



# Description des workflows et des opérations

## Partie 3 : préservation et intégrité

Date	Version
27/01/2023	12.0 (Version 6.RC)

## 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	12/08/2022
Validation	AGR	Équipe Vitam	27/01/2023

## 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	<p>Mise à jour pour tenir compte des fonctionnalités mises en œuvre pendant la <i>Release 10</i> :</p> <ul style="list-style-type: none"> <li>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.</li> <li>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.</li> <li>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 ».</li> </ul>
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	<p>Mise à jour pour tenir compte des fonctionnalités mises en œuvre pendant la <i>Release 11</i> :</p> <ul style="list-style-type: none"> <li>chapitre 2, section V (« Rapport de l'audit de cohérence ») : mise à jour du rapport ;</li> <li>chapitre 11, section II (« Rapport de préservation ») : création de la section.</li> </ul>
3.2	02/09/2019	JPP	<p>Mise à jour pour tenir des fonctionnalités mises en œuvre pendant la Release 11 :</p> <ul style="list-style-type: none"> <li>chapitre 8, section III et IV.</li> </ul>
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

<b>6.0</b>	20/03/2020	AGR	Séparation du document en 2 parties et finalisation pour publication Release 13
<b>6.1</b>	24/06/2020	MVI	Mise à jour pour tenir des fonctionnalités mises en œuvre pendant la Release 14 : <ul style="list-style-type: none"> <li>chapitre 4, section 4.1 (« Workflow d'entrée ») : ajout des tâches CHECK_OBJECT_SIZE et CHECK_ATTACHEMENT ;</li> </ul>
<b>7.0</b>	06/07/2020	AGR	Finalisation pour publication Release 14
<b>7.1</b>	26/10/2020	MVI	Mise à jour pour tenir des fonctionnalités mises en œuvre pendant la Release 15 : <ul style="list-style-type: none"> <li>chapitre 2, section 2.2 (« Workflow d'audit de vérification des journaux sécurisés ») : ajout de la section</li> <li>chapitre 5, section 5.2.3 (« Structure du rapport du référentiel des règles de gestion ») : mise à jour de la modélisation du rapport.</li> </ul>
<b>8.0</b>	26/10/2020	AGR	Finalisation pour publication Release 15
<b>8.1</b>	23/02/2021	MVI	Mise à jour pour tenir des fonctionnalités mises en œuvre pendant la Release 16 : <ul style="list-style-type: none"> <li>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).</li> </ul>
<b>9.0</b>	15/03/2021	AGR	Finalisation pour publication Release 16
<b>9.1</b>	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"> <li>partitionnement de la documentation avec une nouvelle répartition : <ul style="list-style-type: none"> <li>partie 1 : référentiels</li> <li>partie 2 : gestion du cycle de vie des archives</li> <li>partie 3 : préservation et intégrité</li> </ul> </li> </ul> <p>Le présent document couvre les opérations liées à la préservation et à l'intégrité.</p> <ul style="list-style-type: none"> <li>chapitre 4, section 4.1 (« Workflow de suppression d'objets ») : ajout du workflow de suppression des versions d'objets techniques.</li> </ul>
<b>10.0</b>	27/09/2021	AGR	Finalisation pour publication Version 5.RC
<b>10.1</b>	22/02/2022	MVI	Mise à jour pour tenir des fonctionnalités mises en œuvre pendant la Version 5 : <ul style="list-style-type: none"> <li>chapitre 13 (« Traceability ») : ajout de la vérification du tampon d'horodatage ;</li> <li>chapitre 14, section 14.5 (« Workflow de l'audit correctif ») : ajout d'une vérification de la disponibilité des ressources ;</li> <li>chapitre 15 (« Préservation ») : ajout d'une vérification de la disponibilité des ressources.</li> </ul>
<b>11.0</b>	04/03/2022	AGR	Finalisation pour publication Version 5
<b>11.1</b>	12/08/2022	MVI	Relecture
<b>12.0</b>	27/01/2023	AGR	Finalisation pour publication Version 6.RC

## 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

Introduction.....	9
Objectif du document.....	9
Description d'un processus.....	9
Structure d'un fichier Properties du Workflow.....	10
13. Traceability.....	13
13.1. Workflow du processus de sécurisation du journal des opérations.....	13
13.1.1. Processus de sécurisation des journaux des opérations (STP_OP_SECURISATION).....	13
13.1.2. Structure de workflow de sécurisation du journal des opérations.....	14
13.2. Workflow de sécurisation des journaux du cycle de vie des groupes d'objets.....	15
13.2.1. Sécurisation des journaux du cycle de vie des groupes d'objets (LOGBOOK_OBJECTGROUP_LFC_TRACEABILITY).....	15
13.2.2. Processus de la sécurisation des journaux du cycle de vie des groupes d'objets STP_OG_LFC_TRACEABILITY.....	15
13.2.3. Structure du workflow du processus de sécurisation des journaux des cycles de vie des groupes d'objets.....	17
13.3. Workflow de sécurisation des journaux du cycle de vie des unités archivistiques.....	18
13.3.1. Sécurisation des journaux du cycle de vie des unités archivistiques (LOGBOOK_UNIT_LFC_TRACEABILITY).....	18
13.3.2. Processus de la sécurisation des journaux du cycle de vie des unités archivistiques STP_UNIT_LFC_TRACEABILITY.....	18
13.3.3. Structure du workflow du processus de sécurisation des journaux des cycles de vie des unités archivistiques.....	20
13.4. Workflow de sécurisation des journaux des écritures.....	21
13.4.1. Processus de sécurisation des journaux des écritures (STP_STORAGE_SECURISATION).....	21
13.4.2. Structure du workflow du processus de sécurisation des journaux des écritures.....	22
13.5. Sauvegarde des journaux des écritures.....	23
13.5.1. Processus de sauvegarde des journaux des écritures (STORAGE_BACKUP).....	23
13.5.2. Structure du workflow du processus de sauvegarde des journaux des écritures.....	24
13.6. Sauvegarde des logs des accès.....	24
13.6.1. Processus de sauvegarde des logs des accès (STORAGE_ACCESS_BACKUP).....	24
13.6.2. Structure du workflow du processus de sauvegarde des logs des accès.....	25
14. Audit.....	25
14.1. Workflow de contrôle d'intégrité d'un journal sécurisé.....	25
14.1.1. Processus de contrôle d'intégrité d'un journal sécurisé (vision métier).....	25
14.1.2. Processus de préparation de la vérification des journaux sécurisés (STP_PREPARE_TRACEABILITY_CHECK).....	26
14.1.3. Processus de vérification de l'arbre de Merkle (STP_MERKLE_TREE).....	26
14.1.4. Processus de vérification de l'horodatage (STP_VERIFY_STAMP).....	27
14.1.5. Structure du workflow de contrôle d'intégrité d'un journal sécurisé.....	29
14.2. Workflow d'audit de vérification des journaux sécurisés.....	30
14.2.1. Processus d'audit de vérification des journaux sécurisés (vision métier).....	30
14.2.2. Processus de préparation de la vérification des journaux sécurisés (STP_PREPARE_TRACEABILITY_LINKED_CHECK).....	30
14.2.3. Processus de la vérification des journaux sécurisés (STP_TRACEABILITY_LINKED_CHECKS).....	31
14.2.4. Processus de la finalisation de la vérification des journaux sécurisés STP_FINALIZE_TRACEABILITY_LINKED_CHECKS.....	35
14.2.5. Audit de la vérification des journaux sécurisés.....	36
LINKED_CHECK_SECURISATION.....	36

