

**PROJET WebRSA**  
**INTEGRATION DES FLUX CAF**  
**Document d'architecture**

	Nom	Société
Auteurs :	Natuk Turna	Logica
	Dinah Blirando	Conseil Général du 93

Valideurs :	Sammy Shanoune	Conseil Général du 93
	Stéphane Vivien	Logica

Documents de référence

Nom du document	Type	Version
INTEGRATION_FLUX_2_1_Guide_Technique	PDF	2.1
INTEGRATION_FLUX_2_1_Annexes	PDF	2.1

Historique des modifications

Date	Version	Auteur	Description
06/05/2010	1.0.0	Natuk TURNA	Draft
17/09/2010	1.0.1	Natuk TURNA	Finalisation
18/10/10	2.0.0	Dinah BLIRANDO	Modification
01/02/2011	2.1.0	Dinah BLIRANDO	MAJ versions

## SOMMAIRE

<b>1. OBJET DU DOCUMENT.....</b>	<b>4</b>
<b>2. PROJET.....</b>	<b>5</b>
2.1. Contexte.....	5
2.2. Architecture générale.....	6
<b>3. FONCTIONNEMENT GENERAL.....</b>	<b>7</b>
3.1. Introduction.....	7
3.1.1. Objectif.....	7
3.1.2. ETL.....	7
3.2. Généralités.....	7
3.2.1. Schémas STAGING_RSA.....	7
3.2.2. Découpage applicatif.....	8
<b>4. Extraction des données.....</b>	<b>9</b>
4.1. Généralités.....	9
4.2. Alimentation.....	9
4.2.1. Principe.....	9
<b>5. Transformation des données.....</b>	<b>11</b>

5.1. Généralités.....	11
5.2. Alimentation.....	11
5.2.1. Principe.....	11
<b>6. Insertion des données.....</b>	<b>12</b>
6.1. Généralités.....	12
6.2. Alimentation.....	12
6.2.1. Principe.....	12
6.3. Contrôles et rejets.....	13
6.3.1. Fonctionnement.....	13
6.3.2. Types de contrôle.....	13
6.3.3. Tables de Rejet.....	13
6.4. Rapport d'erreur et fichier XML de rejets.....	15
6.4.1. Rapport d'erreur.....	15
6.4.2. Xml de rejet.....	15

## 1. OBJET DU DOCUMENT

Ce document a pour objectif de présenter l'architecture des interfaces, l'architecture de la base de données intermédiaire (Staging, Elementaire et Administration) et les différentes étapes de l'intégration des données dans la base de données Webrsa.

Ce document sera mis à jour en cas d'évolution des interfaces et/ou des différentes sources et cibles.

## 2. PROJET

### 2.1. Contexte

Dans le cadre de la mise en œuvre du RSA, le Conseil Général de la Seine Saint Denis reçoit les éléments constituant les demandes, instructions et virements de la CNAF via des fichiers XML.

Ces fichiers XML sont intégrés à l'application WEB RSA du Conseil Général.

Le Conseil Général de la Seine Saint Denis a retenu l'ETL TALEND pour répondre aux problématiques d'alimentation.

