nuxeo | Professional Open Source ECM

Nuxeo Runtime

Bogdan Stefanescu - Solutions Linux 2007

Qui suis-je ?

- Bogdan Stefanescu
  - mailto:bs@nuxeo.com
- Architecte / Team Leader chez Nuxeo
  - http://www.nuxeo.com/
- Lead Architect et Team Leader du projet Apogée
  - http://apogee.nuxeo.org/
- Lead Architect de Nuxeo Runtime et des projets ECM Core
  - http://www.nuxeo.org/

nuxeo

Plan

- Nuxeo Runtime - Introduction
  - Court historique
  - Inconvénients de J2EE
  - Qu'est-ce ?
  - Fonctionnalités
- OSGi
  - Qu'est-ce que OSGi ?
  - OSGi Bundle Manifest
- Nuxeo Runtime - Implémentation
  - Packaging
  - Composants, points d'extension
  - Plans futurs

nuxeo

Nuxeo Runtime - court historique

- Nuxeo a migré de Python (Zope) vers Java
  - CPS est un produit ECM open source très dynamique et configurable
  - CPS a été complètement réécrit en Java EE
- Nuxeo EP 5 est la nouvelle génération de CPS basée sur les dernières avancées technologiques Java EE
  - JBoss Application Server
  - EJB3, JSF, JBoss Seam
- Nuxeo est donc passé d'une platforme très dynamique à une technologie plus puissante, mais moins flexible
  - Comment retrouver la même flexibilité qu'avant ?
  - Nuxeo Runtime !

