



Description des workflows et des opérations Partie 1 : référentiels

Date	Version
04/03/2022	11.0 (Version 5)

État du document

En projet
 Vérifié
 Validé

Maîtrise du document

Responsabilité	Nom	Entité	Date
Rédaction	Équipe Vitam	Équipe Vitam	09/01/2019
Vérification	MVI	Équipe Vitam	14/02/2022
Validation	AGR	Équipe Vitam	04/03/2022

Suivi des modifications

Version	Date	Auteur	Modifications
1.0	09/01/2019		Génération à partir de l'ancien document en RST
1.1	25/01/2019	NMO	Relecture
1.2	30/01/2019	EVA	Relecture
1.3	30/01/2019	MRE	Relecture
2.0	30/01/2019	MRE	Finalisation du document pour publication de la Release 9
2.1	07/02/2019	MVI	Mise à jour pour tenir compte des fonctionnalités mises en œuvre pendant la <i>Release 10</i> : <ul style="list-style-type: none"> chapitre 5, section III (« Workflow d'administration d'un référentiel des formats ») : précisions sur les cas WARNING (section III.A.1) ; précisions sur le contenu des champs suivants : EvDetDateTime, Warnings et UpdatedPUIDs. chapitre 5, section XII (« Workflow d'administration d'un référentiel des griffons ») : corrections diverses : le champ Identifier fonctionne en mode VITAM esclave, le champ ExecutableVersion n'est pas un entier. chapitre 5, section XIII (« Workflow d'administration d'un référentiel des scénarios de préservation ») : ajout de l'étape « Processus de génération du rapport d'import du référentiel des scénarios de préservation SCENARIO_REPORT » et d'une partie sur la « Structure du rapport d'administration du référentiel des scénarios de préservation ».
2.2	15/04/2019	MAF	Mise à jour et corrections diverses
3.0	24/04/2019	MRE	Finalisation pour publication Release 10
3.1	14/08/2019	MVI	Mise à jour pour tenir compte des fonctionnalités mises en œuvre pendant la <i>Release 11</i> : <ul style="list-style-type: none"> chapitre 2, section V (« Rapport de l'audit de cohérence ») : mise à jour du rapport ; chapitre 11, section II (« Rapport de

			préservation ») : création de la section.
3.2	02/09/2019	JPP	Mise à jour pour tenir des fonctionnalités mises en œuvre pendant la Release 11 : <ul style="list-style-type: none"> • chapitre 8, section III et IV.
4.0	09/09/2019	MAF	Finalisation pour publication Release 11
4.1	24/09/2019	GFO	Modification multi-stratégies
4.2	23/10/2019	GFO	Mise à jour des Audits
4.3	19/11/2019	MAF	Mise à jour pour tenir des fonctionnalités mises en œuvre pendant la Release 12
5.0	29/11/2019	AGR	Finalisation pour publication Release 12
6.0	20/03/2020	AGR	Séparation du document en 2 parties et finalisation pour publication Release 13
6.1	24/06/2020	MVI	Mise à jour pour tenir des fonctionnalités mises en œuvre pendant la Release 14 : <ul style="list-style-type: none"> • chapitre 4, section 4.1 (« Workflow d'entrée ») : ajout des tâches CHECK_OBJECT_SIZE et CHECK_ATTACHEMENT ;
7.0	06/07/2020	AGR	Finalisation pour publication Release 14
7.1	26/10/2020	MVI	Mise à jour pour tenir des fonctionnalités mises en œuvre pendant la Release 15 : <ul style="list-style-type: none"> • chapitre 2, section 2.2 (« Workflow d'audit de vérification des journaux sécurisés ») : ajout de la section • chapitre 5, section 5.2.3 (« Structure du rapport du référentiel des règles de gestion ») : mise à jour de la modélisation du rapport.
8.0	26/10/2020	AGR	Finalisation pour publication Release 15
8.1	23/02/2021	MVI	Mise à jour pour tenir des fonctionnalités mises en œuvre pendant la Release 16 : <ul style="list-style-type: none"> • chapitre 4, section 4.1 (« Workflow d'entrée ») : ajout de la tâche de vérification de l'intégrité du bordereau de transfert (MANIFEST_DIGEST_CHECK) ; refonte du processus de mise à jour des groupes d'objets (STP_UPDATE_OBJECT_GROUP).
9.0	15/03/2021	AGR	Finalisation pour publication Release 16
9.1	09/08/2021	MVI	Mise à jour pour tenir des fonctionnalités mises en œuvre pendant la Version 5.RC : <ul style="list-style-type: none"> • partitionnement de la documentation avec une nouvelle répartition : <ul style="list-style-type: none"> ◦ partie 1 : référentiels ◦ partie 2 : gestion du cycle de vie des archives ◦ partie 3 : préservation et intégrité <p>Le présent document couvre les opérations liées aux référentiels.</p>
10.0	27/09/2021	AGR	Finalisation pour publication Version 5.RC
10.1	14/02/2022	MVI	Mise à jour pour tenir des fonctionnalités mises en œuvre pendant la Version 5 : <ul style="list-style-type: none"> • chapitre 3, section 3.4 (« Workflow d'administration d'un référentiel des services agents ») : ajout d'un contrôle interdisant de lancer un nouvel import de référentiel quand un import est déjà en cours (IMPORT_AGENCIES.CONCURRENCE.KO).

11.0	04/03/2022	AGR	Finalisation pour publication Version 5
-------------	------------	-----	---

Licence

La solution logicielle VITAM est publiée sous la licence CeCILL 2.1 ; la documentation associée (comprenant le présent document) est publiée sous [Licence Ouverte V2.0](#).

Table des matières

1. Introduction.....	7
1.1. Objectif du document.....	7
1.2. Description d'un processus.....	7
1.3. Structure d'un fichier Properties du Workflow.....	8
2. Ingest de plan de classement.....	11
Workflow d'entrée d'un plan de classement.....	12
2.1.1. Processus d'entrée d'un plan de classement (vision métier).....	12
2.1.2. Structure de workflow d'entrée d'un plan de classement.....	13
3. Masterdata.....	17
3.1. Workflow d'import d'un arbre de positionnement.....	17
3.1.1. Processus d'import d'un arbre (HOLDINGScheme - vision métier).....	17
3.1.2. Structure du Workflow d'import d'un arbre de positionnement.....	18
3.2. Workflow d'administration d'un référentiel de règles de gestion.....	21
3.2.1. Processus d'administration d'un référentiel de règles de gestion (STP_IMPORT_RULES).....	21
3.2.2. Structure du Workflow d'import d'un référentiel des règles de gestion.....	24
3.2.3. Structure du rapport d'entrée du référentiel des règles de gestion.....	25
3.3. Workflow d'administration d'un référentiel des formats.....	27
3.3.1. Processus d'import et de mise à jour d'un référentiel de formats (STP_REFERENTIAL_FORMAT_IMPORT).....	27
3.3.2. Structure du Workflow d'import d'un référentiel des formats.....	28
3.3.3. Structure du rapport d'administration d'un référentiel des formats.....	28
3.4. Workflow d'administration d'un référentiel des services agents.....	31
3.4.1. Processus d'import et mise à jour d'un référentiel de services agents (STP_IMPORT_AGENCIES).....	31
3.4.2. Structure du Workflow d'import d'un référentiel des services agents.....	33
3.4.3. Structure du rapport d'import du référentiel des services agents.....	34
3.5. Workflow d'administration d'un référentiel des contrats d'entrée.....	35
3.5.1. Processus d'import d'un contrat d'entrée (STP_IMPORT_INGEST_CONTRACT).....	35
3.5.2. Structure du Workflow d'import d'un référentiel des contrats d'entrée.....	37
3.5.3. Processus de mise à jour d'un contrat d'entrée (STP_UPDATE_INGEST_CONTRACT).....	37
3.5.4. Structure du Workflow de mise à jour d'un référentiel des contrats d'entrée.....	38
3.6. Workflow d'administration d'un référentiel des contrats d'accès.....	39
3.6.1. Processus d'import et mise à jour d'un contrat d'accès (STP_IMPORT_ACCESS_CONTRACT).....	39
3.6.2. Structure du Workflow d'import du référentiel des contrats d'accès.....	40
3.6.3. Processus de mise à jour d'un contrat d'accès STP_UPDATE_ACCESS_CONTRACT.....	41
3.6.4. Structure du Workflow de mise à jour d'un référentiel des contrats d'accès.....	43
3.7. Workflow d'administration d'un référentiel des contrats de gestion.....	44
3.7.1. Processus d'import et mise à jour d'un contrat de gestion (STP_IMPORT_MANAGEMENT_CONTRACT).....	44
3.7.2. Structure du Workflow d'import du référentiel des contrats de gestion.....	45
3.7.3. Processus de mise à jour d'un contrat de gestion (STP_UPDATE_MANAGEMENT_CONTRACT).....	45
3.7.4. Structure du Workflow de mise à jour d'un référentiel des contrats de gestion.....	47
3.8. Workflow d'administration d'un référentiel des profils d'archivage.....	48
3.8.1. Processus d'import d'une notice de profil d'archivage (STP_IMPORT_PROFILE_JSON).....	48
3.8.2. Structure du Workflow d'import d'une notice de profil d'archivage.....	49
3.8.3. Processus de mise à jour d'une notice de profil d'archivage (STP_UPDATE_PROFILE_JSON).....	49
3.8.4. Structure du Workflow de mise à jour d'une notice de profil d'archivage.....	51
3.9. Workflow d'administration d'un référentiel des profils de sécurité.....	52

3.9.1. Processus d'import d'un référentiel des profils de sécurité (STP_IMPORT_SECURITY_PROFILE)	52
3.9.2. Structure du Workflow d'import d'un référentiel des profils de sécurité.....	53
3.9.3. Processus de mise à jour d'un référentiel des profils de sécurité (STP_UPDATE_SECURITY_PROFILE).....	53
3.9.4. Structure du Workflow de mise à jour du référentiel des profils de sécurité.....	54
3.10. Workflow d'administration d'un référentiel des contextes applicatifs.....	55
3.10.1. Processus d'import d'un référentiel des contextes applicatifs (STP_IMPORT_CONTEXT).....	55
3.10.2. Structure du Workflow d'import d'un référentiel des contextes applicatifs.....	56
3.10.3. Processus de mise à jour d'un référentiel des contextes applicatifs (STP_UPDATE_CONTEXT)....	56
3.10.4. Structure du Workflow de mise à jour du référentiel des contextes applicatifs.....	57
3.11. Workflow d'administration du référentiel des profils d'unités archivistiques.....	58
3.11.1. Processus d'import d'un référentiel des profils d'unité archivistique (IMPORT_ARCHIVEUNITPROFILE).....	58
3.11.2. Structure du Workflow d'import d'un référentiel des profils d'unité archivistique.....	59
3.11.3. Processus de mise à jour d'un profil d'unité archivistique (UPDATE_ARCHIVEUNITPROFILE). 59	
3.11.4. Structure du Workflow de mise à jour du référentiel des profils d'unité archivistique.....	61
3.12. Workflow d'administration d'un référentiel des vocabulaires de l'ontologie.....	62
3.12.1. Processus d'import et mise à jour des vocabulaires de l'ontologie (STP_IMPORT_ONTOLOGY)..	62
3.12.2. Structure du Workflow d'import et de mise à jour d'un référentiel des vocabulaires de l'ontologie. 64	
3.12.3. Structure du rapport d'administration des vocabulaires de l'ontologie.....	64
3.13. Workflow d'administration d'un référentiel des griffons.....	67
3.13.1. Processus d'import et mise à jour des griffons (STP_IMPORT_GRIFFIN).....	67
3.13.2. Structure du Workflow d'import d'un référentiel des griffons.....	68
3.13.3. Structure du rapport d'administration du référentiel des griffons.....	68
3.14. Workflow d'administration d'un référentiel des scénarios de préservation.....	70
3.14.1. Processus d'import et de mise à jour des scénarios de préservation (STP_IMPORT_SCENARIO). 70	
3.14.2. Structure du Workflow d'import d'un référentiel des scénarios de préservation.....	72
3.14.3. Structure du rapport d'administration du référentiel des scénarios de préservation.....	72

1. INTRODUCTION

Avertissement : Cette documentation reflète l'état actuel de la solution logicielle Vitam. Elle est susceptible de changer dans les prochaines versions pour tenir compte des développements de la solution logicielle Vitam.

1.1. Objectif du document

Ce document a pour objectif de présenter les différents processus employés par la solution logicielle Vitam. Il est destiné aux administrateurs aussi bien techniques que fonctionnels, aux archivistes souhaitant avoir une connaissance plus avancée du logiciel ainsi qu'aux développeurs.

Il explicite chaque processus (appelé également « workflow »), et pour chacun leurs tâches, traitements et actions.

Ce document comprend du matériel additionnel pour faciliter la compréhension des processus comme des fiches récapitulatives et des schémas. Il explique également la manière dont est formée la structure des fichiers de workflow.

Cette première partie du document décrit les workflows suivants :

- Ingest de plan de classement
- Masterdata

1.2. Description d'un processus

Un workflow est un processus composé d'étapes (macro-workflow), elles-mêmes composées d'une liste de tâches et d'actions à exécuter de manière séquentielle, unitairement ou de manière itérative sur une liste d'éléments (micro-workflow).

Pour chacun de ces éléments, le document décrit :

- La règle générale qui s'applique à cet élément,
- Les statuts de sortie possibles (OK, KO...), avec les raisons de ces sorties et les clés associées,
- Des informations complémentaires, selon le type d'élément traité.

Un « traitement » désigne ci-dessous une opération, une étape ou une tâche. Chaque traitement peut avoir à son issue un des statuts suivant :

- OK : le traitement s'est déroulé comme attendu et le système a été modifié en conséquence.
- Warning : le traitement a atteint son objectif mais le système émet une réserve. Soit :
 - Le système suspecte une anomalie lors du déroulement du traitement sans pouvoir le confirmer lui-même et lève une alerte à destination de l'utilisateur afin que celui-ci puisse valider qu'il s'agit du comportement souhaité.
Exemple : un SIP versé sans objet provoque une opération en warning, car le fait de ne verser qu'une arborescence d'unités archivistiques sans aucun objet peut être suspect (au sens métier).
 - Le système a effectué un traitement entraînant une modification de données initialement non prévue par l'utilisateur.
Exemple : la solution logicielle Vitam a détecté un format de fichier en contradiction avec le format décrit dans le bordereau de transfert. Elle enregistre alors ses propres valeurs en base de données au lieu de prendre celles du bordereau et utilise le warning pour en avertir l'utilisateur.
 - Le système a effectué un traitement dont seule une partie a entraîné une modification de données. L'autre partie de ce traitement s'est terminée en échec sans modification (KO).
Exemple : une modification de métadonnées en masse d'unités archivistiques dont une partie de la modification est OK et une partie est KO : le statut de l'étape et de l'opération sera Warning.
- KO : le traitement s'est terminé en échec et le système n'a pas été modifié en dehors des éléments de

traçabilités tels que les journaux et les logs. L'intégralité du traitement pourrait être rejouée sans provoquer l'insertion de doublons.

- Fatal : le traitement s'est terminé en échec a cause d'un problème technique. L'état du système dépend de la nature du traitement en fatal et une intervention humaine est requise pour expertiser et résoudre la situation. Lorsque le statut FATAL survient à l'intérieur d'une étape (par exemple dans une des tâches ou une des actions de l'étape), c'est toute l'étape qui est mise en pause. Si cette étape est rejouée, les objets déjà traités avant le problème technique ne sont pas traités à nouveau : le workflow reprend exactement là où il s'était arrêté et commence par rejouer l'action sur l'objet qui a provoqué l'erreur.

Un workflow peut être terminé, en cours d'exécution ou être en pause. Un workflow en pause représente le processus arrêté à une étape donnée. Chaque étape peut être mise en pause : ce choix dépend du mode de versement (le mode pas à pas marque une pause à chaque étape), ou du statut (le statut FATAL met l'étape en pause). Les workflows en pause sont visibles dans l'IHM dans l'écran « Gestion des opérations ».

Chaque action peut avoir les modèles d'exécutions suivants (toutes les étapes sont par défaut bloquantes) :

- Bloquant
 - Si une action bloquante est identifiée en erreur, le workflow est alors arrêté en erreur. Seules les actions nécessaires à l'arrêt du workflow sont alors exécutées.
- Non bloquant
 - Si une action non bloquante est identifiée en erreur, elle seule sera en erreur et le workflow continuera normalement.

1.3. Structure d'un fichier *Properties du Workflow*

Les fichiers **Properties** (par exemple *DefaultIngestWorkflow.json*) permettent de définir la structure du Workflow pour les étapes, tâches et traitements réalisés dans le module d'Ingest Interne, en excluant les étapes et traitements réalisés dans le module d'Ingest externe.

Un Workflow est défini en JSON avec la structure suivante :

- un bloc en-tête contenant :
 - **ID** : identifiant unique du workflow,
 - **Identifiant** : clé du workflow,
 - **Name** : nom du workflow,
 - **TypeProc** : catégorie du workflow,
 - **Comment** : description du workflow ou toutes autres informations utiles concernant le workflow.
- une liste d'étapes dont la structure est la suivante :
 - **WorkerGroupId** : identifiant de famille de Workers,
 - **StepName** : nom de l'étape, servant de clé pour identifier l'étape,
 - **Behavior** : modèle d'exécution pouvant avoir les types suivants :
 - **BLOCKING** : le traitement est bloqué en cas d'erreur, il est nécessaire de recommencer à la tâche en erreur. Les étapes **FINALLY** (définition ci-dessous) sont tout de même exécutées,
 - **NOBLOCKING** : le traitement peut continuer malgré les éventuelles erreurs ou avertissements,
 - **FINALLY** : le traitement correspondant est toujours exécuté, même si les étapes précédentes se sont terminées en échec.
 - **Distribution** : modèle de distribution, décrit comme suit :
 - **Kind** : un type pouvant être REF (un élément unique) ou LIST (une liste d'éléments hiérarchisés) ou encore LIST_IN_FILE (liste d'éléments),
 - **Element** : l'élément de distribution indiquant l'élément unique sous forme d'URI (REF) ou la liste d'éléments en pointant vers un dossier (LIST),
 - **Type** : le type des objets traités (ObjectGroup uniquement pour le moment),
 - **StatusOnEmptyDistribution** : permet dans le cas d'un traitement d'une liste vide, de

- surcharger le statut WARNING par un statut prédéfini,
- **BulkSize**: taille de la liste, valeur à spécifier ex: « bulkSize » : 1000. La valeur par défaut est de 16.
- une liste d'Actions :
 - **ActionKey** : nom de l'action
 - **Behavior** : modèle d'exécution pouvant avoir les types suivants :
 - **BLOCKING** : l'action est bloquante en cas d'erreur. Les actions suivantes (de la même étape) ne seront pas exécutées,
 - **NOBLOCKING** : l'action peut continuer malgré les éventuels erreurs ou avertissements.
 - **LifecycleLog**: action indiquant le calcul sur les LFC. Valeur du champ « DISABLED ».
 - **In** : liste de paramètres d'entrées :
 - **Name** : nom utilisé pour référencer cet élément entre différents handlers d'une même étape,
 - **URI** : cible comportant un schéma (WORKSPACE, MEMORY, VALUE) et un path où chaque handler peut accéder à ces valeurs via le handlerIO :
 - **WORKSPACE** : path indiquant le chemin relatif sur le workspace (implicitement un File),
 - **MEMORY** : path indiquant le nom de la clef de valeur (implicitement un objet mémoire déjà alloué par un handler précédent),
 - **VALUE** : path indiquant la valeur statique en entrée (implicitement une valeur String).
- **Out** : liste de paramètres de sorties :
 - **Name** : nom utilisé pour référencer cet élément entre différents handlers d'une même étape,
 - **URI** : cible comportant un schéma (WORKSPACE, MEMORY) et un path où chaque handler peut stocker les valeurs finales via le handlerIO :
 - **WORKSPACE** : path indiquant le chemin relatif sur le workspace (implicitement un File local),
 - **MEMORY** : path indiquant le nom de la clé de valeur (implicitement un objet mémoire).

```

{
  "id": "DEFAULT_WORKFLOW",
  "name": "Default Ingest Workflow",
  "identifiant": "PROCESS_SIP_UNITARY",
  "typeProc": "INGEST",
  "comment": "Default Ingest Workflow V6",
  "steps": [
    {
      "workerGroupId": "DefaultWorker",
      "stepName": "STP_INGEST_CONTROL_SIP",
      "behavior": "BLOCKING",
      "distribution": {
        "kind": "REF",
        "element": "SIP/manifest.xml"
      },
      "actions": [
        {
          "action": {
            "actionKey": "CHECK_SEDA",
            "behavior": "BLOCKING"
          }
        },
        {
          "action": {
            "actionKey": "CHECK_HEADER",
            "behavior": "BLOCKING",
            "in": [
              {
                "name": "checkContract",
                "uri": "VALUE:true"
              },
              {
                "name": "checkOriginatingAgency",
                "uri": "VALUE:true"
              },
              {
                "name": "checkProfile",
                "uri": "VALUE:true"
              }
            ]
          }
        }
      ]
    }
  ]
}

```

Bloc d'en tête

Nom de l'étape

Liste des tâches de l'étape

Illustration 1: Exemple d'un fichier properties - Ingest Interne

2. INGEST DE PLAN DE CLASSEMENT

Workflow d'entrée d'un plan de classement

Cette section décrit le processus d'entrée d'un plan de classement dans la solution logicielle Vitam.

Le processus d'entrée d'un plan de classement est identique au workflow d'entrée d'un SIP. Il débute lors du lancement du téléchargement d'un plan de classement dans la solution logicielle Vitam. Toutes les étapes, tâches et traitements sont journalisés dans le journal des opérations.

Les étapes, tâches et traitements associées ci-dessous décrivent le processus d'entrée d'un plan (clé et description de la clé associée dans le journal des opérations), non encore abordées dans la description de l'entrée d'un SIP.

Toutes les étapes, tâches et traitements sont journalisées dans le journal des opérations. Les étapes et actions associées ci-dessous décrivent le processus (clé et description de la clé associée dans le journal des opérations) tel qu'implémenté dans la version actuelle de la solution logicielle Vitam.

2.1.1. Processus d'entrée d'un plan de classement (vision métier)

La structure d'un plan de classement diffère de celle d'un SIP par l'absence d'objet et de vérification de l'offre de stockage. Il s'agit plus simplement d'une arborescence représentée par des unités archivistiques. Ce processus partage donc certaines étapes avec celui du transfert d'un SIP classique, en ignore certaines et rajoute des tâches additionnelles.

2.1.1.1. Traitement additionnel dans la tâche CHECK_DATAOBJECTPACKAGE

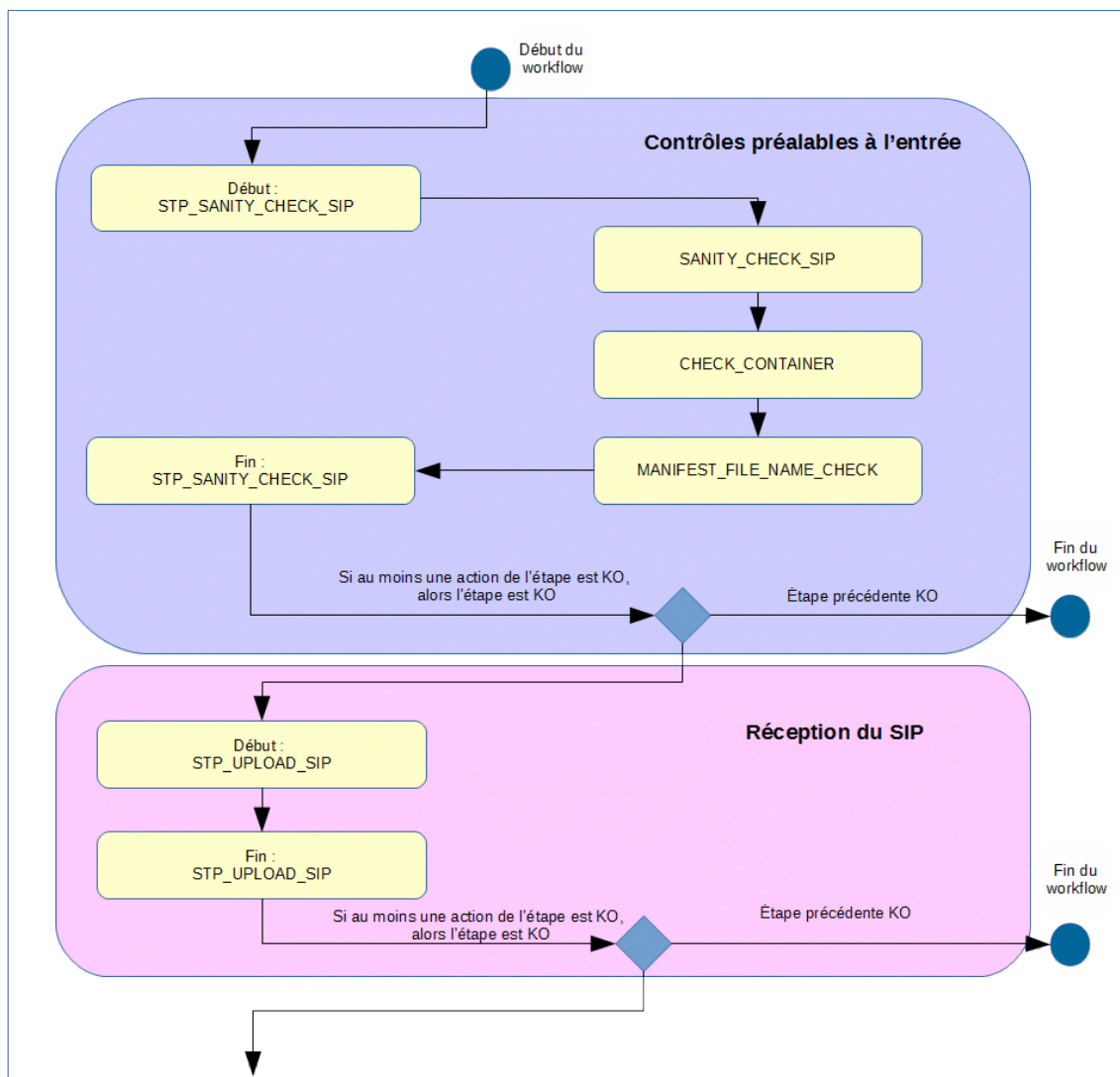
Vérification de la non-existence d'objets (CHECK_NO_OBJECT)