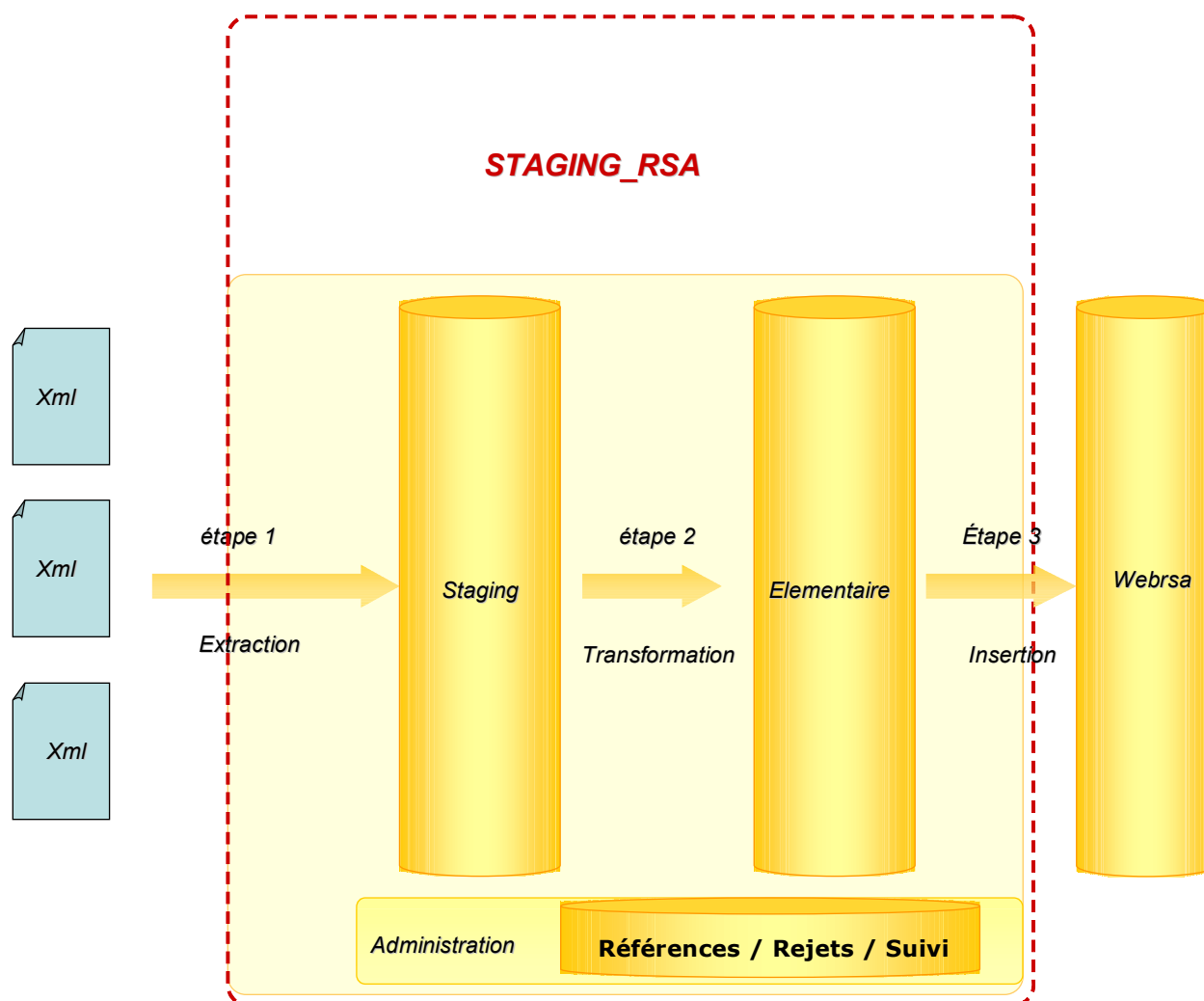
TALEND a mis en œuvre des flux d'alimentation jugés aujourd'hui très complexes et difficilement maintenables.

Plusieurs problèmes ont été remontés faisant suites à de nombreuses évolutions des flux.

Le Conseil Général de la Seine Saint Denis a demandé à Logica de proposer une refonte complète des alimentations.

Logica a opté pour la mise en place d'un entrepôt de données donnant une vision à la fois homogène et consolidée des informations traitées.

## 2.2. Architecture générale



### 3. FONCTIONNEMENT GENERAL

#### 3.1. Introduction

##### 3.1.1. Objectif

Les données envoyées par la CNAF sont au format XML avec une arborescence complexe avec plusieurs sous niveaux de balises imbriquées. De ce fait, difficile à analyser en cas de retour ou de rejet de certaines données.

Ces données seront extraites/centralisées dans la base de données intermédiaire *STAGING\_RSA* avant d'être chargées dans la base de données *webrsa*. La périodicité de ces chargements est journalière et mensuelle.

Ce document a pour objectif de spécifier toutes les phases, en définissant les règles et processus permettant l'alimentation de la base de données *webrsa*.

##### 3.1.2. ETL

Les traitements d'alimentation sont réalisés à l'aide d'un ETL (*Extract, Transform and Load*). L'ETL utilisé est TALEND.

