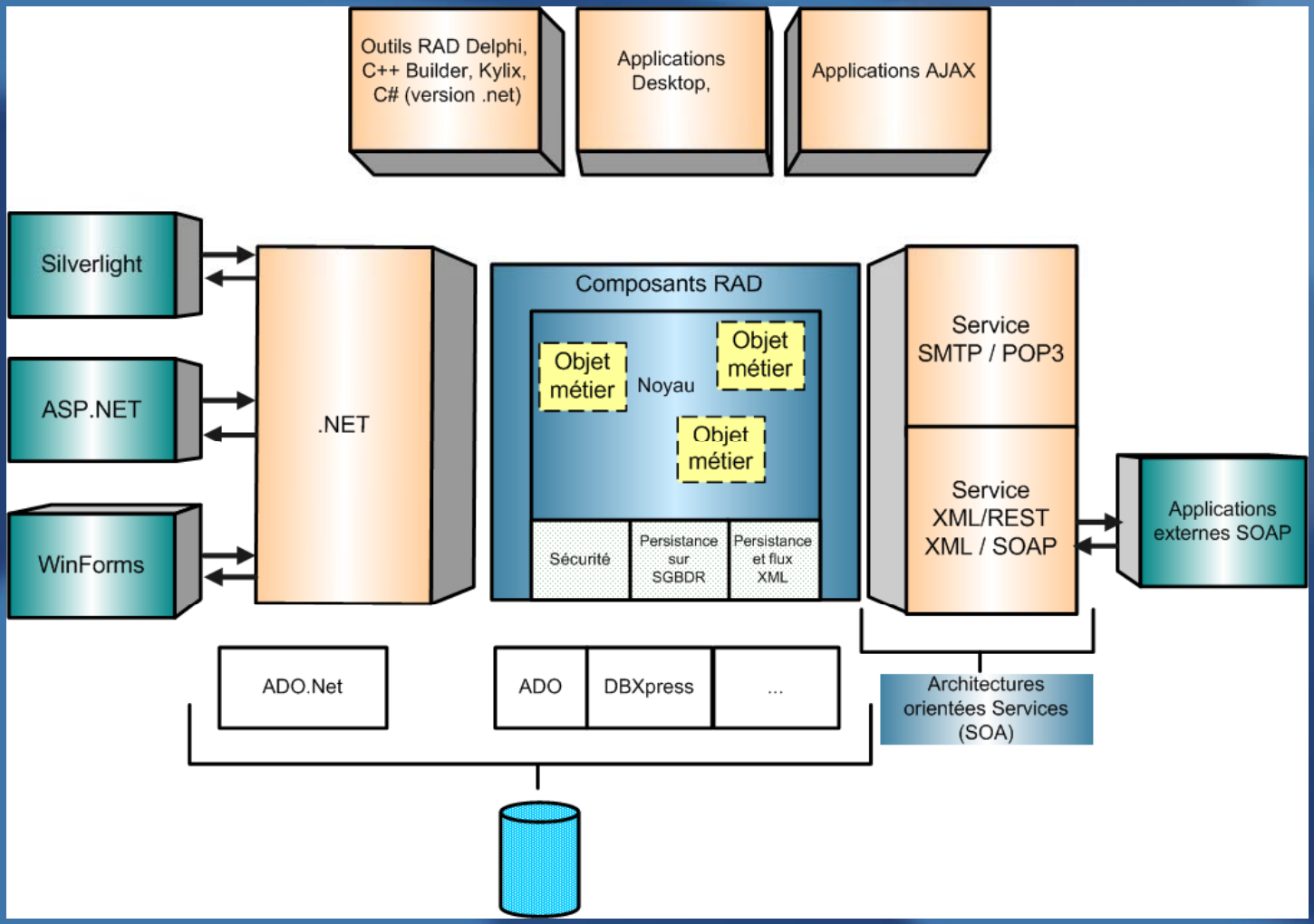
Anil CASSAM-CHENAI
M.I. TECHNOLOGIES

Objectifs du projet

- Réalisation d'une gamme de produits exportables
- Innovation
- Utilisation maximale de ressources locales
  - Compétences
  - Réseau
- En se mettant à un niveau technologique International
  - Importance de la connexion internet (permanence de la connexion plus que débit)