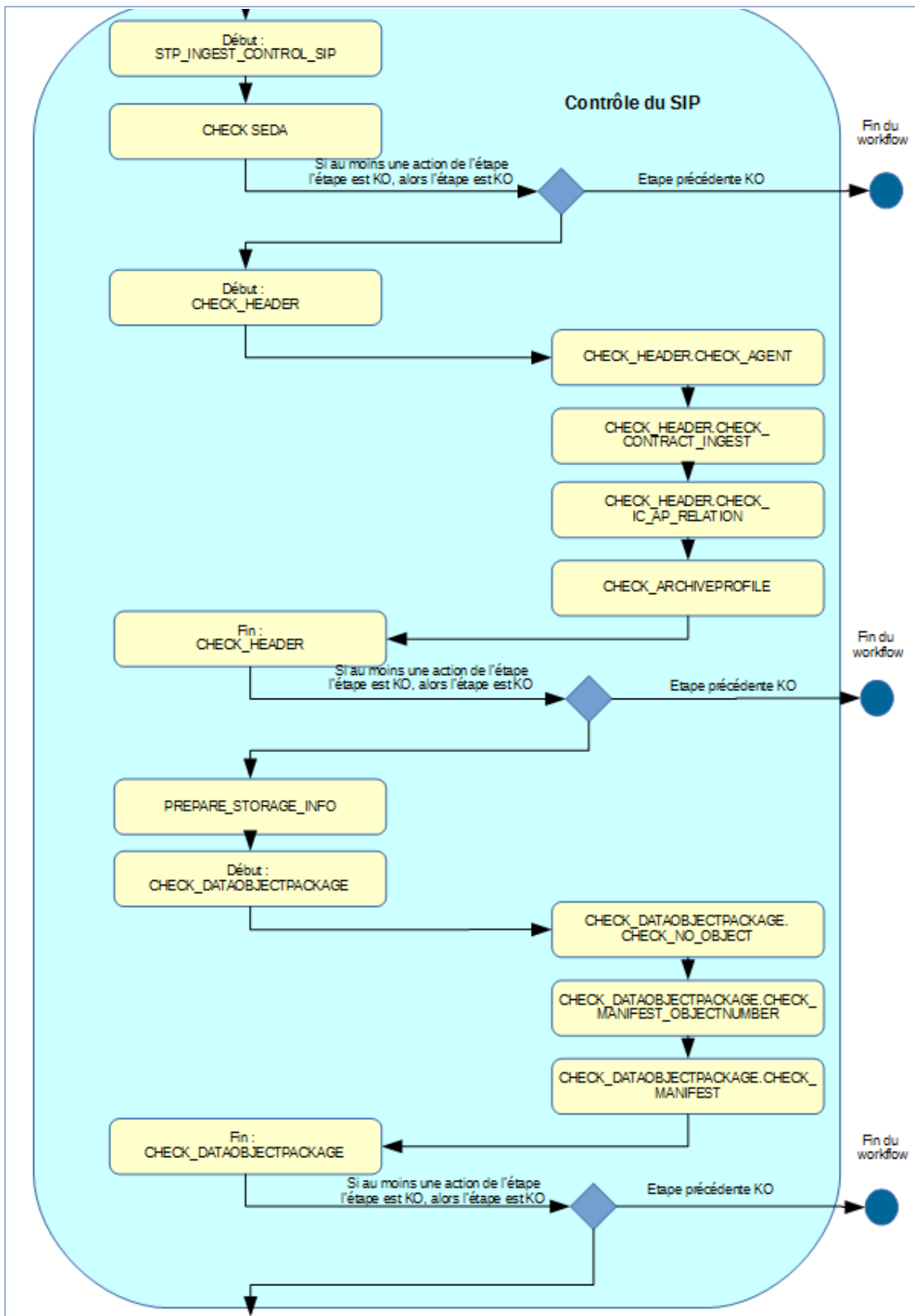
- **Règle** : traitement consistant à vérifier qu'il n'y a pas d'objet binaire dans le bordereau de transfert du plan de classement
- **Type** : bloquant
- **Statuts** :
 - OK : aucun objet binaire n'est présent dans le bordereau de transfert (CHECK_DATAOBJECTPACKAGE.CHECK_NO_OBJECT.OK = Succès de la vérification de l'absence d'objet)
 - KO : des objets binaires sont présents dans le bordereau de transfert (CHECK_DATAOBJECTPACKAGE.CHECK_NO_OBJECT.KO = Échec de la vérification de l'absence d'objet : objet(s) trouvé(s))
 - FATAL : une erreur technique est survenue lors de la vérification de la non-existence d'objet binaire (CHECK_DATAOBJECTPACKAGE.CHECK_NO_OBJECT.FATAL = Erreur technique lors de la vérification de l'absence d'objet)

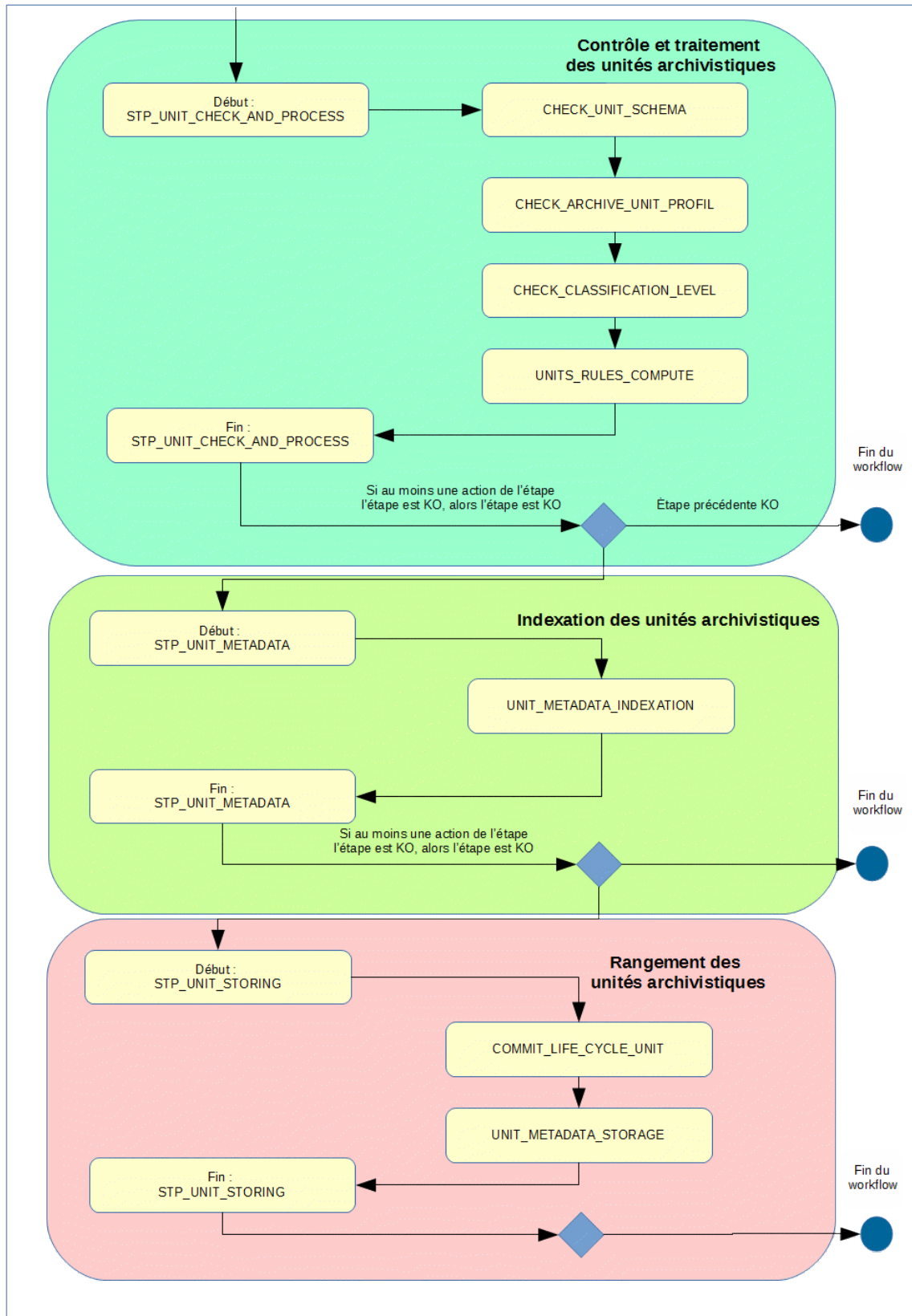
2.1.2. Structure de workflow d'entrée d'un plan de classement

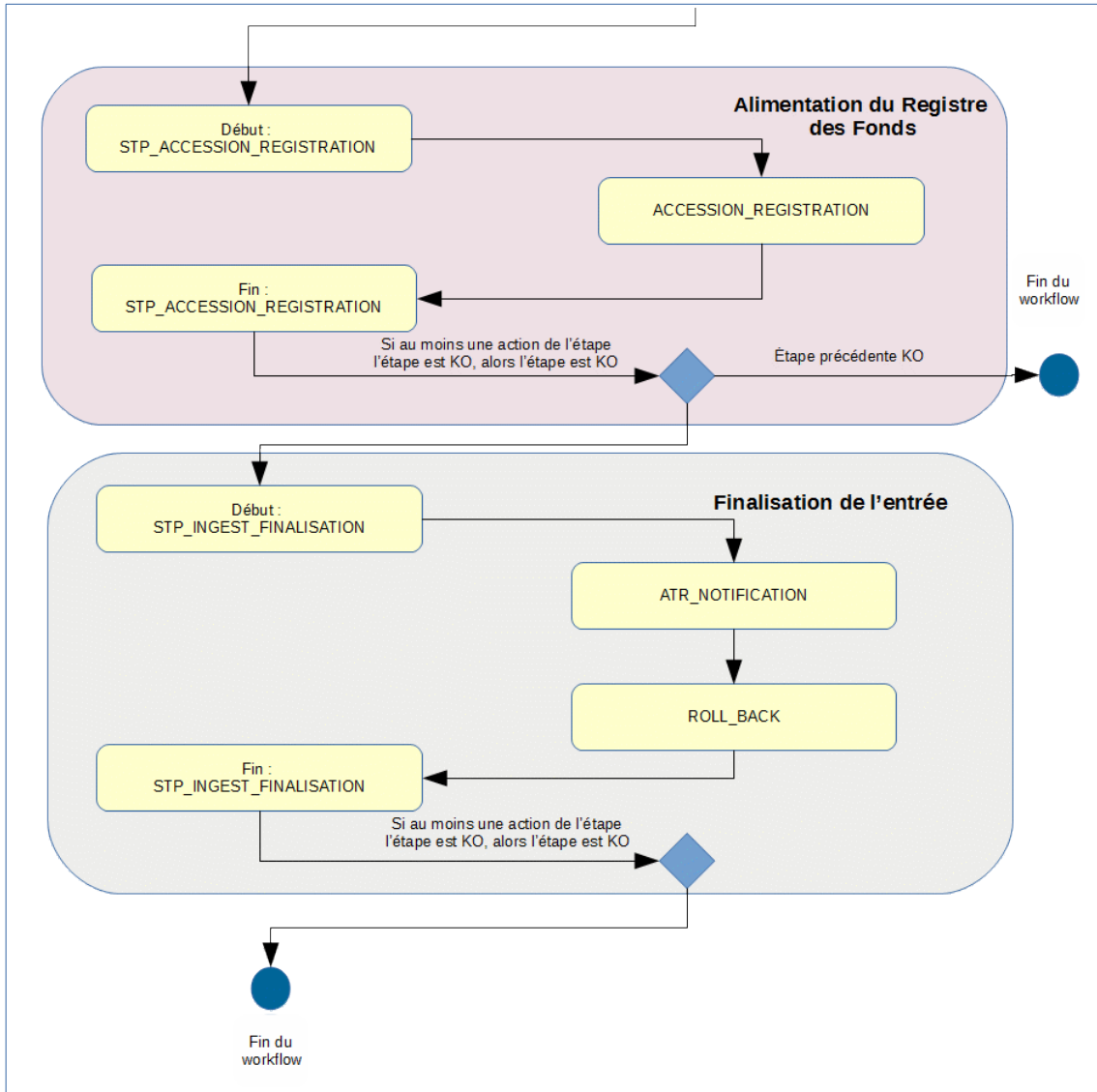
Le workflow actuel mis en place dans la solution logicielle Vitam est défini dans le fichier « *DefaultFilingSchemeWorkflow.json* ». Ce fichier est disponible dans : sources/processing/processing-management/src/main/resources/workflows.

D'une façon synthétique, le workflow est décrit de cette façon :









3. MASTERDATA

Cette section décrit les processus (workflows) d'administration des différents référentiels de la solution logicielle Vitam. Ceux-ci se construisent sur la base de fichiers à importer. La structure de ces fichiers et la description de leurs contenus sont décrites dans la documentation relative au modèle de données. Si un des fichiers importés contient des balises HTML, son contenu sera considéré comme dangereux et l'import sera rejeté. Ce rejet ne fera pas l'objet d'une opération et ne sera donc pas enregistré dans le journal des opérations. En revanche, une alerte de sécurité sera émise dans un log de sécurité de la solution logicielle Vitam, pour informer l'administrateur de cette tentative.

3.1. Workflow d'import d'un arbre de positionnement

Cette section décrit le processus permettant d'importer un arbre de positionnement dans la solution logicielle Vitam. La structure d'un arbre de positionnement diffère de celle d'un SIP en plusieurs points.

Un arbre ne doit pas avoir d'objet, ni de service producteur, ni de règle de gestion associée. Il s'agit plus simplement d'une arborescence représentée par des unités archivistiques. Ce processus partage donc certaines étapes avec celui du transfert d'un SIP classique, en ignore certaines et rajoute des tâches additionnelles.

Toutes les étapes, tâches et traitements sont journalisés dans le journal des opérations et décrivent le processus (clé et description de la clé associée dans le journal des opérations) tel qu'implémenté dans la version actuelle de la solution logicielle Vitam.

3.1.1. Processus d'import d'un arbre (HOLDINGScheme - vision métier)

Le processus d'import d'un arbre est identique au workflow d'entrée d'un SIP. Il débute lors du lancement du téléchargement de l'arbre dans la solution logicielle Vitam. Par ailleurs, toutes les étapes, tâches et traitements sont journalisés dans le journal des opérations.

La fin du processus peut prendre plusieurs statuts :

- **Statuts :**
 - OK : l'arbre de positionnement a été importé (HOLDINGScheme.OK = Succès de l'import de l'arbre de positionnement)
 - KO : l'arbre de positionnement n'a pas été importé (HOLDINGScheme.KO = Échec de l'import de l'arbre de positionnement)
 - FATAL : une erreur technique est survenue lors de l'import de l'arbre de positionnement (HOLDINGScheme.FATAL = Erreur technique lors de l'import de l'arbre de positionnement)

Les étapes, tâches et traitements associées ci-dessous décrivent le processus d'import d'un arbre de positionnement (clé et description de la clé associée dans le journal des opérations), non encore abordées dans la description de l'entrée d'un SIP.

Traitement additionnel dans la tâche CHECK_DATAOBJECTPACKAGE (CheckDataObjectPackageActionHandler.java)

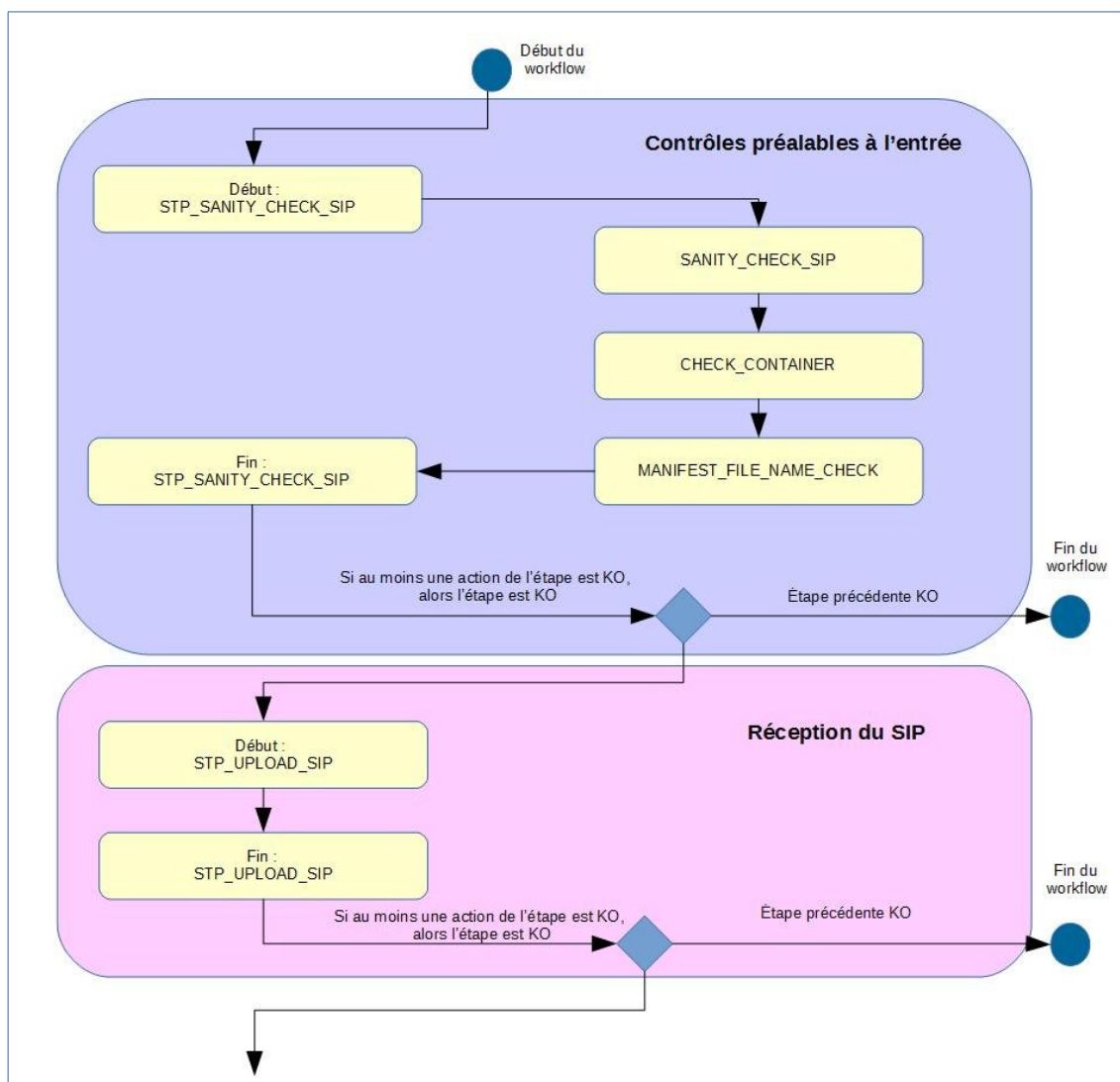
Vérification de la non-existence d'objets CHECK_NO_OBJECT (CheckDataObjectPackageActionHandler)

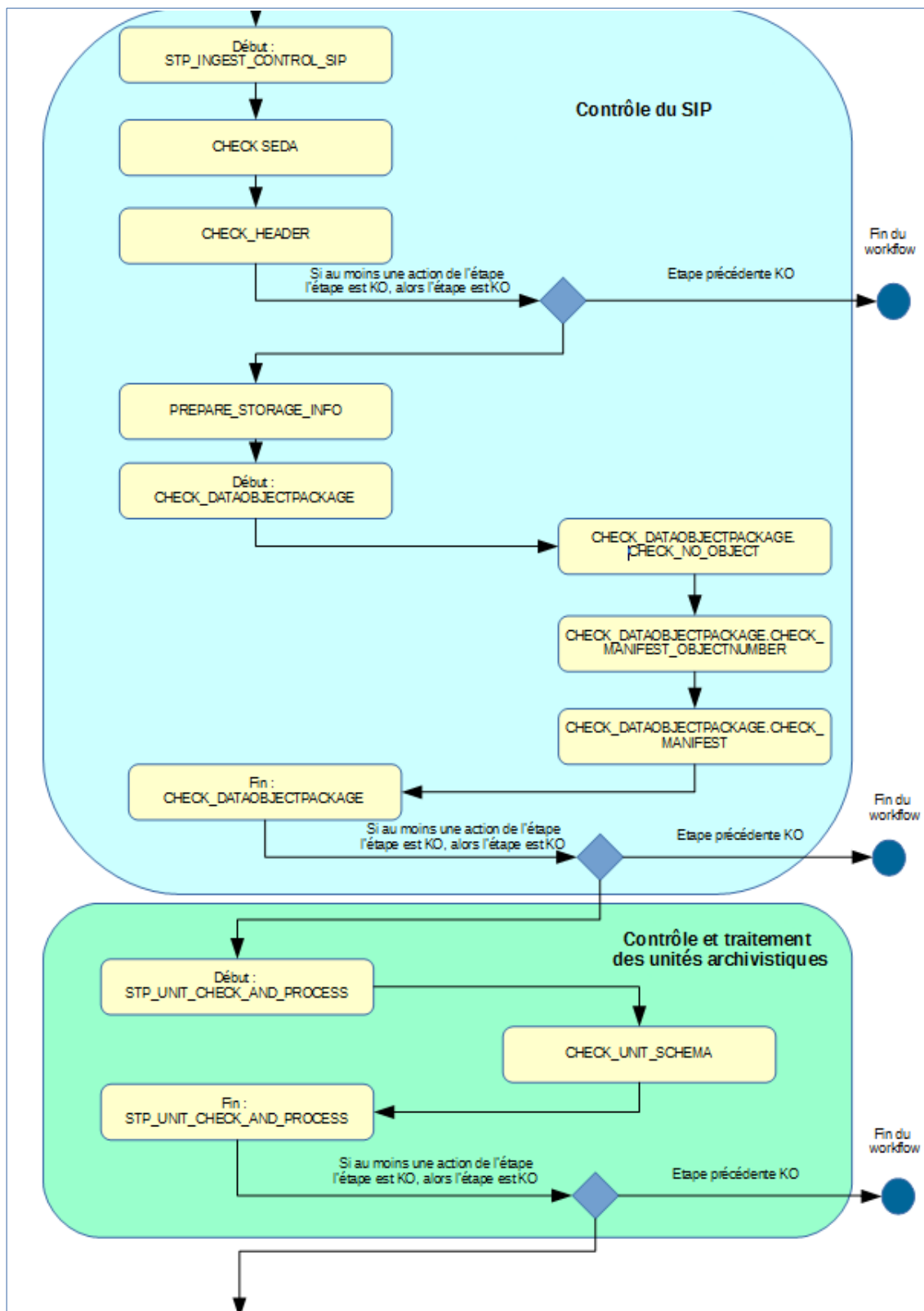
- **Règle :** traitement consistant à vérifier qu'il n'y a pas d'objet binaire dans le bordereau de transfert de l'arbre de positionnement.
 - **Statuts :**
 - OK : aucun objet binaire n'est présent dans le bordereau de transfert (CHECK_DATAOBJECTPACKAGE.CHECK_NO_OBJECT.OK = Succès de la vérification de l'absence d'objet)
 - KO : des objets binaires sont présents dans le bordereau de transfert (CHECK_DATAOBJECTPACKAGE.CHECK_NO_OBJECT.KO = Échec de la vérification de l'absence d'objet : objet(s) trouvé(s))

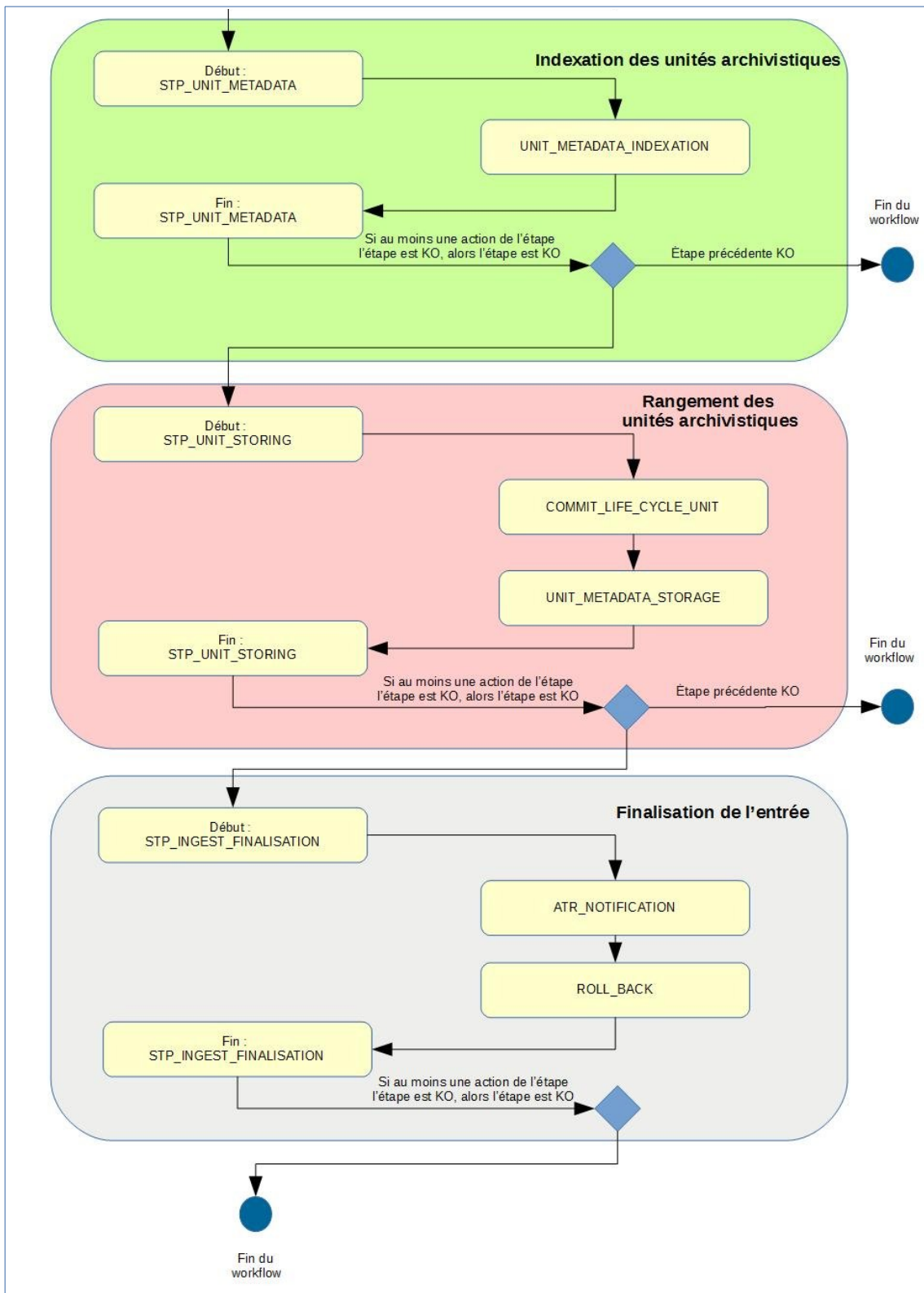
- **FATAL** : une erreur technique est survenue lors de la vérification de la non-existence d'objet binaire (CHECK_DATAOBJECTPACKAGE.CHECK_NO_OBJECT.FATAL = Erreur technique lors de la vérification de l'absence d'objet)

3.1.2. Structure du Workflow d'import d'un arbre de positionnement

D'une façon synthétique, le workflow est décrit de cette façon :







3.2. Workflow d'administration d'un référentiel de règles de gestion

Cette section décrit le processus (workflow) permettant d'importer et de mettre à jour un référentiel de règles de gestion dans la solution logicielle Vitam.

Toutes les étapes, tâches et traitements sont journalisés dans le journal des opérations et décrivent le processus (clé et description de la clé associée dans le journal des opérations) tel qu'implémenté dans la version actuelle de la solution logicielle Vitam.

3.2.1. Processus d'administration d'un référentiel de règles de gestion (STP_IMPORT_RULES)

L'import d'un référentiel de règles de gestion permet de vérifier le formalisme de ce dernier, notamment que les données obligatoires sont bien présentes pour chacune des règles. Tous les éléments réalisés au cours de ce processus sont exécutés dans une seule étape. Cet import concerne aussi bien l'import initial (aucune règle de gestion pré-existante) que la mise à jour du référentiel.

Ce processus d'import débute lors du lancement du téléchargement dans la solution logicielle Vitam du fichier CSV contenant le référentiel.

