



Sword Group

Aramis Nuxeo

Communication Aramis - GED Aramis

NOTE D'ORGANISATION

IDL_TMANUX/NOT_002/1.1

Client : AERMC
Entité : SWORD/IDL
Contrat/Projet : MCO DE LA PLATEFORME GED NUXEO
Identifiant : IDL_TMANUX
Date⁽¹⁾ : 04/07/2014
Etat : ☒ à valider ☐ validé
Diffusion : ☐ interne ☒ contrôlée ☐ libre

(1) Date d'approbation (cf. circuit de validation interne).

CIRCUIT DE VALIDATION

| Version | Rédaction | | | Vérification | | | Approbation | | |
|---------|-----------|------------|------|--------------|------|------|-------------|------|------|
| | Nom | Date | Visa | Nom | Date | Visa | Nom | Date | Visa |
| 1.1 | S. Pesnel | 04/07/2014 | | V. Guezille | | | T. Georges | | |

Figure 1 – Circuit de validation

HISTORIQUE DES EVOLUTIONS

| Version | Objet de la version (citer les fiches de revue de document prises en compte) |
|---------|---|
| 1.1 | Description des modifications |

Figure 2 – Historique des évolutions

LISTE DE DIFFUSION

| Destinataire | Fonction | Nombre d'exemplaires | Support |
|--------------|----------|----------------------|---------|
| | | | |

Figure 3 – Liste de diffusion

DOCUMENTS ASSOCIES

| Origine | N° | Titre | Référence | Usage(*) |
|---------|----|-------|-----------|----------|
| | | | | |
| | | | | |
| | | | | |
| | | | | |
| | | | | |
| | | | | |
| | | | | |

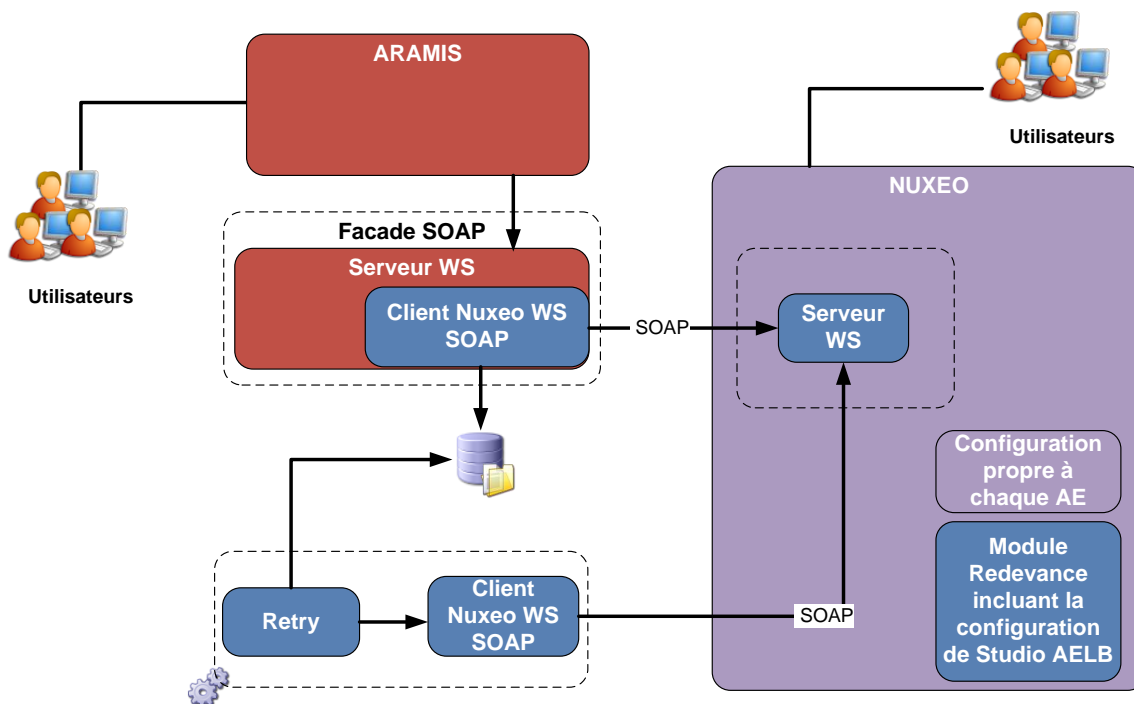
(*) : Indiquer le contexte de citation du document : à lire au préalable, documents de référence, documents complémentaires, ...