#### 3.2. Généralités

##### 3.2.1. Schémas *STAGING\_RSA*

Les données au sein de la base *STAGING\_RSA* sont décomposées de la façon suivante :

- 1) **Staging** : Les données 'brutes' sont extraites depuis les fichiers XML et sont chargées dans le schéma *staging*. L'image des tables *staging* est définie en prenant comme modèle les définitions des schémas des XML source (XSD).
- 2) **Élémentaire** : Les données présentes dans les tables *staging* sont extraites, transformées et mises en forme pour être insérées dans les tables du schéma *élémentaire*. Les tables du schéma *élémentaire* sont définies en prenant comme modèle les tables cibles de la base de données *Webrsa*.
- 3) **Administration** : Les données de référence, les rejets, les logs et les autres informations nécessaires au contrôle d'intégrité des données sont chargées dans le schéma *administration*.

### 3.2.2. *Découpage applicatif*

L'importance des traitements à réaliser implique de découper et d'organiser les processus d'alimentation. La hiérarchie du découpage ci-dessous s'appuie sur le fonctionnement et le vocabulaire de l'outil d'extraction utilisé, *TALEND* :

- **Unité de traitement** : plus petit élément du point de vue de l'exploitant, lancé par l'ordonnanceur (fichiers .sh LINUX). Il s'agit de la granularité de reprise du point de vue de l'exploitant.
- **« Job père »** : traitement qui lance successivement un ensemble cohérent de « Job fils ».
- **« Job fils »** : traitement élémentaire de l'application. Il s'agit de la granularité la plus fine du point de vue de l'application

Un « Job fils » correspond à l'alimentation d'une table, et l'enchaînement de l'alimentation de plusieurs tables est réalisé par un « Job père ».



## 4. Extraction des données

### 4.1. Généralités

Le schéma *staging* de la base de données *STAGING\_RSA* est une zone tampon qui contient les données sources. Ces données sont :

- Stockées sous un format varchar.
- Correspondent à une exécution/extraction de l'application source.

La base de données opérationnelle ne contient aucun historique de données. Les données sont remplacées à chaque chargement.

### 4.2. Alimentation

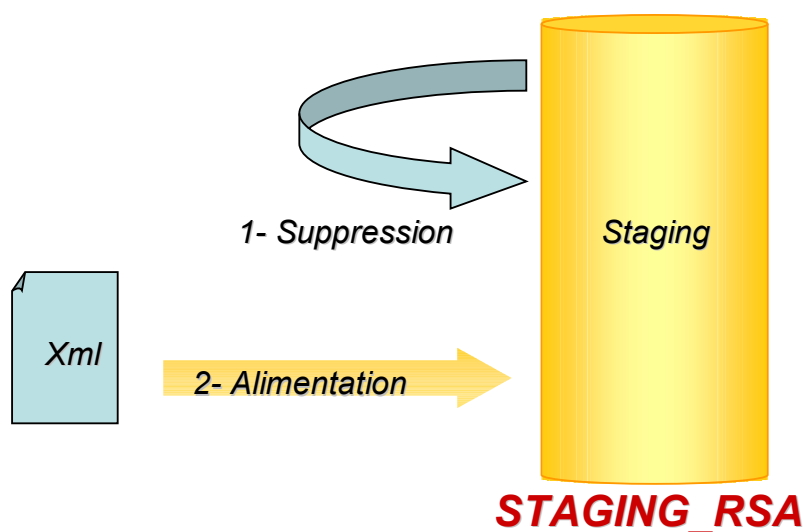
#### 4.2.1. Principe

L'interface entre les fichiers XML sources et la base *STAGING\_RSA* est réalisée de façon à charger l'ensemble des données sources dans un espace unique : le schéma *staging*.

Les données sont chargées dans le schéma sans transformation.

Lors du chargement, pour chaque table du schéma *staging* :

- 1) On supprime tous les enregistrements précédents
- 2) On insère les nouveaux enregistrements en gardant en base uniquement la dernière occurrence d'un dossier (numdemrsa) ou d'une personne dans un dossier (infosfoyerrsa, nomnai, prenom, dtnai, rgnai).



**Principe d'alimentation des Tables Staging**

## 5. Transformation des données

### 5.1. Généralités

Le schéma *élémentaire* de la base de données *STAGING\_RSA* est une zone tampon qui contient les données pour le chargement en base cible. Ces données sont :

- transformées au format des tables cibles.
- Correspondent à une exécution/extraction de l'application source.