La fin du processus peut prendre plusieurs statuts :

- **Statuts :**
 - OK : le référentiel des règles de gestion a été importé (STP_IMPORT_RULES.OK = Succès du processus d'import du référentiel des règles de gestion)
 - WARNING : le référentiel des règles de gestion a été importé et ce nouvel import modifie des règles de gestions préalablement utilisées par des unités archivistiques dans la solution logicielle Vitam (STP_IMPORT_RULES.WARNING = Avertissement lors du processus d'import des règles de gestion, des règles de gestions ont été modifiées et sont utilisées par des unités archivistiques existantes)
 - KO : le référentiel des règles de gestion n'a pas été importé (STP_IMPORT_RULES.KO = Échec du processus d'import du référentiel des règles de gestion)
 - FATAL : une erreur technique est survenue lors de l'import du référentiel des règles de gestion (STP_IMPORT_RULES.FATAL = Erreur technique lors du processus d'import du référentiel des règles de gestion)

3.2.1.1. Contrôle de la conformité du fichier des règles de gestion CHECK_RULES (UnitsRulesComputePlugin.java)

- **Règle :** tâche consistant à
 - contrôler qu'aucune règle supprimée du référentiel n'est utilisée par une unité archivistique déjà présente dans le système,
 - contrôler les règles modifiées utilisées par des unités archivistiques déjà présentes dans le système,
 - vérifier que les informations obligatoires minimales ont bien été remplies pour chacune des règles, conformément aux exigences du référentiel des règles de gestion. La liste de ces exigences est décrite dans le document modèle de données.

De plus le fichier remplit les conditions suivantes :

- il est au format CSV
- les informations suivantes sont toutes décrites dans cet ordre
 - RuleId
 - RuleType
 - RuleValue
 - RuleDescription
 - RuleDuration

- RuleMeasurement
- Aucune règle supprimée n'est utilisée par une unité archivistique
- Aucune opération d'import de référentiel de règle de gestion n'a lieu en même temps
- Si le tenant définit des durées minimales pour des catégories de règles de gestion (configuration de sécurité), les règles de gestion importées doivent avoir des durées supérieures ou égales à ces durées minimales de sécurité
- **Type** : bloquant
- **Statuts** :
 - OK : les règles ci-dessus sont respectées (CHECK_RULES.OK = Succès du contrôle de la conformité du fichier des règles de gestion)
 - WARNING : une règle modifiée par l'import du référentiel est actuellement utilisée par une unité archivistique (CHECK_RULES.WARNING = Avertissement lors du contrôle de la conformité du fichier des règles de gestion)
 - KO :
 - Cas 1 : une des règles ci-dessus n'est pas respectée. Le détail des erreurs est inscrit dans le rapport d'import du référentiel (CHECK_RULES.KO = Échec du contrôle de la conformité du fichier des règles de gestion)
 - Cas 2 : le fichier CSV n'est pas reconnu comme un CSV valide (CHECK_RULES.INVALID_CSV.KO = Échec du contrôle de la conformité du fichier des règles de gestion : fichier CSV invalide)
 - Cas 3 : une opération de mise à jour du référentiel est déjà en cours (CHECK_RULES.IMPORT_IN_PROCESS.KO = Échec du contrôle de la conformité du fichier car une mise à jour du référentiel est déjà en cours)
 - Cas 4 : au moins une règle de gestion a une durée qui est inférieure à la durée minimale requise sur ce tenant. Selon la configuration de la solution logicielle Vitam, ce cas peut provoquer une alerte de sécurité, enregistrée dans les logs de sécurité. (CHECK_RULES.MAX_DURATION_EXCEEDS.KO = Échec lors du contrôle de sécurité des règles de gestion. Les durées des règles de gestion doivent être supérieures ou égales aux durées minimales requises par le tenant)
 - FATAL : une erreur technique est survenue lors du contrôle des règles de gestion (CHECK_RULES.FATAL = Erreur technique lors du contrôle des règles de gestion)
- Dans le rapport de l'import du référentiel des règles de gestion, des clés plus détaillées peuvent y être inscrites, selon les erreurs rencontrées :
 - Le fichier importé n'est pas au format CSV (STP_IMPORT_RULES_NOT_CSV_FORMAT.KO = Le fichier importé n'est pas au format CSV)
 - Il existe plusieurs fois le même RuleId (STP_IMPORT_RULES_RULEID_DUPLICATION.KO = Il existe plusieurs fois le même RuleId. Ce RuleId doit être unique dans l'ensemble du référentiel)
 - Au moins une RuleType est incorrecte (STP_IMPORT_RULES_WRONG_RULETYPE_UNKNOW.KO = Au moins une RuleType est incorrecte. RuleType autorisés : AppraisalRule, AccessRule, StorageRule, DisseminationRule, ReuseRule, ClassificationRule)
 - Au moins une valeur obligatoire est manquante (STP_IMPORT_RULES_MISSING_INFORMATION.KO = Au moins une valeur obligatoire est manquante. Valeurs obligatoires : RuleID, RuleType, RuleValue, RuleDuration, RuleMeasurement)
 - Des valeurs de durée sont incorrectes pour RuleMeasurement (STP_IMPORT_RULES_WRONG_RULEMEASUREMENT.KO = Au moins un champ RuleDuration a une valeur incorrecte. La valeur doit être un entier positif ou nul, ou être indiquée unlimited)
 - Au moins un champ RuleDuration a une valeur incorrecte (STP_IMPORT_RULES_WRONG RULEDURATION.KO = Au moins un champ RuleDuration a une valeur incorrecte. La valeur doit être un entier positif ou nul, ou être indiquée unlimited)

- L'association de RuleDuration et de RuleMeasurement n'est pas inférieure ou égale à 999 ans (STP_IMPORT_RULES_WRONG_TOTALDURATION.KO = L'association de RuleDuration et de RuleMeasurement doit être inférieure ou égale à 999 ans)
- Des règles supprimées sont actuellement utilisées (STP_IMPORT_RULES_DELETE_USED_RULES.KO = Des règles supprimées sont actuellement utilisées)
- Des durées sont inférieures ou égales aux durées minimales autorisées dans la configuration de la plateforme (STP_IMPORT_RULES RULEDURATION_EXCEED.KO = Échec lors du contrôle de sécurité des règles de gestion. Les durées des règles de gestion doivent être supérieures ou égales aux durées minimales requises par le tenant)
- FATAL : une erreur technique est survenue lors du contrôle des règles de gestion (CHECK_RULES.FATAL=Erreur technique lors du contrôle de la conformité du fichier de règles de gestion)

3.2.1.2. Génération du rapport d'analyse du référentiel des règles de gestion RULES_REPORT (RulesManagerFileImpl.java)

- **Règle** : tâche consistant à créer le rapport d'import des règles
- **Type** : bloquant
- **Statuts** :
 - OK : le rapport est généré (RULES_REPORT.OK = Succès de la génération du rapport d'analyse du référentiel des règles de gestion)
 - KO : pas de cas KO
 - FATAL : une erreur technique est survenue lors de la création du rapport (RULES_REPORT.FATAL = Erreur technique lors de la génération du rapport d'analyse du référentiel des règles de gestion)

3.2.1.3. Persistance des données en base COMMIT_RULES (RulesManagerFileImpl.java)

- **Règle** : tâche consistant à enregistrer les données du référentiel en base
- **Type** : bloquant
- **Statuts** :
 - OK : les données sont persistées en base (COMMIT_RULES.OK = Succès de la persistance des données en base)
 - FATAL : une erreur technique est survenue lors de la persistance des données en base (COMMIT_RULES.FATAL = Erreur technique lors de la persistance des données en base)

3.2.1.4. Enregistrement du fichier d'import du référentiel des règles de gestion STP_IMPORT_RULES_BACKUP_CSV (RulesManagerFileImpl.java)

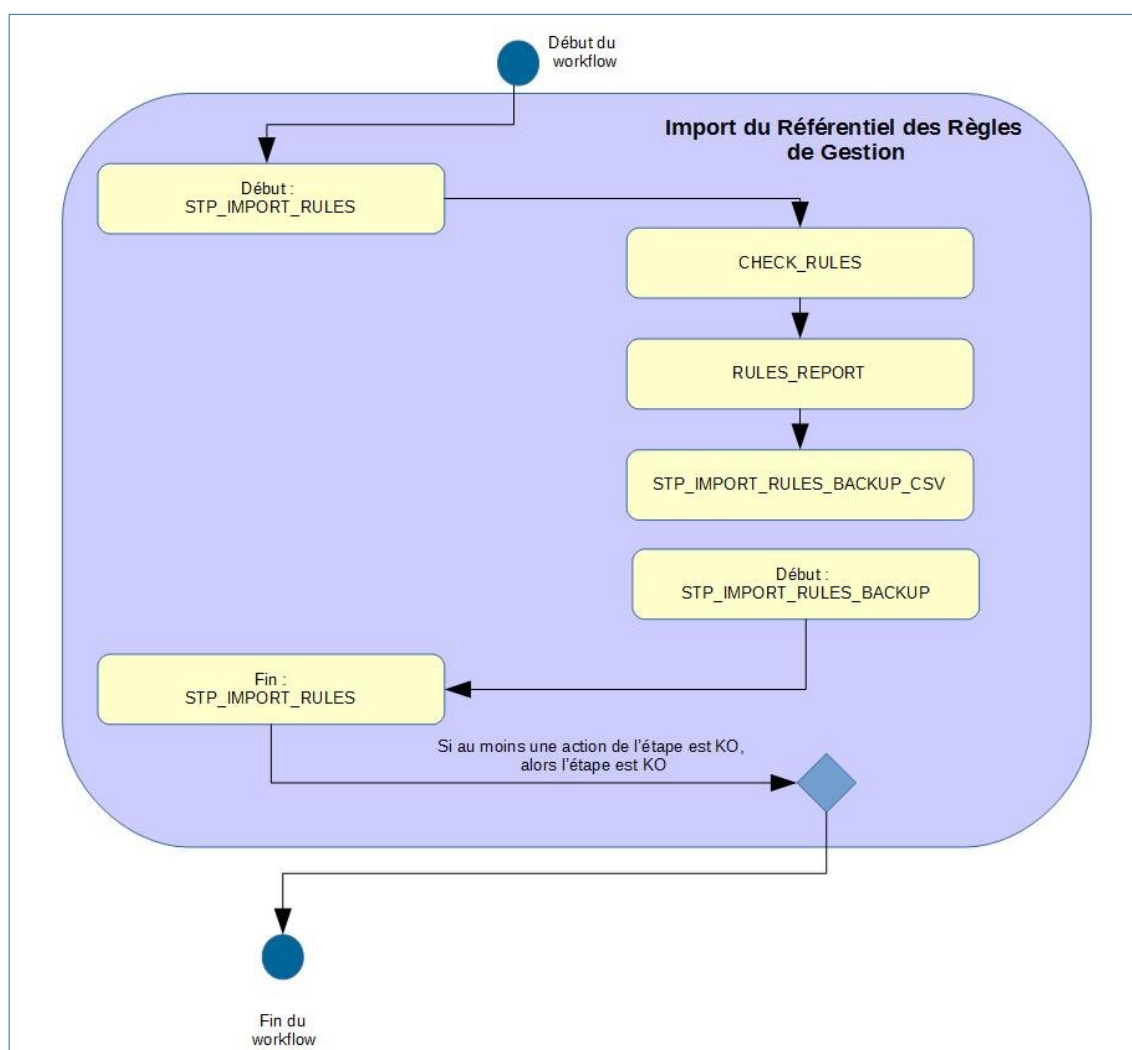
- **Règle** : tâche consistant à écrire le fichier .csv d'import du référentiel des règles de gestion sur les offres de stockage
- **Type** : bloquant
- **Statuts** :
 - OK : le CSV d'import est enregistré (STP_IMPORT_RULES_BACKUP_CSV.OK = Succès du processus d'enregistrement du fichier d'import du référentiel des règles de gestion)
 - KO : pas de cas KO
 - FATAL : une erreur technique est survenue lors de l'enregistrement du CSV d'import (STP_IMPORT_RULES_BACKUP_CSV.FATAL = Erreur technique lors du processus d'enregistrement du fichier d'import du référentiel des règles de gestion)

3.2.1.5. Sauvegarde du JSON STP_IMPORT_RULES_BACKUP (RulesManagerFileImpl.java)

- **Règle** : tâche consistant à écrire le fichier .json correspondant à une copie de la base de données sur les offres de stockage
- **Type** : bloquant
- **Statuts** :
 - OK : une copie de la base de donnée nouvellement importée est enregistrée (STP_IMPORT_RULES_BACKUP.OK = Succès du processus de sauvegarde du référentiel des règles de gestion)
 - KO : pas de cas KO
 - FATAL : une erreur technique est survenue lors de la copie de la base de donnée nouvellement importée (STP_IMPORT_RULES_BACKUP.FATAL = Erreur technique lors du processus de sauvegarde du référentiel des règles de gestion)

3.2.2. Structure du Workflow d'import d'un référentiel des règles de gestion

D'une façon synthétique, le workflow est décrit de cette façon :



3.2.3. Structure du rapport d'entrée du référentiel des règles de gestion

Lorsqu'un nouveau référentiel est importé, la solution logicielle Vitam génère un rapport de l'opération. Ce rapport est en 2 parties :

- « Operation » contient :
 - « evId » : l'identifiant de l'opération.
 - « evDateTime » : la date et l'heure de l'opération d'import.
 - « evType » : le type d'opération. Dans le cadre de ce rapport, il s'agit toujours de « STP_IMPORT_RULES ».
 - « outMessg » : message final de l'opération (Succès/Avertissement/Échec du processus d'import du référentiel des règles de gestion)
- Le détail des opérations effectuées :
 - « updatedRules » : contient l'intégralité des règles dont une mise à jour a été effectuée.
 - « deletedRules » : contient l'intégralité des règles dont une mise à jour a été effectuée.
 - « usedFileRulesToUpdate » : contient l'intégralité des règles en cours d'utilisation dont une mise à jour a été effectuée. Chaque détail précise en plus l'ensemble des informations relatives aux règles de gestion mises à jour, dans la version mise à jour de la règle (identifiant, type, intitulé, description, durée et mesure).
 - « usedFileRulesToDelete » : contient l'intégralité des règles en cours d'utilisation dont la suppression a été demandée lors de la mise à jour du référentiel des règles de gestion. Chaque détail précise en plus l'ensemble des informations relatives aux règles de gestion mises à jour, dans la version mise à jour de la règle (identifiant, type, intitulé, description, durée et mesure).
 - « error » : liste des erreurs rencontrées lors d'un import ou d'une mise à jour, ligne par ligne. Pour chaque ligne du fichier en erreur, le rapport détaille les éléments suivants :
 - « Code » : type de rapport dans la base de données MongoDB
 - « Message » : message exact de l'erreur ou du résultat d'exécution
 - « Information additionnelle » : précision supplémentaire concernant l'erreur rencontrée.

3.2.3.1. Exemples

Exemple 1 : import initial d'un référentiel en succès

Le rapport généré est :

```
"FileRulesToImport": ["APP-00001", "APP-00002", "APP-09002", "APP-09048", "APP-09049",
"APP-09050", "APP-09051", "ACC-00001", "ACC-00002", "ACC-00003", "ACC-00004", "ACC-00005",
"ACC-00006", "ACC-00007", "ACC-00008", "ACC-00009", "ACC-00010", "ACC-00011", "ACC-00012",
"ACC-00013", "ACC-00014", "ACC-00015", "ACC-00016", "ACC-00017", "ACC-00018", "ACC-00019",
"ACC-00020", "ACC-00021", "ACC-00022", "ACC-00023", "ACC-00024", "ACC-00025", "ACC-00026",
"ACC-00027", "ACC-00028", "ACC-00029", "ACC-00030", "ACC-00031", "ACC-00032", "ACC-00033",
"ACC-00034", "ACC-00035", "ACC-00036", "STO-00001", "STO-00002", "STO-00003", "R1", "R2",
"R3", "R4", "R5", "R6", "DIS-00001", "DIS-00002", "DIS-00003", "DIS-00004", "REU-00001",
"CLASS-00001", "APP-1069001", "APP-1069002", "APP-1069003", "APP-1069004", "DIS-1069001"],

"Operation": {
  "evId": "aeaeaaaabchia7juabddealol3qdydiaaaaq",
  "evDateTime": "2019-11-12T09:09:50.676",
  "evType": "STP_IMPORT_RULES",
  "outMessg": "Succès du processus d'enregistrement de la copie du référentiel des
règles de gestion"
```

Exemple 2 : mise à jour d'un référentiel existant en échec

Dans cet exemple, la mise à jour :

- Essayer de modifier une RuleType d'une règle en lui mettant « AccessRulez » au lieu de « AccessRule »
- Met à jour une règle de gestion en cours d'utilisation

Le rapport généré est :

```
{
  "Operation": {
    "evId": "aeeaaaaabchia7juabddealol42lhiaaaaaaq",
    "evDateTime": "2019-11-12T10:42:01.394",
    "evType": "STP_IMPORT_RULES",
    "outMessg": "Échec du processus d'import du référentiel des règles de gestion"
  },
  "error": {
    "line 6": [{
      "Code": "STP_IMPORT_RULES_WRONG_RULETYPE_UNKNOW.KO",
      "Message": "Au moins une RuleType est incorrecte. RuleType
autorisés : AppraisalRule, AccessRule, StorageRule, DisseminationRule, ReuseRule,
ClassificationRule",
      "Information additionnelle": "AccessRulez"
    }]
  }
}
```

3.3.Workflow d'administration d'un référentiel des formats

Cette section décrit le processus (workflow) permettant d'administrer un référentiel des formats. Cette opération n'est réalisable que sur le tenant d'administration.

Toutes les étapes, tâches et traitements sont journalisés dans le journal des opérations et décrivent le processus (clé et description de la clé associée dans le journal des opérations) tel qu'implémenté dans la version actuelle de la solution logicielle Vitam.

3.3.1.Processus d'import et de mise à jour d'un référentiel de formats (STP_REFERENTIAL_FORMAT_IMPORT)

L'import du référentiel des formats permet de vérifier le formalisme de ce dernier, notamment que le fichier doit être au format xml et respecter le formalisme du référentiel PRONOM publié par the National Archives (UK) c'est-à-dire que chaque format contient bien le nombre d'informations minimales attendues. Tous les éléments réalisés au cours de ce processus sont exécutés dans une seule étape.

Ce processus d'import débute lors du lancement du téléchargement dans la solution logicielle Vitam du fichier XML contenant le référentiel.

La fin du processus peut prendre plusieurs statuts

- **Statuts :**
 - OK : les informations correspondant à chaque format sont décrites conformément aux règles émises pour décrire le référentiel PRONOM (STP_REFERENTIAL_FORMAT_IMPORT.OK = Succès du processus d'import du référentiel des formats).
 - KO : la condition ci-dessus n'est pas respectée (STP_REFERENTIAL_FORMAT_IMPORT.KO = Échec du processus d'import du référentiel des formats).
 - WARNING : la version et/ou la date du référentiel PRONOM est/sont validée(s) mais un avertissement signale que la version et/ou la date du référentiel est/sont antérieure(s) à celle(s) du référentiel présent dans la solution logicielle Vitam ou que celle(s)-ci est/sont identique(s) (STP_REFERENTIAL_FORMAT_IMPORT.WARNING = Avertissement lors du processus d'import du référentiel des formats version (n°) du fichier de signature PRONOM)
 - FATAL : une erreur technique est survenue lors de l'import du référentiel des formats (STP_REFERENTIAL_FORMAT_IMPORT.FATAL = Erreur technique lors du processus d'import du référentiel des formats)

3.3.1.1. Processus de sauvegarde du référentiel des formats (STP_BACKUP_REFERENTIAL_FORMAT)

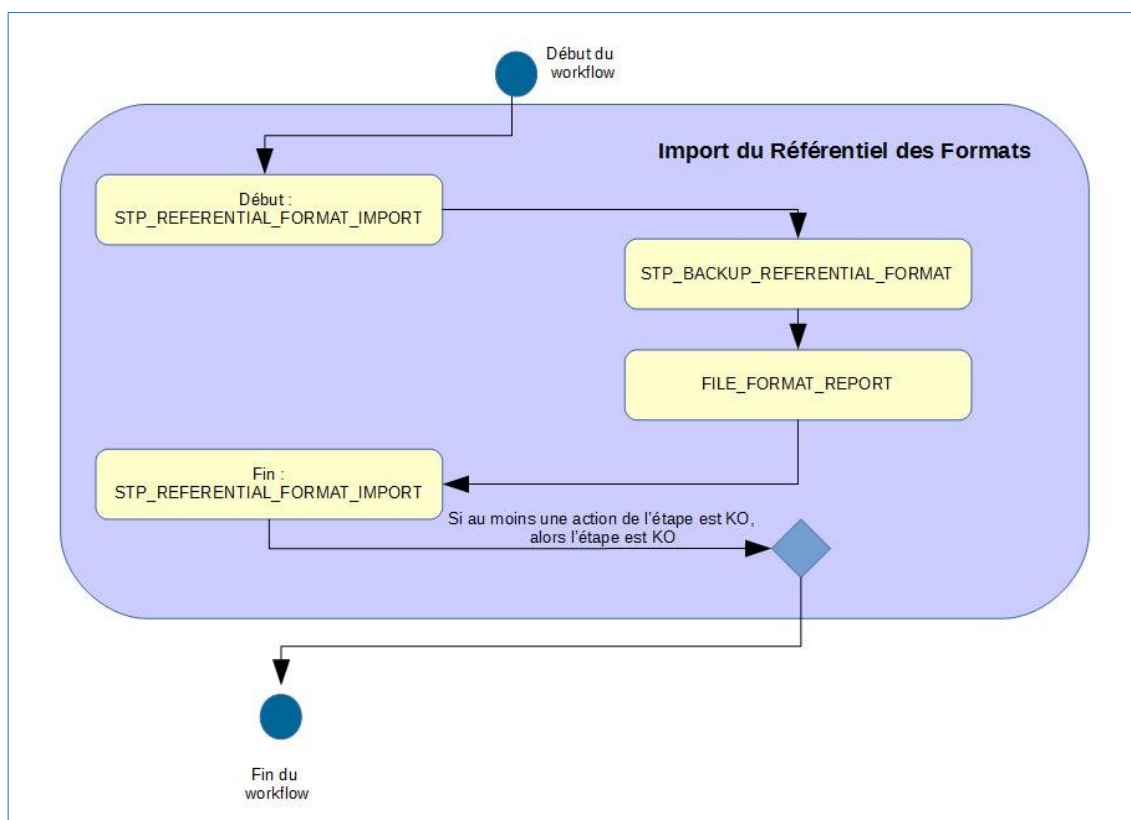
- **Règle :** tâche consistant à sauvegarder le référentiel des formats sur les offres de stockage.
- **Type :** bloquant
- **Statuts :**
 - OK : la sauvegarde du référentiel des formats a bien été effectuée (STP_BACKUP_REFERENTIAL_FORMAT.OK = Succès du processus de sauvegarde du référentiel des formats)
 - KO : la sauvegarde du référentiel des formats n'a pas été effectuée (STP_BACKUP_REFERENTIAL_FORMAT.KO = Échec du processus de sauvegarde du référentiel des formats)
 - FATAL : une erreur technique est survenue lors de la sauvegarde du référentiel des formats (STP_BACKUP_REFERENTIAL_FORMAT.FATAL = Erreur technique lors du processus de sauvegarde du référentiel des formats)

3.3.1.2. Processus de génération du rapport d'import du référentiel des formats FILE_FORMAT_REPORT

- **Règle** : tâche consistant à créer le rapport d'import du référentiel des formats.
- **Type** : bloquant
- **Statuts** :
 - OK : le rapport d'import du référentiel des formats a bien été créé (FILE_FORMAT_REPORT.OK = Succès du processus de génération du rapport d'import du référentiel des formats)
 - KO : pas de KO
 - FATAL : une erreur technique est survenue lors de la création du rapport d'import référentiel des formats (FILE_FORMAT_REPORT.FATAL = Erreur technique lors du processus de génération du rapport du référentiel des formats)

3.3.2. Structure du Workflow d'import d'un référentiel des formats

D'une façon synthétique, le workflow est décrit de cette façon :



3.3.3. Structure du rapport d'administration d'un référentiel des formats

Lorsqu'un nouveau référentiel est importé, la solution logicielle Vitam génère un rapport de l'opération. Ce rapport est en plusieurs parties :

- « Operation » contient :
 - evType : le type d'opération. Dans le cadre de ce rapport, il s'agit toujours de « STP_REFERENTIAL_FORMAT_IMPORT »
 - evDateTime : la date et l'heure du début de la génération du rapport
 - evId : l'identifiant de l'opération

- « StatusCode » : le statut de l’opération (OK, KO, WARNING)
- « PreviousPronomVersion » : le numéro de la version du référentiel précédemment installé dans la solution logicielle Vitam
- « PreviousPronomCreationDate » : la date de création du référentiel précédemment installé dans la solution logicielle Vitam
- « NewPronomVersion » : le numéro de version du référentiel désormais installé
- « NewPronomCreationDate » : la date de création du référentiel désormais installé
- « RemovedPUIDs » : la liste des PUID qui ont été supprimés
- « AddedPUID »s : la liste des PUID qui ont été ajoutés
- « UpdatedPUIDs » : la liste des PUID qui ont été mis à jour, accompagnés d’un différentiel entre les métadonnées précédemment importées et les nouvelles
- « Warnings » : le message concernant le warning : le référentiel des formats importé est plus ancien, en termes de numéro de version et/ou de date de création, que la version du référentiel présente dans la solution, le référentiel des formats importé est identique, en termes de numéro de version et/ou de date de création, à la version précédemment installée dans la solution.