14.2.6. Structure du workflow.....	36
14.3. Workflow de l’audit de l’existence et de l’intégrité des fichiers.....	39
14.3.1. Processus d’audit d’existence des fichiers (vision métier).....	39
14.3.2. Processus d’audit d’intégrité des fichiers (vision métier).....	39
14.3.3. Processus de préparation de l’audit (STP_PREPARE_AUDIT).....	39
14.3.4. Processus d’exécution de l’audit (STP_AUDIT).....	40
14.3.5. Processus de finalisation de l’audit (STP_FINALISE_AUDIT).....	41
14.3.6. Structure du workflow de l’audit de l’existence et de l’intégrité des fichiers.....	42
14.3.7. Rapport d’audit.....	43
14.4. Workflow d’audit de cohérence des fichiers.....	49
14.4.1. Processus d’audit de cohérence des fichiers (vision métier).....	49
14.4.2. Processus de préparation d’audit (STP_EVIDENCE_AUDIT_PREPARE).....	49
14.4.3. Processus de récupération des données de la base (STP_EVIDENCE_AUDIT_CHECK_DATABASE).....	50
14.4.4. Processus de préparation des signatures à partir des fichiers sécurisés (STP_EVIDENCE_AUDIT_LIST_SECURED_FILES_TO_DOWNLOAD).....	50
14.4.5. Processus d’extraction des signatures à partir des fichiers sécurisés (STP_EVIDENCE_AUDIT_EXTRACT_ZIP_FILE).....	51
14.4.6. Processus de préparation des rapports pour chaque objet, groupe d’objets ou unité auditée (STP_EVIDENCE_AUDIT_PREPARE_GENERATE_REPORTS).....	51
14.4.7. Processus de finalisation d’audit et génération du rapport final (STP_EVIDENCE_AUDIT_FINALIZE).....	52
14.4.8. Structure du workflow d’audit de cohérence des fichiers.....	53
14.4.9. Rapport d’audit de cohérence.....	55
14.5. Workflow de l’audit correctif.....	58
14.5.1. Processus d’audit correctif des fichiers (vision métier).....	58
14.5.2. Processus de préparation d’audit (STP_EVIDENCE_AUDIT_PREPARE).....	58
14.5.3. Processus de récupération des données de la base (STP_EVIDENCE_AUDIT_CHECK_DATABASE).....	59
14.5.4. Processus de préparation des signatures à partir des fichiers sécurisés (STP_EVIDENCE_AUDIT_LIST_SECURED_FILES_TO_DOWNLOAD).....	59
14.5.5. Processus d’extraction des signatures à partir des fichiers sécurisés (STP_EVIDENCE_AUDIT_EXTRACT_ZIP_FILE).....	60
14.5.6. Processus de préparation des rapports pour chaque objet, groupe d’objets ou unité auditée (STP_EVIDENCE_AUDIT_PREPARE_GENERATE_REPORTS).....	60
14.5.7. Processus de correction des signatures pour chaque objet, groupe d’objets ou unité auditée défaillante (STP_CORRECTIVE_AUDIT).....	61
14.5.8. Processus de finalisation des corrections des signatures pour chaque objet, groupe d’objets ou unité auditée défaillante (STP_CORRECTION_FINALIZE).....	62
14.5.9. Structure du workflow de correction suite à audit.....	63
14.5.10. Rapport d’audit correctif.....	65
14.6. Workflow de relevé de valeur probante.....	66
14.6.1. Processus de préparation du relevé de valeur probante (STP_PROBATIVE_VALUE_PREPARE).....	66
14.6.2. Processus d’alimentation du relevé de valeur probante (STP_PROBATIVE_VALUE_ACTION).....	67
14.6.3. Processus de finalisation du relevé de valeur probante (STP_PROBATIVE_VALUE_GENERATE_REPORT).....	67
14.6.4. Structure de workflow du relevé de valeur probante.....	68
14.7. Rapport du relevé de valeur probante.....	69
14.7.1. Exemple de JSON : rapport de valeur probante.....	69
14.7.2. Détails du rapport.....	88
15. Préservation.....	90
15.1. Processus de préparation du traitement de la préservation (STP_PREPARATION).....	90
15.1.1. Vérification des seuils de limitation des traitements CHECK_DISTRIBUTION_THRESHOLD	