### 5.2. Alimentation

#### 5.2.1. Principe

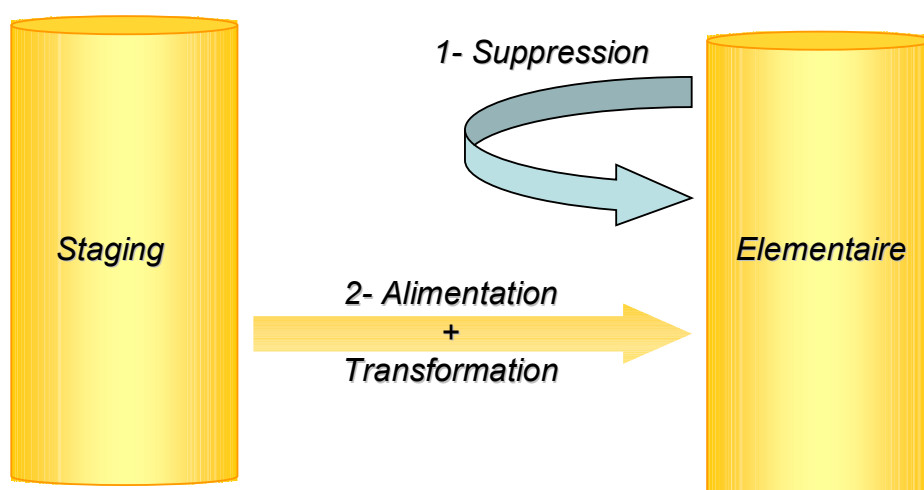
L'interface entre le schéma *staging* et le schéma *élémentaire* est réalisée de façon à charger l'ensemble des données sources dans un espace unique : le schéma *élémentaire*.

Les données sont transformées et mises aux types de la base cible.

Lors du chargement, pour chaque table du schéma *élémentaire* :

- 1) On supprime tous les enregistrements précédents.
- 2) On insère les nouveaux enregistrements.

## STAGING\_RSA



**Principe d'alimentation du schéma élémentaire**

## 6. Insertion des données

### 6.1. Généralités

- Les données des tables *élémentaire* étant prêtes, celles-ci sont insérées dans la base cible *webrsa*.

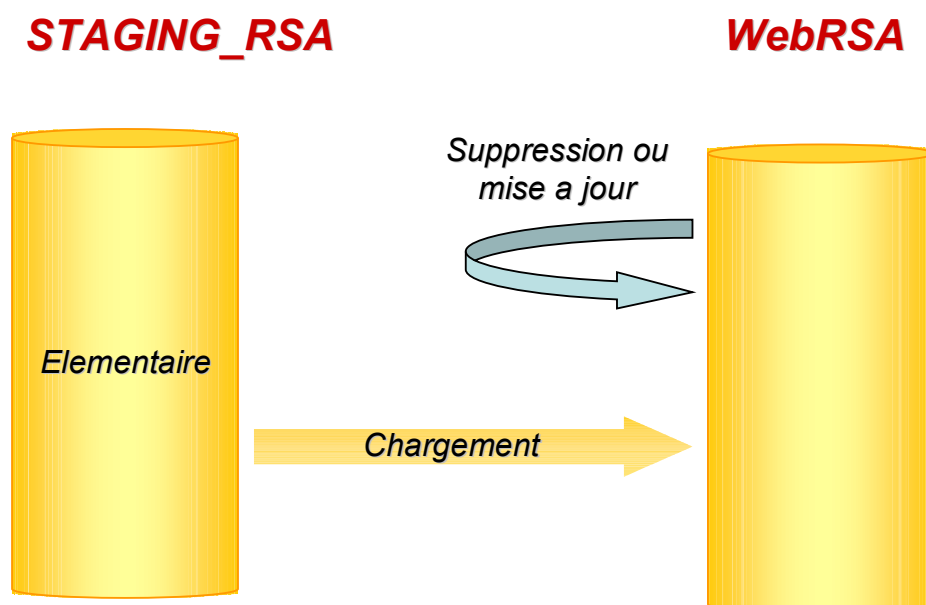
### 6.2. Alimentation

#### 6.2.1. Principe

L'interface entre le schéma *élémentaire* et la base cible *webrsa* effectue les différents contrôles d'intégrité avec les données existantes et gère les différentes règles d'insertion.