– Demo interface Desktop

Structure globale



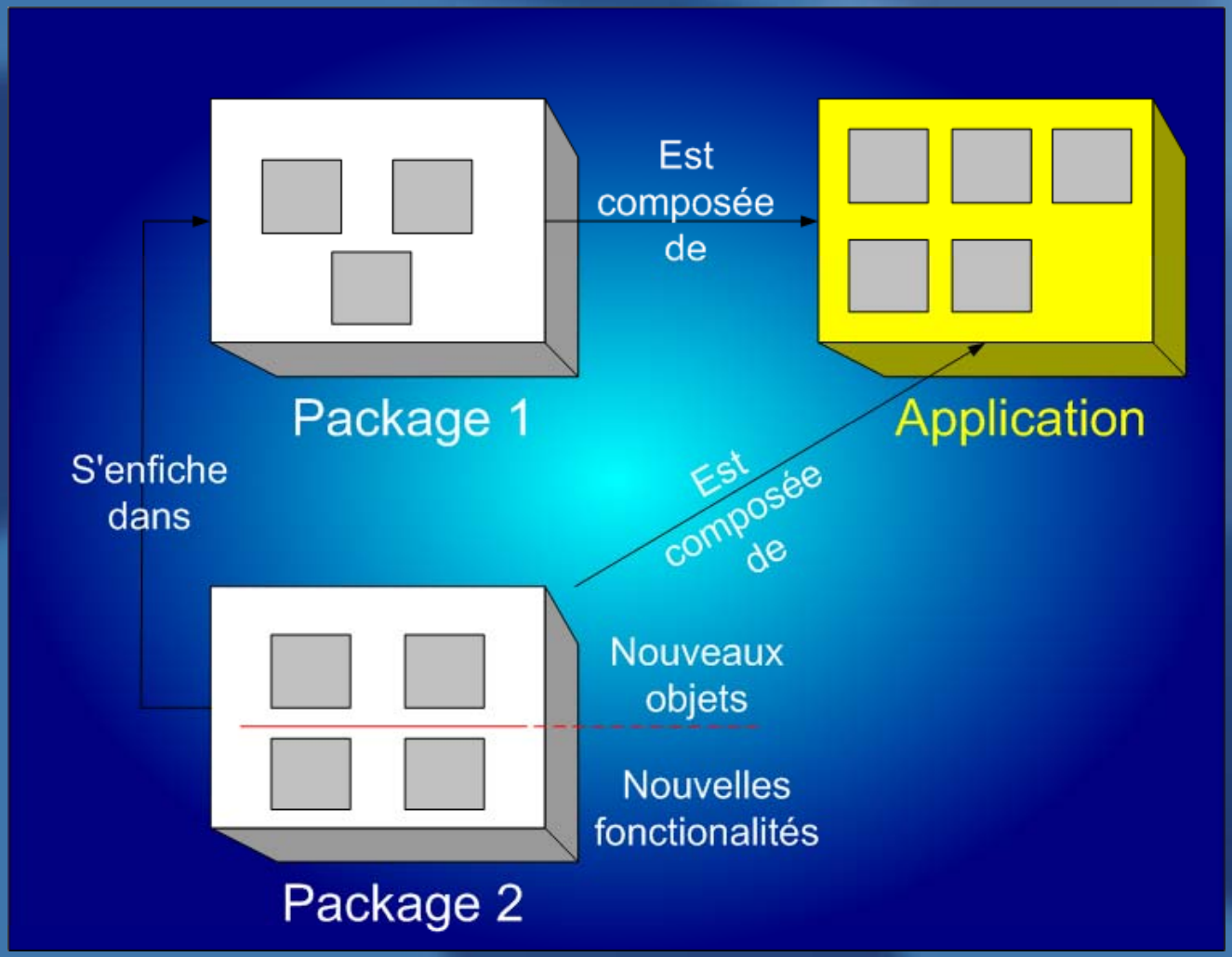
Architecture multiplateforme

- Pascal Objet
  - Natif:
    - Delphi, FPC (Free Pascal Compiler)
    - Portable
  - Compilateur .net
    - Delphi Prism
    - Compatible Mono
- Indépendance de la base de données
  - Base de donnée externe (Oracle, Access, Postgres, etc)
  - Base de donnée intégrée

Structure modulaire (1)

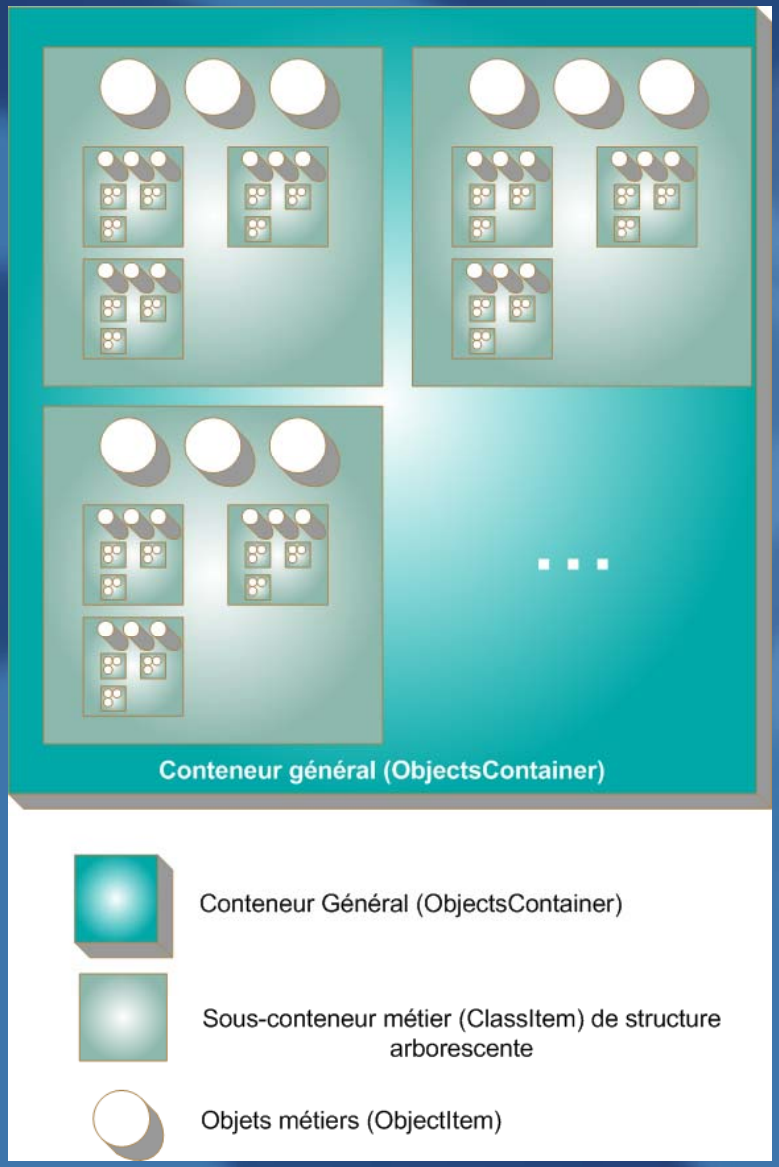
- Logique de plug-in
  - Le serveur d'application en lui-même n'a aucune fonction métier
  - La gestion documentaire, la comptabilité sont des plug-in du serveur d'application
  - Plusieurs plug-in sont chargeables de manière concurrente
  - Les plug-in sont téléchargeables et d'installation automatique (architecture P2P)