Exemple :

```
« Operation » ({}
« evType » : « STP_REFERENTIAL_FORMAT_IMPORT », « evDateTime » : « 2018-12-11T14:46:43.856 », « evId » : « aeeeeaaaaghg7kglaboimalhtw57z7qaaaaq »
}, « StatusCode » : « WARNING », « PreviousPronomVersion » : « 94 »,
« PreviousPronomCreationDate » : « 2018-09-17T12:54:53.000 », « NewPronomVersion » :
« 88 », « NewPronomCreationDate » : « 2016-09-27T15:37:53.000 », « RemovedPUIDs » :
[ « fmt/1216 », « fmt/1217 », « fmt/1214 », « fmt/1215 », « fmt/1212 », « fmt/1213 »,
« fmt/1210 », « fmt/1211 », « fmt/1108 », « fmt/1109 », « fmt/1106 », « fmt/1107 »,
« fmt/1104 », « fmt/1105 », « fmt/1102 », « fmt/1103 », « fmt/1100 », « fmt/1101 »,
« fmt/985 », « fmt/986 », « fmt/987 », « fmt/988 », « fmt/981 », « fmt/982 »,
« fmt/983 », « fmt/984 », « fmt/989 », « fmt/980 », « fmt/975 », « fmt/976 »,
« fmt/977 », « fmt/978 », « fmt/979 », « fmt/1209 », « fmt/1207 », « fmt/1208 »,
« fmt/1205 », « fmt/1206 », « fmt/1203 », « fmt/1204 », « fmt/1201 », « fmt/1202 »,
« fmt/1200 », « fmt/1140 », « fmt/1020 », « fmt/1141 », « fmt/1018 », « fmt/1139 »,
« fmt/1019 », « fmt/1016 », « fmt/1137 », « fmt/1017 », « fmt/1138 », « fmt/1014 »,
« fmt/1135 », « fmt/1015 », « fmt/1136 », « fmt/1012 », « fmt/1133 », « fmt/1013 »,
« fmt/1134 », « fmt/1010 », « fmt/1131 », « fmt/1011 », « fmt/1132 », « fmt/1030 »,
« fmt/1151 », « fmt/996 », « fmt/1031 », « fmt/1152 », « fmt/997 », « fmt/998 »,
« fmt/1150 », « fmt/999 », « fmt/992 », « fmt/993 », « fmt/994 », « fmt/995 »,
« fmt/1029 », « fmt/1027 », « fmt/1148 », « fmt/1028 », « fmt/1149 », « fmt/1025 »,
« fmt/1146 », « fmt/990 », « fmt/1026 », « fmt/1147 », « fmt/991 », « fmt/1023 », « fmt/
1144 », « fmt/1024 », « fmt/1145 », « fmt/1021 », « fmt/1142 », « fmt/1022 »,
« fmt/1143 », « fmt/1119 », « fmt/1117 », « fmt/1118 », « fmt/1115 », « fmt/1116 »,
« fmt/1113 », « fmt/1114 », « fmt/1111 », « fmt/1112 », « fmt/1110 », « fmt/1130 »,
« fmt/1009 », « fmt/1007 », « fmt/1128 », « fmt/1008 », « fmt/1129 », « fmt/1005 »,
« fmt/1126 », « fmt/1006 », « fmt/1127 », « fmt/1003 », « fmt/1124 », « fmt/1004 »,
« fmt/1125 », « fmt/1001 », « fmt/1122 », « fmt/1002 », « fmt/1123 », « fmt/1120 »,
« fmt/1000 », « fmt/1121 », « fmt/1063 », « fmt/1184 », « fmt/1064 », « fmt/1185 »,
« fmt/1061 », « fmt/206 », « fmt/1182 », « fmt/1062 », « fmt/1183 », « fmt/1180 »,
« fmt/1060 », « fmt/1181 », « fmt/1058 », « fmt/1179 », « fmt/1059 », « fmt/1056 »,
« fmt/1177 », « fmt/1057 », « fmt/1178 », « fmt/1054 », « fmt/1175 », « fmt/1055 »,
« fmt/1176 », « fmt/1074 », « fmt/1195 », « fmt/1075 », « fmt/1196 », « fmt/1072 »,
« fmt/1193 », « fmt/1073 », « fmt/1194 », « fmt/1070 », « fmt/1191 », « fmt/1071 »,
« fmt/1192 », « fmt/1190 », « fmt/1069 », « fmt/1067 », « fmt/1188 », « fmt/1068 »,
« fmt/1189 », « fmt/1065 », « fmt/1186 », « fmt/1066 », « fmt/1187 », « fmt/1041 »,
« fmt/1162 », « fmt/1042 », « fmt/1163 », « fmt/1160 », « fmt/1040 », « fmt/1161 »,
« fmt/1038 », « fmt/1159 », « fmt/1039 », « fmt/1036 », « fmt/1157 », « fmt/1037 »,
« fmt/1158 », « fmt/1034 », « fmt/1155 », « fmt/1035 », « fmt/1156 », « fmt/1032 »,
« fmt/1153 », « fmt/1033 », « fmt/1154 », « fmt/1052 », « fmt/1173 », « fmt/1053 »,
« fmt/1174 », « fmt/1050 », « fmt/1171 », « fmt/1051 », « fmt/1172 », « fmt/1170 »,
« fmt/1049 », « fmt/1047 », « fmt/1168 », « fmt/1048 », « fmt/1169 », « fmt/1045 »,
« fmt/1166 », « fmt/1046 », « fmt/1167 », « fmt/1043 », « fmt/1164 », « fmt/1044 »,
```

```
« fmt/1165 », « fmt/1098 », « fmt/1099 », « fmt/1085 », « fmt/1086 », « fmt/1083 »,
« fmt/1084 », « fmt/1081 », « fmt/1082 », « fmt/1080 », « fmt/1078 », « fmt/1199 »,
« fmt/1079 », « fmt/1076 », « fmt/1197 », « fmt/1077 », « fmt/1198 », « fmt/1096 »,
« fmt/1097 », « fmt/1094 », « fmt/1095 », « fmt/1092 », « fmt/1093 », « fmt/1090 »,
« fmt/1091 », « fmt/1089 », « fmt/1087 », « fmt/1088 » ], « AddedPUIDs » : [ ],
« UpdatedPUIDs » : {
  « fmt/563 » : [ « + MimeType : « , « - MimeType : application/postscript » ],
  « fmt/641 » : [ « + HasPriorityOverFileFormatID : [ fmt/154 ] », « -
HasPriorityOverFileFormatID : [ fmt/154, fmt/353 ] » ], « fmt/245 » : [ « + Extension :
[ \n ] », « - Extension : [ ] » ], « fmt/899 » : [ « + MimeType : application/octet-
stream », « - MimeType : application/vnd.microsoft.portable-executable » ],
« x-fmt/430 » : [ « + Extension : [ msg ] », « - Extension : [ msg, oft ] » ],
« fmt/417 » : [ « + MimeType : « , « - MimeType : application/postscript » ],
« fmt/616 » : [ « + MimeType : application/font-woff », « + Version : « , « - MimeType :
font/woff », « - Version : 1.0 » ], « fmt/418 » : [ « + MimeType : « , « - MimeType :
application/postscript » ], « fmt/419 » : [ « + MimeType : « , « - MimeType :
application/postscript » ], « fmt/570 » : [ « + HasPriorityOverFileFormatID : [ ] », « -
HasPriorityOverFileFormatID : [ fmt/986 ] » ]
}, « Warnings » : [ "Same referential version: 94", "Same referential date: 2018-09-
17T12:54:53.000", "1669 puids removed." ]
}
```

3.4.Workflow d'administration d'un référentiel des services agents

Cette section décrit le processus (workflow) permettant d'importer un référentiel de services agents.

Toutes les étapes, tâches et traitements sont journalisées dans le journal des opérations. Et décrivent le processus (clé et description de la clé associée dans le journal des opérations) tel qu'implémenté dans la version actuelle de la solution logicielle Vitam.

L'import d'un référentiel de services agents permet de vérifier le formalisme de ce dernier, notamment que les données obligatoires sont bien présentes pour chacun des agents. Tous les éléments réalisés au cours de ce processus sont exécutés dans une seule étape. Cet import concerne aussi bien l'import initial (pas de services agents pré-existant) que la mise à jour du référentiel.

Ce processus d'import débute lors du lancement du téléchargement dans la solution logicielle Vitam du fichier CSV contenant le référentiel.

La fin du processus peut prendre plusieurs statuts :

3.4.1.Processus d'import et mise à jour d'un référentiel de services agents (STP_IMPORT_AGENCIES)

- **Règle** : étape consistant à vérifier que le fichier importé remplit les conditions suivantes :
 - il est au format CSV
 - les informations suivantes sont toutes décrites dans l'ordre exact pour chacun des services agents :
 - Identifier
 - Name
 - Description (optionnel)
 - L'identifiant doit être unique
- **Type** : bloquant
- **Statuts** :
 - OK : le fichier respecte les règles (STP_IMPORT_AGENCIES.OK = Succès du processus d'import du référentiel des services agents)
 - KO :
 - Cas 1 : une information concernant les services agents est manquante (Identifier, Name, Description) (STP_IMPORT_AGENCIES.KO = Échec du processus d'import du référentiel des services agents). .
 - Cas 2 : un service agent qui était présent dans la base a été supprimé (STP_IMPORT_AGENCIES.DELETION.KO = Échec du processus d'import du référentiel des services agents : Des services agents absents du fichier d'import sont utilisés par au moins une unité archivistique présente dans le système)
 - Cas 3 : une autre opération d'import de référentiel des services agents est en cours (STP_IMPORT_AGENCIES.CONCURRENCE.KO = Échec du processus d'import du référentiel des services agents : une autre opération d'import des services agents est en cours)
 - FATAL : une erreur technique est survenue lors de l'import du référentiel des services agents (STP_IMPORT_AGENCIES.FATAL = Erreur technique lors du processus d'import du référentiel des services agents)

3.4.1.1. Vérification des contrats utilisés IMPORT_AGENCIES.USED_CONTRACT

- **Règle** : tâche consistant à contrôler les contrats d'accès utilisant des services agents modifiés
- **Type** : bloquant
- **Statuts** :

- OK : aucun des services agents utilisés par des contrats d'accès n'a été modifié (STP_IMPORT_AGENCIES.USED_CONTRACT.OK = Succès du processus de vérification des services agents utilisés dans les contrats d'accès)
- WARNING : un ou plusieurs services agents utilisés par des contrats d'accès ont été modifiés (STP_IMPORT_AGENCIES.USED_CONTRACT.WARNING = Avertissement lors du processus de vérification des services agents utilisés dans les contrats d'accès)
- KO : pas de cas KO
- FATAL : une erreur technique est survenue lors de la vérification des services agents utilisés dans les contrats d'accès (STP_IMPORT_AGENCIES.USED_CONTRACT.FATAL = Erreur technique lors du processus de vérification des services agents utilisés dans les contrats d'accès)

3.4.1.2. Vérification des unités archivistiques IMPORT_AGENCIES.USED_AU

- **Règle** : tâche consistant à contrôler les unités archivistiques référençant des services agents modifiés
- **Type** : bloquant
- **Statuts** :
 - OK : aucun service agent référencé par les unités archivistiques n'a été modifié (STP_IMPORT_AGENCIES.USED_AU.OK = Succès du processus de vérification des services agents référencés par les unités archivistiques)
 - WARNING : au moins un service agent référencé par une unité archivistique a été modifié (STP_IMPORT_AGENCIES.USED_AU.WARNING = Avertissement lors du processus de vérification des services agents référencés par les unités archivistiques)
 - KO : pas de cas KO
 - FATAL : une erreur technique est survenue lors de la vérification des services agents utilisés par les unités archivistiques (STP_IMPORT_AGENCIES.USED_AU.FATAL = Erreur technique lors du processus de vérification des services agents référencés par les unités archivistiques)

3.4.1.3. Création du rapport au format JSON AGENCIES_REPORT (AgenciesService.java)

- **Règle** : tâche consistant à créer le rapport d'import de référentiel des services agents
- **Type** : bloquant
- **Statuts** :
 - OK : le rapport d'import du référentiel des services agents a bien été créé (STP_AGENCIES_REPORT.OK = Succès du processus de génération du rapport d'import du référentiel des services agents)
 - KO : pas de cas KO
 - FATAL : une erreur technique est survenue lors de la création du rapport d'import du référentiel des services agents (STP_AGENCIES_REPORT.FATAL = Erreur technique lors du processus de génération du rapport d'import du référentiel des services agents)

3.4.1.4. Sauvegarde du CSV d'import IMPORT_AGENCIES_BACKUP_CSV (AgenciesService.java)

- **Règle** : tâche consistant à sauvegarder le fichier d'import de référentiel des services agents sur les offres de stockage
- **Type** : bloquant
- **Statuts** :
 - OK : le fichier d'import du référentiel des services agent a bien été sauvegardé (STP_IMPORT_AGENCIES_BACKUP_CSV.OK = Succès du processus de sauvegarde du fichier d'import du référentiel des services agents)
 - KO : pas de cas KO

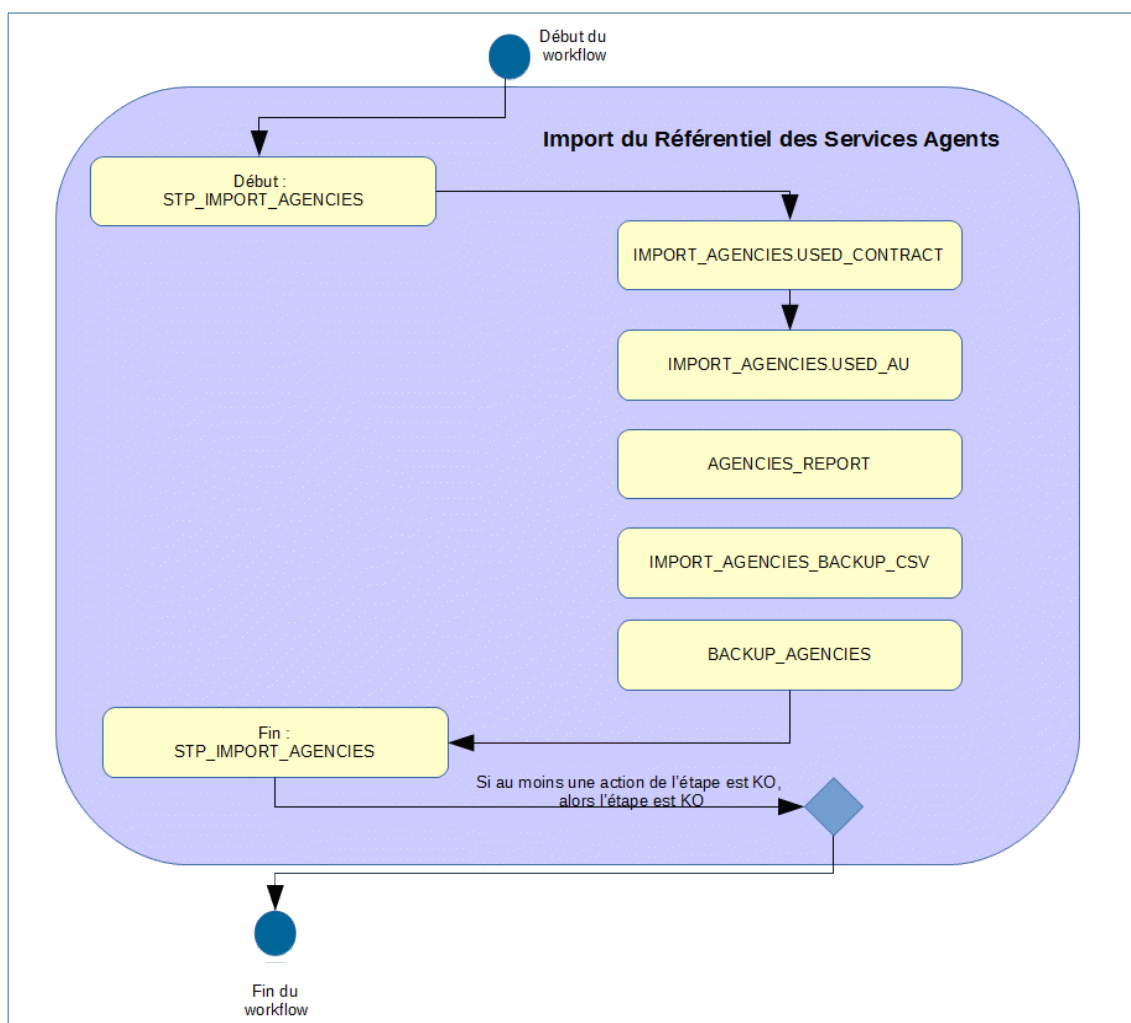
- FATAL : une erreur technique est survenue lors de la sauvegarde de fichier d'import du référentiel des services agents (STP_AGENCIES_REPORT.FATAL = Erreur technique lors du processus de sauvegarde du fichier d'import du référentiel des services agents)

3.4.1.5. Sauvegarde d'une copie de la base de données BACKUP_AGENCIES (AgenciesService.java)

- **Règle** : tâche consistant à créer une copie de la base de données contenant le référentiel des services agents sur les offres de stockage
- **Type** : bloquant
- **Statuts** :
 - OK : la copie de la base de donnée contenant le référentiel des services agents a été créé avec succès (STP_BACKUP_AGENCIES.OK = Succès du processus de sauvegarde du référentiel des services agents)
 - KO : pas de cas KO
 - FATAL : une erreur technique est survenue lors de la création d'une copie de la base de données contenant le référentiel des services agent (STP_BACKUP_AGENCIES.FATAL = Erreur technique lors du processus de sauvegarde du référentiel des services agents)

3.4.2. Structure du Workflow d'import d'un référentiel des services agents

D'une façon synthétique, le workflow est décrit de cette façon :



3.4.3. Structure du rapport d'import du référentiel des services agents

Lorsqu'un nouveau référentiel est importé, la solution logicielle Vitam génère un rapport de l'opération. Ce rapport est en plusieurs parties :

- « Operation » contient :
 - evType : le type d'opération. Dans le cadre de ce rapport, il s'agit toujours de « STP_IMPORT_AGENCIES »
 - evDateTime : la date et l'heure de l'opération d'import
 - evId : l'identifiant de l'opération
- « AgenciesToImport » : contient la liste des identifiants contenue dans le fichier
- « InsertAgencies » : contient les identifiants des services agents ajoutés
- « UpdatedAgencies » : liste les identifiants des services agents modifiés
- « UsedAgencies By Contrat » : liste les identifiants des services agents modifiés qui sont ou pas utilisés par des contrats d'accès
- « UsedAgencies By AU » : liste les identifiants des services agents modifiés qui sont ou pas ont utilisés dans des unités archivistiques
- « UsedAgencies to Delete » : liste les identifiants des services agents supprimés qui sont pas non utilisés dans des unités archivistiques
- **Exemple 1 : modification et suppression d'un service agent**

Le rapport généré est :

```
{
  "AgenciesToImport": ["test", "Identifieur0", "Identifieur1", "Identifieur2", "Identifieur3",
"Identifieur4", "Identifieur5", "Identifieur6", "Identifieur7", "Identifieur8", "Identifieur9",
"Identifieur10", "Vitam", "FRAN_NP_009913", "FRAN_NP_009941", "producteur1",
"FRAN_NP_050500", "FRAN_NP_050770", "BBBBBB", "FRAN_NP_050758", "FRAN_NP_050056",
"FRAN_NP_005568", "ABCDEFG", "OAIJAZ", "ZAJFKNDSC", "Service_producteur",
"SOC_ARCHEO_TOURRAINE", "EEEEEE", "FRAN_NP_050392", "FRAN_NP_050313", "FRAN_NP_051314",
"FR_ORG_AGEN", "FRAAA", "CCCC", "ASP", "producteur2", "FRAN_NP_000001", "FRAN_NP_005061",
"FRAN_NP_009915", "KARINE_GILBERT", "MICHEL_MERCIER", "DGFIP", "FRAN_NP_005164",
"SOC_ARCHEO_TOURRAINE", "Service_versant", "XXXXXXX", "YYYYYYY", "FRAN_NP_009196",
"FRAN_NP_005761", "FFFFFF", "FRAN_NP_005479", "FRAN_NP_000992", "FRAN_NP_000002",
"FRAN_NP_000013", "FRAN_NP_000015", "FRAN_NP_000008", "FRAN_NP_000017", "FRAN_NP_000024",
"FRAN_NP_000018", "FRAN_NP_000023", "FRAN_NP_000010", "FRAN_NP_000019", "FRAN_NP_000020",
"FRAN_NP_000021", "FRAN_NP_000022", "FRAN_NP_000009", "RATP"],

  "UsedAgencies to Delete": ["FRAN_NP_009734", "Test_VITAM"],

  "Operation": {
    "evId": "aeaaaaabchia7juabddealol5nlueaaaaaq",
    "evDateTime": "2019-11-12T11:23:36.061",
    "evType": "IMPORT_AGENCIES"
  }
}
```

Exemple 2 : ajout en erreur d'un service agent, causé par un champ obligatoire qui est manquant

Le rapport généré est :

```
{
  "Operation": {
    "evId": "aeaaaaaacflvhgbabrs6alb6vdoehyaaaaq",
    "evType": "STP_IMPORT_AGENCIES",
    "evDateTime": "2017-11-02T15:36:03.976"
  }
}
```

```

},
"AgenciesToImport": ["AG-TNR0002"],
"UsedAgencies to Delete":["AG-TNR0002"]
}

```

3.5.Workflow d'administration d'un référentiel des contrats d'entrée

Cette section décrit le processus (workflow) permettant d'importer et de mettre à jour un contrat d'entrée.

Toutes les étapes, tâches et traitements sont journalisées dans le journal des opérations et décrivent le processus (clé et description de la clé associée dans le journal des opérations) tel qu'implémenté dans la version actuelle de la solution logicielle Vitam.

3.5.1.Processus d'import d'un contrat d'entrée (STP_IMPORT_INGEST_CONTRACT)

Le processus d'import d'un contrat d'entrée permet à la fois de vérifier qu'il contient les informations minimales obligatoires, que l'ensemble des informations sont cohérentes, et de lui affecter des éléments peuplés automatiquement.

Tous les éléments réalisés au cours de ce processus sont exécutés dans une seule étape.

- **Règle** : étape consistant à vérifier la présence des informations minimales dans le contrat d'entrée, à s'assurer de la cohérence des informations saisies, à affecter des données aux champs peuplés par la solution logicielle Vitam et à enregistrer le contrat :
 - Le champ « Name » est peuplé d'une chaîne de caractères
 - Si le tenant concerné est en mode « esclave », le champ « Identifiant » doit être rempli. Sinon, il est automatiquement complété par la solution logicielle Vitam
 - Les données suivantes optionnelles si elles sont remplies le sont en respectant les règles énoncées pour chacune :
 - Le champ « Description » doit être une chaîne de caractères
 - Le champ « Status » doit avoir la valeur ACTIVE ou INACTIVE
 - Le champ « ArchiveProfile » doit être un tableau d'une ou plusieurs chaînes de caractère. Chacune de ces chaînes de caractère doit correspondre au champ « Identifiant » d'un profil d'archivage contenu dans le référentiel des profils
 - Le champ « CheckParentLink » : doit avoir la valeur ACTIVE ou INACTIVE
 - Le champ « LinkedParentId » doit être une chaîne de caractères devant correspondre au GUID d'une AU de plan de classement ou d'arbre de positionnement déjà pris en charge par la solution logicielle Vitam sur le même tenant
 - Le champ « MasterMandatory » doit avoir la valeur « true » ou « false »
 - Le champ « EveryDataObjectVersion » doit avoir la valeur « true » ou « false »
 - Le champ « DataObjectVersion » devrait être un tableau dont chaque élément est dans l'énumération suivante (ou être vide) : « BinaryMaster », « TextContent », « Thumbnail », « PhysicalMaster », « Dissemination »
 - Le champ « FormatUnidentifiedAuthorized » doit avoir la valeur « true » ou « false »
 - Le champ « EveryFormatType » doit avoir la valeur « true » ou « false »
 - Le champ « FormatType » doit être un tableau dont chaque élément est une PUID du référentiel des formats (exemple : « fmt/17 »)
 - Le champ « ManagementContractId » est soit vide soit un « Identifiant » de contrat de gestion existant dans le référentiel de contrats de gestion
- **Type** : bloquant
- **Statuts** :
 - OK : le contrat répond aux exigences des règles (STP_IMPORT_INGEST_CONTRACT.OK = Succès du processus d'import du contrat d'entrée)
 - KO : une des règles ci-dessus n'a pas été respectée (STP_IMPORT_INGEST_CONTRACT.KO =

Échec du processus d'import du contrat d'entrée)

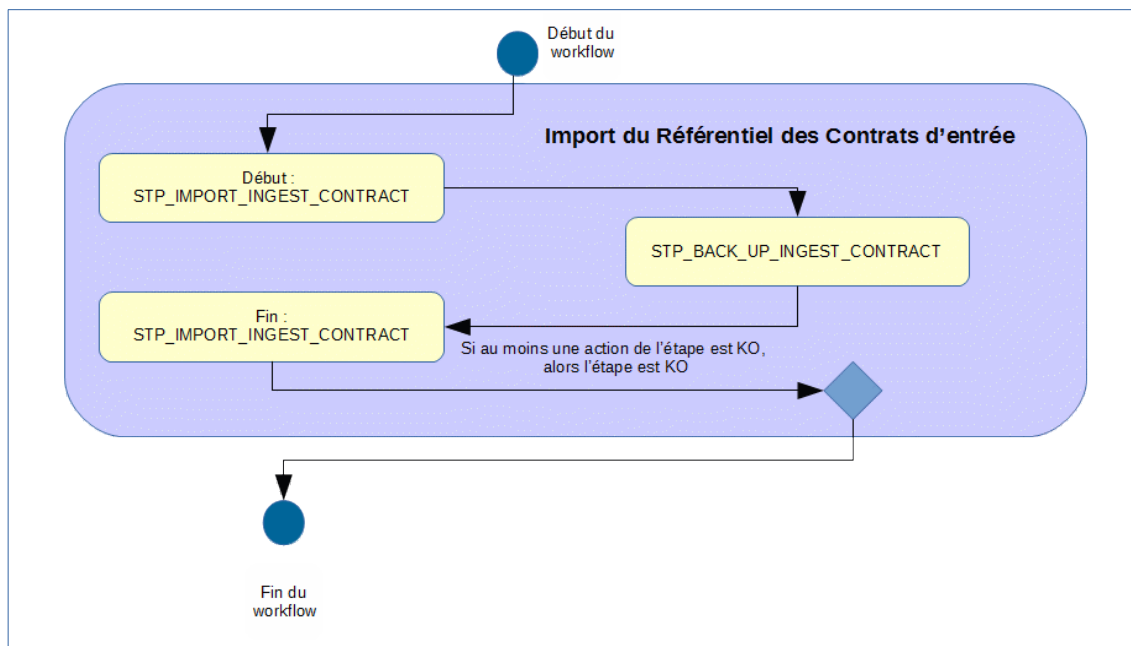
- Cas n°1 : l'identifiant utilisé existe déjà
(STP_IMPORT_INGEST_CONTRACT.IDENTIFIER_DUPLICATION.KO = Échec de l'import : l'identifiant est déjà utilisé)
- Cas n°2 : un champ obligatoire n'est pas renseigné
(STP_IMPORT_INGEST_CONTRACT.EMPTY_REQUIRED_FIELD.KO = Échec de l'import : au moins un des champs obligatoires n'est pas renseigné)
- Cas n°3 : le profil d'archivage mentionné n'existe pas
(STP_IMPORT_INGEST_CONTRACT.PROFILE_NOT_FOUND.KO = Échec de l'import : profil d'archivage non trouvé)
- Cas n°4 : le contrat d'entrée autorise tous les formats mais déclare des formats dans la liste des formats autorisés (STP_IMPORT_INGEST_CONTRACT.FORMAT_MUST_BE_EMPTY.KO= Échec de l'import : la liste blanche des formats doit être vide lorsque tous les formats sont autorisés) ;
- Cas n°5 : un ou plusieurs formats déclarés ne sont pas référencés dans le référentiel des formats (STP_IMPORT_INGEST_CONTRACT.FORMAT_NOT_FOUND.KO = Échec de l'import : un ou plusieurs formats ne sont pas référencés dans le référentiel des formats) ;
- Cas n°6 : le contrat d'entrée n'autorise pas tous les formats mais ne déclare pas de formats dans la liste des formats autorisés
(STP_IMPORT_INGEST_CONTRACT.FORMAT_MUST_NOT_BE_EMPTY.KO = Échec de l'import : la liste blanche des formats ne peut pas être vide lorsque tous les formats ne sont pas autorisés)
- Cas n°7 : le contrat de gestion déclaré n'est pas référencé dans le référentiel des contrats de gestion
(STP_IMPORT_INGEST_CONTRACT.MANAGEMENT_CONTRAT_NOT_FOUND.KO = Échec de l'import : le contrat de gestion n'est pas référencé dans le référentiel des contrats de gestion) ;
- FATAL : une erreur technique est survenue lors de l'import du contrat
(STP_IMPORT_INGEST_CONTRACT.FATAL = Erreur technique du processus d'import du contrat d'entrée)
- WARNING : avertissement lors du processus d'import du contrat d'entrée
(STP_IMPORT_INGEST_CONTRACT.WARNING = Avertissement lors du processus d'import du contrat d'entrée)

Sauvegarde du JSON (STP_BACKUP_INGEST_CONTRACT)

- **Règle** : étape consistant à enregistrer une copie de la base de données des contrats d'entrée sur les offres de stockage
- **Type** : bloquant
- **Statuts** :
 - OK : une copie de la base de données nouvellement importée est enregistrée
(STP_BACKUP_INGEST_CONTRACT.OK = Succès du processus de sauvegarde des contrats d'entrée)
 - KO : pas de cas KO
 - FATAL : une erreur technique est survenue lors de la copie de la base de données nouvellement importée (STP_BACKUP_INGEST_CONTRACT.FATAL = Erreur technique lors du processus de sauvegarde des contrats d'entrée)

3.5.2. Structure du Workflow d'import d'un référentiel des contrats d'entrée

D'une façon synthétique, le workflow est décrit de cette façon :



3.5.3. Processus de mise à jour d'un contrat d'entrée (STP_UPDATE_INGEST_CONTRACT)

Le processus de mise d'un contrat d'entrée permet à la fois de vérifier qu'il contient les informations minimales obligatoires, que de vérifier la cohérence de l'ensemble des informations sont cohérentes, et de lui affecter des éléments peuplés automatiquement.

Tous les éléments réalisés au cours de ce processus sont exécutés dans une seule étape.