Figure 4 – Documents associés

Sommaire

| | |
|---|-----------|
| 1 ORGANISATION DES ÉLÉMENTS..... | 4 |
| 2 PROTOCOLE DE COMMUNICATION | 6 |
| 2.1 OBJECTIF | 6 |
| 2.2 COMMUNICATION SOAP | 6 |
| 2.3 MISE EN PLACE | 6 |
| 2.3.1 Côté serveur..... | 6 |
| 2.3.2 Côté client..... | 7 |
| 3 COUCHE DE SERVICE | 9 |
| 3.1 OBJECTIF | 9 |
| 3.2 MISE EN PLACE | 9 |
| 4 DÉCOUPAGE DU PROJET STUDIO | 10 |
| 4.1 OBJECTIF | 10 |
| 4.2 DÉCOUPAGE | 10 |

Ce deuxième schéma illustre l'organisation mise en place dans le cadre du projet avec l'AERMC :

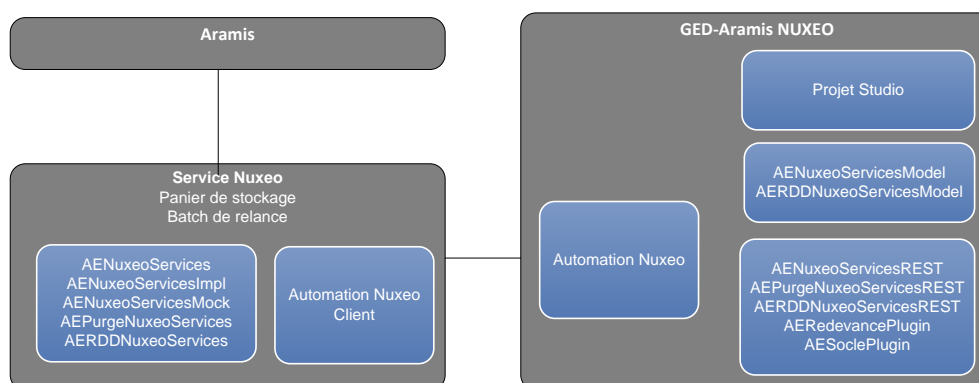


Avec :

- En rouge : les éléments gérés par AKKA
- En bleu : les éléments gérés par ATOS pour le compte de l'AELB (et des autres agences),
- En mauve : les éléments gérés spécifiquement par chaque agence.

Le protocole de communication utilisé est SOAP. Les interfaces de communication client / serveur sont générées sur la base d'un fichier WSDL.

Cette troisième illustration présente l'organisation précédente des éléments composant l'architecture.



L'ensemble des modifications listées dans ce document seront réalisées par Sword. La version compilée et non compilée des différents modules sera ensuite transmis à l'AELB.

2 Protocole de communication

Cette partie réalise un zoom sur la solution envisagée pour le protocole de communication entre la GED Nuxeo et l'application Aramis.

2.1 Objectif

L'objectif visé est de s'abstraire de la couche de communication Nuxeo. Avec cette solution, le protocole ne dépend plus de Nuxeo. En cas d'évolution de ce protocole sur une montée de version Nuxeo, la communication Aramis/GED-Aramis ne sera pas impactée. Actuellement, ceci pourrait poser problème au cas où la communication impliquerait plusieurs serveurs Nuxeo dans des versions différentes dont des versions pour lesquels le protocole aurait évolué.

2.2 Communication SOAP

Une communication en protocole SOAP a été mise en place. Les interfaces de communication client et serveur sont générées à partir d'un fichier WSDL. C'est ce fichier qui décrit le contenu des messages échangés.