Structure modulaire (2)

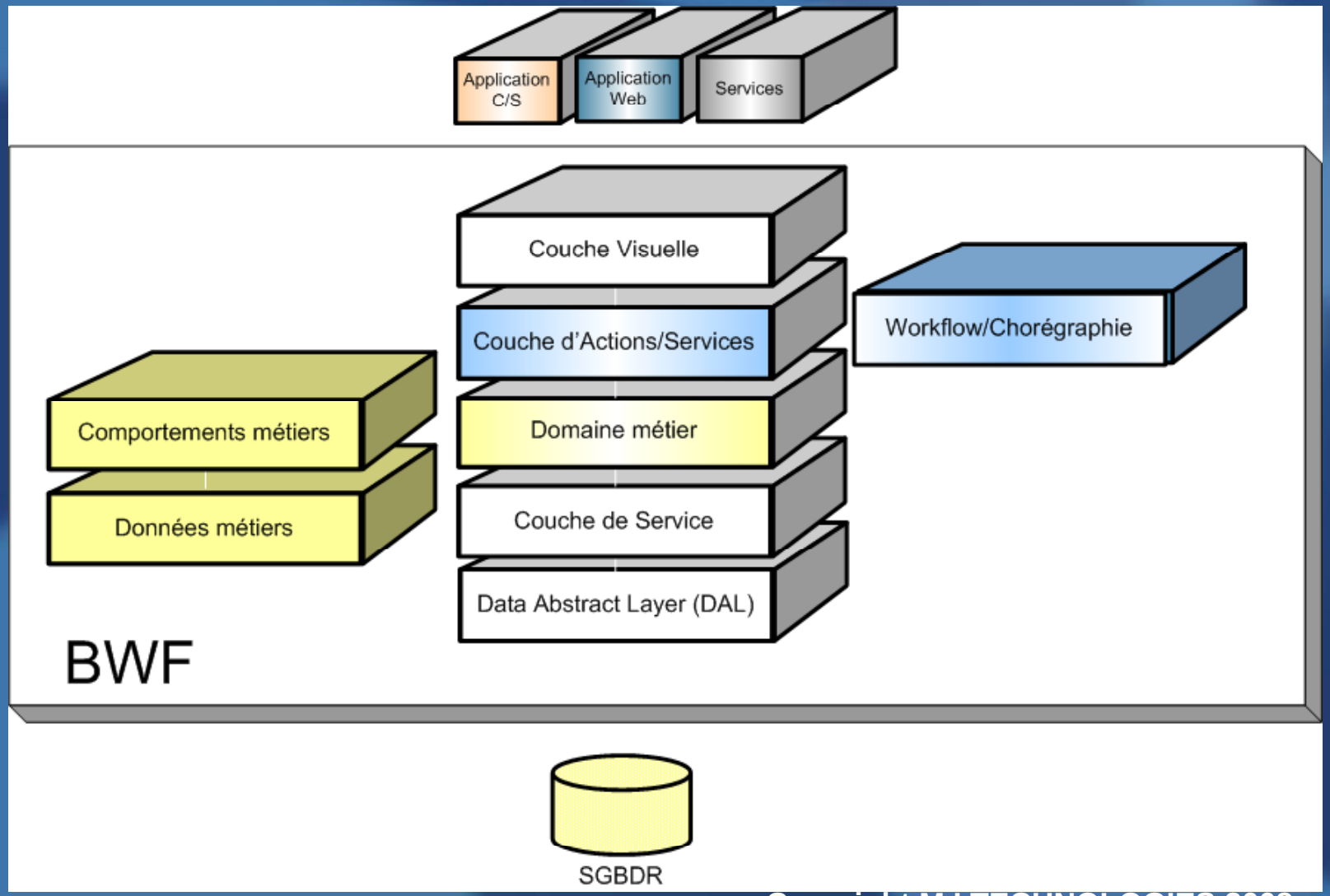


Des plug-ins standard

- Des applications métiers
  - Comptabilité
  - Gestion documentaire
  - Gestion commerciale
- Qui servent de preuve de concept
- Utilisables dans différents modes :
  - client serveur traditionnel
  - Interface Web
  - Sous forme de service Web (ASP)