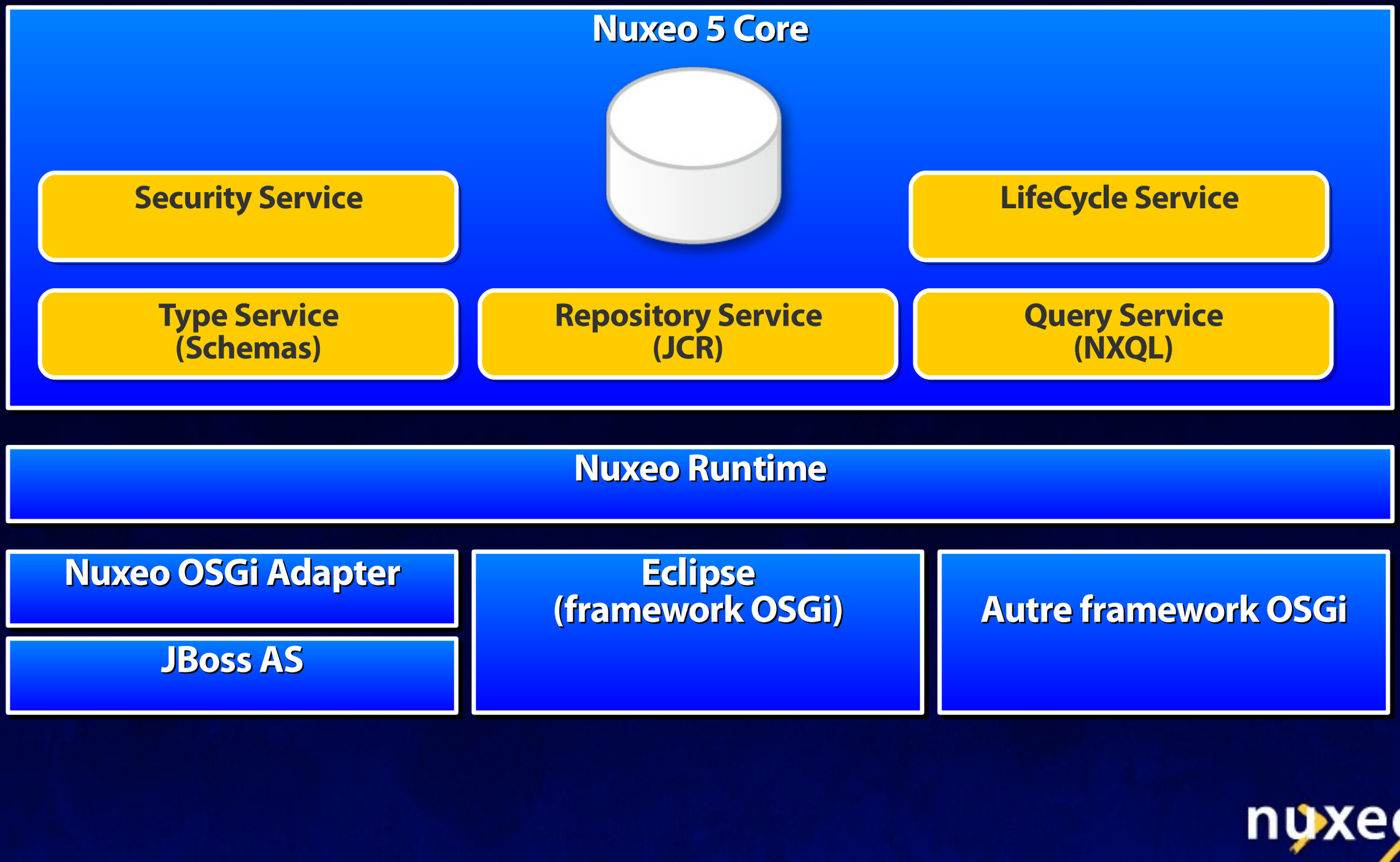
Inconvénients de Java EE

- Manque de flexibilité au niveau du déploiement
  - Les descripteurs des modules (XML) ne permettent pas la fragmentation des modules
  - Références en dur vers les noms des packages dans les descripteurs XML des modules ou dans le code
  - Impossible de rajouter une fonctionnalité sans devoir reconstruire et redéployer entièrement l'application
