« Presentation Layer »

Nuxeo 5 Web requirements - *Generation engine*

- **Powerful templating system**
  - Want to keep a templating engine with a composition system
- **MVC model, Widget and Layout system**
  - Clear separation of concerns *(Presentation / Validation / Model)*
  - Reusable and customizable widgets *(like in CPS)*
  - Complete layout system like with CPSSkins
- **Not tied to HTML**
  - Keep the possibility to generate non HTML data (ie : XForm)

nuxeo

Nuxeo 5 Web requirements - *the choice of JSF / Facelets*

- Great standard even if complex
  - Java EE Standard
  - Very pluggable and extensible

- Growing libraries of Widgets
  - from Ajax/HTML to DataModel

- IDE support is growing

- Facelets are close to Zope's PageTemplates
  - Templating and slot model
  - Pure presentation technology
  - Clean for designers

Java Server Faces

- JSF is used as rendering framework
  - High level and plugable model for web UI
  - JEE Standard
  - Facelets is a rendering engine for JSF like PageTemplates
- We use apache JSF implementation
  - Apache MyFaces 1.1.5
  - Tomahawk 1.1.5
- We use Ajax4JSF for Ajax integration
- NXThemes provides the CPSSkins features for JSF

Nuxeo 5 Web requirements - *Web Framework req.*

- **JSF and Facelets compliant**

- **Cache aware state management**
  - Each page rendering needs to fetch a lot of documents
    - But most documents are the same between two requests

    ➡ keep state of the user context
  - Data needs to be fresh and invalidated

- **Remote acces to the application controlers**
  - Access via Ajax
  - Access via Browser helper
  - Access via Office Plugin

Nuxeo 5 Web requirements - *Web Framework req.*

- **Clean EJB3 integration**
  - Easy access to ejb3 NXP Services
  - JSF binding
- **Loosely couple UI and application components**
  - Nuxeo 5 must provide hot-pluggables Web components
    - Shared context
    - Dependencies injections
    - Events system
- **Multiscreen and wizards management**
  - ECM provides a lot of use cases
  - Manual pageflow management is boring

JBoss Seam

- The missing link between JSF and ejb3
  - Uses ejb3 as Action Listeners (event handlers)
- Extends JSF to bring real context management
  - Keep all conversation related objects in cache
  - Context Bijection
- Extends JSF for navigation rules
  - Uses JBPM to define PageFlow
- Integrate JSON and prototype
  - Every ActionListener can be accessed via AJAX
- Part of Seam is in normalization
- A hype JBoss framework searching for it's killer app