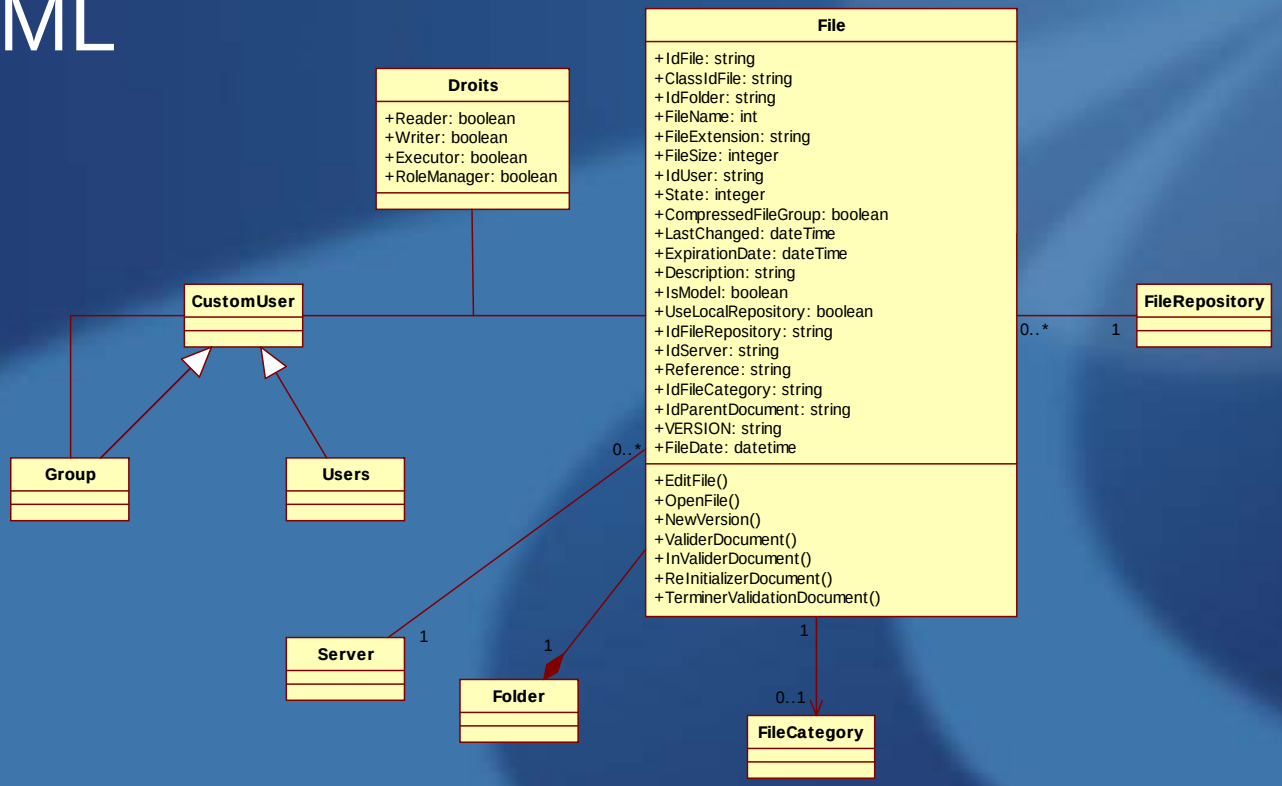


Architecture en couches



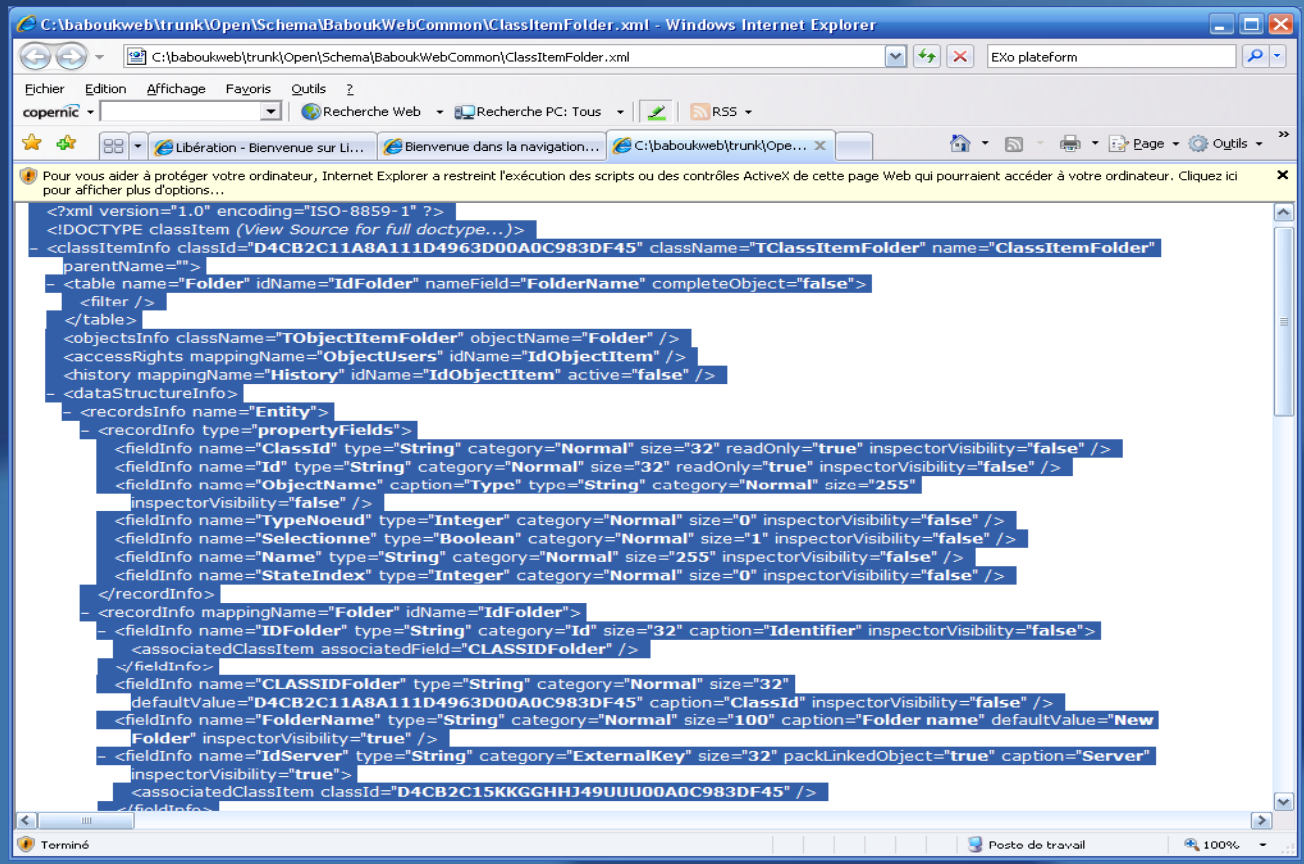
Métadonnées XML et MDA (1)

- Capacité de réaliser la structure de données en mode MDA : méta-données UML



Métadonnées XML et MDA (2)