(CheckDistributionThreshold.java).....	90
15.1.2. Préparation du traitement de préservation PRESERVATION_PREPARATION (PreservationPreparationPlugin.java).....	90
15.2. Processus de lancement du griffon de la préservation (STP_PRESERVATION_ACTION).....	91
15.2.1. Vérification de la disponibilité des ressources PRESERVATION_CHECK_RESOURCE_AVAILABILITY (PreservationCheckResourceAvailability.java) .....	91
15.2.2. Lancement du griffon PRESERVATION_ACTION (PreservationActionPlugin.java).....	91
15.2.3. Identification du format des objets binaires PRESERVATION_SIEGFRIED_IDENTIFICATION (PreservationSiegfriedPlugin.java).....	91
15.2.4. Calcul de l’empreinte des nouveaux objets binaires PRESERVATION_BINARY_HASH (PreservationGenerateBinaryHash.java).....	91
15.2.5. Écriture des objets binaires générés sur les offres de stockage PRESERVATION_STORAGE_BINARY (PreservationStorageBinaryPlugin.java).....	92
15.2.6. Vérification de la sécurité des unités archivistiques PRESERVATION_OBJECTGROUP_METADATA_SECURITY_CHECKS.....	92
15.2.7. Indexation des métadonnées de préservation PRÉSERVATION_INDEXATION_METADATA (PreservationUpdateObjectGroupPlugin.java).....	92
15.2.8. Sauvegarde des métadonnées et des journaux de cycle de vie sur les offres de stockage PRESERVATION_STORAGE_METADATA_LFC (PreservationStorageMetadataAndLfc.java).....	93
15.2.9. Vérification de la sécurité des unités archivistiques UNIT_METADATA_SECURITY_CHECKS. .	93
15.2.10. Extraction des métadonnées descriptives au moyen du griffon PRESERVATION_EXTRACTION_AU.....	94
15.3. Processus de préparation de l’insertion de métadonnées descriptives (STP_PRESERVATION_PREPARATION_INSERTION_AU_METADATA) (PreservationPreparationInsertionAuMetadata.java).....	94
15.4. Processus d’insertion de métadonnées descriptives dans les unités archivistiques (STP_PRESERVATION_INSERTION_AU_METADATA).....	94
15.5. Processus d’alimentation du registre des fonds (STP_ACCESSION_REGISTRATION).....	95
15.6. Processus de création du rapport de préservation (STP_PRESERVATION_FINALIZATION). .	96
15.7. Structure du workflow de préservation.....	97
15.8. Rapport de préservation.....	101
15.8.1. Exemple de fichier JSONL : rapport de préservation OK.....	102
15.8.2. Détail du rapport.....	104
16. Suppression des versions des GOT.....	107
16.1. Processus de préparation de la suppression des versions des GOT (STP_DELETE_GOT_VERSIONS_PREPARATION ).....	107
16.1.1. Vérification des seuils de la suppression des versions des GOT (CHECK_DISTRIBUTION_THRESHOLD :CheckDistributionThreshold.java).....	107
16.1.2. Préparation de la suppression des versions des GOT (DELETE_GOT_VERSIONS_PREPARATION :DeleteGotVersionsPreparationPlugin.java).....	108
16.2. Processus de suppression des versions des GOT (STP_DELETE_GOT_VERSIONS_ACTION ) (Distribution sur LIST GUID/distributionFileOG.jsonl.....	108
16.2.1. Action de suppression (DELETE_GOT_VERSIONS_ACTION : DeleteGotVersionsActionPlugin.java).....	108
16.2.2. Stockage des groupes d’objets dans workspace (STORE_METADATA_AND_LFC_WORKSPACE : DeleteGotVersionsStoreMetadataAndLfcPlugin.java) .....	108
16.2.3. Suppression des groupes d’objets depuis les offres (DELETE_GOT_VERSIONS_STORAGE : DeleteGotVersionsStoragePlugin.java).....	109
16.3. Processus de mise à jour du registre des fonds	

(STP_DELETE_GOT_VERSIONS_ACCESSION_REGISTER_UPDATE).....	109
Alimentation du Registre des Fonds (DELETE_GOT_VERSIONS_ACCESSION_REGISTER_UPDATE : DeleteGotVersionsAccessionRegisterUpdatePlugin.java).....	110
16.4. Processus de finalisation de la suppression des versions des groupes d'objets (STP_DELETE_GOT_VERSIONS_FINALIZATION).....	110
16.5. Structure du Workflow de suppression des versions des GOTS.....	111
16.6. Rapport de suppression des versions des groupes d'objets.....	113
16.6.1. Exemple de JSON : rapport de suppression des versions des got.....	113
16.6.2. Détail du rapport.....	113



## 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.*

### 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 :

- Traçabilité
- Audits
- Préservation
- Suppression de versions d'objets techniques

### 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.

## 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 éventuels 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*

## 13. TRACEABILITY

### 13.1. Workflow du processus de sécurisation du journal des opérations

Cette section décrit le processus (workflow) de sécurisation des journaux mis en place dans la solution logicielle Vitam pour le journal des opérations.

Celui-ci est défini dans le fichier « LogbookAdministration.java » situé ici : `sources/logbook/logbook-administration/src/main/java/fr/gouv/vitam/logbook/administration/core/`

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.

#### 13.1.1. Processus de sécurisation des journaux des opérations (STP\_OP\_SECURISATION)

Le processus de sécurisation des journaux consiste en la création d'un fichier .zip contenant l'ensemble des journaux à sécuriser, ainsi que le tampon d'horodatage calculé à partir de l'arbre de Merkle de la liste de ces mêmes journaux. Les journaux concernés par cette sécurisation sont le journal des opérations et le journal des écritures.

Ce fichier zip est ensuite enregistré sur les offres de stockage, en fonction de la stratégie de stockage.