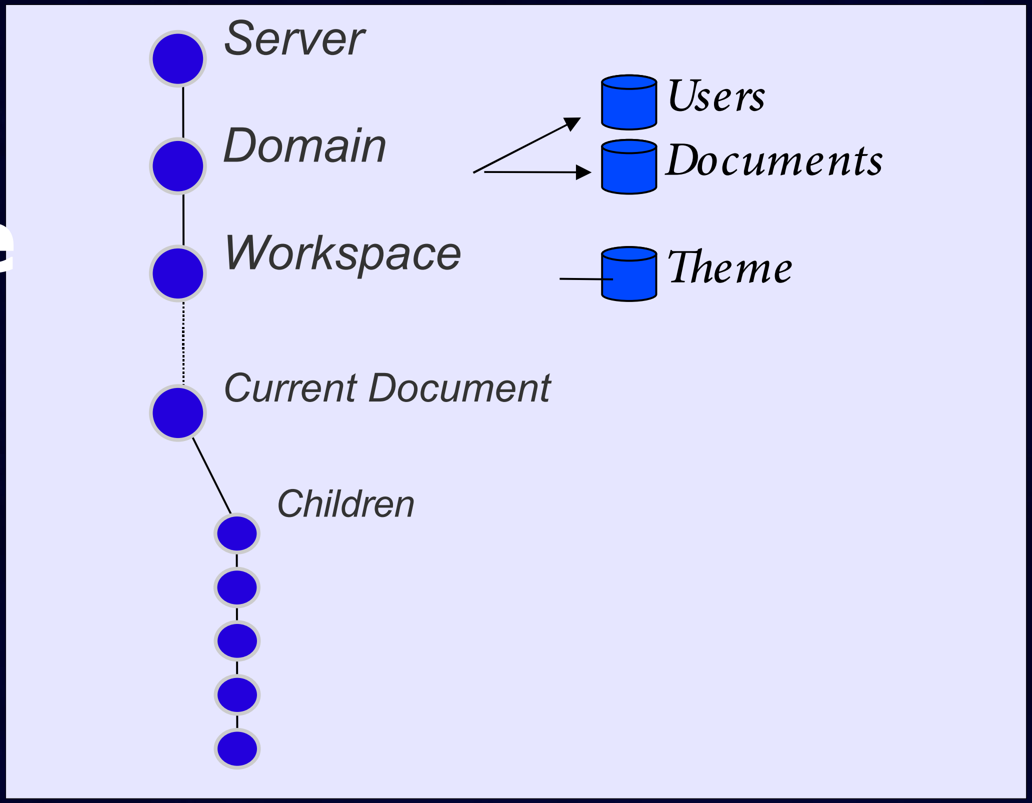
Nuxeo 5 Web Layer – *Why Seam ?*

- **Seam provides a lot of the needed features “out of the box”**
  - Native EJB3 integration
  - Bijection
  - Seam context management
  - Seam remoting and Ajax4JSF Integration
  - Pageflows via JBPM integration
- **Good fit with other selected infrastructure components**
  - JBoss, JBPM and JSF
- **Great project**
  - Good references for lead developers of Seam
  - Great vision

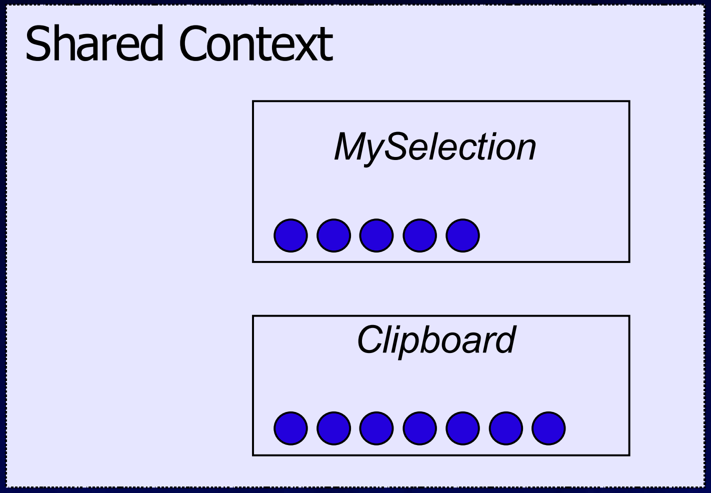
Nuxeo 5 - Context Management with Seam

- **Nuxeo 5 uses Seam so store contextual documents**

- **Partial content location tree**
  - Reading placeful configuration
  - Retrieving informations to display



- **Current working context**
  - Current document and childs
  - Clipboard
  - Working set of documents



Nuxeo 5 - Context Management with Seam

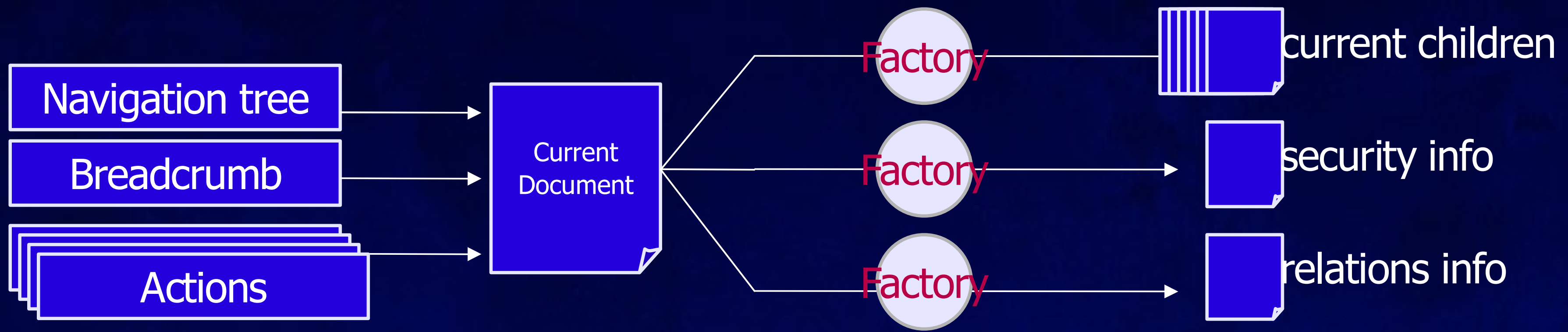
- **Seam context stores results of service calls**
  - DocumentModel (Core calls)
  - List of DocumentModels
  - Facets of a given DocumentModel
    - Security descriptors, Relations informations, Subcription configuration, ...
- **Context variable are usually pulled via factory**
  - Use of the Seam @Factory annotation
  - Use of @DataModel annotation to feed DataLists from context
- **Each component can access easily the contextual documents**
  - Ask for injection via @In