– Transformation en méta-données BabboukWeb



Contenu des méta données

- Les méta données BaboukWeb contiennent à la fois :
  - Des informations sur la structure de données et le Mappage OR
    - Entités, Associations, Index, Requêtes standards (Collections)
  - Les états de l'objet (et les actions qui peuvent le faire migrer d'un état à un autre)
  - Les actions réalisables sur cet objet
  - Les droits d'accès à un objet
  - Les évènements d'un objet

Interêt des méta-données

- Paramétrables à la main
  - Données
  - actions
    - compilées
    - Scriptables
  - Etats et icône d'état
- Deux notions essentielles pour les plug-ins
  - Héritage des méta-données XML
  - Redéfinition des méta-données

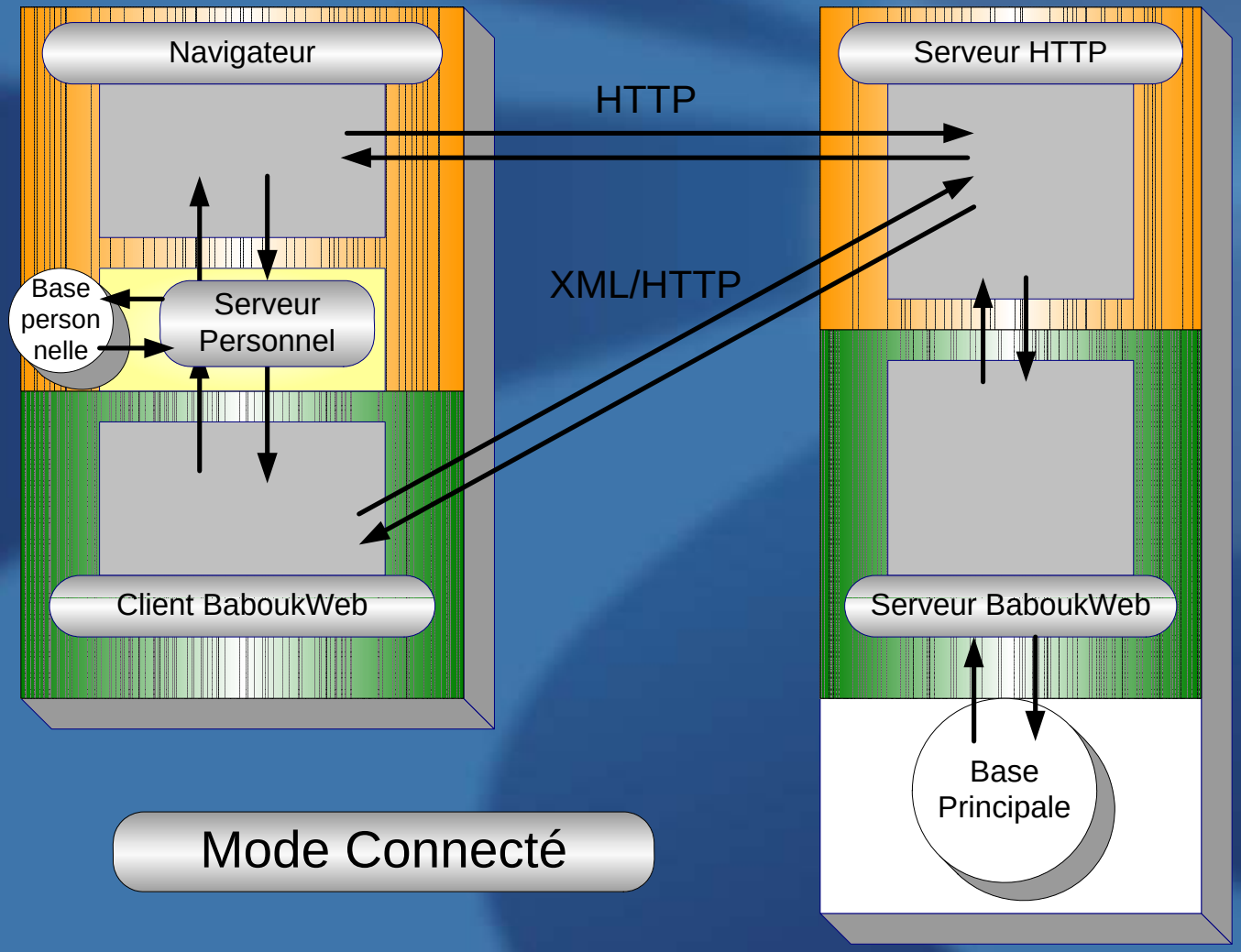
Et on peut aller plus loin …

- Définition d’applications via XML
  - On décrit l’assemblage de plug-ins qui donnent l’application
- Définition de plug-in via XML
  - On liste les métadonnées qui vont constituer les objets métiers, les requêtes et même des scripts
- Définition des interfaces via XML
  - Assemblage de frame prédéfinies
  - Au niveau application et fiche d’objet