- **Règle** : opération consistant à sécuriser le journal des opérations
- **Type** : bloquant
- **Statuts** :
  - OK : le journal des opérations a été sécurisé (STP\_OP\_SECURISATION.OK = Succès du processus de sécurisation du journal des opérations)
  - KO : pas de cas KO
  - FATAL : une erreur technique est survenue lors de la sécurisation du journal des opérations (STP\_OP\_SECURISATION.FATAL = Erreur technique lors du processus de sécurisation du journal des opérations)

##### 13.1.1.1. Calcul du tampon d'horodatage OP\_SECURISATION\_TIMESTAMP (LogbookAdministration.java)

- **Règle** : tâche consistant à calculer le tampon d'horodatage à partir de la racine de l'arbre de Merkle constitué de la liste des journaux qui sont en train d'être sécurisés
- **Type** : bloquant
- **Statuts** :
  - OK : le tampon d'horodatage est calculé (OP\_SECURISATION\_TIMESTAMP.OK = Succès de la création du tampon d'horodatage de l'ensemble des journaux)
  - FATAL : une erreur technique est survenue lors de l'horodatage (OP\_SECURISATION\_TIMESTAMP.FATAL = Erreur technique lors de la création du tampon d'horodatage de l'ensemble des journaux)

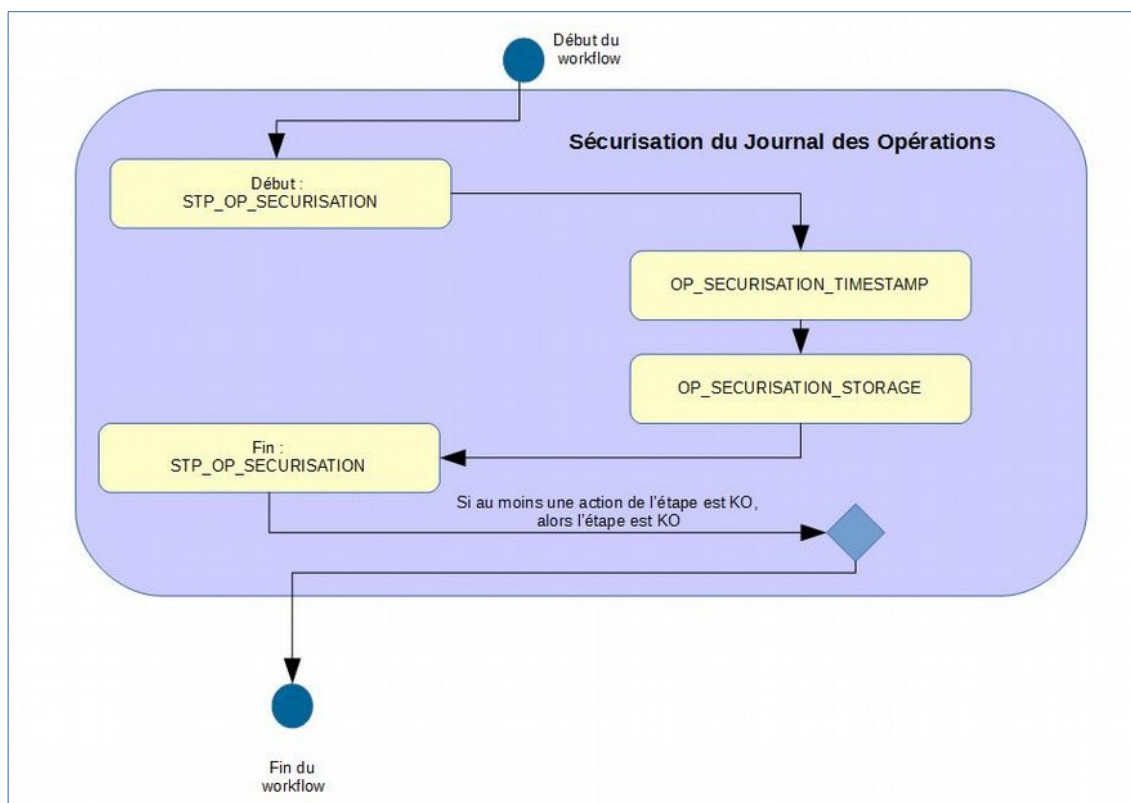
##### 13.1.1.2. Stockage des journaux sécurisés (OP\_SECURISATION\_STORAGE) (LogbookAdministration.java)

- **Règle** : tâche consistant à écrire les journaux sécurisés sur les offres de stockage, en fonction de la stratégie de stockage
- **Type** : bloquant
- **Statuts** :

- OK : le journal sécurisé est écrit sur les offres de stockage (OP\_SECURISATION\_STORAGE.OK = Succès de l'enregistrement des journaux sur les offres de stockage)
- FATAL : une erreur technique est survenue lors de l'écriture du journal sécurisé (OP\_SECURISATION\_STORAGE.FATAL = Erreur technique lors de l'enregistrement des journaux sur les offres de stockage)

### 13.1.2. Structure de workflow de sécurisation du journal des opérations

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



## 13.2.Workflow de sécurisation des journaux du cycle de vie des groupes d'objets

Cette section décrit le processus (workflow) permettant la sécurisation des journaux du cycle de vie des groupes d'objets mis en place dans la solution logicielle Vitam. Le workflow mis en place dans la solution logicielle Vitam est défini dans le fichier « DefaultObjectGroupLifecycleTraceability.json ». Ce fichier est disponible dans : sources/processing/processing-management/src/main/resources/workflows.

*Note : Le traitement permettant la sécurisation des journaux du cycle de vie procède par des tranches de lots de 100 K. La solution Vitam à la fin de ce premier lot enclenche un autre traitement de 100K et ce jusqu'à avoir traités l'ensemble des groupes d'objets.*

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.

### 13.2.1.Sécurisation des journaux du cycle de vie des groupes d'objets (LOGBOOK\_OBJECTGROUP\_LFC\_TRACEABILITY)

Le processus de sécurisation des journaux du cycle de vie consiste en la création d'un fichier .zip contenant l'ensemble des journaux du cycle de vie à sécuriser, ainsi que le tampon d'horodatage.