Nuxeo 5 - Context Management with Seam

- **AL get the current document by injection**

```
public abstract class InputController{

   @In(required = true)
   protected ECServer ecServer;

   @In(required = true)
   protected ECDomain ecDomain;

   @In(required = true)
   protected ECContentRoot ecContentRoot;

   @In(required = false)
   protected DocumentModel currentItem;
```

- Navigation controlers outject the currentDocument

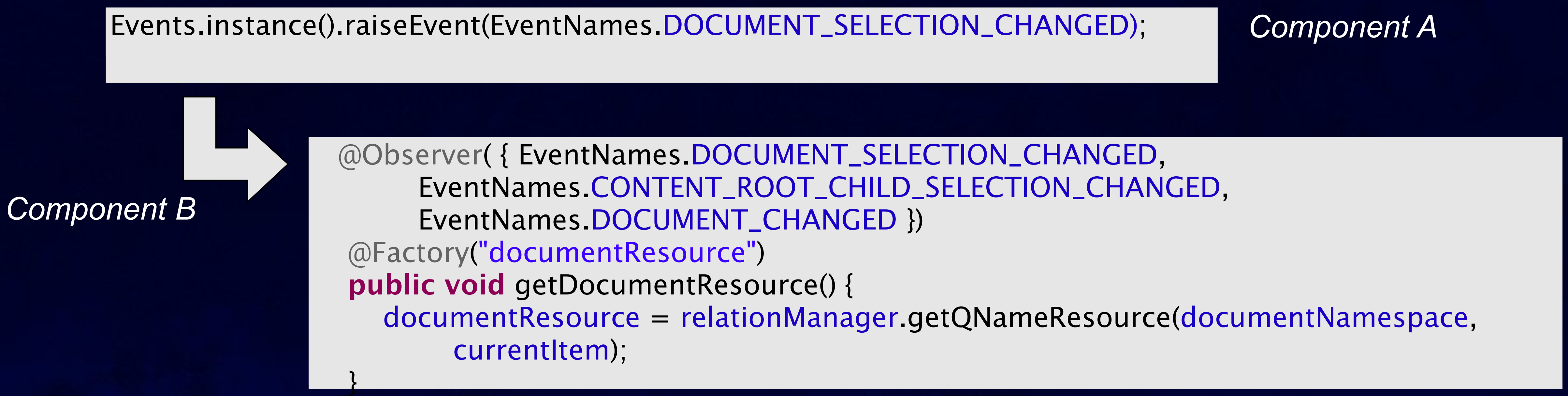


Nuxeo 5 - Context Management with Seam

- **Context Invalidation**
  - Context variables are dependent
    update currentDocument => update currentDocumentChilds
  - Invalidation needs to
    - nullify the variable to get @Factory called the next time (lazy refresh)
    - call explicitly the factory
  - Dependencies are spread across components that don't know each others
    - Distributed invalidation

Nuxeo 5 - Context Management with Seam

- **Seam messaging system is used to propagate context changes**



```
Events.instance().raiseEvent(EventNames.DOCUMENT_SELECTION_CHANGED);
```

Component A

```
@Observer( { EventNames.DOCUMENT_SELECTION_CHANGED,
        EventNames.CONTENT_ROOT_CHILD_SELECTION_CHANGED,
        EventNames.DOCUMENT_CHANGED })
 @Factory("documentResource")
 public void getDocumentResource() {
    documentResource = relationManager.getQNameResource(documentNamespace,
           currentItem);
 }
```