Règles d'insertions *webrsa* possibles pour la partie Insertion de données :

- 1) On supprime tous les enregistrements précédents ayant les mêmes clés fonctionnelles et on insert.
- 2) On met à jour les données existantes avec comme clé de mise à jour la clé fonctionnelle.
- 3) On insert sans effacer les données.



**Principe d'alimentation de la base cible WebRSA**

## 6.3. Contrôles et rejets

### 6.3.1. Fonctionnement

Pour une meilleure gestion des flux de webRSA, des contrôles de données ont été mis en place avant l'insertion des données dans le but de minimiser les erreurs et les retours lors de l'intégration des données dans le système.

Ces contrôles antérieurs à l'insertion des données dans la base de données *webrsa* permettent de mettre de côté les données erronées et celles qui ne sont pas fonctionnellement valides.

Ces données erronées sont insérées dans la table de rejet *administration.rejet* et aussi archivées dans la table *administration.rejet\_historique*.

### 6.3.2. Types de contrôle

Aucun contrôle particulier n'a été mis en place entre les fichiers XML et la base de données STAGING\_RSA, ni entre la base STAGING\_RSA et la base WebRSA. Seules les contraintes d'intégrités définies au niveau des tables sont susceptibles de produire des rejets.

Toutes les données intégrées dans la base STAGING\_RSA sont soumises aux contrôles d'intégrité !

### 6.3.3. Tables de Rejet

Les données rejetées sont insérées dans la table de rejet décrite ci-dessous :

Nom Colonnes	Type et taille	Description
cleinfodemandersa	Integer	Le numero de dossier correspondant au fichier XML.
flux	Varchar (20)	Le nom du flux sur lequel s'est produite l'erreur. INSTRUCTION, BENEFICIAIRE ou FINANCIER
etape	Integer	Le numéro de l'étape ou s'est produite l'erreur :

Nom Colonnes	Type et taille	Description
		1= XML->staging 2= staging->elementaire 3= elementaire-> WebRSA
table_en_erreur	Varchar (50)	Nom de la table ou s'est produite l'erreur d'insertion
log	Varchar (1000)	Log d'erreur SQL retournée par la BDD

A la fin de chaque exécution une copie des données de la table rejet est effectuée pour pouvoir avoir une archive des données erronées.

Celles ci sont copiées dans la table *administration.rejet\_historique* décrite ci dessous :

Nom Colonnes	Type et taille	Description
cleinfodemandersa	Integer	Le numéro de dossier correspondant au fichier XML.
flux	Varchar (20)	Le nom du flux sur lequel s'est produite l'erreur. INSTRUCTION, BENEFICIAIRE ou FINANCIER
etape	Integer	Le numéro de l'étape ou s'est produite l'erreur : 1= XML->staging 2= staging->elementaire 3= elementaire-> WebRSA
table_en_erreur	Varchar (50)	Nom de la table ou s'est produite l'erreur d'insertion
log	Varchar (1000)	Log d'erreur SQL retournée par la BDD
numdemrsa	Varchar (20)	Le numéro de demande RSA.
matricule	Varchar (20)	Le matricule
DT_INSERT	Timestamp	La date de rejet.

## **6.4. Rapport d'erreur et fichier XML de rejets.**

### **6.4.1. Rapport d'erreur**

A la fin de chaque exécution d'un flux, un rapport d'erreur est généré contenant le nom du fichier XML source ainsi que toutes les données rejetées et les erreurs qui ont causé le rejet.

### **6.4.2. Xml de rejet**

Pour pouvoir réintégrer les données en erreur, à la fin de chaque flux, l'interface crée un XML de rejet pour chaque dossier ayant une ou plusieurs données rejetées.

Ainsi ces XML de rejet pourront être envoyés à la CAF avec le rapport qui leurs correspond et pourront être réintégrés dans la base *webrsa* après correction.

Cette procédure de contrôle de rejet permet l'intégration des dossiers qui ont des données valides en mettant de côté uniquement les dossiers ayant leurs données non valides sur un fichier XML en entrée. Un fichier ne sera pas réintégré entièrement.

Cela facilite aussi la recherche et correction des données non valides pour la CAF.