Ce fichier zip est ensuite enregistré sur les offres de stockage, en fonction de la stratégie de stockage.

- **Règle** : opération consistant à sécuriser les journaux du cycle de vie des groupes d'objets
- **Type** : bloquant
- **Statuts** :
  - OK : les journaux du cycle de vie ont été sécurisés (LOGBOOK\_OBJECTGROUP\_LFC\_TRACEABILITY.OK = Succès de la sécurisation des journaux du cycle de vie des groupes d'objets)
  - WARNING : il n'y pas de nouveaux journaux à sécuriser depuis la dernière sécurisation (LOGBOOK\_OBJECTGROUP\_LFC\_TRACEABILITY.WARNING = Avertissement lors de la sécurisation des journaux du cycle de vie des groupes d'objets)
  - KO : pas de cas KO
  - FATAL : une erreur technique est survenue lors de la sécurisation des journaux du cycle de vie des groupes d'objets (LOGBOOK\_OBJECTGROUP\_LFC\_TRACEABILITY.FATAL = Erreur technique lors de la sécurisation des journaux du cycle de vie des groupes d'objets)

### 13.2.2.Processus de la sécurisation des journaux du cycle de vie des groupes d'objets STP\_OG\_LFC\_TRACEABILITY

#### 13.2.2.1. Vérification des processus concurrents CHECK\_CONCURRENT\_WORKFLOW\_LOCK

- **Règle** : tâche consistant à vérifier s'il n'y a pas d'autres processus de traçabilité des journaux de cycle de vie des groupes d'objets concurrents
- **Type** : bloquant
- **Statuts** :
  - OK : le contrôle de processus de traçabilité des journaux du cycle de vie des groupes d'objets concurrents s'est terminé avec succès (CHECK\_CONCURRENT\_WORKFLOW\_LOCK.OK = Succès de la vérification des processus concurrents)
  - KO : des processus concurrents de traçabilité des groupes d'objets sont en cours d'exécution (CHECK\_CONCURRENT\_WORKFLOW\_LOCK.KO = Échec lors de la vérification des processus concurrents)

- FATAL : une erreur technique est survenue lors de la vérification des processus concurrents (CHECK\_CONCURRENT\_WORKFLOW\_LOCK.FATAL = Erreur technique lors de la vérification des processus concurrents)

### 13.2.2.2. Préparation de la liste des journaux du cycle de vie et des métadonnées des groupes d'objets PREPARE\_OG\_LFC\_TRACEABILITY

- **Règle** : tâche consistant à récupérer les journaux du cycle de vie à sécuriser et récupération des informations concernant les dernières opérations de sécurisation
- **Type** : bloquant
- **Statuts** :
  - OK : les fichiers des journaux du cycle de vie ont été exportés (dans ObjectGroup) ainsi que les informations concernant les dernières opérations de sécurisation (PREPARE\_OG\_LFC\_TRACEABILITY.OK = Succès de la préparation des journaux du cycle de vie et des métadonnées des groupes d'objets)
  - KO : les informations sur la dernière opération de sécurisation n'ont pas pu être obtenues / exportées, ou un problème a été rencontré avec un journal du cycle de vie (PREPARE\_OG\_LFC\_TRACEABILITY.KO = Échec de la préparation des journaux du cycle de vie et des métadonnées des groupes d'objets)
  - FATAL : une erreur technique est survenue (PREPARE\_LC\_TRACEABILITY.FATAL = Erreur technique lors de la préparation des journaux du cycle de vie et des métadonnées des groupes d'objets)

### 13.2.2.3. Sécurisation des journaux du cycle de vie des groupes d'objets BUILD\_OG\_LFC\_TRACEABILITY

- **Règle** : tâche consistant à appliquer l'algorithme pour créer les fichiers sécurisés des journaux du cycle de vie des groupes d'objets, journal par journal, et à générer le fichier sécurisé
- **Type** : bloquant
- **Statuts** :
  - OK : le fichier sécurisé pour le journal du cycle de vie en cours a été généré (BUILD\_OG\_LFC\_TRACEABILITY.STARTED.OK = Succès de la sécurisation des journaux du cycle de vie des groupes d'objets)
  - WARNING : il n'y a pas de nouveaux journaux à sécuriser (BUILD\_OG\_LFC\_TRACEABILITY.WARNING = Avertissement lors de la sécurisation des journaux du cycle de vie des groupes d'objets)
  - KO : le fichier pour le groupe d'objet n'a pas pu être trouvé (BUILD\_OG\_LFC\_TRACEABILITY.KO = Échec de la sécurisation des journaux du cycle de vie des groupes d'objets)
  - FATAL : une erreur technique est survenue lors de la génération des fichiers sécurisés (BUILD\_OG\_LFC\_TRACEABILITY.FATAL = Erreur technique lors de la sécurisation des journaux du cycle de vie des groupes d'objets)

### 13.2.2.4. Finalisation de la sécurisation des journaux du cycle de vie des groupes d'objets FINALIZE\_OG\_LFC\_TRACEABILITY