Component B

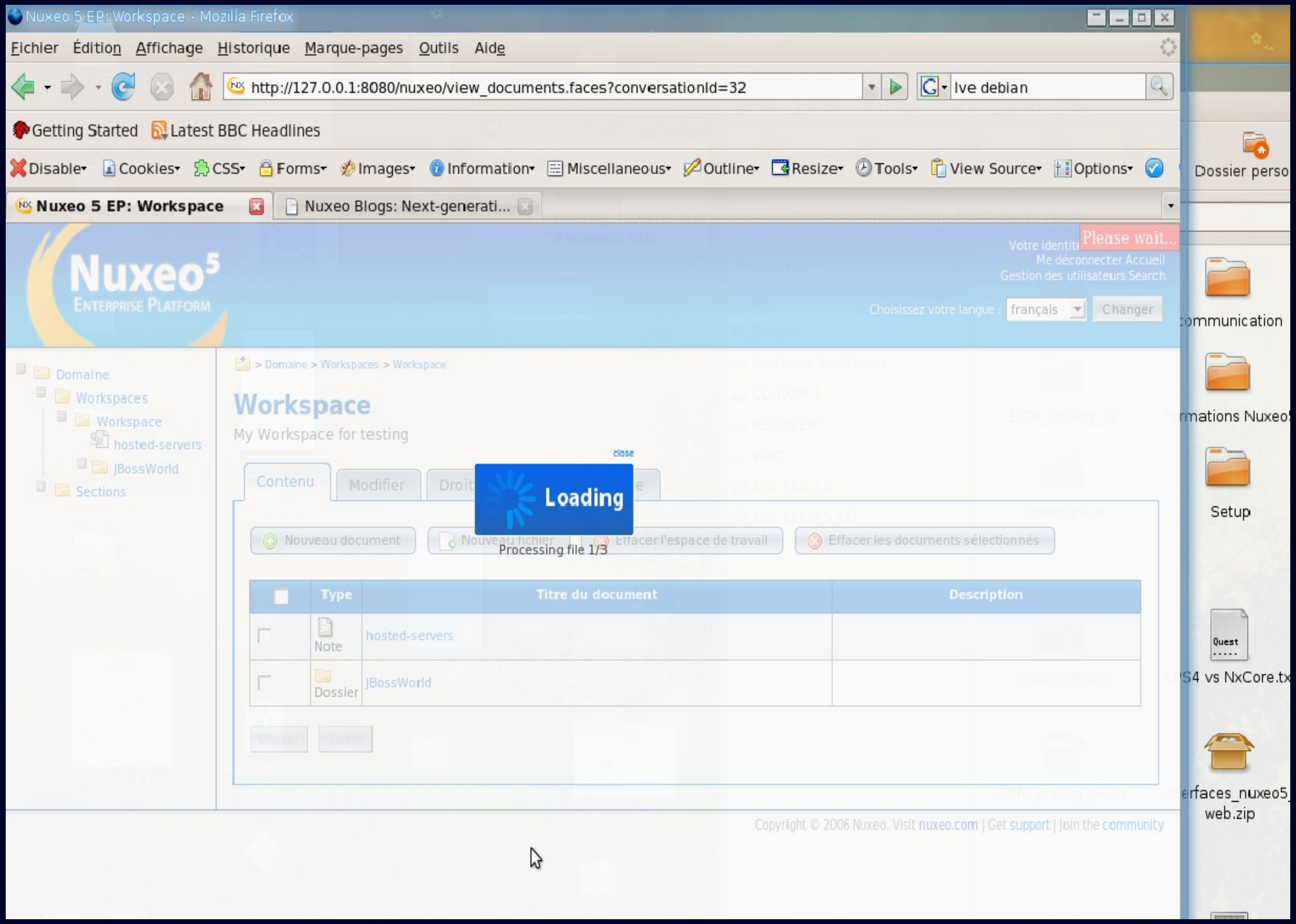
Nuxeo 5 - Context Management with Seam

- **“Stalled copies” problem**
  - Long running contexts act as a cache (session/conversation)
    - invalidation needs to be propagated when modifications occur *(document edit, security update, new child ...)*
  - There may be a lots of users using the same document
    - efficient to share the state of a document across user's contexts *(Domain, Workspace ...)*
- **Nuxeo 5 Cache and Context management**
  - The documents are actually stored in JBoss cache
    - Cache is invalidated by core events
  - The context holds *Ghost*DocumentModels or lists of *Ghosts*
    - acts as transparent stubs to the real DocumentModel in the shared cache