Afin de faciliter les tests, la mise en place, la communication via Nuxeo automation a été maintenue. La couche de service nécessaire au traitement des messages a été mutualisée entre les deux modes de communication. Cela couvre

- Côté client :
 - la validation des données envoyées,
 - le mapping vers les objets attendus par Aramis
- Côté serveur
 - les traitements impliqués par les messages

L'utilisation de l'un ou l'autre des protocoles se fait via un paramètre de configuration du fichier « nuxeo-connector.properties ». La propriété « client.impl » permet de choisir l'implémentation à utiliser :

- Soit « fr.ae.redevances.services.externes.ged.NuxeoServiceSoapImpl » pour le soap
- Soit « fr.ae.redevances.services.externes.ged.NuxeoServiceRestImpl » pour l'utilisation de l'automation Nuxeo

L'url configurée dans le fichier de propriétés (paramètre « connector.url ») doit être définie en adéquation avec l'implémentation retenu, c'est-à-dire

- Pour le SOAP : « nuxeo/webservices/aramis/GedAramisService?wsdl »
- Pour l'automation nuxeo : « nuxeo/site/automation »

2.3 Mise en place

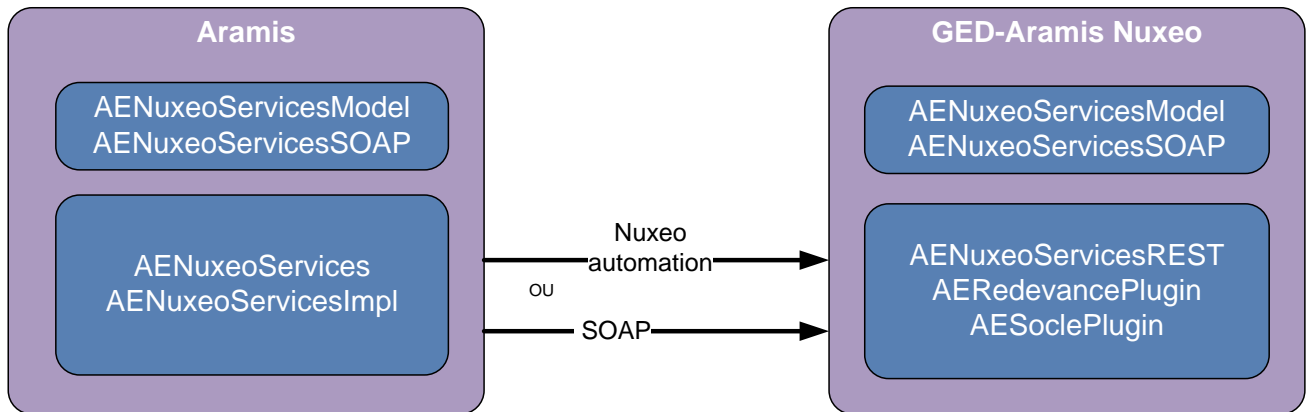
2.3.1 Coté serveur

La mise en place consiste en

- la création d'un fichier wsdl énumérant les messages envoyés dans le cadre des échanges entre la GED et Aramis.
- l'implémentation de l'interface de service soap : mapping des données en entrées et appel de la couche de service mutualisé
- la déclaration du webservice soap dans Nuxeo (AENuxeoServicesREST/src/main/resources/OSGI-INF/deployment-fragment.xml)

Les points d'entrée SOAP comme « Nuxeo automation » sont regroupés dans le projet AENuxeoServicesREST (fr.ae.redevances.services.externes.ged.rest.jar).

Un projet AENuxeoServicesSoap (fr.ae.redevances.services.externes.ged.soap.jar) a été ajouté contenant les classes nécessaires à la communication SOAP. Ce package est utilisé aussi bien coté client que serveur. Son contenu est automatiquement généré sur la base du fichier WSDL.