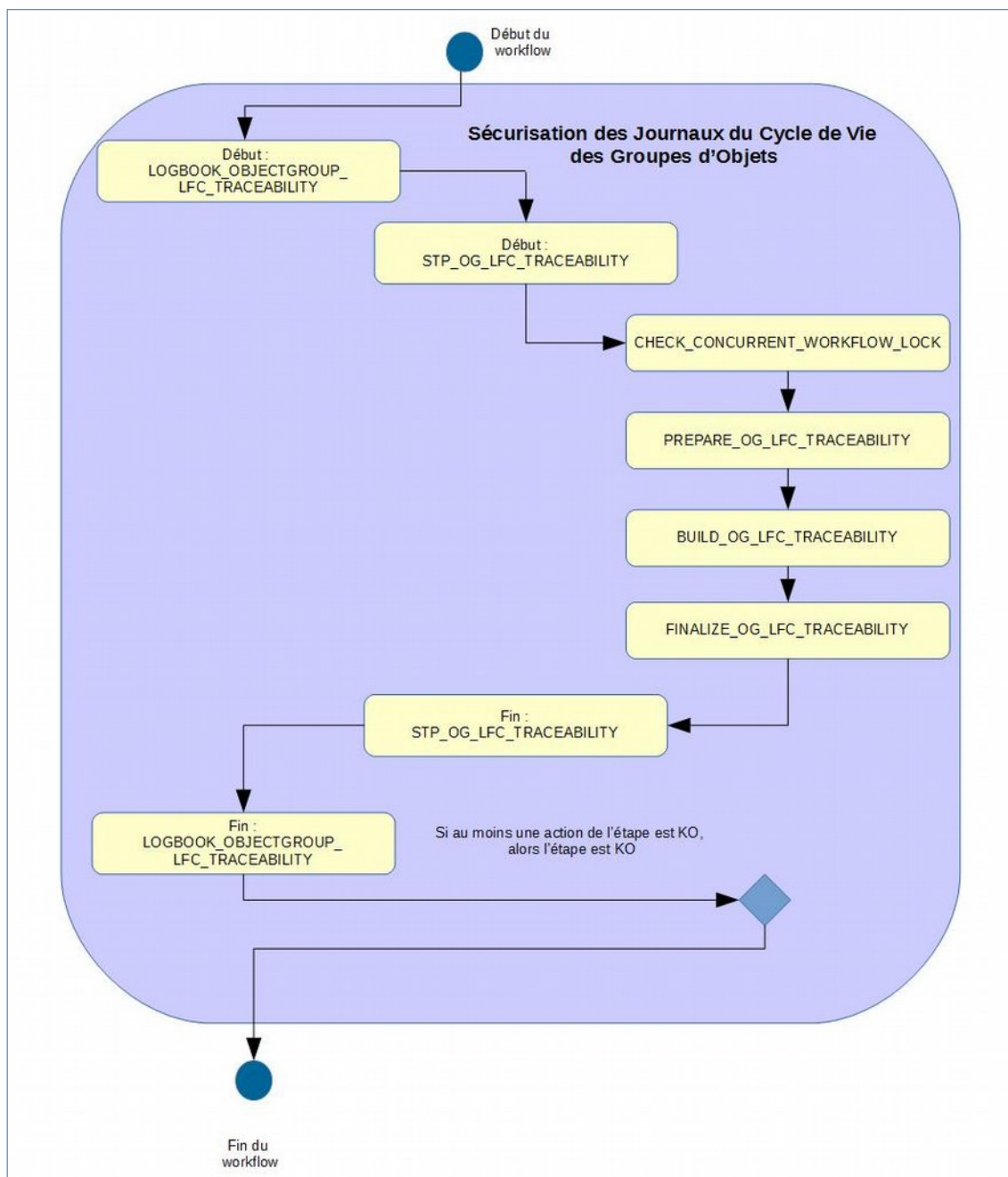
- **Règle** : tâche consistant à récupérer les différents fichiers générés puis à calculer le tampon d'horodatage
- **Type** : non applicable
- **Statuts** :
  - OK : la finalisation de la sécurisation des journaux du cycle de vie des groupes d'objets a bien été effectuée (FINALIZE\_OG\_LFC\_TRACEABILITY.OK = Succès de la finalisation de la sécurisation des journaux du cycle de vie des groupes d'objets)
  - KO : la finalisation de la sécurisation des journaux du cycle de vie des groupes d'objets n'a pas été effectuée (FINALIZE\_OG\_LFC\_TRACEABILITY.KO = Échec de la finalisation de la sécurisation des journaux du cycle de vie des groupes d'objets)



- FATAL : une erreur technique est survenue lors de la finalisation de la la sécurisation des journaux du cycle de vie des groupes d'objets (FINALIZE\_OG\_LFC\_TRACEABILITY.FATAL = Erreur technique lors de la finalisation de la sécurisation des journaux du cycle de vie des groupes d'objets)

### 13.2.3. Structure du workflow du processus de sécurisation des journaux des cycles de vie des groupes d'objets

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



### 13.3. Workflow de sécurisation des journaux du cycle de vie des unités archivistiques

Cette section décrit le processus (workflow) permettant la sécurisation des journaux du cycle de vie des unités archivistiques mis en place dans la solution logicielle Vitam des unités archivistiques. Le workflow mis en place dans la solution logicielle Vitam est défini dans le fichier « DefaultUnitLifecycleTraceability.json ». Ce fichier est disponible dans : sources/processing/processing-management/src/main/resources/workflows.

*Note : Le traitement permettant la sécurisation des journaux du cycle de vie procède par des tranches de lots de 100 K. La solution Vitam à la fin de ce premier lot enclenche un autre traitement de 100 K et ce jusqu'à avoir traité l'ensemble des unités archivistiques.*

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.

#### 13.3.1. Sécurisation des journaux du cycle de vie des unités archivistiques (LOGBOOK\_UNIT\_LFC\_TRACEABILITY)

Le processus de sécurisation des journaux du cycle de vie consiste en la création d'un fichier .zip contenant l'ensemble des journaux du cycle de vie à sécuriser, ainsi que le tampon d'horodatage.

Ce fichier zip est ensuite enregistré sur les offres de stockage, en fonction de la stratégie de stockage.