Nuxeo 5 & Seam Remoting

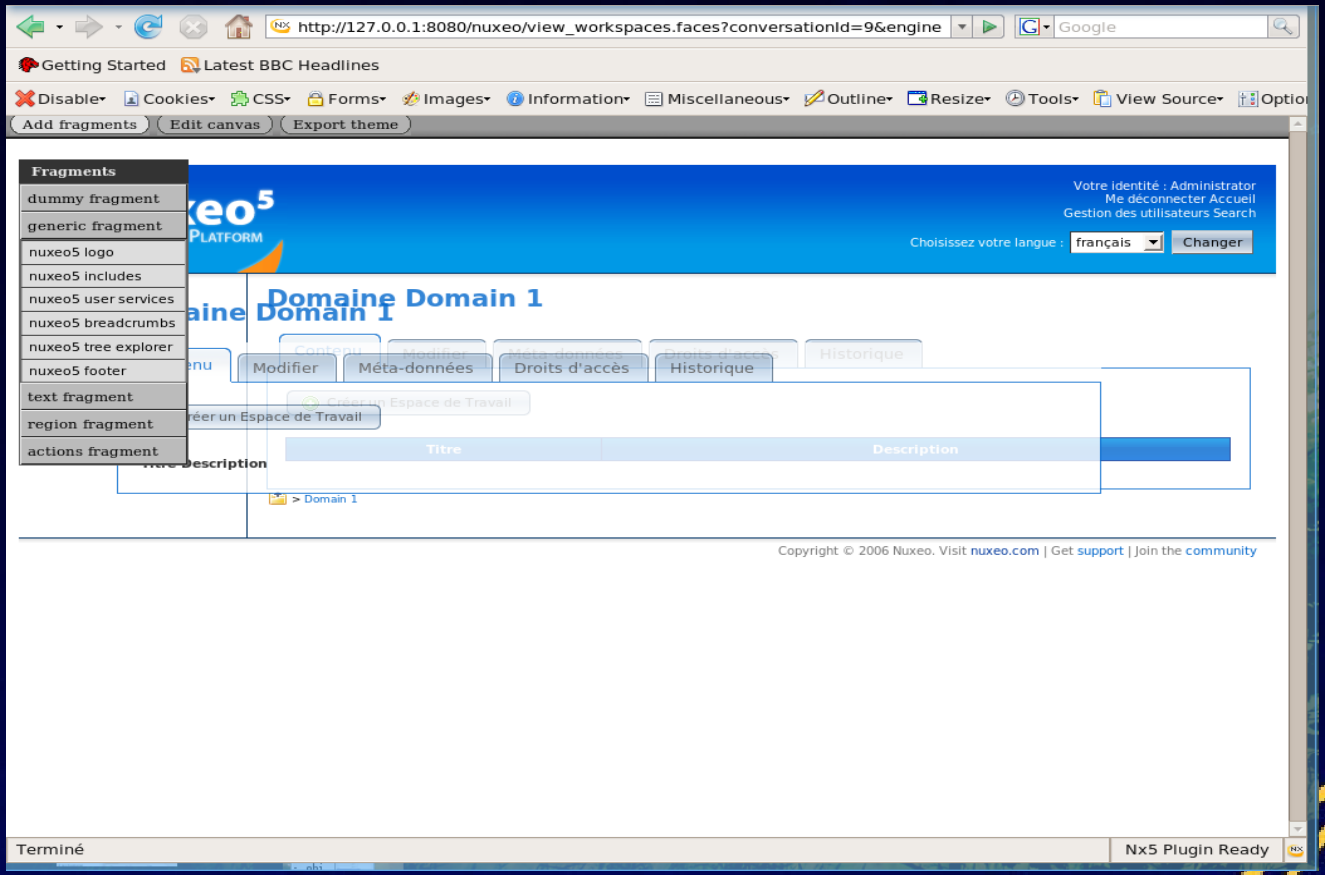
- **Some Nuxeo 5 Seam Components are accessed remotely**
  - ActionListener
  - Specific Seam Components
- **By some PlugIns like LiveEdit**
  - Use SOAP to integrate Nuxeo5 with MS Office and OpenOffice
- **By the browser helpers (Firefox and IE)**
  - Use Seam remoting to upload files by Drag&Drop
- **The theme Editor**
  - Uses Seam remoting to edit the theme from JS WYSIWYG editor
- **Is used for internal Drag&Drop and AJAX features**

Nuxeo 5 & Seam – *Remoting - examples*



Drag &Drop filesf romt heD esktopt ot heB rowser
Firefox / Internet Explorer
Win32 / MacOS / Linux

Nuxeo Theme Editor
Modify page layout from the browser using AJAX
Add fragments WYSIWYG