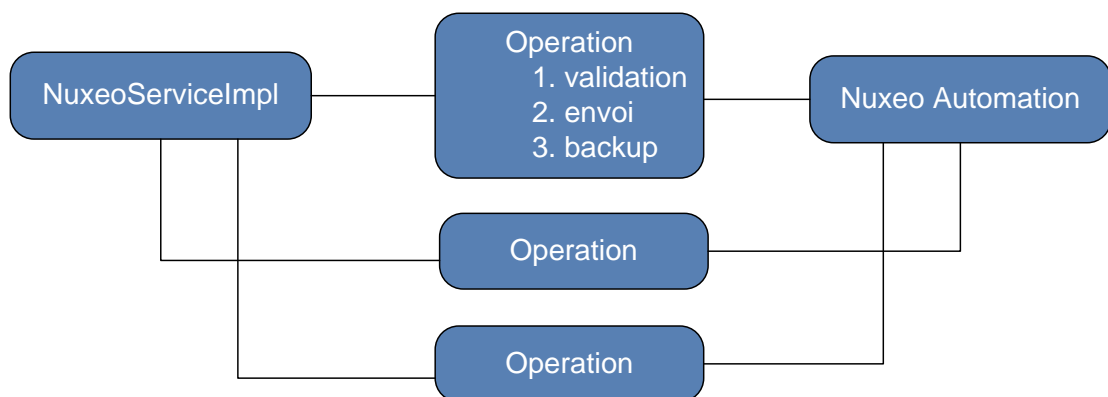


2.3.2 Côté client

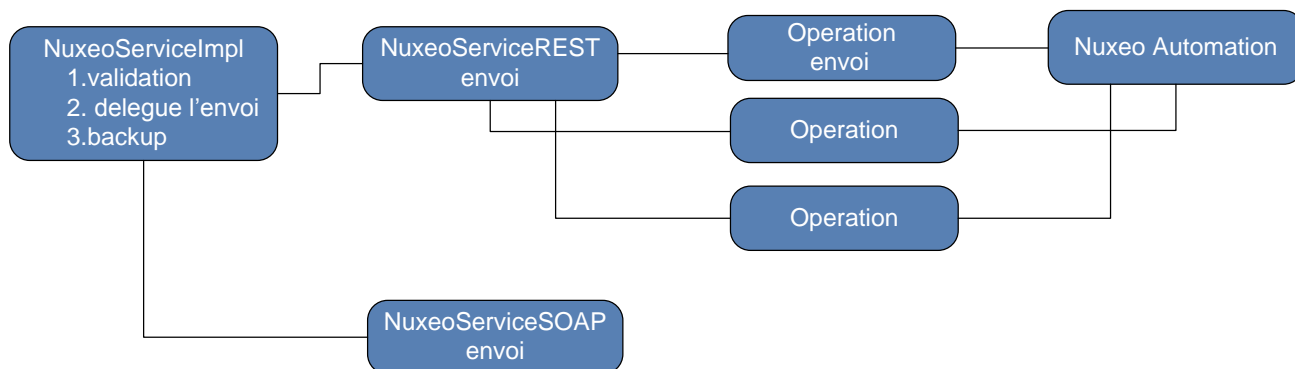
Une implémentation de l'interface `NuxeoService` a été implémentée pour passer par le protocole SOAP. Les validations et autre opération faite en amont aux appels à Nuxeo automation ont été refactorés et déplacés dans des classes externes aux opérations Nuxeo afin d'être mutualisé entre les deux modes de communication. Cela permet de valider les modifications en deux temps :

- validation des traitements qui seront utilisé par le client SOAP en continuant à passer par Nuxeo automation,
- puis utilisation du protocole SOAP.

Dans la version initiale, l'implémentation de l'interface `NuxeoService` sollicite plusieurs opérations suivant les requêtes à effectuer. Chacune de ces opérations valide les données, envoie la requête à la GED et enfin au besoin fait un backup de la requête (au cas où la GED est indisponible).



Le découpage a été modifié de la façon suivante : La validation et le backup éventuel est porté par la classe `NuxeoServiceImpl`. En revanche l'envoi est délégué soit à la classe `NuxeoServiceRESTImpl`, soit à la classe `NuxeoServiceSoapImpl`.