- **Règle** : opération consistant à sécuriser les journaux du cycle de vie des unités archivistiques
- **Type** : bloquant
- **Statuts** :
  - OK : les journaux du cycle de vie ont été sécurisés (LOGBOOK\_UNIT\_LFC\_TRACEABILITY.OK = Succès de la sécurisation des journaux du cycle de vie des unités archivistiques)
  - WARNING : il n'y pas de nouveaux journaux à sécuriser depuis la dernière sécurisation (LOGBOOK\_UNIT\_LFC\_TRACEABILITY.WARNING = Avertissement lors de la sécurisation des journaux du cycle de vie des unités archivistiques)
  - KO : pas de cas KO
  - FATAL : une erreur technique est survenue lors de la sécurisation du journal des opérations (LOGBOOK\_UNIT\_LFC\_TRACEABILITY.FATAL = Erreur technique lors de la sécurisation des journaux du cycle de vie des unités archivistiques)

#### 13.3.2. Processus de la sécurisation des journaux du cycle de vie des unités archivistiques STP\_UNIT\_LFC\_TRACEABILITY

##### 13.3.2.1. Vérification des processus concurrents CHECK\_CONCURRENT\_WORKFLOW\_LOCK

- **Règle** : tâche consistant à vérifier s'il n'y a pas d'autres processus de traçabilité des journaux du cycle de vie des unités archivistiques concurrents.
- **Type** : bloquant
- **Statuts** :
  - OK : le contrôle de processus de traçabilité des journaux du cycle de vies des unités archivistiques concurrents s'est terminé avec succès (CHECK\_CONCURRENT\_WORKFLOW\_LOCK.OK = Succès de la vérification des processus concurrents)
  - KO : des processus concurrents de traçabilité des unités archivistiques sont en cours d'exécution (CHECK\_CONCURRENT\_WORKFLOW\_LOCK.KO = Échec lors de la vérification des processus concurrents)
  - FATAL : une erreur technique est survenue lors de la vérification des processus concurrents

(CHECK\_CONCURRENT\_WORKFLOW\_LOCK.FATAL = Erreur technique lors de la vérification des processus concurrents)

### 13.3.2.2. Préparation de la liste des journaux du cycle de vie et des métadonnées des unités archivistiques PREPARE\_UNIT\_LFC\_TRACEABILITY

- **Règle** : tâche consistant à récupérer les journaux des cycles de vie à sécuriser et à récupérer les informations concernant les dernières opérations de sécurisation
- **Type** : bloquant
- **Statuts** :
  - OK : les fichiers des journaux des cycles de vie ont été exportés ainsi que les informations concernant les dernières opérations de sécurisation (PREPARE\_UNIT\_LFC\_TRACEABILITY.OK = Succès de la préparation des journaux du cycle de vie et des métadonnées des unités archivistiques)
  - KO : les informations sur la dernière opération de sécurisation n'ont pas pu être obtenues / exportées, ou un problème a été rencontré avec un journal de cycle de vie (PREPARE\_UNIT\_LFC\_TRACEABILITY.KO = Échec de la préparation des journaux du cycle de vie et des métadonnées des unités archivistiques)
  - FATAL : une erreur technique est survenue (PREPARE\_UNIT\_LFC\_TRACEABILITY.FATAL = Erreur technique lors de la préparation des journaux du cycle de vie et des métadonnées des unités archivistiques)

### 13.3.2.3. Sécurisation des journaux du cycle de vie des groupes d'objets BUILD\_UNIT\_LFC\_TRACEABILITY

- **Règle** : application de l'algorithme pour créer les fichiers sécurisés des journaux du cycle de vie des unités archivistiques, journal par journal, et génération du fichier sécurisé
- **Type** : bloquant
- **Statuts** :
  - OK : le fichier sécurisé pour le journal du cycle de vie en cours a été généré (BUILD\_UNIT\_LFC\_TRACEABILITY.STARTED.OK = Succès de la sécurisation des journaux du cycle de vie des unités archivistiques)
  - WARNING : il n'y a pas de nouveaux journaux à sécuriser (BUILD\_UNIT\_LFC\_TRACEABILITY.WARNING = Avertissement lors de la sécurisation des journaux du cycle de vie des unités archivistiques)
  - KO : le fichier pour le groupe d'objet n'a pas pu être trouvé (BUILD\_UNIT\_LFC\_TRACEABILITY.KO = Échec de la sécurisation des journaux du cycle de vie des unités archivistiques)
  - FATAL : une erreur technique est survenue lors de la génération des fichiers sécurisés (BUILD\_UNIT\_LFC\_TRACEABILITY.FATAL = Erreur technique lors de la sécurisation des journaux du cycle de vie des unités archivistiques)