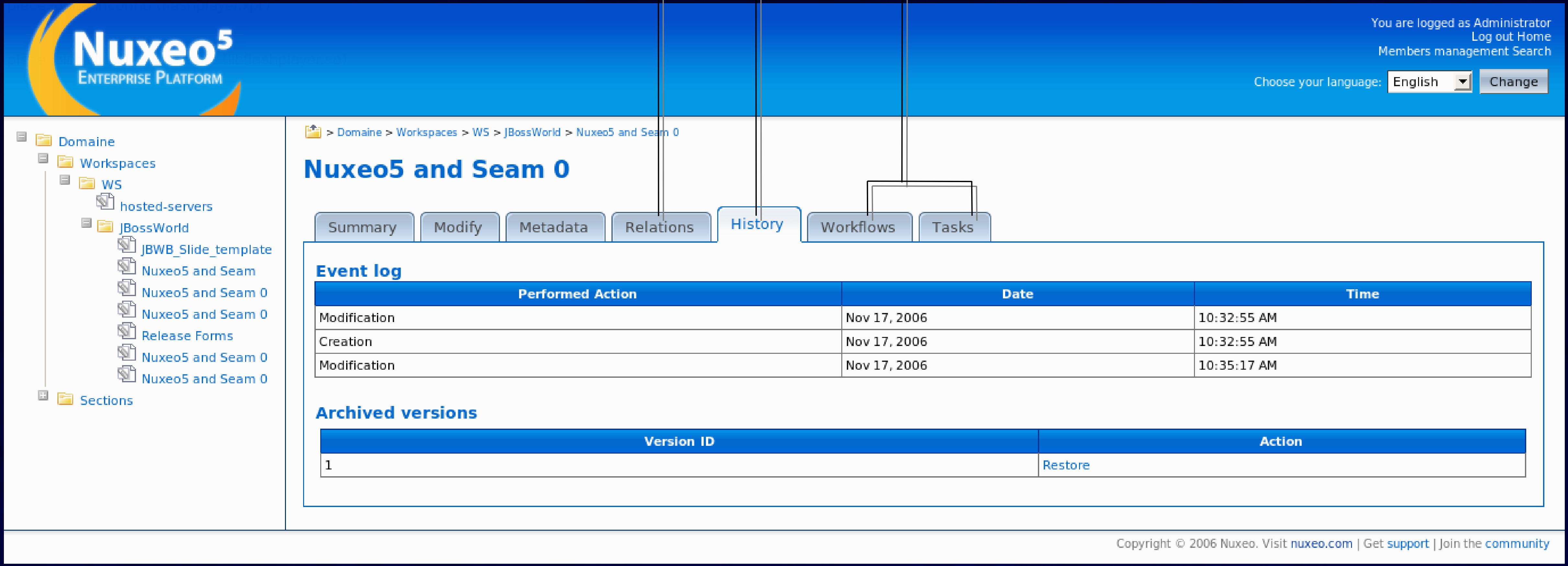
Nuxeo 5 & Seam – *additional views and actions*

NXRelations

NXAudit

NXWorkflow

Contributed views



You are logged as Administrator
Log out Home
Members management Search

Choose your language: English Change

> Domaine > Workspaces > WS > JBossWorld > Nuxeo5 and Seam 0

Nuxeo5 and Seam 0

Event log

Performed Action	Date	Time
Modification	Nov 17, 2006	10:32:55 AM
Creation	Nov 17, 2006	10:32:55 AM
Modification	Nov 17, 2006	10:35:17 AM

Archived versions

Version ID	Action
1	Restore

Nuxeo 5 Loosely coupled components

- **The Nuxeo 5 UI is build from several components**
  - Base components (DocumentManager, SecurityManager ...)
  - Additional optional components
    - NXAuditClient
    - NXRelationClient
    - NXWorkflowClient
    - ...
- **Each additional component uses NXRuntime**
  - to contribute Views
    - Facelets that provide a new view on Documents : AuditView, RelationView ...
  - to contribute Action
    - Buttons or Links to achieve additional processing : Create a relation, ...