Packaging

En terme de packaging, l'archive RDVStockage.ear doit subir les modifications suivantes :

- Mise à jour des jars
 - fr.ae.redevances.services.externes.ged.impl.jar
 - lib/fr.ae.redevances.services.externes.ged.interface.jar
 - lib/fr.ae.redevances.services.externes.ged.model.jar
- Ajout des jars
 - lib/fr.ae.redevances.services.externes.ged.soap.jar
 - lib/commons-codec-1.4.jar

3 Couche de service

Dans ce chapitre, un zoom va être réalisé sur les processus relatifs à la communication entre la GED Aramis et l'application Aramis.

3.1 Objectif

L'objet de la couche de service est de s'occuper des traitements. Ainsi la couche de communication (par exemple les opérations Nuxeo) se contente d'appeler les méthodes du service qui s'occupe du traitement métier.

Une implémentation par défaut reprenant les traitements existants peut ainsi être mutualisée. Une gestion spécifique est alors aisée en déclarant une nouvelle implémentation du service via une contribution XML. Cette nouvelle implémentation vient surcharger le service par défaut. Si les modifications sont minimales, ce service peut étendre le service par défaut et ne surcharger que quelques méthodes.

Surcharger le service plutôt que l'opération permet de conserver l'extraction des paramètres et juste d'affecter le traitement fait dessus. De plus il est plus aisé d'appeler la méthode du service par défaut depuis le nouveau service, que d'appeler l'opération qui a été surchargé dans une nouvelle opération (l'appel d'opération impliquant la gestion des paramètres et du contexte).

3.2 Mise en place

Les services suivants ont été créés :

- RedevanceNavigationService
- GedAramisService
- GedAramisListService
- GedAramisLinkService

Le service RedevanceNavigationService est déclaré au sein du package AERedevancePlugin-api et l'implémentation est dans AERedevancePlugin-core. Il regroupe les méthodes d'aide à la navigation au sein de l'arborescence. Ces méthodes étaient gérées dans la version précédente au sein d'une classe utilitaire. Il permet d'accéder au différent type de document à partir d'identifiants.

Les services GedAramisService, GedAramisListService et GedAramisLinkService sont déclarés dans le package AENuxeoServicesREST. GedAramisService s'occupe de la gestion des requêtes de création, modification. GedAramisListService s'occupe des requêtes de listing. GedAramisLinkService s'occupe de la génération des liens associés au document, liens au travers desquels ils sont accédés depuis Aramis.

4 Découpage du Projet Studio

Ce chapitre va traiter du projet Studio utilisé par l'AELB qui, en l'état, peut être un risque pour la mutualisation du code.

4.1 Objectif

Un projet Studio peut être utilisé pour gérer la configuration des redevances, mais peut également servir à la personnalisation de l'interface graphique. A moins que le projet Studio ne soit dédié au module « Redevance », le risque est de trouver des configurations spécifiques à une agence dans le module mutualisé. C'est pourquoi nous préconisons d'avoir un module redevance construit sur la base du projet Studio et du module de redevance actuel. Les configurations propres à chaque agence viendront dans des modules complémentaires (Studio ou autre).

4.2 Découpage

Le projet studio, présent sous la forme d'une archive JAR, contient un fichier de configuration XML de plus de 12000 lignes. Le module a été découpé, les contributions ont été réparti entre les package

- AERedevancePlugin-core : schéma, cycle de vie
- AERedevancePlugin-web : vocabulaire, layout
- AERedevancePlugin-contrib : contribution utilisateurs, layout modifiant peu le standard, suppression de l'arborescence documentaire
- AERedevancePlugin-theme : Thème et page de login

Le thème regroupait dans un package dédié peut être retiré en enlevant simplement le jar, sans avoir à faire de configuration particulière, le thème utilisé sera alors extrait du standard, d'un jar ou d'un autre projet studio.

Le jar AERedevancePlugin-contrib vient compléter les contributions présentes dans les autres jar, afin de couvrir la totalité de la configuration qui été faite dans Studio. Ce jar n'est pas inclus dans la configuration déployée à l'AERMC.