- **Règle** : opération consistant à modifier un contrat d'entrée, selon un processus identique à celui de l'import initial
- **Type** : bloquant
- **Statuts** :
 - OK : la mise à jour du contrat d'entrée a bien été effectuée (STP_UPDATE_INGEST_CONTRACT.OK = Succès du processus de mise à jour du contrat d'entrée)
 - KO : la mise à jour du contrat d'entrée n'a pas été effectuée (STP_UPDATE_INGEST_CONTRACT.KO = Échec du processus de mise à jour du contrat d'entrée)
 - Cas n°1 : l'identifiant du contrat d'entrée est inconnu (STP_UPDATE_INGEST_CONTRACT.CONTRACT_NOT_FOUND.KO = Échec du processus de mise à jour du contrat d'entrée : contrat d'entrée non trouvé)
 - Cas n°2 : l'identifiant du contrat d'entrée mis à jour est déjà utilisé (STP_UPDATE_INGEST_CONTRACT.IDENTIFIER_DUPLICATION.KO = Échec du processus de mise à jour du contrat d'entrée : l'identifiant est déjà utilisé)
 - Cas n°3 : une des valeurs saisies dans le contrat d'entrée ne correspond pas aux valeurs attendues (STP_UPDATE_INGEST_CONTRACT.NOT_IN_ENUM.KO = Échec du processus de mise à jour du contrat d'entrée : une valeur ne correspond pas aux valeurs attendues)
 - Cas n°4 : le profil d'archivage mentionné n'existe pas (STP_UPDATE_INGEST_CONTRACT.PROFILE_NOT_FOUND.KO = Échec du processus de mise à jour du contrat d'entrée : au moins un profil d'archivage est inconnu)

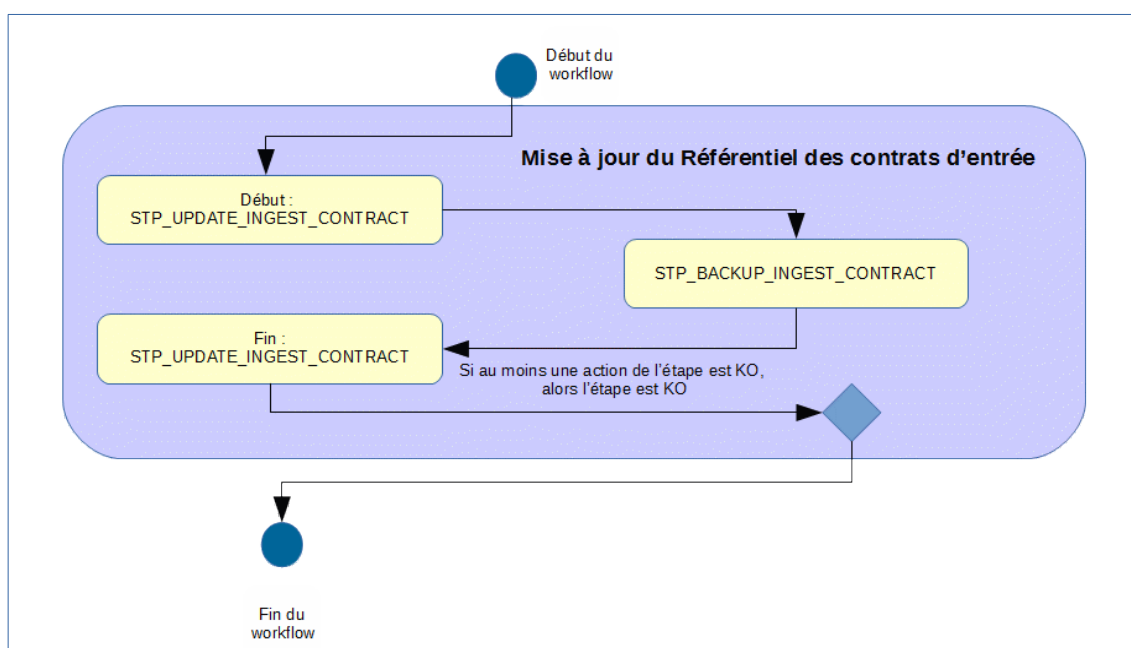
- Cas n°5 : la requête de mise à jour du contrat d'entrée est mal formatée (STP_UPDATE_INGEST_CONTRACT.BAD_REQUEST.KO = Échec du processus de mise à jour du contrat d'entrée : mauvaise requête)
- Cas n°6 : un ou plusieurs formats déclarés ne sont pas référencés dans le référentiel des formats (STP_UPDATE_INGEST_CONTRACT.FILEFORMAT_NOT_FOUND.KO = Échec du processus de mise à jour du contrat d'entrée : au moins un identifiant de format est inconnu)
- Cas n°7 : le contrat de gestion déclaré n'est pas référencé dans le référentiel des contrats de gestion (STP_UPDATE_INGEST_CONTRACT.MANAGEMENT_CONTRAT_NOT_FOUND.KO = Échec du processus de mise à jour : le contrat de gestion est inconnu)
- FATAL : une erreur technique est survenue lors du processus de mise à jour du contrat d'entrée (STP_UPDATE_INGEST_CONTRACT.FATAL = Erreur technique lors du processus de mise à jour du contrat d'entrée)

3.5.3.1. Sauvegarde du JSON (STP_BACKUP_INGEST_CONTRACT)

- **Règle** : tâche consistant à enregistrer une copie de la base de données des contrats d'entrée sur les offres de stockage
- **Type** : bloquant
- **Statuts** :
 - OK : une copie de la base de données nouvellement importée est enregistrée (STP_BACKUP_INGEST_CONTRACT.OK = Succès du processus de sauvegarde des contrats d'entrée)
 - KO : pas de cas KO
 - FATAL : une erreur technique est survenue lors de la copie de la base de données nouvellement importée (STP_BACKUP_INGEST_CONTRACT.FATAL = Erreur technique lors du processus de sauvegarde des contrats d'entrée)

3.5.4. Structure du Workflow de mise à jour d'un référentiel des contrats d'entrée

D'une façon synthétique, le workflow est décrit de cette façon :



3.6.Workflow d'administration d'un référentiel des contrats d'accès

Cette section décrit le processus (workflow) permettant d'importer et de mettre à jour un contrat d'accès.

Toutes les étapes, tâches et traitements sont journalisés dans le journal des opérations et décrivent le processus (clé et description de la clé associée dans le journal des opérations) tel qu'implémenté dans la version actuelle de la solution logicielle Vitam.

3.6.1.Processus d'import et mise à jour d'un contrat d'accès (STP_IMPORT_ACCESS_CONTRACT)

Le processus d'import d'un contrat d'accès permet à la fois de vérifier qu'il contient les informations minimales obligatoires, de vérifier la cohérence de l'ensemble des informations et de lui affecter des éléments peuplés automatiquement.

Tous les éléments réalisés au cours de ce processus sont exécutés dans une seule étape.

- **Règle** : opération consistant à vérifier la présence des informations minimales dans le contrat d'accès, à s'assurer de la cohérence des informations saisies, à affecter des données aux champs peuplés par la solution logicielle Vitam et à enregistrer le contrat
 - Les données suivantes sont obligatoirement remplies :
 - Le champ « Name » est peuplé d'une chaîne de caractères
 - Le champ « Identifier » est peuplé d'une chaîne de caractères si le référentiel des contrats d'accès est configuré en mode esclave sur le tenant sélectionné
 - Le champ « _tenant » : identifiant du tenant.
 - Le champ « Status » est peuplé avec la valeur « ACTIVE » ou la valeur « INACTIVE »
 - Le champ « AccessLog » peut être renseigné avec la valeur « ACTIVE » ou la valeur « INACTIVE »
 - Les données suivantes optionnelles, si elles sont remplies, le sont en respectant les règles énoncées pour chacune :
 - Le champ « Description » est peuplé avec une chaîne de caractères
 - Le champ « DataObjectVersion » est soit vide, soit peuplé avec un tableau d'une ou plusieurs chaînes de caractères. Chacune de ces chaînes de caractères doit correspondre à un des usages définis dans les groupe d'objets techniques pris en charge dans la solution logicielle Vitam, « BinaryMaster », « TextContent », « Thumbnail », « PhysicalMaster », « Dissemination »
 - Le champ « OriginatingAgencies » est soit vide soit peuplé avec un tableau d'une ou plusieurs chaînes de caractères. Chacune de ces chaînes de caractères doit correspondre au champ « Identifier » d'un service agent contenu dans le référentiel des services agents.
 - Le champ « WritingPermission » doit être à « true » ou « false »
 - Le champ « EveryOriginatingAgency » doit être à « true » ou « false »
 - Le champ « EveryDataObjectVersion » doit être à « true » ou « false »
 - Le champ « WritingRestrictedDesc » : droit de modification des métadonnées descriptives seulement
 - Le champ « RootUnit » est soit vide, soit peuplé avec un tableau d'une ou plusieurs chaînes de caractère. Chacune des chaînes de caractère doit correspondre au GUID d'une unité archivistique prise en charge dans la solution logicielle Vitam
- **Type** : bloquant
- **Statuts** :
 - OK : le contrat répond aux exigences des règles (STP_IMPORT_ACCESS_CONTRACT.OK = Succès du processus d'import du contrat d'accès)
 - KO : une des règles ci-dessus n'a pas été respectée (STP_IMPORT_ACCESS_CONTRACT.KO = Échec du processus d'import du contrat d'accès)

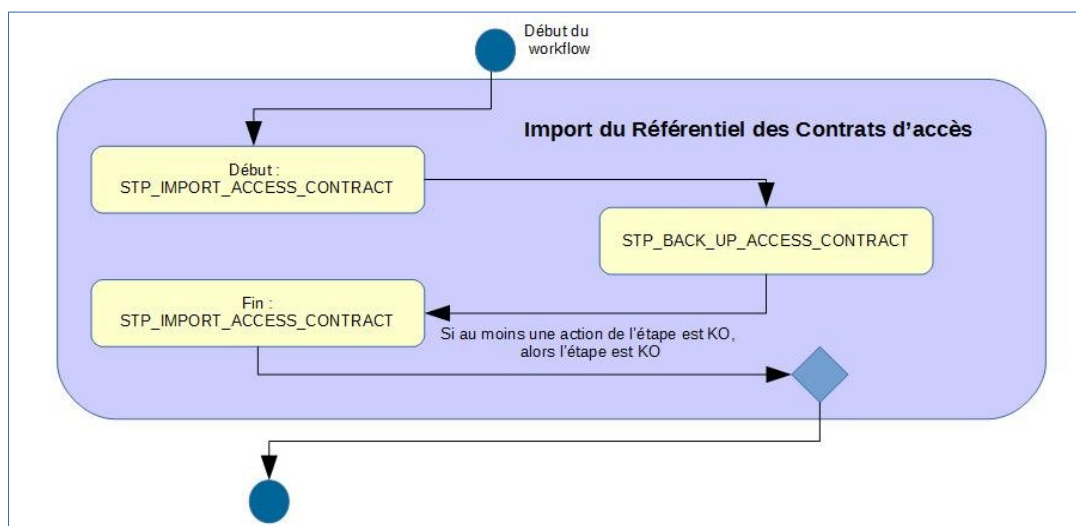
- Cas n°1 : l'identifiant du contrat est déjà utilisé
(STP_IMPORT_ACCESS_CONTRACT.IDENTIFIER_DUPLICATION.KO = Échec de l'import du contrat d'accès : l'identifiant est déjà utilisé)
- Cas n°2 : au moins un des champs obligatoires n'est pas renseigné
(STP_IMPORT_ACCESS_CONTRACT.EMPTY_REQUIRED_FIELD.KO = Échec de l'import du contrat d'accès : au moins un des champs obligatoires n'est pas renseigné)
- Cas n°3 : service producteur inconnu
(STP_IMPORT_ACCESS_CONTRACT.AGENCY_NOT_FOUND.KO = Échec de l'import du contrat d'accès : au moins un service producteur est inconnu)
- Cas n°4 : erreur de validation du contrat
(STP_IMPORT_ACCESS_CONTRACT.VALIDATION_ERROR.KO = Échec de l'import du contrat d'accès : erreur de validation du contrat du contrat d'accès)
- FATAL : une erreur technique est survenue lors de la vérification de l'import du contrat d'accès
(STP_IMPORT_ACCESS_CONTRACT.FATAL = Erreur technique lors du processus d'import du contrat d'accès)
- WARNING : avertissement lors du processus d'import du contrat d'accès
(STP_IMPORT_ACCESS_CONTRACT.WARNING = Avertissement lors du processus d'import du contrat d'accès)

3.6.1.1. Sauvegarde du JSON STP_BACKUP_ACCESS_CONTRACT (IngestContractImpl.java)

- **Règle** : tâche consistant à enregistrer une copie de la base de données des contrats d'accès sur les offres de stockage
- **Type** : bloquant
- **Statuts** :
 - OK : une copie de la base de données nouvellement importée est enregistrée
(STP_BACKUP_ACCESS_CONTRACT.OK = Succès du processus de sauvegarde des contrats d'accès)
 - KO : pas de cas KO
 - FATAL : une erreur technique est survenue lors de la copie de la base de données nouvellement importée (STP_BACKUP_ACCESS_CONTRACT.FATAL = Erreur technique lors du processus de sauvegarde des contrats d'accès)

3.6.2. Structure du Workflow d'import du référentiel des contrats d'accès

D'une façon synthétique, le workflow est décrit de cette façon :



3.6.3. Processus de mise à jour d'un contrat d'accès STP_UPDATE_ACCESS_CONTRACT

- **Règle** : opération consistant à modifier un contrat d'accès, selon un processus identique à celui de l'import initial
- **Type** : bloquant
- **Statuts** :
 - OK : le contrat répond aux exigences des règles (STP_UPDATE_ACCESS_CONTRACT.OK = Succès du processus de mise à jour du contrat d'accès)
 - KO : une des règles ci-dessus n'a pas été respectée (STP_UPDATE_ACCESS_CONTRACT.KO = Échec du processus de mise à jour du contrat d'accès)
 - Cas n° 1 : l'identifiant du contrat d'entrée est inconnu (STP_UPDATE_ACCESS_CONTRACT.CONTRACT_NOT_FOUND.KO = Échec du processus de mise à jour du contrat d'accès : contrat d'accès non trouvé)
 - Cas n°2 : une des valeurs saisies dans le contrat d'accès ne correspond pas aux valeurs attendues (STP_UPDATE_ACCESS_CONTRACT.NOT_IN_ENUM.KO = Échec du processus de mise à jour du contrat d'accès : une valeur ne correspond pas aux valeurs attendues)
 - Cas n°3 : service producteur inconnu (STP_UPDATE_ACCESS_CONTRACT.AGENCY_NOT_FOUND.KO = Échec du processus de mise à jour du contrat d'accès : au moins un service agent est inconnu)
 - Cas n°4 : la requête de mise à jour du contrat d'accès est mal formatée (STP_UPDATE_ACCESS_CONTRACT.BAD_REQUEST.KO = Échec du processus de mise à jour du contrat d'accès : mauvaise requête)
 - FATAL : une erreur technique est survenue lors de la vérification de l'import du contrat d'accès (STP_UPDATE_ACCESS_CONTRACT.FATAL = Erreur technique lors du processus de mise à jour du contrat d'accès)

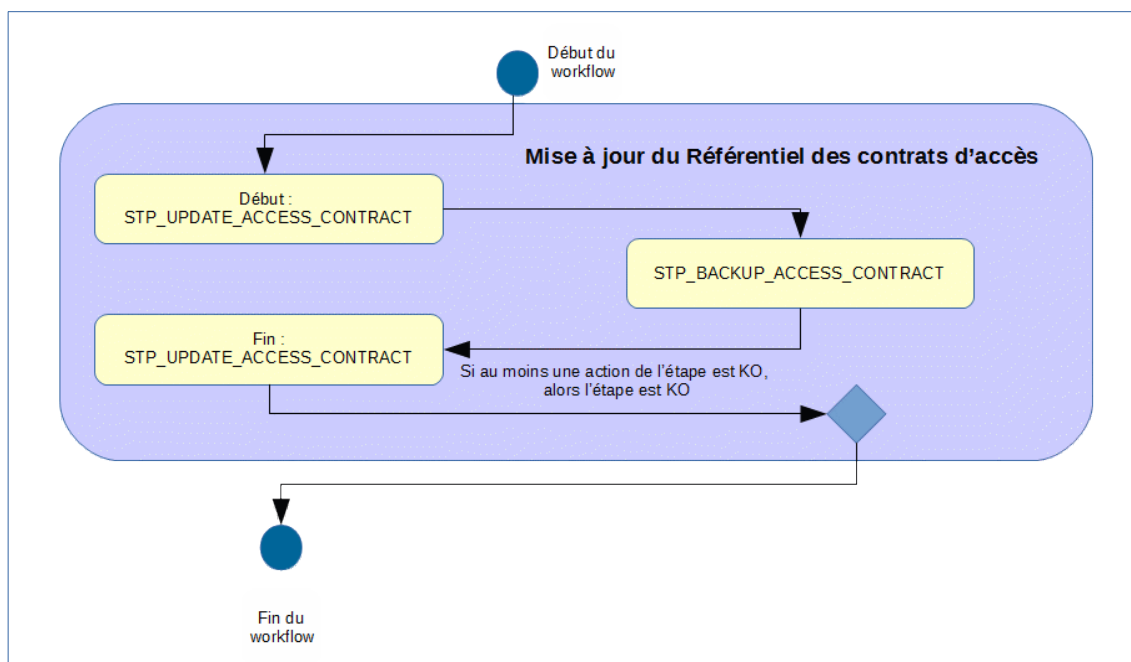
3.6.3.1. Sauvegarde du JSON STP_BACKUP_ACCESS_CONTRACT (IngestContractImpl.java)

- **Règle** : tâche consistant à enregistrer une copie de la base de données des contrats d'accès sur les offres de stockage
- **Type** : bloquant
- **Statuts** :
 - OK : une copie de la base de données nouvellement importée est enregistrée (STP_BACKUP_ACCESS_CONTRACT.OK = Succès du processus de sauvegarde des contrats)

- d'accès)
- KO : pas de cas KO
 - FATAL : une erreur technique est survenue lors de la copie de la base de données nouvellement importée (STP_BACKUP_ACCESS_CONTRACT.FATAL = Erreur technique lors du processus de sauvegarde des contrats d'accès)

3.6.4. Structure du Workflow de mise à jour d'un référentiel des contrats d'accès

D'une façon synthétique, le workflow est décrit de cette façon :



3.7.Workflow d'administration d'un référentiel des contrats de gestion

Cette section décrit le processus (workflow) permettant d'importer et de mettre à jour un contrat de gestion.

Toutes les étapes, tâches et traitements sont journalisés dans le journal des opérations et décrivent le processus (clé et description de la clé associée dans le journal des opérations) tel qu'implémenté dans la version actuelle de la solution logicielle Vitam.

3.7.1.Processus d'import et mise à jour d'un contrat de gestion (STP_IMPORT_MANAGEMENT_CONTRACT)

Le processus d'import d'un contrat de gestion permet à la fois de vérifier qu'il contient les informations minimales obligatoires, de vérifier la cohérence de l'ensemble des informations et de lui affecter des éléments peuplés automatiquement.

Tous les éléments réalisés au cours de ce processus sont exécutés dans une seule étape.

- **Règle** : opération consistant à vérifier la présence des informations minimales dans le contrat de gestion, à s'assurer de la cohérence des informations saisies, à affecter des données aux champs peuplés par la solution logicielle Vitam et à enregistrer le contrat
 - Les données suivantes sont obligatoirement remplies :
 - Le champ « Name » est peuplé d'une chaîne de caractères
 - Le champ « Identifier » est peuplé d'une chaîne de caractères si le référentiel des contrats de gestion est configuré en mode esclave sur le tenant sélectionné
 - Le champ « _tenant » : identifiant du tenant.
 - Le champ « Status » est peuplé avec la valeur « ACTIVE » ou la valeur « INACTIVE »
 - Les données suivantes optionnelles, si elles sont remplies, le sont en respectant les règles énoncées pour chacune :
 - Le champ « Description » est peuplé avec une chaîne de caractères
 - Le sous-objet « Storage » est soit vide, soit peuplé avec les champs suivants
 - Le champ « UnitStrategy » est soit vide soit peuplé avec un identifiant de stratégie de stockage déployé sur la plateforme VITAM et contenant l'offre de référence définie dans la stratégie de plateforme VITAM (« default »)
 - Le champ « ObjectGroupStrategy » est soit vide soit peuplé avec un identifiant de stratégie de stockage déployé sur la plateforme VITAM et contenant l'offre de référence définie dans la stratégie de plateforme VITAM (« default »)
 - Le champ « ObjectStrategy » est soit vide soit peuplé avec un identifiant de stratégie de stockage déployé sur la plateforme VITAM
- **Type** : bloquant
- **Statuts** :
 - OK : le contrat répond aux exigences des règles (STP_IMPORT_MANAGEMENT_CONTRACT.OK = Succès du processus d'import du contrat de gestion)
 - KO : une des règles ci-dessus n'a pas été respectée (STP_IMPORT_MANAGEMENT_CONTRACT.KO = Échec du processus d'import du contrat de gestion)
 - Cas n°1 : l'identifiant du contrat est déjà utilisé (STP_IMPORT_MANAGEMENT_CONTRACT.IDENTIFIER_DUPLICATION.KO = Échec de l'import du contrat de gestion : l'identifiant est déjà utilisé)
 - Cas n°2 : au moins un des champs obligatoires n'est pas renseigné (STP_IMPORT_MANAGEMENT_CONTRACT.EMPTY_REQUIRED_FIELD.KO = Échec de l'import du contrat de gestion : au moins un des champs obligatoires n'est pas renseigné)
 - Cas n°3 : une des stratégies est invalide

(STP_IMPORT_MANAGEMENT_CONTRACT.STRATEGY_VALIDATION_ERROR.KO = Échec de l'import du contrat de gestion : au moins une stratégie est invalide)

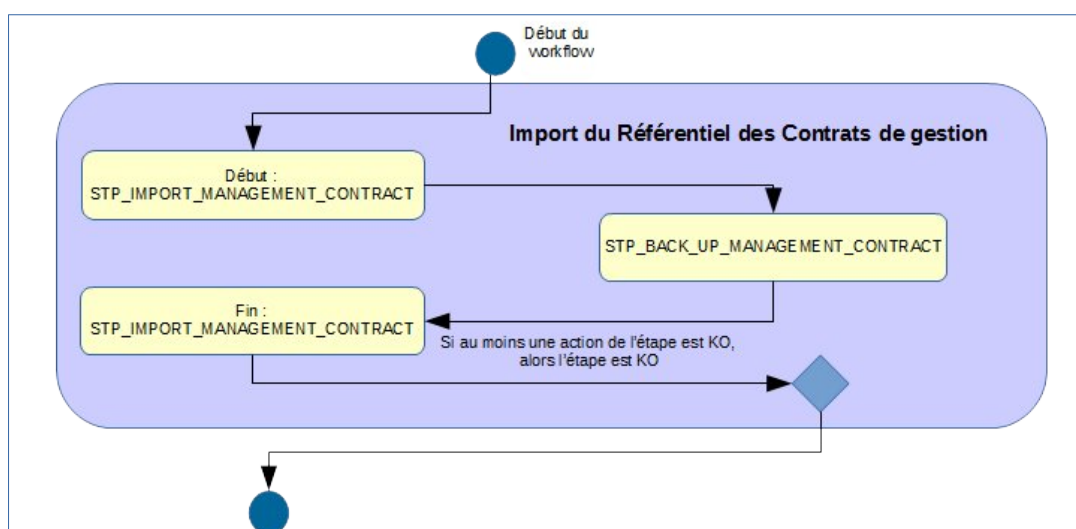
- Cas n°4 : erreur de validation du contrat
(STP_IMPORT_MANAGEMENT_CONTRACT.VALIDATION_ERROR.KO = Échec de l'import du contrat de gestion : erreur de validation du contrat du contrat de gestion)
- FATAL : une erreur technique est survenue lors de la vérification de l'import du contrat de gestion
(STP_IMPORT_MANAGEMENT_CONTRACT.FATAL = Erreur technique lors du processus d'import du contrat de gestion)
- WARNING : avertissement lors du processus d'import du contrat de gestion
(STP_IMPORT_MANAGEMENT_CONTRACT.WARNING = Avertissement lors du processus d'import du contrat de gestion)

3.7.1.1. Sauvegarde du JSON STP_BACKUP_MANAGEMENT_CONTRACT (ManagementContractImpl.java)

- **Règle** : tâche consistant à enregistrer une copie de la base de données des contrats d'accès sur les offres de stockage
- **Type** : bloquant
- **Statuts** :
 - OK : une copie de la base de données nouvellement importée est enregistrée
(STP_BACKUP_MANAGEMENT_CONTRACT.OK = Succès du processus de sauvegarde des contrats de gestion)
 - KO : pas de cas KO
 - FATAL : une erreur technique est survenue lors de la copie de la base de données nouvellement importée
(STP_BACKUP_MANAGEMENT_CONTRACT.FATAL = Erreur technique lors du processus de sauvegarde des contrats de gestion)

3.7.2. Structure du Workflow d'import du référentiel des contrats de gestion

D'une façon synthétique, le workflow est décrit de cette façon :



3.7.3. Processus de mise à jour d'un contrat de gestion (STP_UPDATE_MANAGEMENT_CONTRACT)

- **Règle** : opération consistant à modifier un contrat de gestion, selon un processus identique à celui de l'import initial

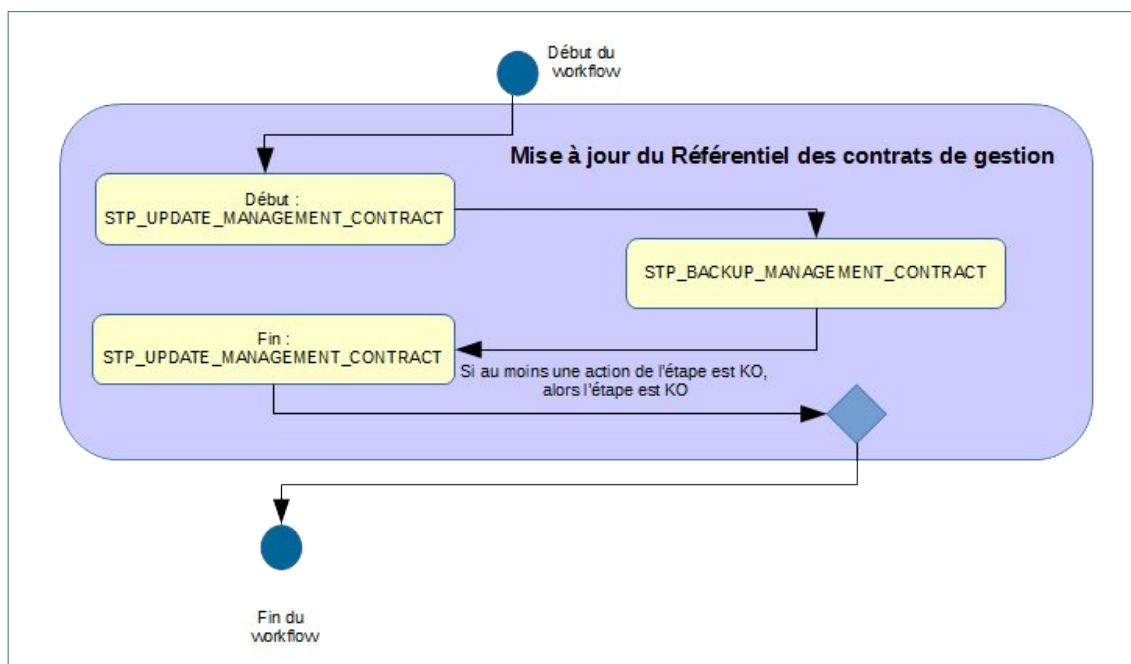
- **Type** : bloquant
- **Statuts** :
 - OK : le contrat répond aux exigences des règles (STP_UPDATE_MANAGEMENT_CONTRACT.OK = Succès du processus de mise à jour du contrat de gestion)
 - KO : une des règles ci-dessus n'a pas été respectée (STP_UPDATE_MANAGEMENT_CONTRACT.KO = Échec du processus de mise à jour du contrat de gestion)
 - Cas n° 1 : l'identifiant du contrat d'entrée est inconnu (STP_UPDATE_MANAGEMENT_CONTRACT.CONTRACT_NOT_FOUND.KO = Échec du processus de mise à jour du contrat de gestion : contrat de gestion non trouvé)
 - Cas n°2 : une des valeurs saisies dans le contrat de gestion ne correspond pas aux valeurs attendues (STP_UPDATE_MANAGEMENT_CONTRACT.NOT_IN_ENUM.KO = Échec du processus de mise à jour du contrat de gestion : une valeur ne correspond pas aux valeurs attendues)
 - Cas n°3 : une des stratégies est invalide (STP_UPDATE_MANAGEMENT_CONTRACT.STRATEGY_VALIDATION_ERROR.KO = Échec du processus de mise à jour du contrat de gestion : erreur de validation du contrat du contrat de gestion)
 - Cas n°4 : la requête de mise à jour du contrat de gestion est mal formatée (STP_UPDATE_MANAGEMENT_CONTRACT.BAD_REQUEST.KO = Échec du processus de mise à jour du contrat de gestion : mauvaise requête)
 - FATAL : une erreur technique est survenue lors de la vérification de l'import du contrat de gestion (STP_UPDATE_MANAGEMENT_CONTRACT.FATAL = Erreur technique lors du processus de mise à jour du contrat de gestion)

3.7.3.1. Sauvegarde du JSON STP_BACKUP_MANAGEMENT_CONTRACT (ManagementContractImpl.java)

- **Règle** : tâche consistant à enregistrer une copie de la base de données des contrats de gestion sur les offres de stockage
- **Type** : bloquant
- **Statuts** :
 - OK : une copie de la base de données nouvellement importée est enregistrée (STP_BACKUP_MANAGEMENT_CONTRACT.OK = Succès du processus de sauvegarde des contrats de gestion)
 - KO : pas de cas KO
 - FATAL : une erreur technique est survenue lors de la copie de la base de données nouvellement importée (STP_BACKUP_MANAGEMENT_CONTRACT.FATAL = Erreur technique lors du processus de sauvegarde des contrats de gestion)

3.7.4. Structure du Workflow de mise à jour d'un référentiel des contrats de gestion

D'une façon synthétique, le workflow est décrit de cette façon :



3.8.Workflow d'administration d'un référentiel des profils d'archivage

Cette section décrit le processus (workflow) permettant d'importer et de mettre à jour un profil d'archivage.