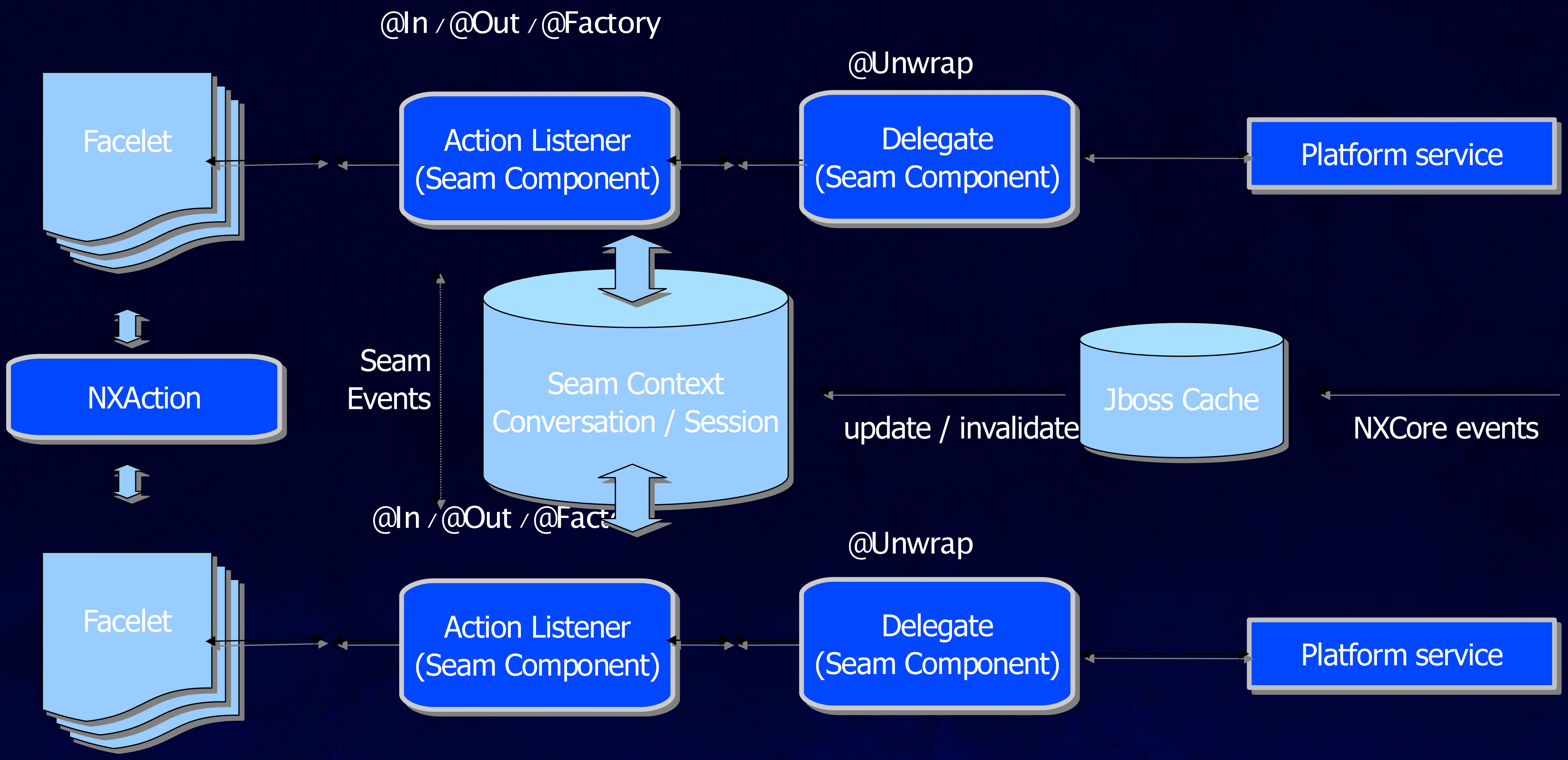
Nuxeo 5 Loosely coupled components

- **Components share some state via the context**
  - At least the currentDocument
  - Most ActionListers use @In to get some contextual documents
- **Components may notify each other**
  - Usage of the @Observer to become an event listener
- **New components may be easily added**
  - Only cares about the shared data and events
    - not “who” the other components are
  - New views can be added transparently
- Nuxeo Runtime take care of the deployment

Nuxeo 5 & Service access

- **Action listeners need to access one or more NX services**
- **Services are accessed via a Delegate Layer**
  - Wraps remote or local calls
- **Delegates use the Seam component manager pattern**
  - use @Unwrap to access services via a transparent Factory
- **Base components are created at application start-up**
  - use @Startup to load services automatically