- Pas de modèle d'extensions (plugins)
  - Impossible de rajouter dynamiquement des extensions sans modifier les modules déjà déployés
  - Pas de notion de dépendance entre modules
  - Pas de support pour le versioning des modules
- Modules difficilement réutilisables sur autres platformes non Java EE

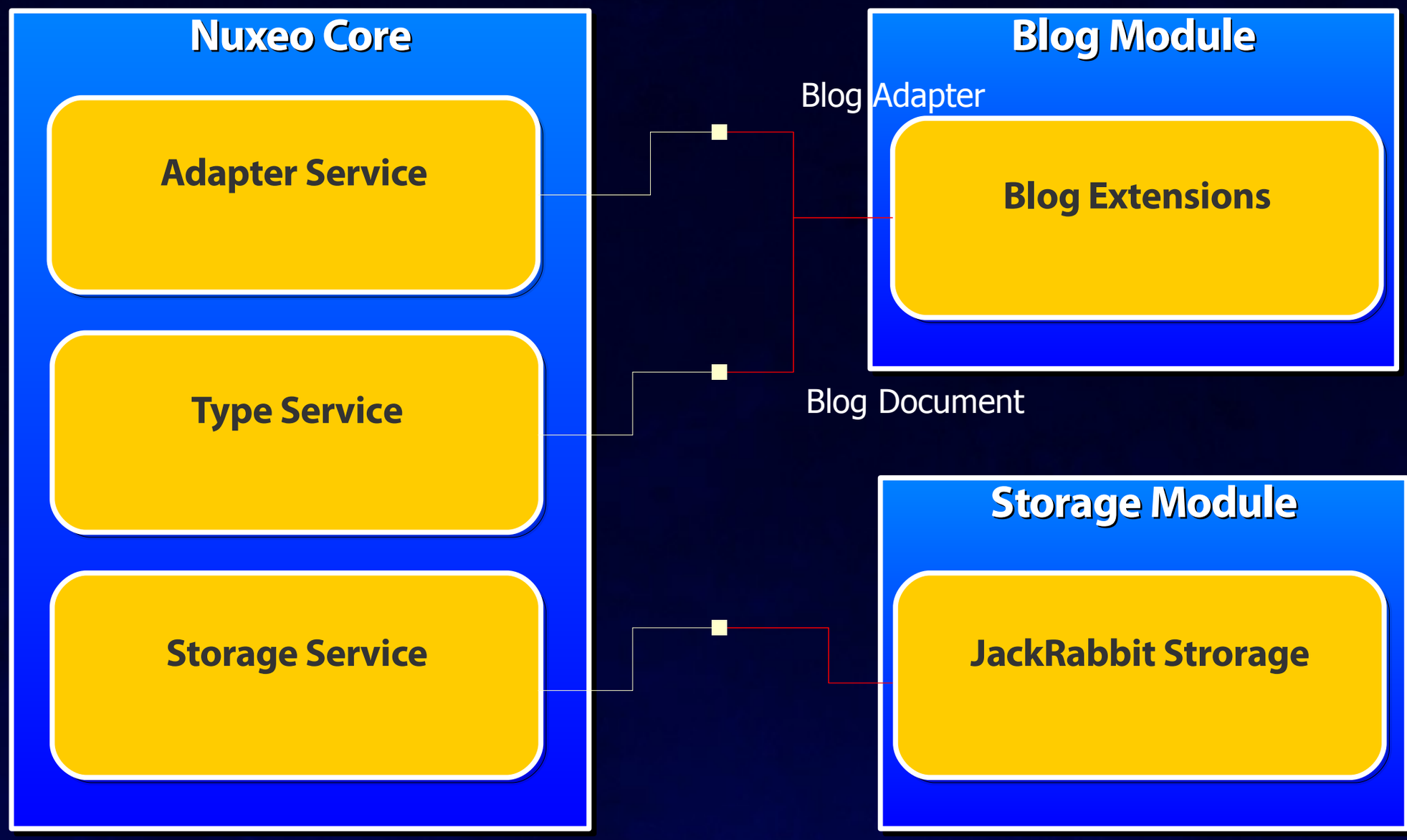
Nuxeo Runtime ?