Toutes les étapes, tâches et traitements sont journalisées dans le journal des opérations et décrivent le processus (clé et description de la clé associée dans le journal des opérations) tel qu'implémenté dans la version actuelle de la solution logicielle Vitam.

3.8.1.Processus d'import d'une notice de profil d'archivage (STP_IMPORT_PROFILE_JSON)

Le processus d'import d'une notice de profil d'archivage permet à la fois de vérifier qu'il contient les informations minimales obligatoires, de vérifier la cohérence de l'ensemble des informations, et de lui affecter des éléments peuplés automatiquement. Tous les éléments réalisés au cours de ce processus sont exécutés dans une seule étape.

Règle : opération consistant à vérifier la présence des informations minimales dans la notice de profil d'archivage, à s'assurer de la cohérence des informations saisies, à affecter des données aux champs peuplés par la solution logicielle Vitam et à enregistrer le profil

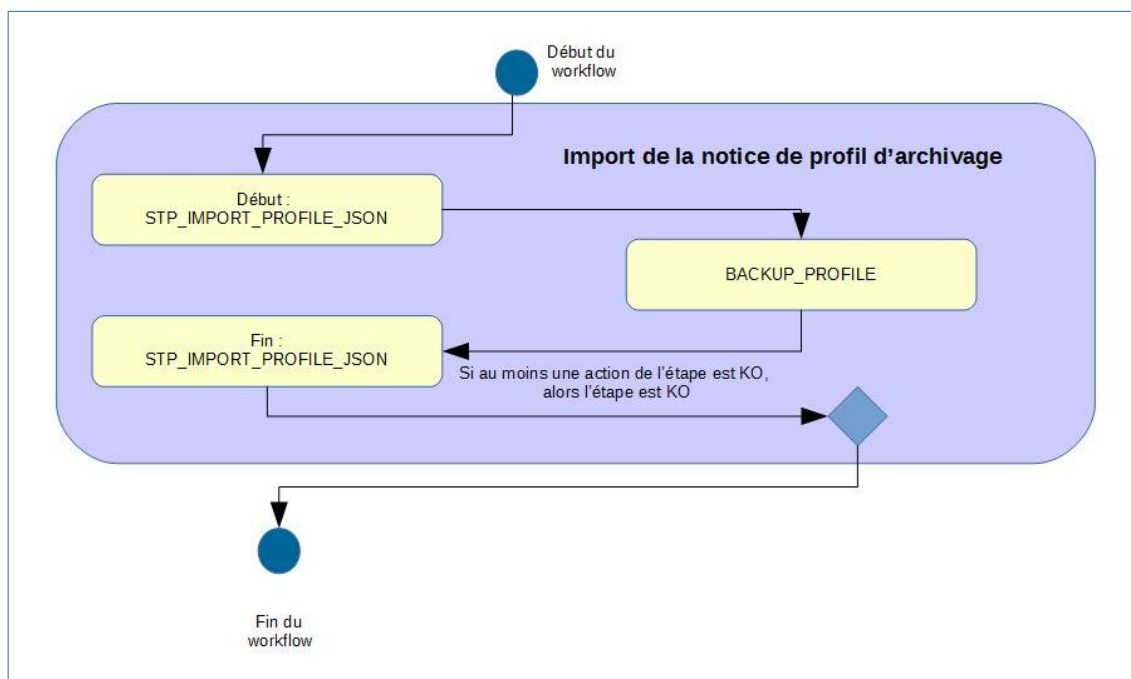
- Les données suivantes sont obligatoirement remplies :
 - Le champ « Name » est peuplé d'une chaîne de caractères
 - Le champ « Identifier » est peuplé d'une chaîne de caractère si le référentiel des profils d'archivage est configuré en mode esclave sur le tenant sélectionné
 - Le champ « Format » doit être renseigné avec la valeur RNG ou XSD
- Les données suivantes optionnelles si elles sont remplies le sont en respectant les règles énoncées pour chacune :
 - Le champ « Description » est peuplé avec une chaîne de caractères
 - Le champ « Status » est peuplé avec la valeur ACTIVE ou la valeur INACTIVE
- **Type** : bloquant
- **Statuts** :
 - OK : les règles ci-dessus sont respectées (STP_IMPORT_PROFILE_JSON.OK=Succès du processus d'import du profil d'archivage)
 - KO :
 - Cas n°1 : une des règles ci-dessus n'a pas été respectée (STP_IMPORT_PROFILE_JSON.KO = Échec du processus d'import du profil d'archivage)
 - Cas n°2 : l'identifiant est déjà utilisé (STP_IMPORT_PROFILE_JSON.IDENTIFIER_DUPLICATION.KO = Échec de l'import du profil d'archivage : l'identifiant est déjà utilisé)
 - Cas n°3 : au moins un des champs obligatoires n'est pas renseigné (STP_IMPORT_PROFILE_JSON.EMPTY_REQUIRED_FIELD.KO = Échec de l'import du profil d'archivage : au moins un des champs obligatoires n'est pas renseigné)
 - Cas n°4 : le profil d'archivage est inconnu (STP_IMPORT_PROFILE_JSON.PROFILE_NOT_FOUND.KO = Échec de l'import du profil d'archivage : profil d'archivage non trouvé)
 - WARNING : avertissement lors du processus d'import du profil d'archivage (STP_IMPORT_PROFILE_JSON.WARNING = Avertissement lors du processus d'import du profil d'archivage)
 - FATAL : une erreur technique est survenue lors de la vérification de l'import du profil d'archivage (STP_IMPORT_PROFILE_JSON.FATAL = Erreur technique lors du processus d'import du profil d'archivage)

3.8.1.1. Sauvegarde du JSON BACKUP_PROFILE (ProfileServiceImpl.java)

- **Règle** : tâche consistant à enregistrer une copie de la base de données des métadonnées de profils d'archivage sur les offres de stockage
- **Type** : bloquant
- **Statuts** :
 - OK : une copie de la base de données nouvellement importée est enregistrée (BACKUP_PROFILE.OK = Succès du processus de sauvegarde des profils d'archivage)
 - KO : pas de cas KO
 - FATAL : une erreur technique est survenue lors de la copie de la base de données nouvellement importée (BACKUP_PROFILE.FATAL = Erreur technique lors du processus de sauvegarde des profils d'archivage)

3.8.2. Structure du Workflow d'import d'une notice de profil d'archivage

D'une façon synthétique, le workflow est décrit de cette façon :



3.8.3. Processus de mise à jour d'une notice de profil d'archivage (STP_UPDATE_PROFILE_JSON)

Le processus de mise à jour d'une notice de profil d'archivage permet à la fois de vérifier qu'il contient les informations minimales obligatoires, de vérifier la cohérence de l'ensemble des informations, et de lui affecter des éléments peuplés automatiquement.

Tous les éléments réalisés au cours de ce processus sont exécutés dans une seule étape.

- **Règle** : opération consistant à mettre à jour la notice d'un profil d'archivage, selon les mêmes règles que celles décrites pour la création.
L'association d'un fichier de profil avec les métadonnées d'un profil provoque également une opération de mise à jour du profil d'archivage.
- **Type** : bloquant
- **Statuts** :
 - OK : le fichier importé est au même format que celui décrit dans le champ « Format »

- (STP_UPDATE_PROFILE_JSON.OK = Succès du processus de mise à jour du profil d'archivage (fichier xsd ou rng))
- KO :
 - Cas n°1 : le fichier importé n'est pas au même format que celui décrit dans le champ « Format » (STP_UPDATE_PROFILE_JSON.KO = Échec du processus de mise à jour du profil d'archivage (fichier xsd ou rng))
 - Cas n°2 : le profil d'archivage est inconnu (STP_UPDATE_PROFILE_JSON.PROFILE_NOT_FOUND.KO = Échec du processus de mise à jour du profil d'archivage : profil non trouvé)
 - Cas n°3 : une des valeurs saisies dans le profil d'archivage ne correspond pas aux valeurs attendues (STP_UPDATE_PROFILE_JSON.NOT_IN_ENUM.KO = Échec du processus de mise à jour du profil d'archivage : une valeur ne correspond pas aux valeurs attendues)
 - Cas n°4 : l'identifiant du profil d'archivage est déjà utilisé (STP_UPDATE_PROFILE_JSON.IDENTIFIER_DUPLICATION.KO = Échec du processus de mise à jour du profil d'archivage : l'identifiant est déjà utilisé)
 - FATAL : une erreur technique est survenue lors de la vérification de l'import du profil d'archivage (STP_UPDATE_PROFILE_JSON.FATAL = Erreur technique lors du processus de mise à jour du profil d'archivage (fichier xsd ou rng))

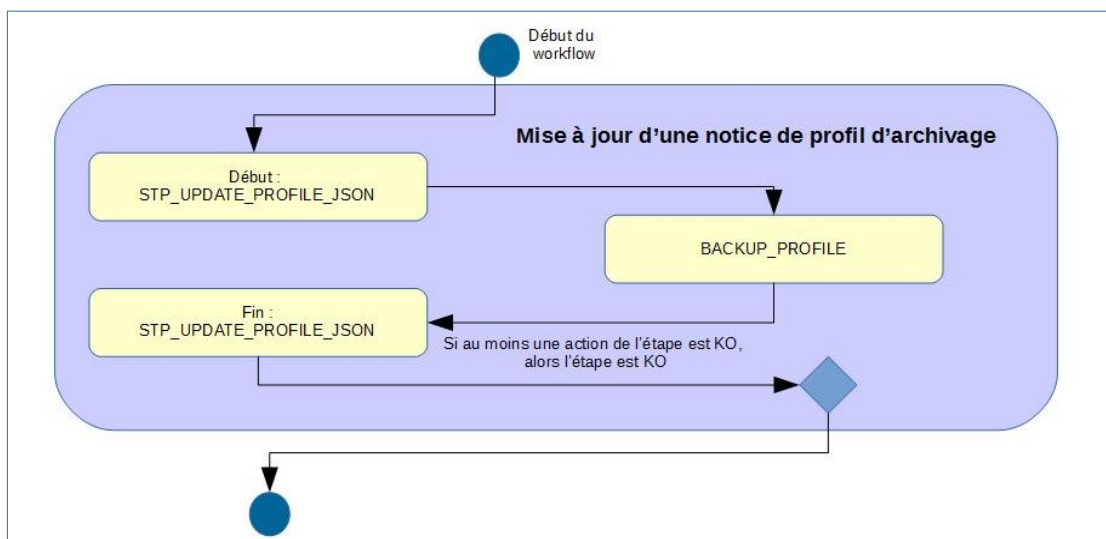
3.8.3.1. Sauvegarde du JSON BACKUP_PROFILE (ProfileServiceImpl.java)

Cette tâche est appelée que ce soit en import initial ou lors de la modification des métadonnées de profils

- **Règle** : tâche consistant à enregistrer enregistrement d'une copie de la base de données des métadonnées de profils d'archivage sur les offres de stockage
- **Type** : bloquant
- **Statuts** :
 - OK : une copie de la base de données nouvellement importée est enregistrée (BACKUP_PROFILE.OK = Succès du processus de sauvegarde des profils d'archivage)
 - KO : pas de cas KO
 - FATAL : une erreur technique est survenue lors de la copie de la base de données nouvellement importée (BACKUP_PROFILE.FATAL = Erreur technique lors du processus de sauvegarde des profils d'archivage)

3.8.4. Structure du Workflow de mise à jour d'une notice de profil d'archivage

D'une façon synthétique, le workflow est décrit de cette façon :



3.9.Workflow d'administration d'un référentiel des profils de sécurité

Cette section décrit le processus (workflow) de création et de mise à jour d'un profil de sécurité. Cette opération n'est réalisable que sur le tenant d'administration.

Toutes les étapes, tâches et traitements sont journalisées dans le journal des opérations et décrivent le processus (clé et description de la clé associée dans le journal des opérations) tel qu'implémenté dans la version actuelle de la solution logicielle Vitam.

3.9.1.Processus d'import d'un référentiel des profils de sécurité (STP_IMPORT_SECURITY_PROFILE)

Le processus d'import d'un profil de sécurité permet à la fois de vérifier qu'il contient les informations minimales obligatoires, de vérifier la cohérence de l'ensemble des informations, et de lui affecter des éléments peuplés automatiquement.

Tous les éléments réalisés au cours de ce processus sont exécutés dans une seule étape.

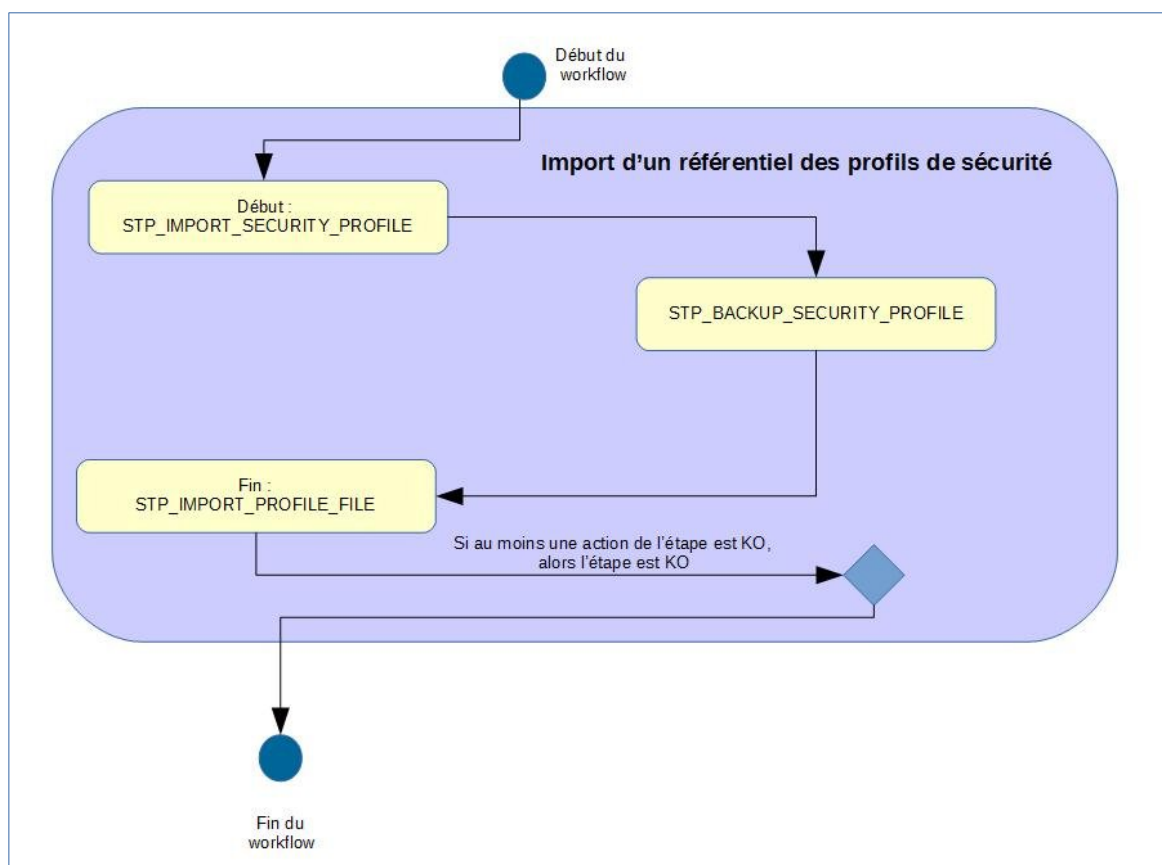
- **Règle** : étape consistant à vérifier la présence des informations minimales dans le profil de sécurité, à s'assurer de la cohérence des informations saisies, à affecter des données aux champs peuplés par la solution logicielle Vitam et à enregistrer le profil de sécurité. Les données suivantes sont obligatoirement remplies :
 - Le champ « Name » doit être peuplé avec une chaîne de caractères unique
 - Le champ « Identifier » doit être unique
 - Le champ « FullAccess » doit être à « true » ou « false »
- **Type** : bloquant
- **Statuts** :
 - OK : les règles ci-dessus sont respectées (STP_IMPORT_SECURITY_PROFILE.OK = Succès du processus d'import du profil de sécurité)
 - KO : une des règles ci-dessus n'est pas respectée (STP_IMPORT_SECURITY_PROFILE.KO = Échec du processus d'import du profil de sécurité)
 - FATAL : une erreur technique est survenue lors de l'import du profil de sécurité (STP_IMPORT_SECURITY_PROFILE.FATAL = Erreur technique lors du processus d'import du profil de sécurité)

3.9.1.1. Sauvegarde du JSON STP_BACKUP_SECURITY_PROFILE (SecurityProfileService.java)

- **Règle** : tâche consistant à enregistrer une copie de la base de données des profils de sécurité sur le stockage
- **Type** : bloquant
- **Statuts** :
 - OK : une copie de la base de données nouvellement importée est enregistrée (STP_BACKUP_SECURITY_PROFILE.OK = Succès du processus de sauvegarde des profils de sécurité)
 - KO : pas de cas KO
 - FATAL : une erreur technique est survenue lors de la copie de la base de données nouvellement importée (STP_BACKUP_SECURITY_PROFILE.FATAL = Erreur technique lors du processus de sauvegarde des profils de sécurité)

3.9.2. Structure du Workflow d'import d'un référentiel des profils de sécurité

D'une façon synthétique, le workflow est décrit de cette façon :



3.9.3. Processus de mise à jour d'un référentiel des profils de sécurité (STP_UPDATE_SECURITY_PROFILE)

Le processus de mise à jour d'un profil de sécurité permet à la fois de vérifier qu'il contient les informations minimales obligatoires, de vérifier la cohérence de l'ensemble des informations, et de lui affecter des éléments peuplés automatiquement.

Tous les éléments réalisés au cours de ce processus sont exécutés dans une seule étape.

- **Règle** : opération consistant à mettre à jour le profil de sécurité, selon les mêmes règles que celles décrites pour la création
- **Type** : bloquant
- **Statuts** :
 - OK : les règles ci-dessus sont respectées (STP_UPDATE_SECURITY_PROFILE.OK = Succès du processus de mise à jour du profil de sécurité)
 - KO : une des règles ci-dessus n'est pas respectée (STP_UPDATE_SECURITY_PROFILE.KO = Échec du processus de mise à jour du profil de sécurité)
 - FATAL : une erreur technique est survenue lors de l'import du profil de sécurité (STP_UPDATE_SECURITY_PROFILE.FATAL = Erreur technique lors du processus de mise à jour du profil de sécurité)

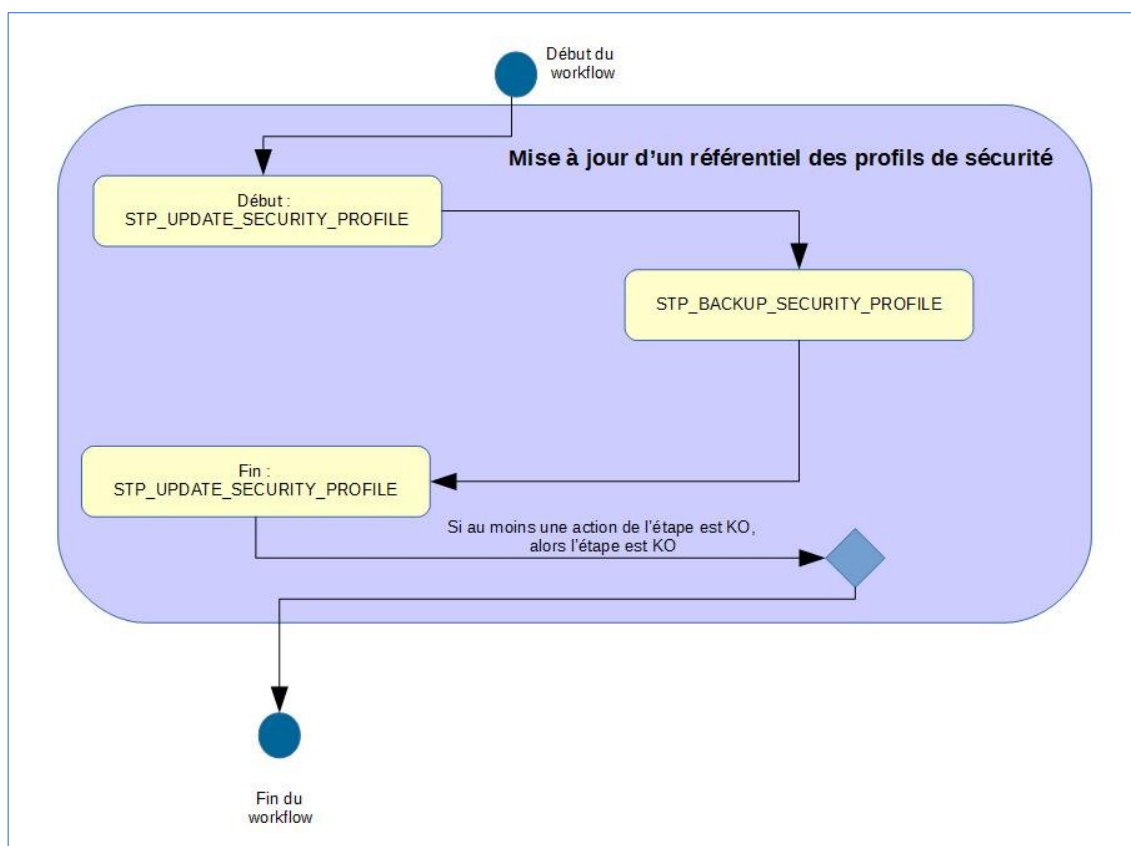
3.9.3.1. Sauvegarde du JSON STP_BACKUP_SECURITY_PROFILE (SecurityProfileService.java)

Cette tâche est appelée que ce soit en import initial ou en modification.

- **Règle** : tâche consistant à enregistrer une copie de la base de données des profils de sécurité sur le stockage
- **Type** : bloquant
- **Statuts** :
 - OK : une copie de la base de données nouvellement importée est enregistrée (STP_BACKUP_SECURITY_PROFILE.OK = Succès du processus de sauvegarde des profils de sécurité)
 - KO : pas de cas KO
 - FATAL : une erreur technique est survenue lors de la copie de la base de données nouvellement importée (STP_BACKUP_SECURITY_PROFILE.FATAL = Erreur technique lors du processus de sauvegarde des profils de sécurité)

3.9.4. Structure du Workflow de mise à jour du référentiel des profils de sécurité

D'une façon synthétique, le workflow est décrit de cette façon :



3.10.Workflow d'administration d'un référentiel des contextes applicatifs

Cette section décrit le processus (workflow) d'import et de mise à jour du référentiel des contextes. Cette opération n'est réalisable que sur le tenant d'administration.

Toutes les étapes, tâches et traitements sont journalisés dans le journal des opérations et décrivent le processus (clé et description de la clé associée dans le journal des opérations) tel qu'implémenté dans la version actuelle de la solution logicielle Vitam.

3.10.1.Processus d'import d'un référentiel des contextes applicatifs (STP_IMPORT_CONTEXT)

Le processus d'import d'un référentiel des contextes applicatifs permet à la fois de vérifier qu'il contient les informations minimales obligatoires, de vérifier la cohérence de l'ensemble des informations, et de lui affecter des éléments peuplés automatiquement.

Tous les éléments réalisés au cours de ce processus sont exécutés dans une seule étape.