Architecture de communication

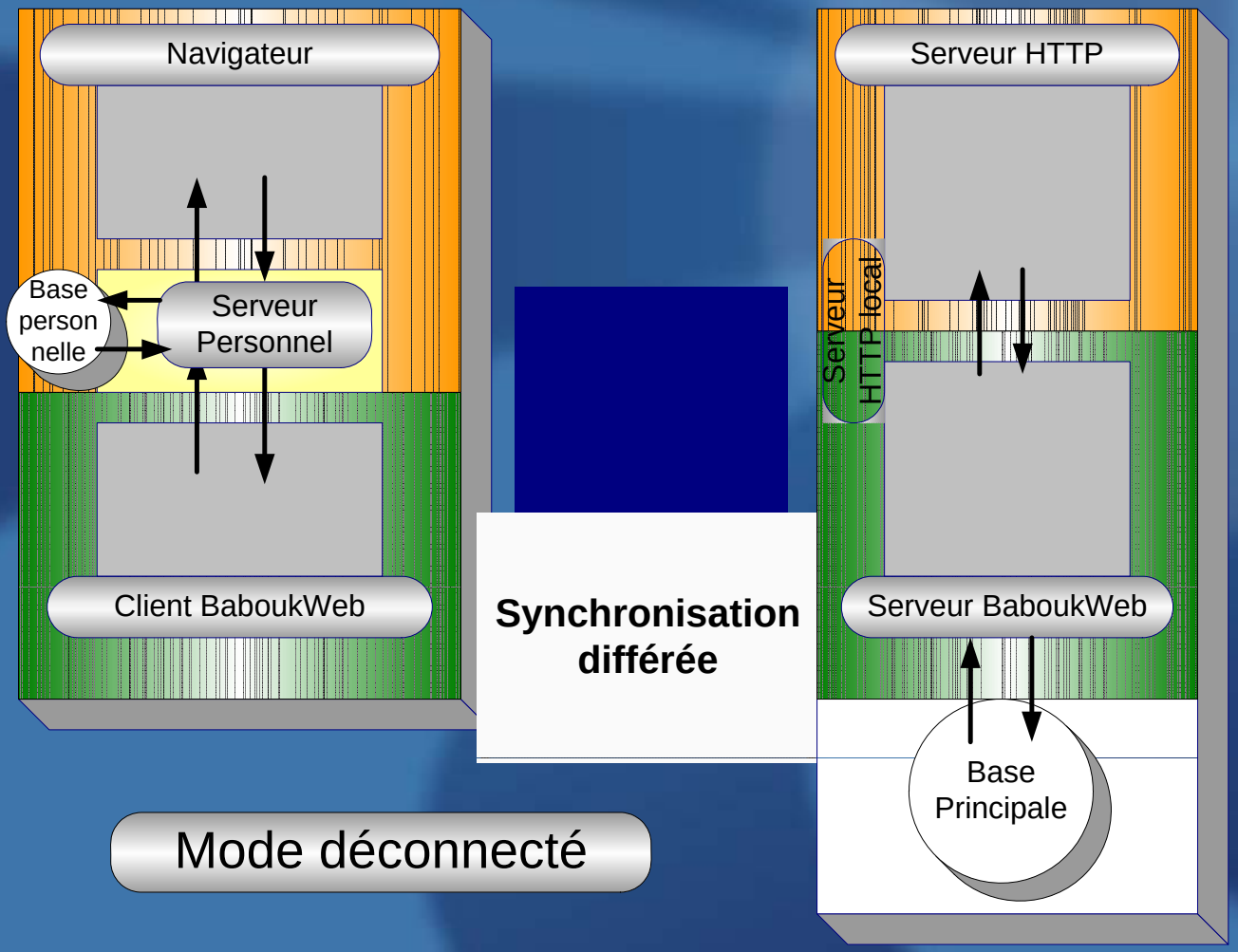
- Mode client/serveur
  - Architecture connectée
  - Architecture déconnectée
- Mode P2P (en cours de conception)
  - Information répartie sur un réseau de pairs
  - Echanges
    - De données
    - D’actions
    - D’évènements
    - De plug-in