- Un modèle de composants extensibles et *déployables partout*
- Rassemble des concepts déjà existants pour un gain en flexibilité et en dynamisme dans les applications Java (EE)
  - Composants descriptifs et points d'extensions
  - OSGi
- Infrastructure technique de Nuxeo 5
  - Definit le modèle de composants utilisé par Nuxeo 5
  - Assure le dynamisme et la flexibilité perdus lors du passage à JEE
- Rend les composants multi-plateformes
  - Le cœur de Nuxeo 5 fonctionne sur JBoss AS et Eclipse RCP (Equinox) *sans aucune modification ni recompilation*

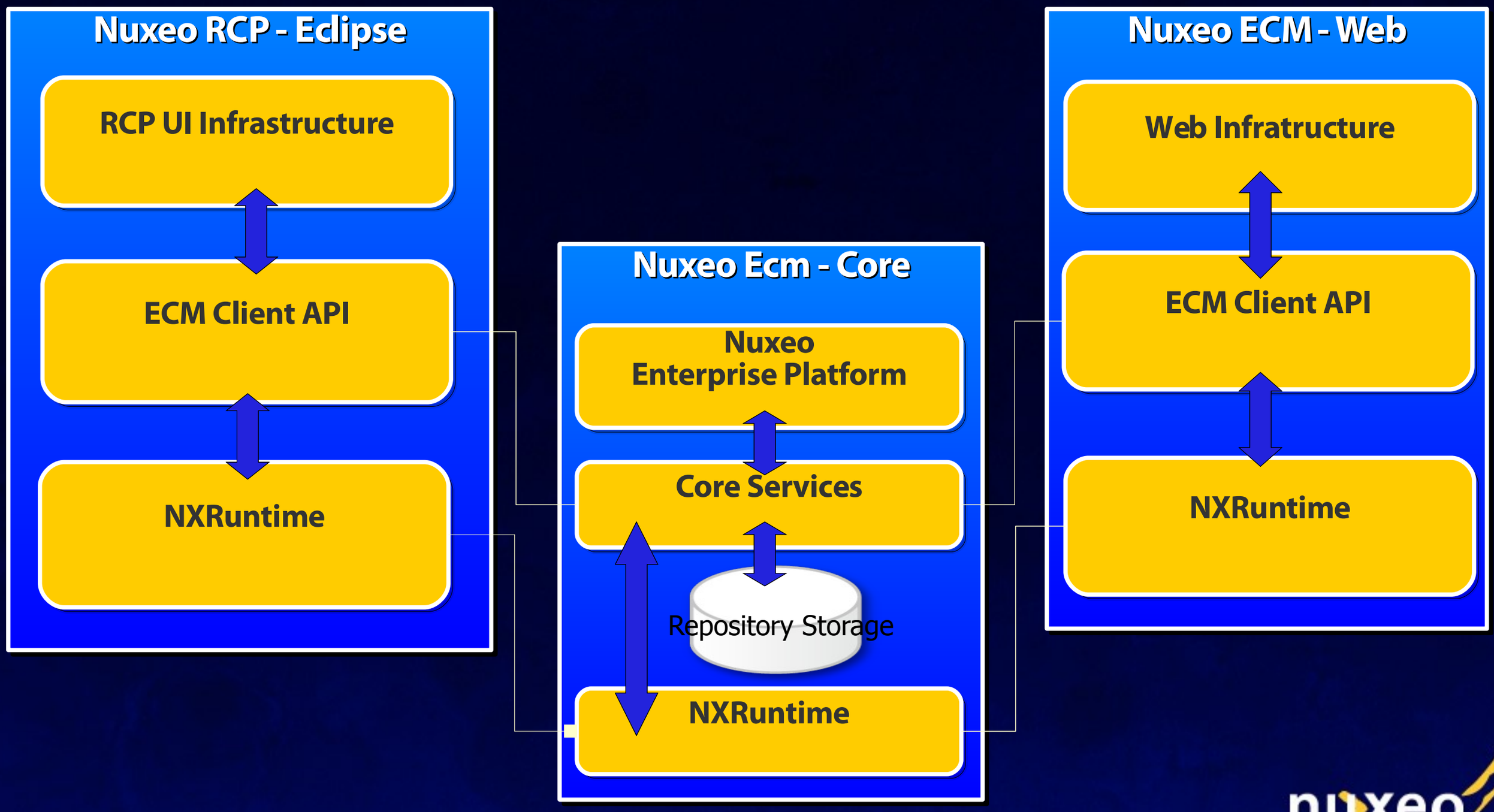
Nuxeo Runtime – Nuxeo 5



Nuxeo Runtime – Nuxeo 5



Nuxeo Runtime – Nuxeo 5



Nuxeo Runtime – fonctionnalités

- Modularité – les composants peuvent être installés et désinstallés à tout moment et dans n'importe quel ordre sans affecter le reste de l'application
- Extensibilité – les composants peuvent contribuer de nouvelles fonctionnalités à d'autres composants ou même les remplacer
- Compatibilité OSGi – Nuxeo Runtime peut être installé sur n'importe quelle platforme compatible OSGi et utilise le modèle de bundle OSGi
- *Deployable partout* – Nuxeo Runtime peut être installé sur n'importe quelle platforme dès qu'un adapteur OSGi existe

Nuxeo Runtime – remoting

- NXRuntime apporte un système de déploiement multi-machines pour faciliter la mise en place d'architecture Java EE modulaires et distribuée
  - Registry de composant partagée
  - Distribution des extensions et des ressources associées aux machines du cluster
- Détection automatique des nouveaux noeuds
  - Posibilite de remplacer le système de communication et detection par des protocoles tels que Jgroups ou JXTA
- Cluster de composants Nuxeo Runtime
- Accès aux composants depuis des machines distantes

nuxeo

OSGi – c'est quoi ?

- OSGi = Open Services Gateway initiative
  - http://www.osgi.org/
- Un ensemble de spécifications définissant une plateforme de services pour Java
- Principales fonctionnalités
  - Déploiement modulaire basé sur deux types de modules nommés *bundles* (archives .JAR) et *fragments* (fragments d'un bundle maître) qui contiennent un MANIFEST spécifique
  - Un module (ou *bundle*) OSGi peut définir plusieurs services (ou composants)
  - Un module est identifié par un nom symbolique unique. Exemple: org.nuxeo.ecm.core
  - Un module peut déclarer des dépendances sur d'autres modules