- **Règle** : opération consistant à vérifier la présence des informations minimales dans le contexte applicatif, à s'assurer de la cohérence des informations saisies, à affecter des données aux champs peuplés par la solution logicielle Vitam et à enregistrer le contexte applicatif :
 - Le champ « Name » doit être peuplé avec une chaîne de caractères unique
 - Le champ « Status » doit être à « ACTIVE » ou « INACTIVE »
 - Le champ « EnableControl » doit être à « true » ou « false »
 - Le champ « Permissions » doit être peuplé avec un tableau contenant des fichiers JSON
 - Le champ « SecurityProfile » doit être peuplé avec une chaîne de caractères
 - Le champ « Identifier » doit être unique
- **Type** : bloquant
- **Statuts** :
 - OK : Les règles ci-dessus sont respectées (STP_IMPORT_CONTEXT.OK = Succès du processus d'import du contexte)
 - KO :
 - Cas n°1 : une des règles ci-dessus n'est pas respectée (STP_IMPORT_CONTEXT.KO = Échec du processus d'import du contexte)
 - Cas n°2 : l'identifiant est déjà utilisé (STP_IMPORT_CONTEXT.IDENTIFIER_DUPLICATION.KO = Échec de l'import : l'identifiant est déjà utilisé)
 - Cas n°3 : un des champs obligatoires n'est pas renseigné (STP_IMPORT_CONTEXT.EMPTY_REQUIRED_FIELD.KO = Échec de l'import : au moins un des champs obligatoires n'est pas renseigné)
 - Cas n°4 : le profil de sécurité mentionné est inconnu du système (STP_IMPORT_CONTEXT.SECURITY_PROFILE_NOT_FOUND.KO = Échec de l'import : profil de sécurité non trouvé)
 - Cas n°5 : au moins un objet déclare une valeur inconnue (STP_IMPORT_CONTEXT.UNKNOWN_VALUE.KO = Échec de l'import : au moins un objet déclare une valeur inconnue)
 - FATAL : une erreur technique est survenue lors de l'import du contexte (STP_IMPORT_CONTEXT.FATAL=Erreur technique lors du processus d'import du contexte)

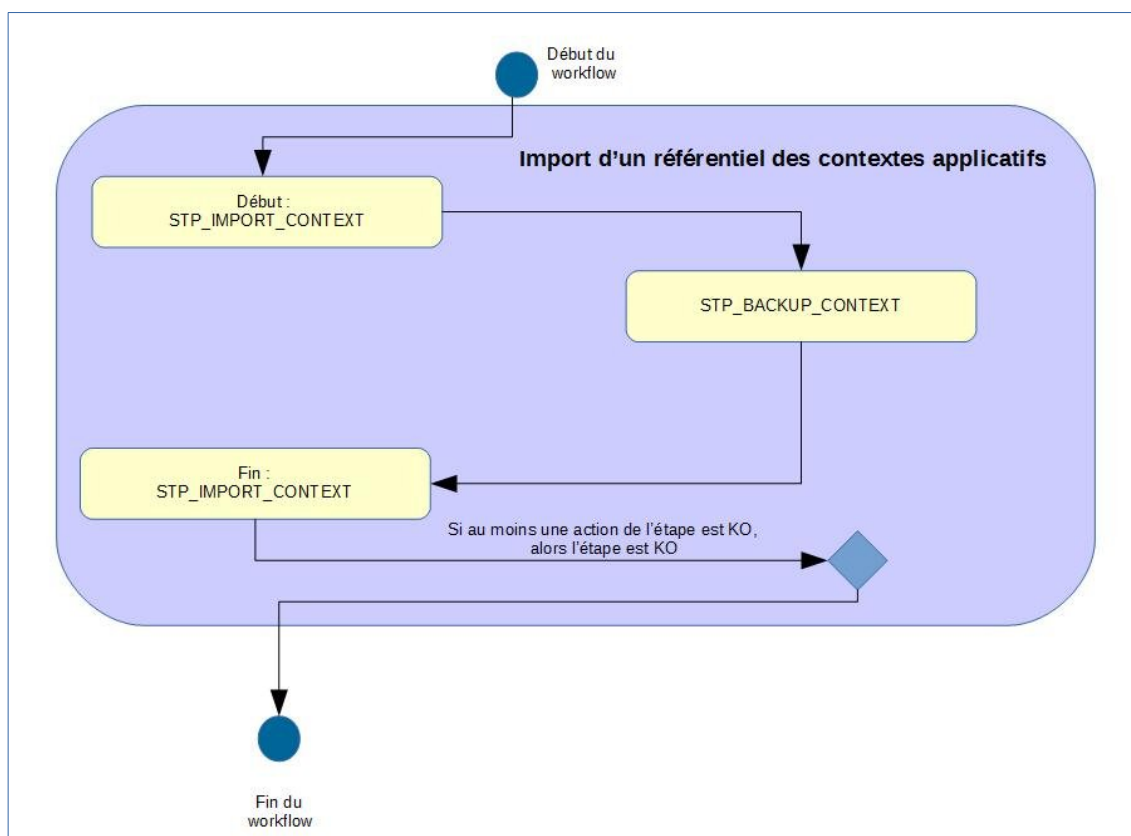
3.10.1.1. Sauvegarde du JSON STP_BACKUP_CONTEXT (ContextServiceImpl.java)

- **Règle** : tâche consistant à enregistrer une copie de la base de données des contextes applicatifs sur le stockage

- **Type** : bloquant
- **Statuts** :
 - OK : une copie de la base de données nouvellement importée est enregistrée (STP_BACKUP_CONTEXT.OK = Succès du processus de sauvegarde des contextes)
 - KO : pas de cas KO
 - FATAL : une erreur technique est survenue lors de la copie de la base de données nouvellement importée (STP_BACKUP_CONTEXT.FATAL = Erreur technique lors du processus de sauvegarde des contextes)

3.10.2. Structure du Workflow d'import d'un référentiel des contextes applicatifs

D'une façon synthétique, le workflow est décrit de cette façon :



3.10.3. Processus de mise à jour d'un référentiel des contextes applicatifs (STP_UPDATE_CONTEXT)

Le processus de mise à jour d'un contexte applicatif permet à la fois de vérifier qu'il contient les informations minimales obligatoires, de vérifier la cohérence de l'ensemble des informations, et de lui affecter des éléments peuplés automatiquement.

Tous les éléments réalisés au cours de ce processus sont exécutés dans une seule étape.

- **Règle** : opération consistant à mettre à jour le contexte applicatif, selon les mêmes règles que celles décrites pour la création
- **Type** : bloquant
- **Statuts** :
 - OK : les règles ci-dessus sont respectées (STP_UPDATE_CONTEXT.OK = Succès du processus de mise à jour du contexte)

- KO : une des règles ci-dessus n'est pas respectée (STP_UPDATE_CONTEXT.KO = Échec du processus mise à jour du contexte)
 - Cas n°1 : une des valeurs saisies dans le contexte applicatif ne correspond pas aux valeurs attendues (STP_UPDATE_CONTEXT.UNKNOWN_VALUE.KO = Échec du processus de mise à jour du contexte : au moins un objet déclare une valeur inconnue)
- FATAL : une erreur technique est survenue lors de l'import du contexte (STP_UPDATE_CONTEXT.FATAL = Erreur technique lors du processus de mise à jour du contexte)

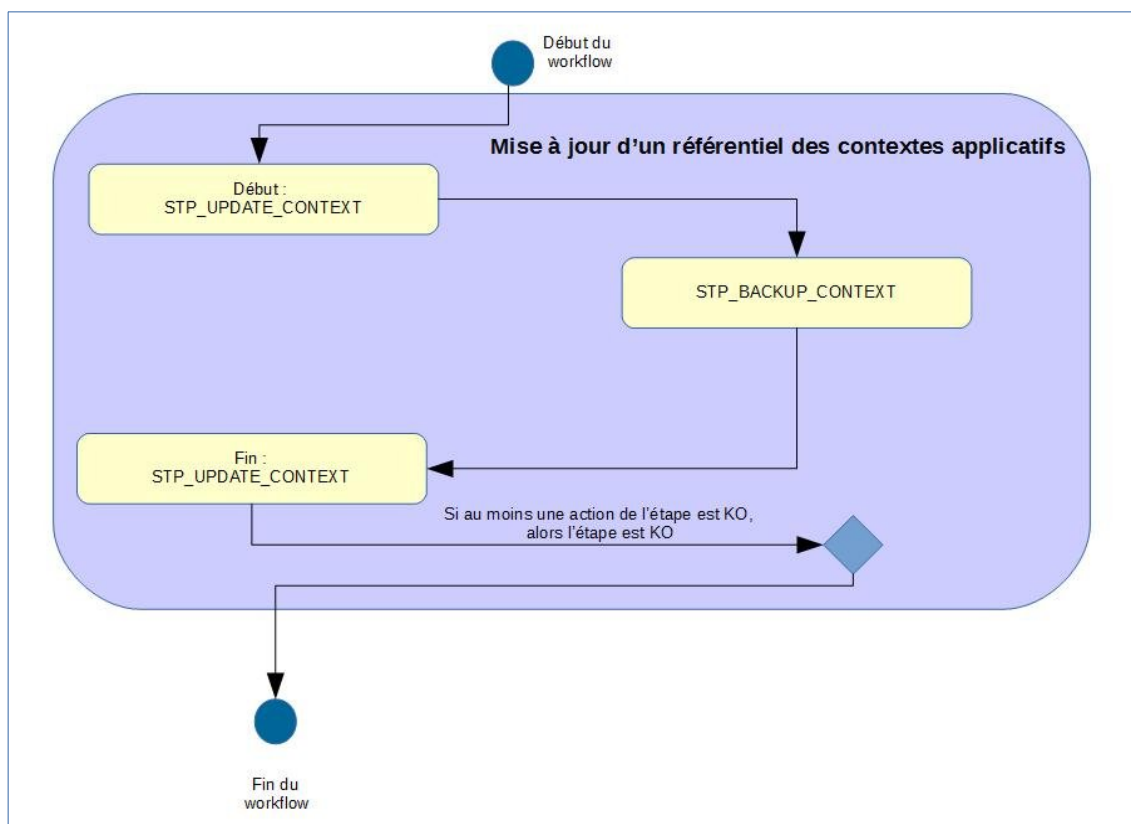
3.10.3.1. Sauvegarde du JSON STP_BACKUP_CONTEXT (ContextServiceImpl.java)

Cette tâche est appelée que ce soit en import initial ou en modification.

- **Règle** : enregistrement d'une copie de la base de données des contextes applicatifs sur le stockage
- **Type** : bloquant
- **Statuts** :
 - OK : une copie de la base de données nouvellement importée est enregistrée (STP_BACKUP_CONTEXT.OK = Succès du processus de sauvegarde des contextes)
 - KO : pas de cas KO
 - FATAL : une erreur technique est survenue lors de la copie de la base de données nouvellement importée (STP_BACKUP_CONTEXT.FATAL = Erreur technique lors du processus de sauvegarde des contextes)

3.10.4. Structure du Workflow de mise à jour du référentiel des contextes applicatifs

D'une façon synthétique, le workflow est décrit de cette façon :



3.11.Workflow d'administration du référentiel des profils d'unités archivistiques

Cette section décrit le processus (workflow) permettant d'importer et de mettre à jour un profil d'unité archivistique (documents type).

Toutes les étapes, tâches et traitements sont journalisés dans le journal des opérations et décrivent le processus (clé et description de la clé associée dans le journal des opérations) tel qu'implémenté dans la version actuelle de la solution logicielle Vitam.

3.11.1.Processus d'import d'un référentiel des profils d'unité archivistique (IMPORT_ARCHIVEUNITPROFILE)

Le processus d'import d'un profil d'unité archivistique (document type) permet à la fois de vérifier qu'il contient les informations minimales obligatoires, de vérifier la cohérence de l'ensemble des informations, et de lui affecter des éléments peuplés automatiquement.

Tous les éléments réalisés au cours de ce processus sont exécutés dans une seule étape.

- **Règle** : opération consistant à vérifier la présence des informations minimales dans le profil d'unité archivistique, à s'assurer de la cohérence des informations saisies, à affecter des données aux champs peuplés par la solution logicielle Vitam et à enregistrer le profil d'unité archivistique :

Les données suivantes sont obligatoirement remplies :

- Le champ « Name » est peuplé d'une chaîne de caractères
- Le champ « Identifier » est peuplé d'une chaîne de caractères si le référentiel des profils d'unité archivistique est configuré en mode esclave sur le tenant sélectionné

Les données suivantes, optionnelles, si elles sont remplies, le sont en respectant les règles énoncées pour chacune :

- Le champ « Description » est peuplé avec une chaîne de caractères
- Le champ « Status » est peuplé avec la valeur « ACTIVE » ou la valeur « INACTIVE »

- **Type** : bloquant
- **Statuts** :
- **OK** : les règles ci-dessus sont respectées (IMPORT_ARCHIVEUNITPROFILE.OK = Succès du processus d'import du profil d'unité archivistique)
- **KO** :
 - Cas n°1 : une des règles ci-dessus n'a pas été respectée (IMPORT_ARCHIVEUNITPROFILE.KO = Échec du processus d'import du profil d'unité archivistique)
 - Cas n°2 : l'identifiant est déjà utilisé (IMPORT_ARCHIVEUNITPROFILE.IDENTIFIER_DUPLICATION.KO = Échec de l'import du profil d'unité archivistique : l'identifiant est déjà utilisé)
 - Cas n°3 : au moins un des champs obligatoires n'est pas renseigné (IMPORT_ARCHIVEUNITPROFILE.EMPTY_REQUIRED_FIELD.KO = Échec de l'import du profil d'unité archivistique : au moins un des champs obligatoires n'est pas renseigné)
 - Cas n°4 : Schéma JSON invalide (IMPORT_ARCHIVEUNITPROFILE.INVALID_JSON_SCHEMA.KO = Échec de l'import du profil d'unité archivistique : schéma JSON non valide)
- **FATAL** : une erreur technique est survenue lors de la vérification de l'import du profil d'unité archivistique (document type) (IMPORT_ARCHIVEUNITPROFILE.FATAL = Erreur technique lors du processus d'import du profil d'unité archivistique (document type))

3.11.1.1. Sauvegarde du JSON STP_BACKUP_ARCHIVEUNITPROFILE (ArchiveUnitProfileManager.java)

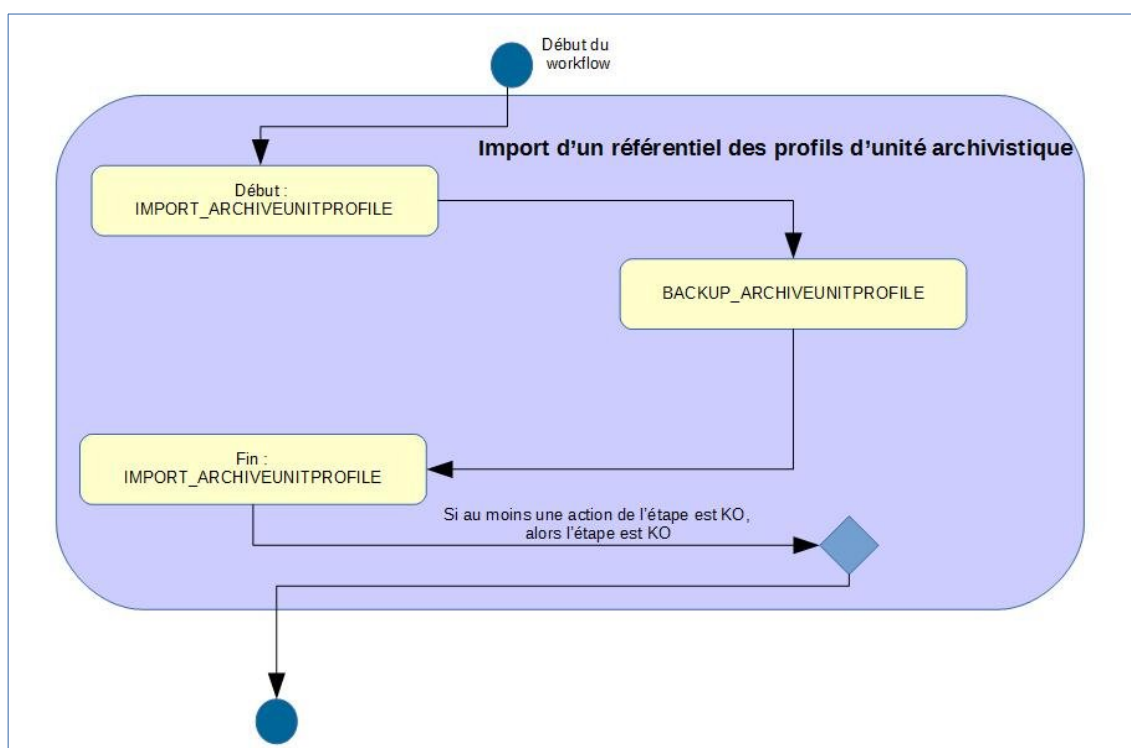
- **Règle** : tâche consistant à enregistrer une copie de la base de données des profils d'unités archivistiques

sur les offres de stockage

- **Type** : bloquant
- **Statuts** :
 - OK : une copie de la base de données nouvellement importée est enregistrée (BACKUP_ARCHIVEUNITPROFILE.OK = Succès du processus de sauvegarde des profils d'unité archivistique (document type))
 - KO : pas de cas KO
 - FATAL : une erreur technique est survenue lors de la copie de la base de données des profils d'unités archivistiques (document type) (BACKUP_ARCHIVEUNITPROFILE.FATAL = Erreur technique lors du processus de sauvegarde des profils d'unité archivistique (document type))

3.11.2. Structure du Workflow d'import d'un référentiel des profils d'unité archivistique

D'une façon synthétique, le workflow est décrit de cette façon :



3.11.3. Processus de mise à jour d'un profil d'unité archivistique (UPDATE_ARCHIVEUNITPROFILE)

Le processus d'import d'un profil d'unité archivistique (document type) permet à la fois de vérifier qu'il contient les informations minimales obligatoires, de vérifier la cohérence de l'ensemble des informations, et de lui affecter des éléments peuplés automatiquement.

Tous les éléments réalisés au cours de ce processus sont exécutés dans une seule étape.

- **Règle** : opération consistant à mettre à jour le profil d'unité archivistique, selon les mêmes règles que celles décrites pour la création
- **Type** : bloquant
- **Statuts** :
 - OK : les règles ci-dessus sont respectées (UPDATE_ARCHIVEUNITPROFILE.OK = Succès du processus de mise à jour du profil d'unité archivistique (document type))

- KO : une des règles ci-dessus n'a pas été respectée (UPDATE_ARCHIVEUNITPROFILE.KO = Échec du processus d'import du profil d'unité archivistique (document type)
 - Cas n°1 : l'identifiant du profil d'unité archivistique est inconnu (UPDATE_ARCHIVEUNITPROFILE.AUP_NOT_FOUND.KO = Échec du processus de mise à jour du profil d'unité archivistique : profil d'unité archivistique non trouvé)
 - Cas n°2 : une des valeurs saisies dans le profil d'unité archivistique ne correspond pas aux valeurs attendues (UPDATE_ARCHIVEUNITPROFILE.NOT_IN_ENUM.KO = Échec du processus de mise à jour du profil d'unité archivistique : une valeur ne correspond pas aux valeurs attendues)
 - Cas n°3 : l'identifiant est déjà utilisé (UPDATE_ARCHIVEUNITPROFILE.IDENTIFIER_DUPLICATION.KO = Échec du processus de mise à jour du profil d'unité archivistique : l'identifiant est déjà utilisé)
- FATAL : une erreur technique est survenue lors du processus de mise à jour du profil d'unité archivistique (UPDATE_ARCHIVEUNITPROFILE.FATAL = Erreur technique lors du processus de mise à jour du profil d'unité archivistique (document type)

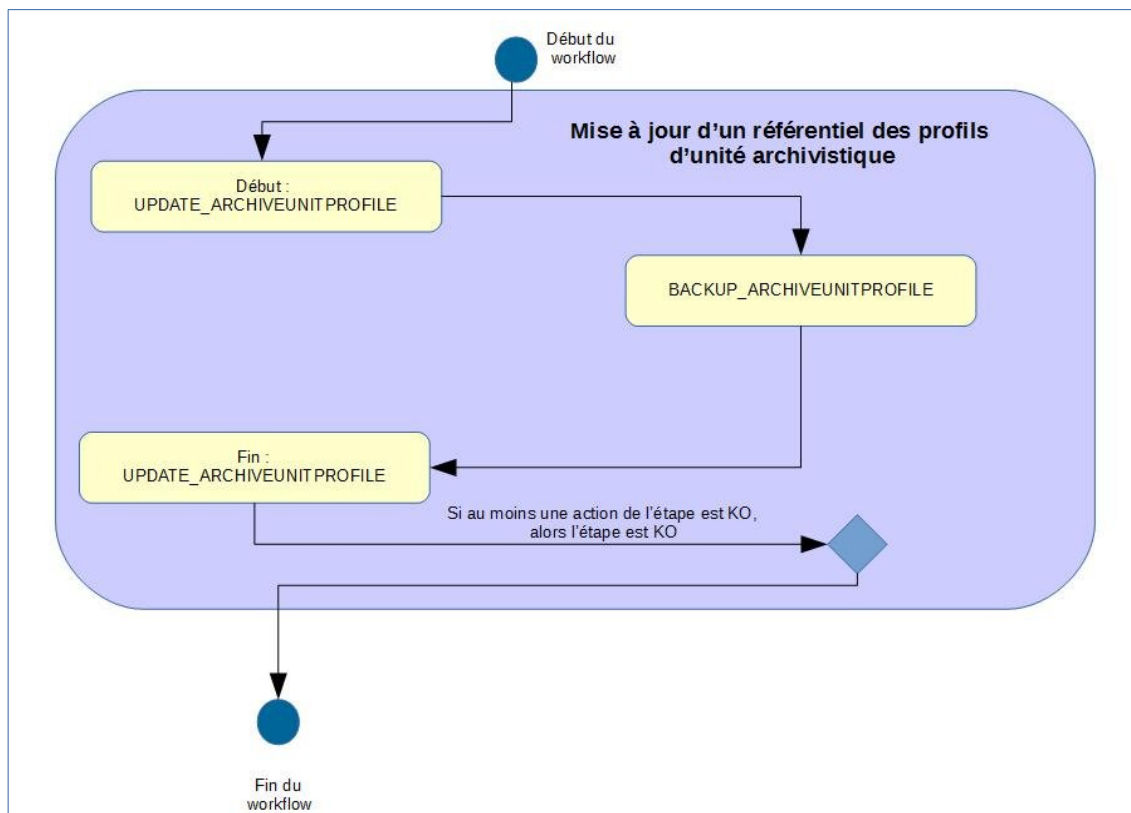
3.11.3.1. Sauvegarde du JSON STP_BACKUP_ARCHIVEUNITPROFILE (ArchiveUnitProfileManager.java)

Cette tâche est appelée que ce soit en import initial ou lors de la modification des métadonnées de profil d'unité archivistique (document type).

- **Règle** : tâche consistant à enregistrer un enregistrement d'une copie de la base de données des métadonnées des profils d'unités archivistiques sur le stockage les offres de stockage
- **Type** : bloquant
- **Statuts** :
 - OK : une copie de la base de données nouvellement importée est enregistrée (BACKUP_ARCHIVEUNITPROFILE.OK = Succès du processus de sauvegarde des profils d'unité archivistique (document type)
 - KO : pas de cas KO
 - FATAL : une erreur technique est survenue lors de la copie de la base de données échec du processus de sauvegarde dudes profils d'unités archivistiques (document type) (BACKUP_ARCHIVEUNITPROFILE.KOFATAL = Erreur technique lors du processus de sauvegarde des profils d'unité archivistique (document type)

3.11.4. Structure du Workflow de mise à jour du référentiel des profils d'unité archivistique

D'une façon synthétique, le workflow est décrit de cette façon :



3.12. Workflow d'administration d'un référentiel des vocabulaires de l'ontologie

Cette section décrit le processus permettant d'importer des vocabulaires de l'ontologie. Cette opération n'est réalisable que sur le tenant d'administration.

Toutes les étapes, tâches et traitements sont journalisés dans le journal des opérations et décrivent le processus (clé et description de la clé associée dans le journal des opérations) tel qu'implémenté dans la version actuelle de la solution logicielle Vitam.

3.12.1. Processus d'import et mise à jour des vocabulaires de l'ontologie (STP_IMPORT_ONTOLOGY)

Le processus d'import d'une ontologie permet d'ajouter des vocabulaires qui seront utilisés dans les profils d'unité archivistique (documents types) Tous les éléments réalisés au cours de ce processus sont exécutés dans une seule étape.

- **Règle** : opération consistant à vérifier la présence des informations minimales dans le référentiel des vocabulaires de l'ontologie, à s'assurer de la cohérence des informations saisies, à affecter des données aux champs peuplés par la solution logicielle Vitam et à enregistrer le référentiel des vocabulaires de l'ontologie.

L'ontologie répond aux exigences suivantes :

- Le fichier est au format Json.
- Les données suivantes sont obligatoires :
 - Le champ « Identifier » est peuplé d'une chaîne de caractères
 - Le champ « Type » est peuplé par une valeur comprise dans la liste :
 - TEXT
 - KEYWORD
 - DATE
 - LONG
 - DOUBLE
 - BOOLEAN
 - GEO_POINT
 - ENUM
 - Le champ « Origin » est peuplé par la valeur « EXTERNAL » ou « INTERNAL ». L'INTERNAL correspond à l'ontologie interne de la solution logicielle Vitam
- Les données suivantes sont facultatives; si elles sont remplies, elles respectent les règles énoncées pour chacune :
 - Le champ « SedaField » est peuplé d'une chaîne de caractères
 - Le champ « ApiField » est peuplé d'une chaîne de caractères
 - Le champ « Description » est peuplé d'une chaîne de caractères
 - Le champ « ShortName » correspond au champ traduction, il est peuplé par une chaîne de valeur
 - Le champ « Collections » indique la collection dans laquelle le vocabulaire est rattaché, ex : « Unit »

Exemple d'ontologie :

```
[ {
  "Identifier" : "AcquiredDate",
  "SedaField" : "AcquiredDate",
  "ApiField" : "AcquiredDate",
  "Description" : "unit-es-mapping.json",
  "Type" : "DATE",
  "Origin" : "INTERNAL",
  "ShortName" : "AcquiredDate",
```

```

"Collections" : [ "Unit" ]
}, {
  "Identifier" : "BirthDate",
  "SedaField" : "BirthDate",
  "ApiField" : "BirthDate",
  "Description" : "unit-es-mapping.json",
  "Type" : "DATE",
  "Origin" : "INTERNAL",
  "ShortName" : "BirthDate",
  "Collections" : [ "Unit" ]
}]
    
```

- **Type** : bloquant
- **Statuts** :
 - OK : les règles ci-dessus sont respectées (IMPORT_ONTOLOGY.OK = Succès du processus d'import de l'ontologie)
 - KO : une des règles ci-dessus n'a pas été respectée (IMPORT_ONTOLOGY.KO = Échec du processus d'import de l'ontologie)
 - FATAL : une erreur technique est survenue lors de la vérification de l'import de l'ontologie (IMPORT_ONTOLOGY.FATAL = Erreur technique lors du processus d'import de l'ontologie)
 - WARNING : avertissement lors du processus d'import de l'ontologie (IMPORT_ONTOLOGY.WARNING = Avertissement lors du processus d'import de l'ontologie)

La modification d'une ontologie s'effectue par ré-import du fichier JSON. Le nouvel import annule et remplace l'ontologie précédente. Ce ré-import observe les règles décrites dans le processus d'import, décrit plus haut.