Nuxeo 5 & Service access



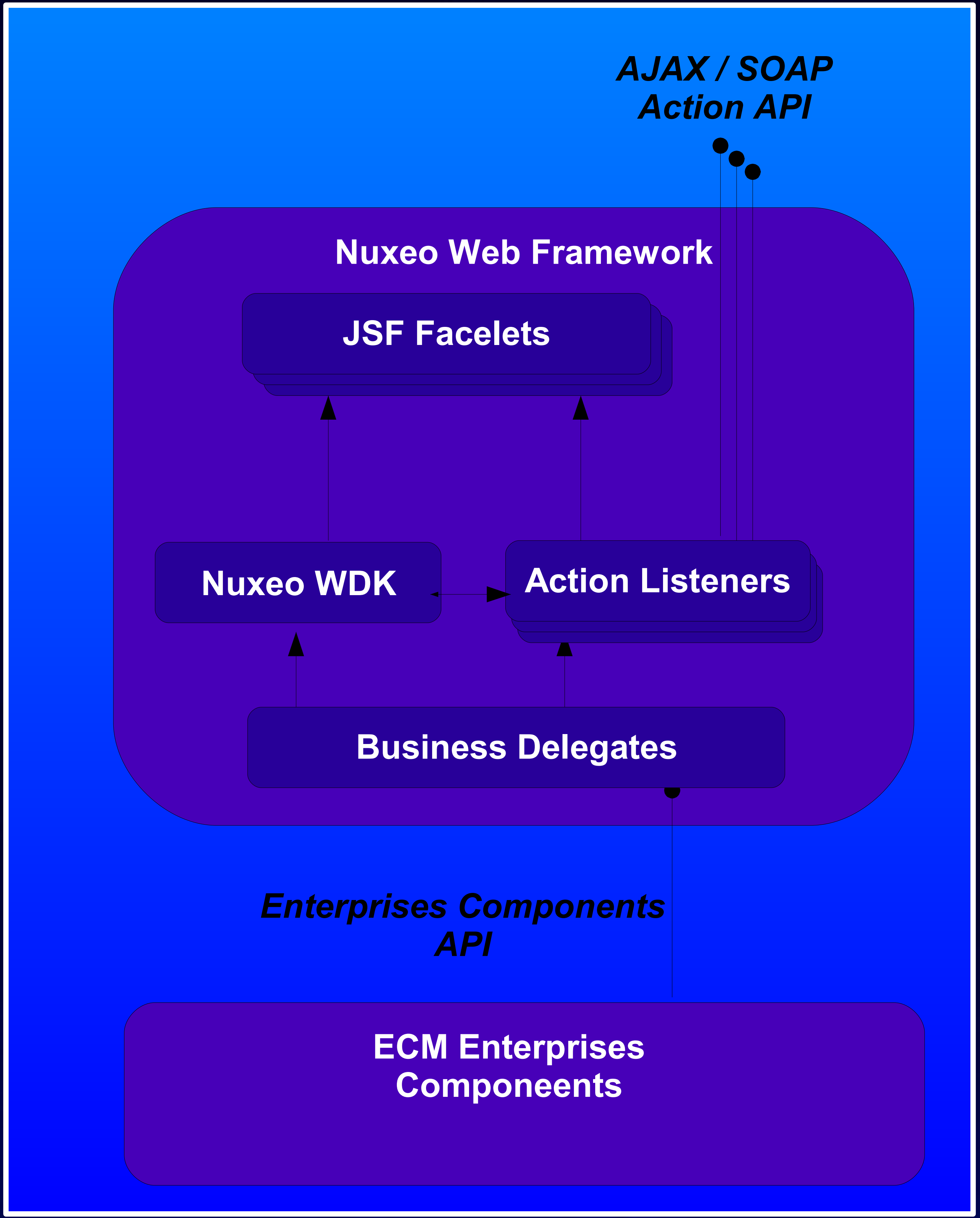
Nuxeo WDK

- Ui Services
  - TypeManager
    - Binds Widgets and Layout to the documents defined in the Core
  - ActionManager
    - Defines Actions with EL enabled guards and filters
  - Cache Invalidation Management
    - Invalidate Seam context
    - Invalidate stalled DocumentModel

Nuxeo WDK

- Make JSF and Nuxeo integration easier
  - EL resolver
    - enable direct access from JSF : #{currentItem.dublincore.modificationDate}
  - NXTagLibs : Encapsulate JSF tags
    - JSF Lib transparency
    - Resolutions
    - Mode (view/edit)
    - Validation management
  - Base classes and patterns
    - Action Listeners
    - Navigation Management
    - Delegate Layer

- JSF / Facelets
  - as presentation
- Seam
  - for context management
  - for Action Listener
- Nuxeo WDK
  - for ECM UI
- Business Delegate
  - to encapsulate remote calls



nuxeo

Enterprise Components

- Enterprise components have their UI part
  - JSF Views
  - ejb3 Seam Action Listeners
  - Document view definition (Layout)

- Examples
  - NXRelationsClient
  - NXWorkflowClient
  - ...