OSGi - Bundle Manifest

```
Manifest-Version: 1.0
Bundle-ManifestVersion: 2
Bundle-Name: Nuxeo ECM Core
Bundle-SymbolicName: org.nuxeo.ecm.core;singleton:=true
Bundle-Version: 1.0.0
Bundle-Vendor: Nuxeo
Bundle-Localization: bundle
Bundle-Activator: org.nuxeo.ecm.core.NXCoreActivator
Bundle-ClassPath: ., lib/xsom.jar,
 lib/connector-api.jar,
 lib/java-cup-v11a.jar
Export-Package: org.nuxeo.ecm.core,
 org.nuxeo.ecm.core.api,
 org.nuxeo.ecm.core.api.local,
 org.nuxeo.ecm.core.jca,
 org.nuxeo.ecm.core.lifecycle,
 org.nuxeo.ecm.core.model
Require-Bundle: org.nuxeo.ecm.core.api,
 org.nuxeo.runtime
Nuxeo-Component: OSGI-INF/CoreService.xml,
 OSGI-INF/TypeService.xml,
 OSGI-INF/RepositoryService.xml,
 OSGI-INF/CoreExtensions.xml,
 OSGI-INF/SecurityService.xml
```

nuxeo

Nuxeo Runtime – implémentation

- Packaging
- Les composants
- Points d'extension
- Platformes supportées
  - Toutes platformes compatible OSGi (dont Eclipse)
  - JBoss Application Server

Nuxeo Runtime – packaging

- Le packaging se fait sous forme de *bundles* OSGi
- Un module (ou *bundle*) peut définir plusieurs composants
- Un composant est défini par l'intermédiaire d'un fichier descriptif XML
- Nuxeo Runtime peut aussi être utilisé en dehors d'une platforme OSGi
  - Ceci simplifie le test des composants mais aussi l'intégration dans n'importe quel type d'application Java
  - Ouvre la possibilité d'intégration avec des plateformes basiques telles que Tomcat

Nuxeo Runtime – les composants

- Chaque composant est défini par un descripteur XML
- Un composant peut déclarer des dépendances sur d'autres composants
  - Le composant va être installé et démarré seulement quand toutes ses dependances auront été resolues
  - Les composants sont desinstallés dans l'ordre inverse de leur installation
- Un composant peut déclarer plusieurs point d'extensions pour permettre à d'autres composants de contribuer des extensions
- Un composant peut déclarer plusieurs extensions pour n'importe quel composant qui définit des points d'extension