***Note** : la mise à jour des vocabulaires de l'ontologie doit respecter certaines règles de compatibilité concernant la valeur du « Type » :*

- *Le champ Type TEXT peut être modifié en KEYWORD, TEXT*
- *Le champ Type KEYWORD peut être modifié en KEYWORD, TEXT*
- *Le champ Type DATE peut être modifié en KEYWORD, TEXT*
- *Le champ Type LONG peut être modifié en KEYWORD, TEXT, DOUBLE*
- *Le champ Type DOUBLE peut être modifié en KEYWORD, TEXT*
- *Le champ Type BOOLEAN peut être modifié en KEYWORD, TEXT*
- *Le champ Type GEO-POINT peut être modifié en KEYWORD, TEXT*
- *Le champ Type ENUM de valeur peut être modifié en KEYWORD, TEXT*

3.12.1.1. Sauvegarde du JSON STP_BACKUP_ONTOLOGY

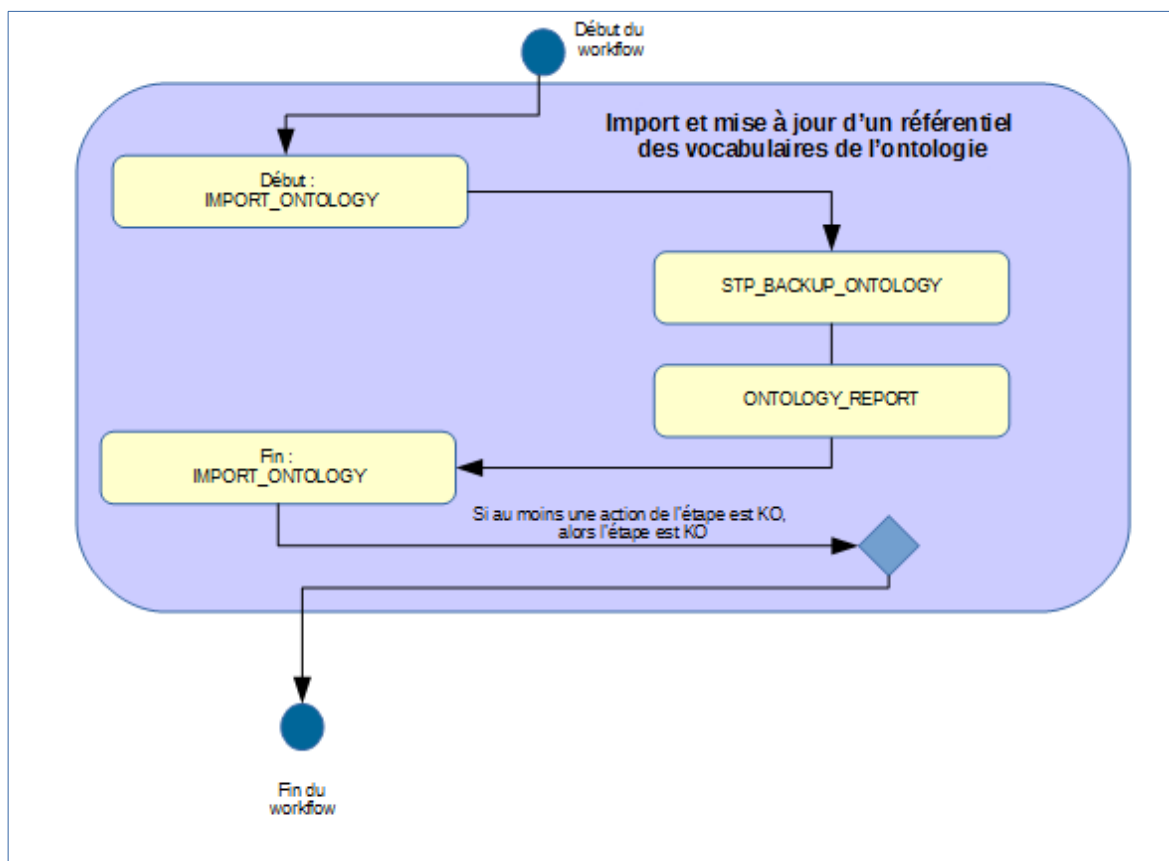
- **Règle** : tâche consistant à enregistrer une copie de la base de données des métadonnées des ontologies sur les offres de stockage
- **Type** : bloquant
- **Statuts** :
 - OK : une copie de la base de données nouvellement importée est enregistrée (STP_BACKUP_ONTOLOGY.OK = Succès du processus de sauvegarde du référentiel des ontologies)
 - KO : pas de cas KO
 - FATAL : une erreur technique est survenue lors de la copie de la base de données (STP_BACKUP_ONTOLOGY.FATAL = Erreur technique lors du processus de sauvegarde du référentiel des ontologies)

3.12.1.2. Processus de génération du rapport d'import du référentiel des vocabulaires de l'ontologie ONTOLOGY_REPORT

- **Règle** : tâche consistant à créer le rapport d'import du référentiel des vocabulaires de l'ontologie
- **Type** : bloquant
- **Statuts** :
 - OK : le rapport d'import du référentiel des vocabulaires de l'ontologie a bien été créé (ONTOLOGY_REPORT.OK = Succès du processus de génération du rapport d'import du référentiel des vocabulaires de l'ontologie)
 - KO : pas de KO
 - FATAL : une erreur technique est survenue lors de la création du rapport d'import du référentiel des vocabulaires de l'ontologie (ONTOLOGY_REPORT.FATAL = Erreur technique lors du processus de génération du rapport d'import du référentiel des vocabulaires de l'ontologie)

3.12.2. Structure du Workflow d'import et de mise à jour d'un référentiel des vocabulaires de l'ontologie

D'une façon synthétique, le workflow est décrit de cette façon :



3.12.3. Structure du rapport d'administration des vocabulaires de l'ontologie

Lorsqu'un référentiel est importé ou mis à jour, la solution logicielle Vitam génère un rapport de l'opération.

Ce rapport est en plusieurs parties :

- « Operation » contient :
 - « evType » : le type d'opération. Dans le cadre de ce rapport, il s'agit toujours de « IMPORT_ONTOLOGY »

- « evDateTime » : la date et l'heure de l'opération d'import
- « evId » : l'identifiant de l'opération
- « outMessage » : le type d'opération. Dans le cadre de ce rapport, il s'agit toujours de « IMPORT_ONTOLOGY »
- « deletedOntologies » : liste des vocabulaires supprimés
- « updatedOntologies » : liste des vocabulaires mis à jour
- « createdOntologies » : liste des vocabulaires ajoutés

Exemple :

```
{
  "Operation": {
    "evType": "IMPORT_ONTOLOGY",
    "evDateTime": "2019-11-09T18:54:23.097",
    "evId": "aeaaaaaaghia7juabddealokgb75gyaaaaaq",
    "outMessg": "IMPORT_ONTOLOGY"
  },
  "deletedOntologies": [],
  "updatedOntologies": ["AcquiredDate", "BirthDate", "BirthName", "Address", "City",
"Country", "Geogname", "PostalCode", "Region", "Corpname", "DeathDate", "FirstName",
"Gender", "GivenName", "Identifrier", "Nationality",
"ArchivalAgencyArchiveUnitIdentifier", "ArchiveUnitProfile", "Jurisdictional", "Spatial",
"Temporal", "CreatedDate", "DataObjectGroupReferenceId", "CustodialHistoryItem",
"Description", "DescriptionLanguage", "DescriptionLevel", "DocumentType", "EndDate",
"evTypeDetail", "FilePlanPosition", "GpsAltitude", "GpsAltitudeRef", "GpsDateStamp",
"GpsLatitude", "GpsLatitudeRef", "GpsLongitude", "GpsLongitudeRef", "GpsVersionID",
"KeywordContent", "KeywordReference", "KeywordType", "Language",
"OriginatingAgencyArchiveUnitIdentifier", "OriginatingSystemId", "ReceivedDate",
"RegisteredDate", "ArchiveUnitRefId", "DataObjectReferenceId",
"RepositoryArchiveUnitPID", "RepositoryObjectPID", "ExternalReference", "Activity",
"ExecutableName", "ExecutableVersion", "Function", "Position", "Role", "Mandate",
"SentDate", "Algorithm", "SignedObjectId", "FullName", "SigningTime", "ValidationTime",
"Source", "StartDate", "Status", "SystemId", "Tag", "Title", "TransactedDate",
"TransferringAgencyArchiveUnitIdentifier", "Type", "Version", "_glpd", "_graph", "_max",
"PreventInheritance", "PreventRulesId", "Rule", "FinalAction", "ClassificationLevel",
"ClassificationOwner", "ClassificationAudience", "ClassificationReassessingDate",
"NeedReassessingAuthorization", "NeedAuthorization", "_min", "_nbc", "_og", "_opi",
"_ops", "_opts", "_sp", "_sps", "offerIds", "strategyId", "_tenant", "_unitType", "_up",
"_us", "_v", "_av", "CreatingApplicationName", "CreatingApplicationVersion",
"CreatingOs", "CreatingOsVersion", "DateCreatedByApplication", "Filename",
"LastModified", "_profil", "qualifier", "DataObjectGroupId", "DataObjectVersion",
"encoding", "FormatId", "FormatLitteral", "MimeType", "MessageDigest", "dValue", "unit",
"NumberOfPage", "Shape", "PhysicalId", "Size", "Uri", "_id", "AccessLog",
"ActivationDate", "CreationDate", "DeactivationDate", "EveryDataObjectVersion",
"EveryOriginatingAgency", "ExcludedRootUnits", "LastUpdate", "Name",
"OriginatingAgencies", "RuleCategoryToFilter", "RootUnits", "WritingPermission",
"WritingRestrictedDesc", "AcquisitionInformation", "ArchivalAgreement", "deleted",
>ActionList", "FormatList", "GriffinIdentifier", "Timeout", "MaxSize", "Debug", "Args",
"ingested", "remained", "Opi", "Opc", "OpType", "Gots", "Units", "Objects", "ObjSize",
"OperationIds", "OriginatingAgency", "SubmissionAgency", "LegalStatus", "ControlSchema",
"Fields", "EnableControl", "AccessContracts", "IngestContracts", "tenant",
"SecurityProfile", "Alert", "Comment", "Extension", "Group",
"HasPriorityOverFileFormatID", "PUID", "VersionPronom", "ArchiveProfiles",
"CheckParentLink", "EveryFormatType", "FormatType", "FormatUnidentifiedAuthorized",
"ComputeInheritedRulesAtIngest", "LinkParentId", "MasterMandatory",
"ManagementContractId", "_lastPersistedDate", "agId", "evDateTime", "evId", "evDetData",
"evIdProc", "SecurisationVersion", "MaxEntriesReached", "ServiceLevel", "evIdReq",
"evParentId", "evType", "evTypeProc", "ArchivalAgency", "TransferringAgency",
"AgIfTrans", "EvDateTimeReq", "EvDetailReq", "Hash", "LogType",
```

```
"MinusOneMonthLogbookTraceabilityDate", "MinusOneYearLogbookTraceabilityDate",  
"NumberOfElements", "PreviousLogbookTraceabilityDate", "TimeStampToken", "diff",  
"errors", "reports", "loadingURI", "pointer", "obId", "outDetail", "outMessg",  
"outcome", "ApiField", "Collections", "Origin", "SedaField", "ShortName", "Format",  
"Path", "RuleDescription", "RuleDuration", "RuleId", "RuleMeasurement", "RuleType",  
"RuleValue", "UpdateDate", "FullAccess", "Permissions", "ud", "OperationId",  
"GlobalStatus", "DestroyableOriginatingAgencies", "NonDestroyableOriginatingAgencies",  
"ExtendedInfoType", "ParentUnitId", "OriginatingAgenciesInConflict", "_sedaVersion",  
"_implementationVersion", "Compressed", "RawMetadata", "BinaryObjectSize",  
"BinaryObject", "ObjectGroup", "ArchiveUnit", "FilteredExtractedObjectGroupData",  
"FilteredExtractedUnitData", "CheckParentId", "ArchivalProfile", "DataObjectSystemId",  
"DataObjectGroupSystemId", "_validComputedInheritedRules", "MaxEndDate",  
"indexationDate", "UnitStrategy", "ObjectGroupStrategy", "ObjectStrategy"],  
  
  "createdOntologies": ["MyKeyword", "MyText", "MyDate", "MyBoolean", "MyLong",  
"MyDouble", "MyGeoPoint", "MyEnum"]  
}
```

3.13.Workflow d'administration d'un référentiel des griffons

Cette section décrit le processus permettant d'importer des griffons. Cette opération n'est réalisable que sur le tenant d'administration.

Toutes les étapes, tâches et traitements sont journalisées dans le journal des opérations et décrivent le processus (clé et description de la clé associée dans le journal des opérations) tel qu'implémenté dans la version actuelle de la solution logicielle Vitam.

3.13.1.Processus d'import et mise à jour des griffons (STP_IMPORT_GRIFFIN)

Le processus d'import d'un référentiel des griffons permet d'ajouter des griffons qui seront utilisés dans les opérations de préservation. Tous les éléments réalisés au cours de ce processus sont exécutés dans une seule étape.

- **Règle** : opération consistant à vérifier la présence des informations minimales dans le référentiel des griffons, à s'assurer de la cohérence des informations saisies, à affecter des données aux champs peuplés par la solution logicielle Vitam et à enregistrer le référentiel des griffons.

Les données suivantes sont obligatoirement remplies :

- Le champ « Name » est peuplé d'une chaîne de caractères ;
- Le champ « Identifier » est peuplé d'une chaîne de caractères devant correspondre à un identifiant signifiant donné au griffon ;
- Le champ « ExecutableName » est peuplé d'une chaîne de caractères devant correspondre au nom technique du griffon utilisé pour lancer l'exécutable dans le système ;
- Le champ « ExecutableVersion » est peuplé d'une valeur devant correspondre à la version du griffon utilisé. Un même exécutable (ExecutableName) peut être associé à plusieurs versions.
- Les données suivantes optionnelles si elles sont remplies le sont en respectant les règles énoncées pour chacune :
 - Le champ « Description » doit être une chaîne de caractères,
 - Le champ « CreationDate » doit être une date est au format ISO 8601,
 - Le champ « LastUpdate » doit être une date est au format ISO 8601,
 - Le champ « _tenant » doit être un entier,
 - Le champ « _v » doit être un entier.

La modification d'un référentiel des griffons s'effectue par ré-import du fichier Json. Le nouvel import annule et remplace le référentiel précédent. Ce ré-import observe les règles décrites dans le processus d'import.

3.13.1.1. Sauvegarde du JSON STP_BACKUP_GRIFFIN

- **Règle** : tâche consistant à enregistrer une copie de la base de données des métadonnées des griffons sur les offres de stockage
- **Type** : bloquant
- **Statuts** :
 - OK : une copie de la base de données nouvellement importée est enregistrée (STP_BACKUP_GRIFFIN.OK = Succès du processus de sauvegarde des griffons)
 - KO : pas de cas KO
 - FATAL : une erreur technique est survenue lors de la copie de la base de données (STP_BACKUP_GRIFFIN.FATAL = Erreur technique lors du processus de sauvegarde des griffons)

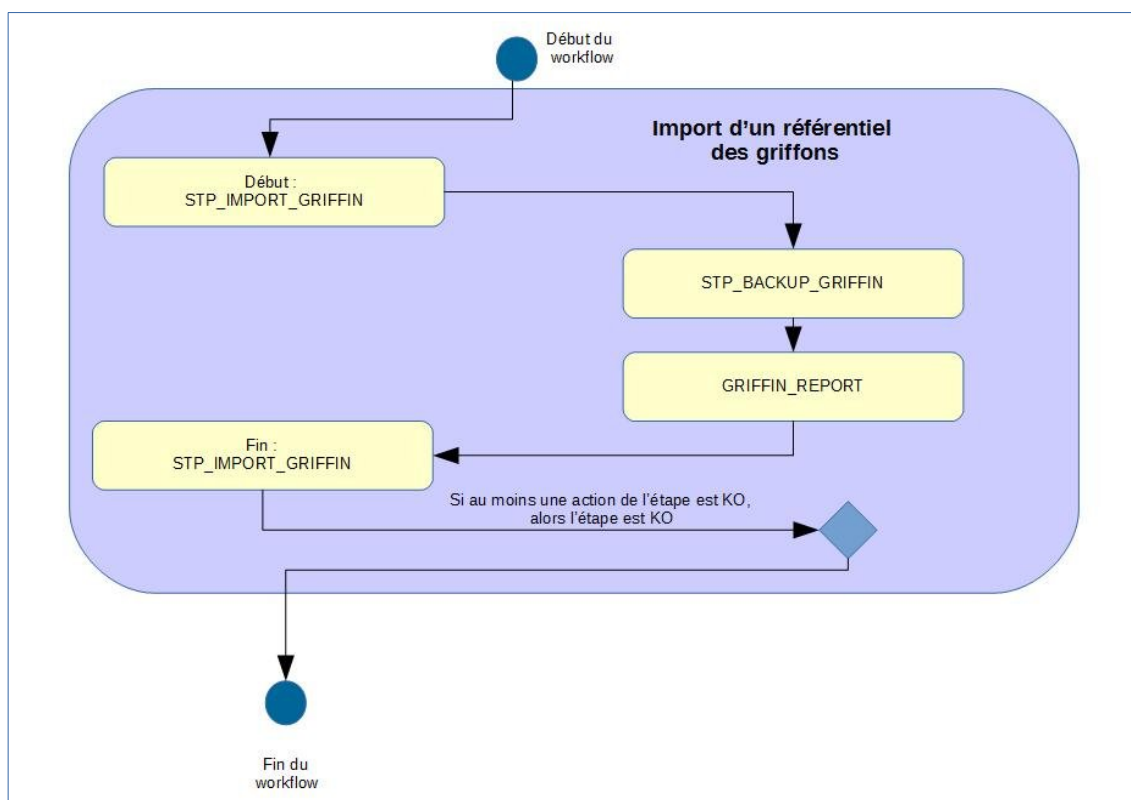
3.13.1.2. Processus de génération du rapport d'import du référentiel des griffons GRIFFIN_REPORT

- **Règle** : tâche consistant à créer le rapport d'import du référentiel des griffons

- **Type** : bloquant
- **Statuts** :
 - OK : le rapport d'import du référentiel des griffons a bien été créé (GRIFFIN_REPORT.OK = Succès du processus de génération du rapport d'import du référentiel des griffons)
 - KO : pas de KO
 - FATAL : une erreur technique est survenue lors de la création du rapport d'import du référentiel des griffons (GRIFFIN_REPORT.FATAL = Erreur technique lors du processus de génération du rapport d'import du référentiel des griffons)

3.13.2. Structure du Workflow d'import d'un référentiel des griffons

D'une façon synthétique, le workflow est décrit de cette façon :



3.13.3. Structure du rapport d'administration du référentiel des griffons

Lorsqu'un référentiel des griffons est importé ou mis à jour, la solution logicielle Vitam génère un rapport de l'opération.

Ce rapport est en plusieurs parties :

- « Operation » contient :
 - « evType » : le type d'opération. Dans le cadre de ce rapport, il s'agit toujours de « IMPORT_GRIFFIN » ;
 - « evDateTime » : la date et l'heure de l'opération d'import ;
 - « evId » : l'identifiant de l'opération ;
- « StatusCode » : le statut de l'opération OK, KO, WARNING ;
- « PreviousGriffinsVersion » : le numéro de la version précédemment installée dans le référentiel ;

- « PreviousGriffinCreationDate » : la date de la version précédemment installée dans le référentiel ;
- « NewGriffinsVersion » : le numéro de version désormais installée ;
- « RemovedIdentifiers » : la liste des griffons qui ont été supprimés ;
- « AddedIdentifiers » : la liste des griffons qui ont été ajoutés ;
- « UpdatedIdentifiers » : la liste des griffons qui ont été mis à jour ;
- « Warnings » : les messages d'avertissement.

Exemple :

```
{ "Operation" : { "evType" : "IMPORT_GRIFFIN", "evDateTime" : "2019-01-30T12:58:05.176", "evId" : "aeaaaaaaaaaghfj4pcabeloalit3lip3yaaaaaq"}, "StatusCode" : "WARNING", "PreviousGriffinsVersion" : "V1", "PreviousGriffinsCreationDate" : "2019-01-30T12:58:05.377", "NewGriffinsVersion" : "V1.0.0", "RemovedIdentifiers" : [ "GRIFFIN2", "GRIFFIN1", "GRI-000005", "GRIFFIN3" ], "AddedIdentifiers" : [ ], "UpdatedIdentifiers" : { "GRI-000002" : [ ], "GRI-000001" : [ ] }, "Warnings" : [ "4 identifiers removed." ] }
```

3.14.Workflow d'administration d'un référentiel des scénarios de préservation

Cette section décrit le processus permettant d'importer un référentiel des scénarios de préservation.

Toutes les étapes, tâches et traitements sont journalisées dans le journal des opérations et décrivent le processus (clé et description de la clé associée dans le journal des opérations) tel qu'implémenté dans la version actuelle de la solution logicielle Vitam.

3.14.1.Processus d'import et de mise à jour des scénarios de préservation (STP_IMPORT_SCENARIO)

Le processus d'import d'un référentiel de scénarios de préservation permet d'ajouter des scénarios de préservation qui seront utilisés dans les opérations de préservation. Tous les éléments réalisés au cours de ce processus sont exécutés dans une seule étape.

- **Règle** : opération consistant à vérifier la présence des informations minimales dans le référentiel des scénarios de préservation, à s'assurer de la cohérence des informations saisies, à affecter des données aux champs peuplés par la solution logicielle Vitam et à enregistrer le référentiel des scénarios de préservation.

Les données suivantes sont obligatoirement remplies :

- Le champ « Name » doit être peuplé avec une chaîne de caractères unique ;
- Le champ « Identifier » doit être unique ;
- Les champs « ActionList » et « Type » doivent avoir pour valeur « ANALYSE », « GENERATE », « IDENTIFY », « EXTRACT » ou « EXTRACT » ;
- Le champ « FormatList » doit avoir pour valeur une suite de chaînes de caractères correspondant à une valeur valide issue du champ « PUID » du référentiel des formats ;
- le champ « GriffinIdentifier » doit avoir pour valeur une chaîne de caractère correspondant à une valeur valide issue du champ « Identifier » du référentiel des griffons ;
- Les champs « Timeout » et « MaxSize » doivent avoir pour valeur un entier ;
- Le champ « Debug » doit être un booléen et avoir pour valeur « true » ou « false » ;
- Le champ « Extension » doit avoir pour valeur une chaîne de caractères ;
- Le champ « Args » doit avoir pour valeur une chaîne de caractères ;
- Les données suivantes sont facultatives ; si elles sont remplies, elles respectent les règles énoncées pour chacune :
 - Le champ « Description » est peuplé d'une chaîne de caractères.

Exemple :

```
[
{
  "Identifier": "PSC-000002",
  "Name": "Transformation en GIF MINI",
  "Description": "Ce scenario transforme une image JPEG en GIF mini",
  "CreationDate": "2018-11-16T15:55:30.721",
  "LastUpdate": "2018-11-20T15:34:21.542",
  "ActionList": [
    "GENERATE"
  ],
  "GriffinByFormat": [
    {
      "FormatList": ["fmt/41", "fmt/43"],
      "GriffinIdentifier": "GRI-000001",
      "TimeOut": 20,
      "MaxSize": 10000000,
      "Debug":true,
      "ActionDetail": [
        {
          "Type": "GENERATE",
```

```
        "Values": {
          "Extension": "GIF",
          "Args": [
            "-thumbnail",
            "100x100"
          ]
        }
      ]
    },
    "DefaultGriffin": null
  }
]
```

3.14.1.1. Sauvegarde du JSON STP_BACKUP_SCENARIO

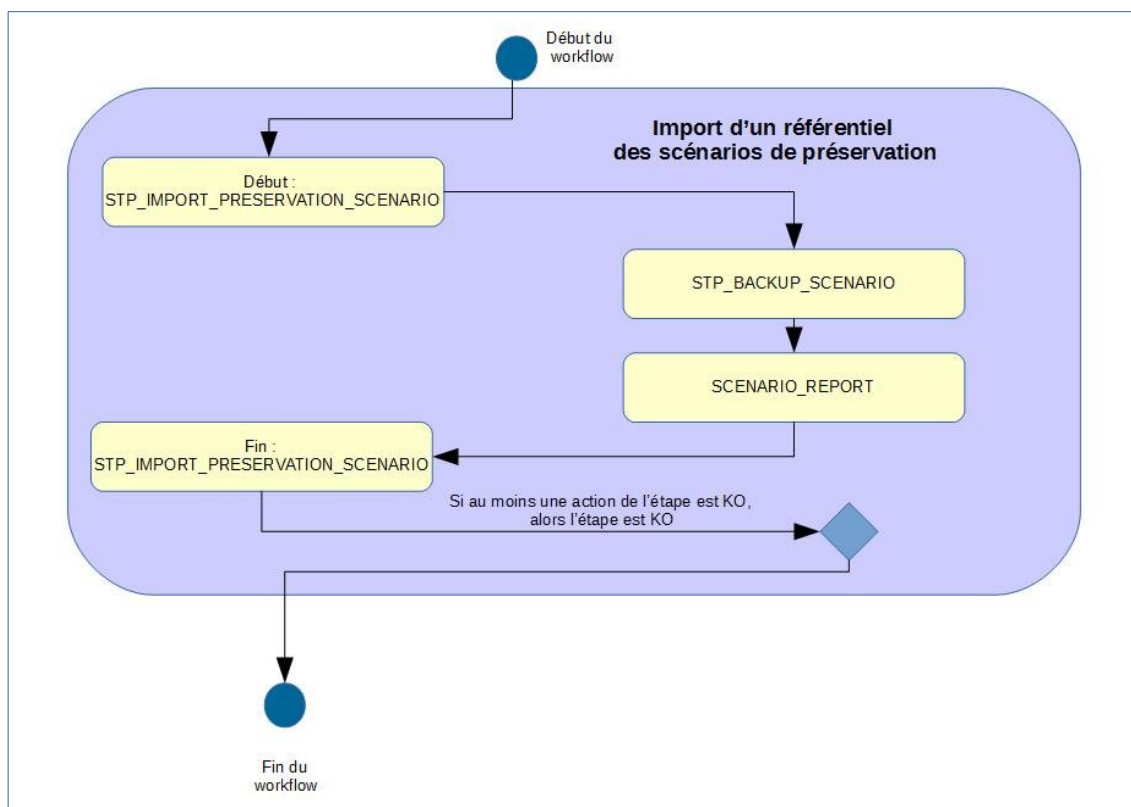
- **Règle** : tâche consistant à enregistrer une copie de la base de données des scénarios de préservation sur les offres de stockage
- **Type** : bloquant
- **Statuts** :
 - OK : une copie de la base de données nouvellement importée est enregistrée (STP_BACKUP_SCENARIO.OK = Succès du processus de sauvegarde des scénarios de préservation)
 - KO : pas de KO
 - FATAL : une erreur technique est survenue lors de la sauvegarde des scénarios de préservation (BACKUP_SCENARIO.FATAL = Erreur technique lors du processus de sauvegarde des scénarios de préservation)

3.14.1.2. Processus de génération du rapport d'import du référentiel des scénarios de préservation SCENARIO_REPORT

- **Règle** : tâche consistant à créer le rapport d'import du référentiel des scénarios de préservation
- **Type** : bloquant
- **Statuts** :
 - OK : le rapport d'import du référentiel des scénarios de préservation a bien été créé (SCENARIO_REPORT.OK = Succès du processus de génération du rapport d'import du référentiel des scénarios de préservation)
 - KO : pas de KO
 - FATAL : une erreur technique est survenue lors de la création du rapport d'import du référentiel des scénarios de préservation (SCENARIO_REPORT.FATAL = Erreur technique lors du processus de génération du rapport d'import du référentiel des scénarios de préservation)

3.14.2. Structure du Workflow d'import d'un référentiel des scénarios de préservation

D'une façon synthétique, le workflow est décrit de cette façon :



3.14.3. Structure du rapport d'administration du référentiel des scénarios de préservation

Lorsqu'un référentiel des scénarios de préservation est importé ou mis à jour, la solution logicielle Vitam génère un rapport de l'opération.

Ce rapport est en plusieurs parties :

- « Operation » contient :
 - « evType » : le type d'opération. Dans le cadre de ce rapport, il s'agit toujours de « STP_IMPORT_PRESERVATION_SCENARIO » ;
 - « evDateTime » : la date et l'heure de l'opération d'import ;
 - « evId » : l'identifiant de l'opération ;
- « StatusCode » : le statut de l'opération OK, KO, WARNING ;
- « PreviousScenariosCreationDate » : la date de la version précédemment installée dans le référentiel ;
- « NewScenariosCreationDate » : la date de la version installée dans le référentiel ;
- « RemovedIdentifiers » : la liste des scénarios qui ont été supprimés ;
- « AddedIdentifiers » : la liste des scénarios qui ont été ajoutés ;
- « UpdatedIdentifiers » : la liste des scénarios qui ont été mis à jour ;
- « Warnings » : les messages d'avertissement.

Exemple :

```

{ "Operation" : { "evType" : "IMPORT_GRIFFIN", "evDateTime" : "2019-01-30T12:58:05.176", "evId" : "aeaaaaaaaaaghfj4pcabeloalit3lip3yaaaaq" },
  "StatusCode" : "WARNING",

```



```
"PreviousScenariosCreationDate" : "2019-01-30T12:58:05.377",  
"NewScenariosCreationDate" : "2019-01-30T12:58:05.377",  
"RemovedIdentifiers" : [ "SC2", "SC1", "SCENARIO5", "SCENARIO3" ],  
"AddedIdentifiers" : [ ],  
"UpdatedIdentifiers" : {"SCE-000002" : [ ], "SCE-000001" : [ ]},  
"Warnings" : [ "4 identifiers removed." ]}
```