Architecture C/S connectée

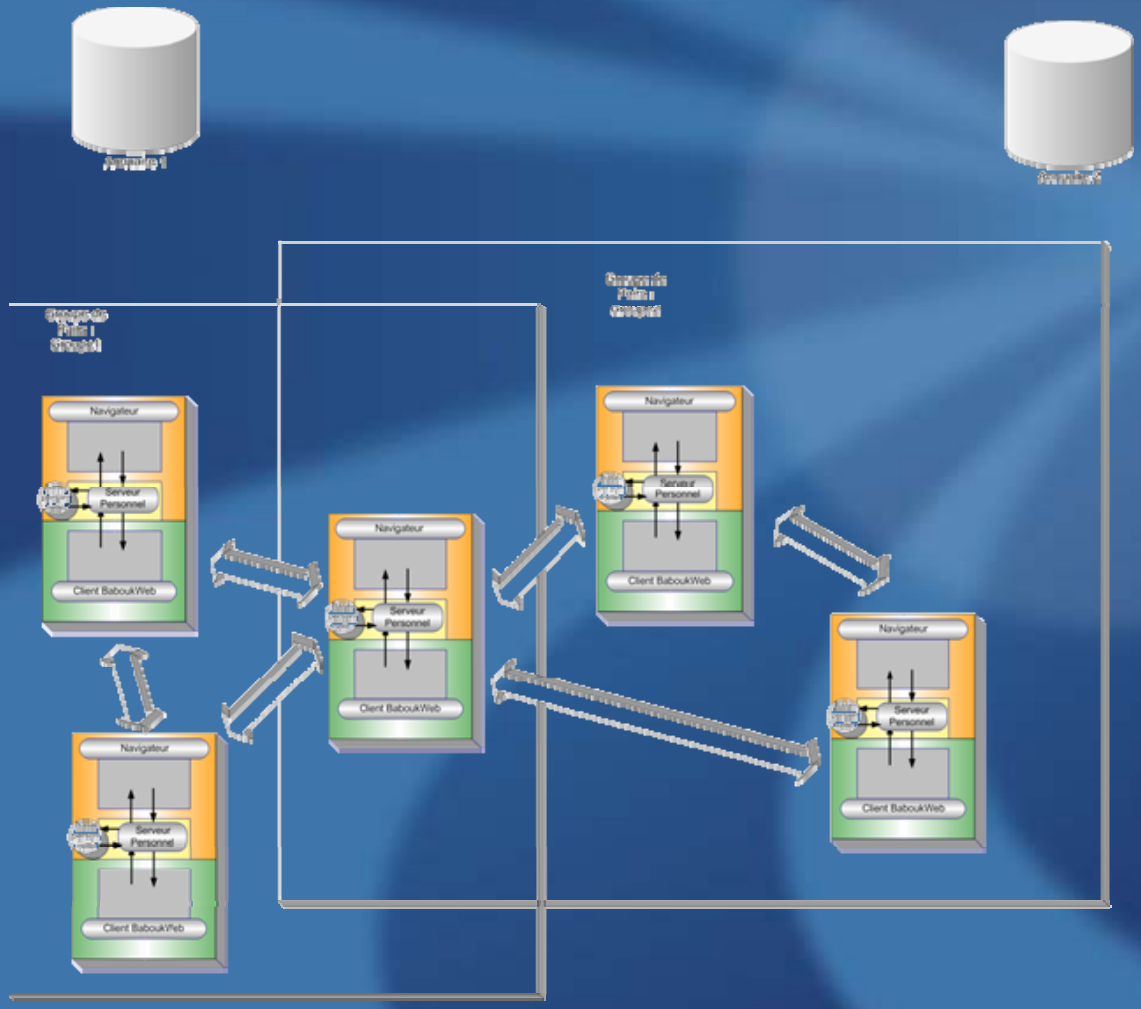


Mode Connecté

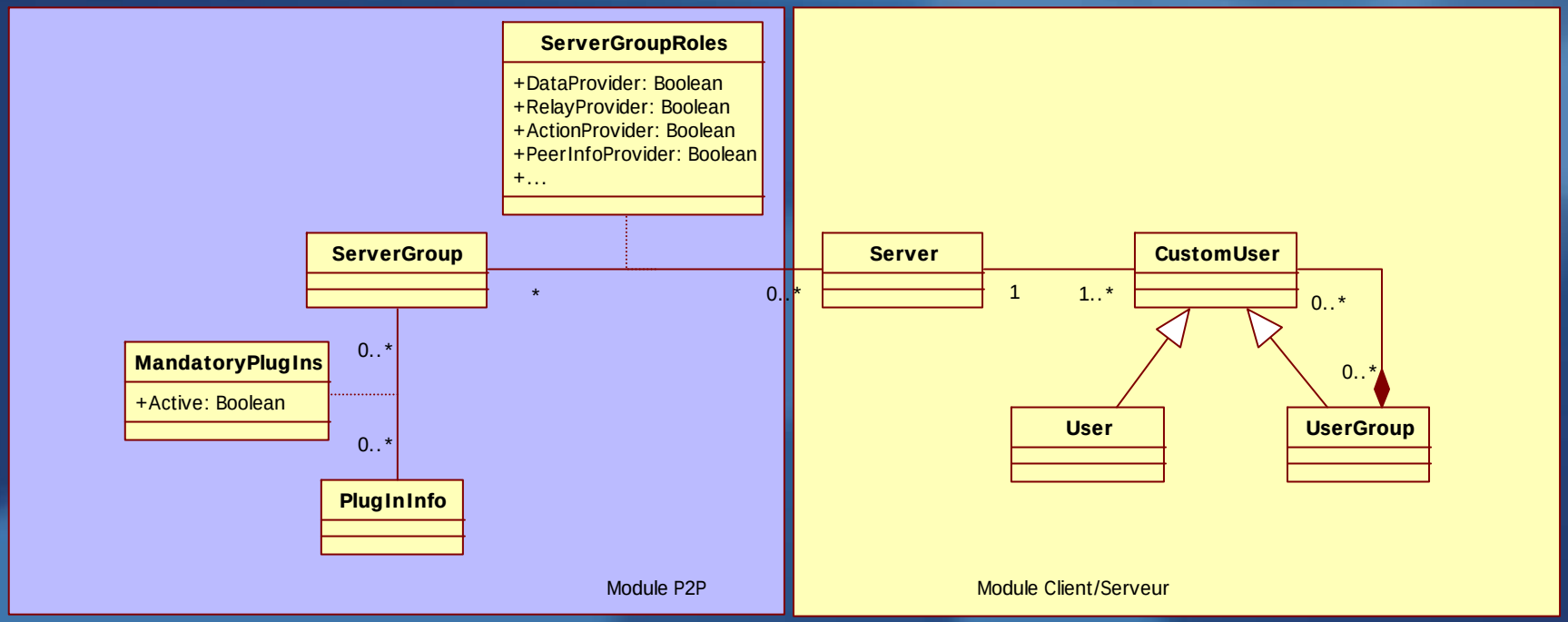
Architecture C/S déconnectée



Structure P2P



Architecture P2P (en cours de conception)



Différents rôles de pairs

- Fournisseur de données (DataProvider)
  - Fourni des données aux autres pairs (en respectant les droits de l'utilisateur connecté sur le pair distant)
- Serveur d'operation (ActionProvider)
  - Peut exécuter des traitements sur la demande du poste distant (scripts et tâches de workflow)
- Serveur de relai (RelayProvider)
  - Permet de relayer des demandes d'autres pairs dans le cadre d’un groupe de pairs. Permet de contourner des firewall
- Serveur d'annuaire (PeerInfoProvider)
  - Contient des informations sur les pairs connectés. Permet aussi l’inscription à un groupe de pairs.

Architecture de communication

- Framework de développement apportant une abstraction au niveau des services et des échanges
  - Canaux de communication
    - En standard utilisation de canal de communication HTTP/HTTPS
    - Flux de données au format XML
  - Groupes de services
- Mais potentiellement
  - Serialization+Canaux binaires sans modification du framework

Système de vue (1)

- **Séparé du modèle : MVC**
- Les vues sont définies en XML
  - Comme Xul, Xaml, etc
- **“migrable”**
  - Elle sont faites pour migrer sur le réseau entre serveur et client ou entre pairs
    - Via la migration de plug-in avec de vues fortement liées
    - Via une requête sur une vue ayant un chemin non local (possibilité de centralisation de vues,…) dans une architecture C/S

Système de vue (2)