Descripteur XML

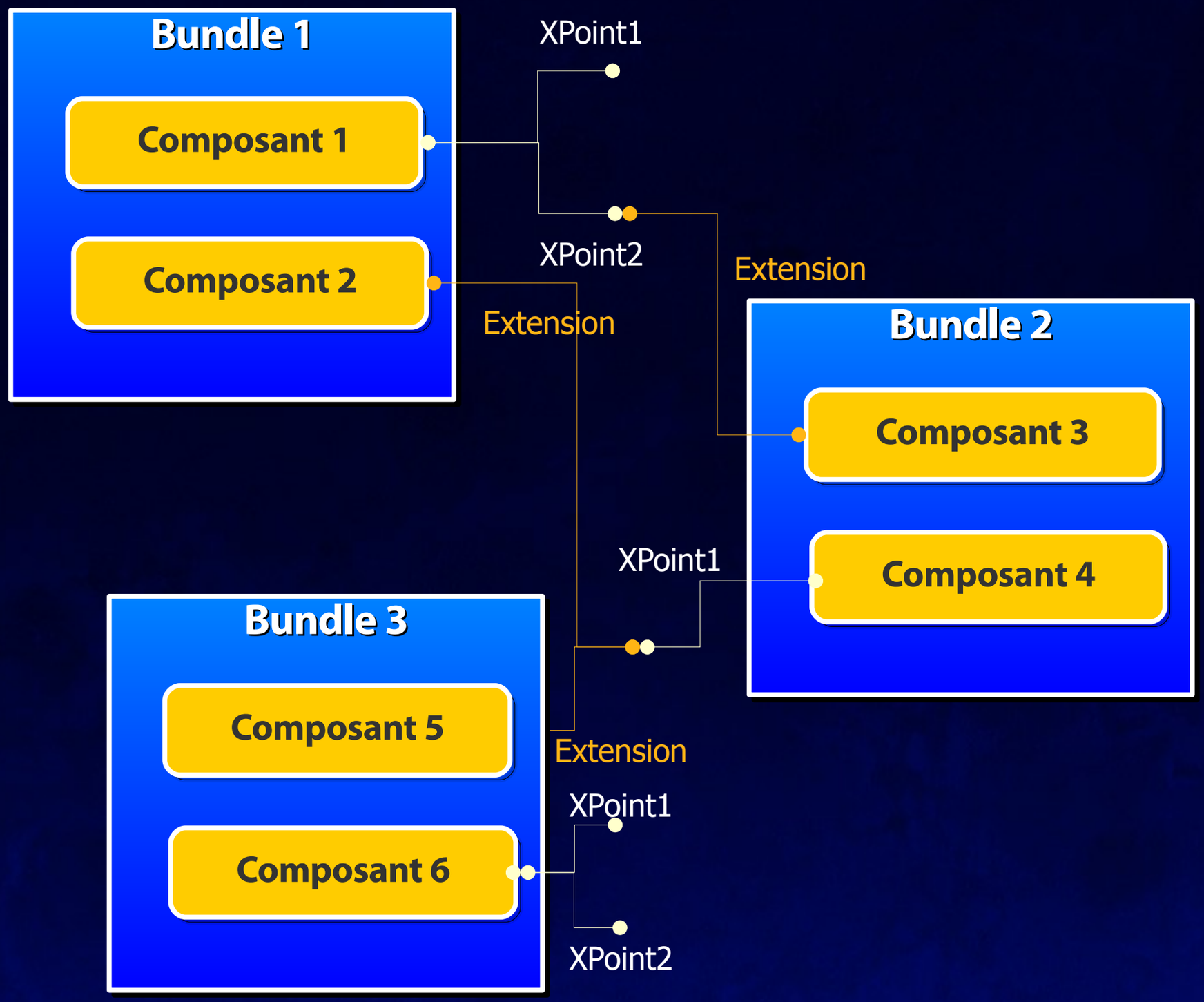
```
<?xml version="1.0"?>

<component name="org.nuxeo.ecm.core.schema.TypeService">

    <implementation class="org.nuxeo.ecm.core.schema.TypeService"/>

    <extension-point name="doctype">
        <object class="org.nuxeo.ecm.core.schema.DocumentTypeDescriptor"/>
    </extension-point>

    <extension-point name="schema">
        <object class="org.nuxeo.ecm.core.schema.SchemaBindingDescriptor"/>
    </extension-point>

    <extension-point name="schema">
        <object class="org.nuxeo.ecm.core.schema.SchemaBindingDescriptor"/>
    </extension-point>

    <extension target="org.nuxeo.ecm.core.api.CoreService"
            point="sessionFactory">
        <factory class="org.nuxeo.ecm.core.api.local.LocalSessionFactory" />
    </extension>

</component>
```

nuxeo

Nuxeo Runtime – points d'extension

- Les points d'extension sont inspirés du mécanisme d'extension utilisé par Eclipse
- Un point d'extension est déclaré par un composant et permet l'extension du composant depuis l'extérieur
- D'autres composants peuvent ajouter des extensions au composant qui déclare des points d'extensions
- Cela permet l'ajout des nouvelles fonctionnalités ou la configuration d'un composant depuis l'extérieur et d'une manière flexible, sans modifier le code du composant ou le réinstaller
- Une application devient un ensemble de composants qui s'étendent mutuellement!

Points d'extensions



nuxeo

Nuxeo Runtime – l'avenir

- Versioning
  - Gérer les versions des composants dans les dépendances
- Alignement avec les services descriptifs OSGi
- Plus de serveurs d'application supportés
  - GlassFish (Reference Implementation)
  - Apache Geronimo (OSGi)
  - BEA WebLogic

nuxeo

nuxeo | Professional Open Source ECM

Merci !