- **“Fonctionnelle”**
  - Une décomposition fonctionnelle. Une fonction peut être représentée par deux composants différents
- **Mutiplateforme**
  - On choisit le composant dans une bibliothèque compatible avec la plateforme cible
- Compatible Web 2.0
  - Interface Desktop et Web (en cours basé sur Tibco)

Structures de donnée

- BaboukWeb offre une structure de données répartie entre plusieurs postes
- Les données sont protégées par les règles de sécurité du serveur d'application
- On dispose de deux modes de requêtage locaux et distants sur les objets
  - Mode type Xpath (vision arborescente. Grosse base XML répartie)
  - Mode utilisant l'algébre des collections/providers
  - Les deux peuvent être mélangés
- Illustration des concepts

Les actions

- On manipule des actions au niveau le plus haut de la structure en couche avant les vues
- Intérêt :
  - Undo/Redo
  - Rôles et limitation des accès
  - Opérations distribuées (dont même opérations séparable en éléments traités indépendemments sur divers serveurs)
  - Scriptable

Le moteur de script (1)

- Raison d’être :
  - Compléter le système des vues XML, en permettant la création d’actions scriptées
  - Ces actions scriptées sont accessible dans un autre script en connaissant son chemin. Donc on a une richesse de conception similaire à celle d’un langage objet
  - Les action scriptées peuvent être distantes
- Le moteur de scripts donne aussi des actions élémentaires de bas niveau chorégraphiables dans un deuxième temps par un moteur de Workflow

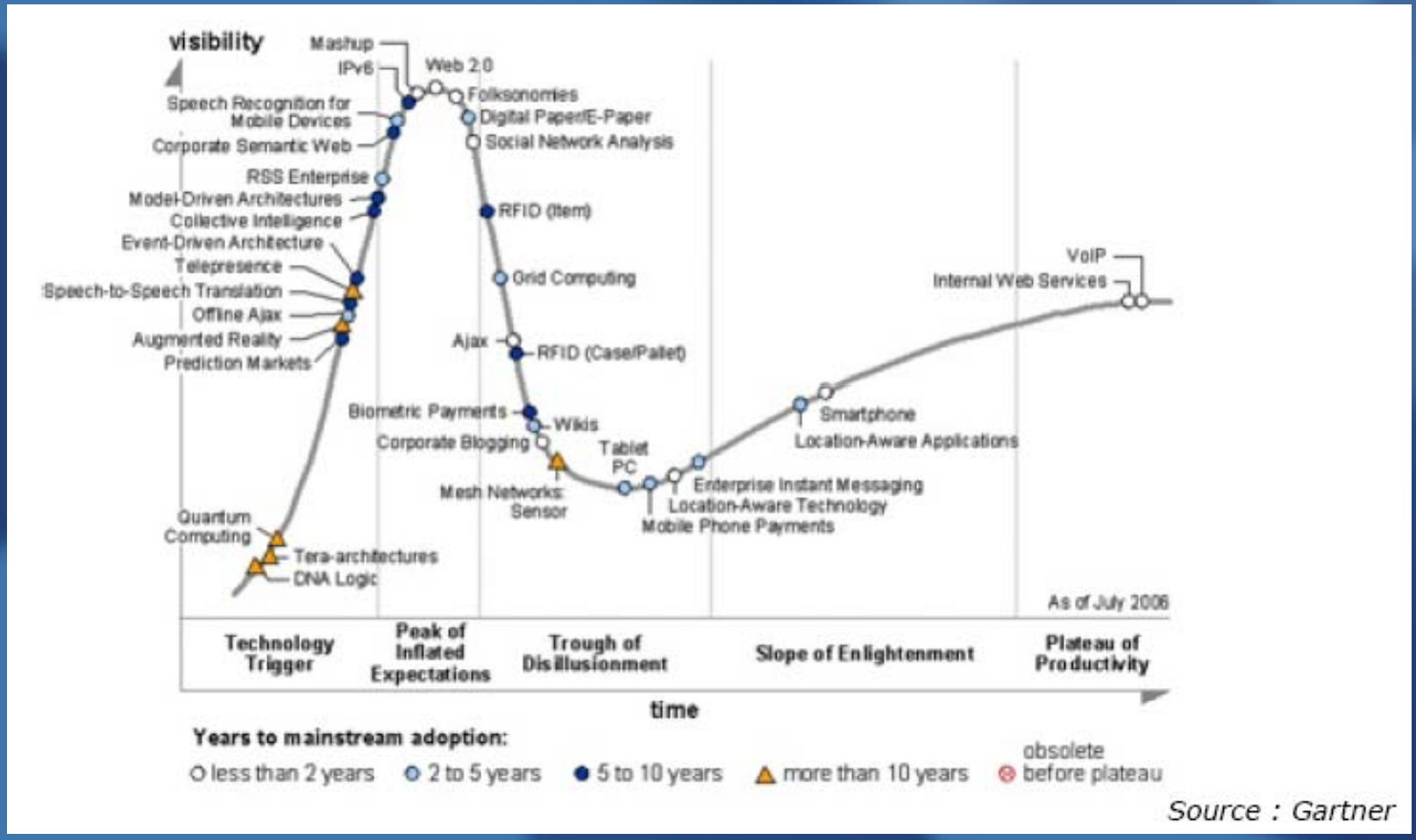
Le moteur de script (2)

- Exemple
  - *If /ClassItem[‘BaboukWebCommon.Folder’]/ObjectItem[@name=‘racine’]/*
    - *Field[@name=‘selected’]=true then*
  - *begin*

    /ClassItem[‘BaboukWeb.Common’]/ObjectItem[@name=‘racine’]/

    Action[@name=‘createSubFolder’](‘Mon dossier’);
  - *end*
- Il existe la possibilité une notation abrégée en posant comme principe que la notation pointée correspond à l’appel d’un élément de chemin (paramétré)stocké dans un dictionnaire associé à un objet

Le futur proche



Source : Gartner