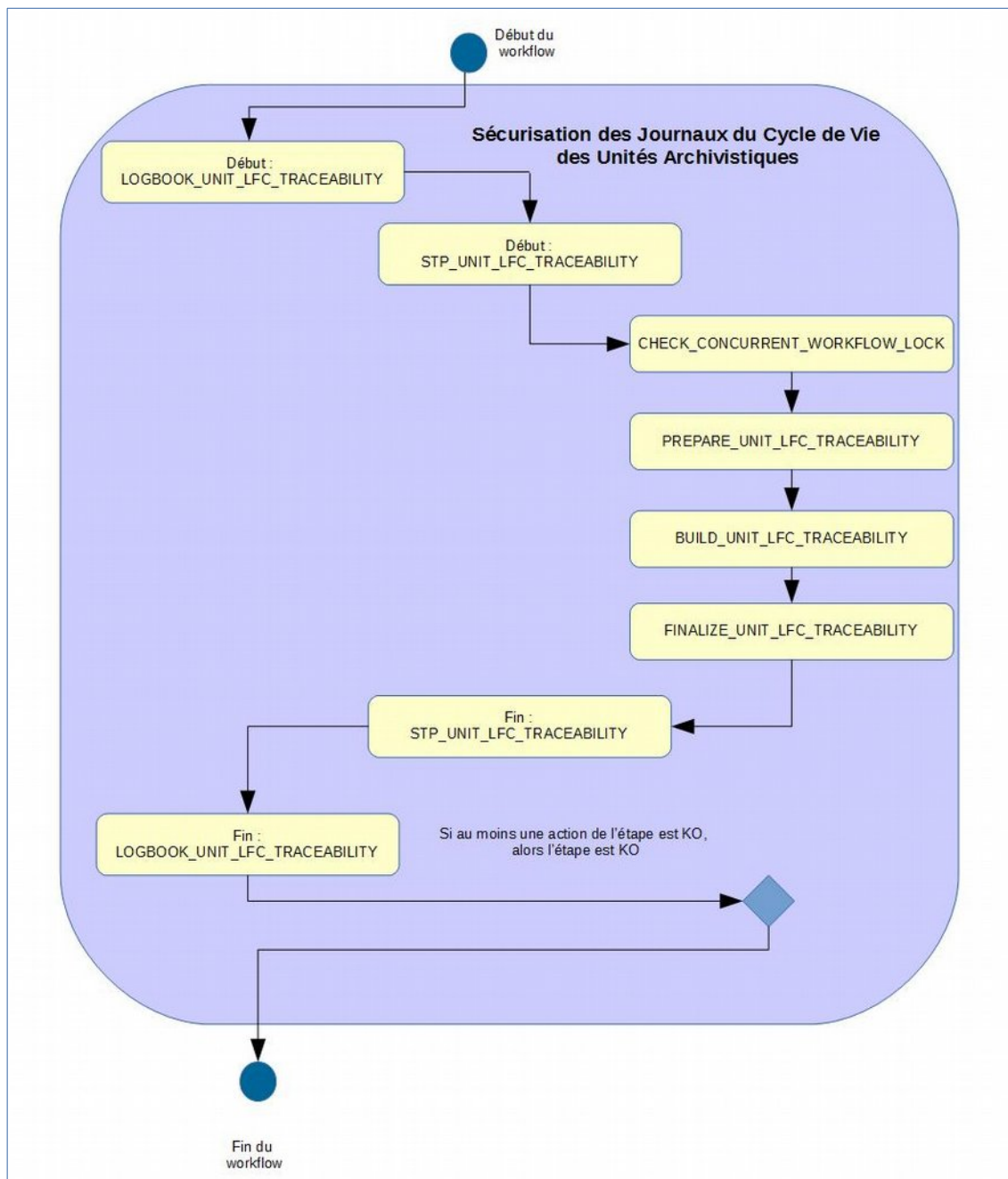
### 13.3.2.4. Finalisation de la sécurisation des journaux du cycle de vie des groupes d'objets FINALIZE\_UNIT\_LFC\_TRACEABILITY

- **Règle** : tâche consistant à récupérer les différents fichiers générés puis à calculer le tampon d'horodatage
- **Type** : non applicable
- **Statuts** :
  - OK : la finalisation de la sécurisation des journaux du cycle de vie des unités archivistiques a bien été effectuée (FINALIZE\_UNIT\_LFC\_TRACEABILITY.OK = Succès de la finalisation de la sécurisation des journaux du cycle de vie des unités archivistiques)
  - KO : la finalisation de la sécurisation des journaux du cycle de vie des unités archivistiques n'a pas été effectuée (FINALIZE\_UNIT\_LFC\_TRACEABILITY.KO = Échec de la finalisation de la sécurisation des journaux du cycle de vie des unités archivistiques)

- FATAL : une erreur technique est survenue lors de la finalisation de la la sécurisation des journaux du cycle de vie des unités archivistiques (FINALIZE\_UNIT\_LFC\_TRACEABILITY.FATAL = Erreur technique lors de la finalisation de la sécurisation des journaux du cycle de vie des unités archivistiques)

### 13.3.3. Structure du workflow du processus de sécurisation des journaux des cycles de vie des unités archivistiques

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



## 13.4.Workflow de sécurisation des journaux des écritures

Cette section décrit la sécurisation des journaux des écritures mis en place 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.

### 13.4.1.Processus de sécurisation des journaux des écritures (STP\_STORAGE\_SECURISATION)

Le processus de sécurisation des journaux des écritures consiste en la création d'un fichier .zip contenant :

- Des logs des journaux sécurisés (logFile.log). Ces logs comportent un certain nombre d'informations comme la date des écritures, l'empreinte des fichiers concernés, le tenant, l'adresse des offres... Ces logs sont un extrait des logs du moteur de stockage, sélectionnés entre deux intervalles de dates.
- Un fichier d'information décrivant le périmètre du fichier des logs des journaux sécurisés associé : date de début et date de fin définissant l'intervalle de sélection des logs à sécuriser, ainsi que l'empreinte du fichier logFile et la date de création du .zip

Au niveau du journal des écritures, cette action est entièrement réalisée dans une seule étape (STP\_STORAGE\_SECURISATION)

- **Règle** : opération consistant à sécuriser le journal des écritures
- **Type** : bloquant
- **Statuts** :
  - OK : le journal des écritures ont été sécurisés (STP\_STORAGE\_SECURISATION.OK = Succès du processus de sécurisation du journal des écritures)
  - WARNING : il n'y a pas de nouveaux journaux à sécuriser (STP\_STORAGE\_SECURISATION.WARNING = Avertissement lors du processus de sécurisation du journal des écritures)
  - KO : pas de cas KO
  - FATAL : une erreur technique est survenue lors de la sécurisation du journal des écritures (STP\_STORAGE\_SECURISATION.FATAL = Erreur technique lors du processus de sécurisation du journal des écritures)

#### 13.4.1.1. Création du tampon d'horodatage de l'ensemble des journaux d'écriture STORAGE\_SECURISATION\_TIMESTAMP

- **Règle** : tâche consistant à calculer le tampon d'horodatage à partir de la racine de l'arbre de Merkle constitué de la liste des journaux qui sont en train d'être sécurisés
- **Type** : bloquant
- **Statuts** :
  - OK : la création du tampon d'horodatage de l'ensemble des journaux d'écriture à été réalisé avec succès (STORAGE\_SECURISATION\_TIMESTAMP.OK = Succès de la création du tampon d'horodatage de l'ensemble des journaux d'écriture)
  - FATAL : une erreur technique est survenue lors de l'horodatage de l'ensemble des journaux d'écriture (STORAGE\_SECURISATION\_TIMESTAMP.FATAL = Erreur technique lors de la création du tampon d'horodatage de l'ensemble des journaux d'écriture)

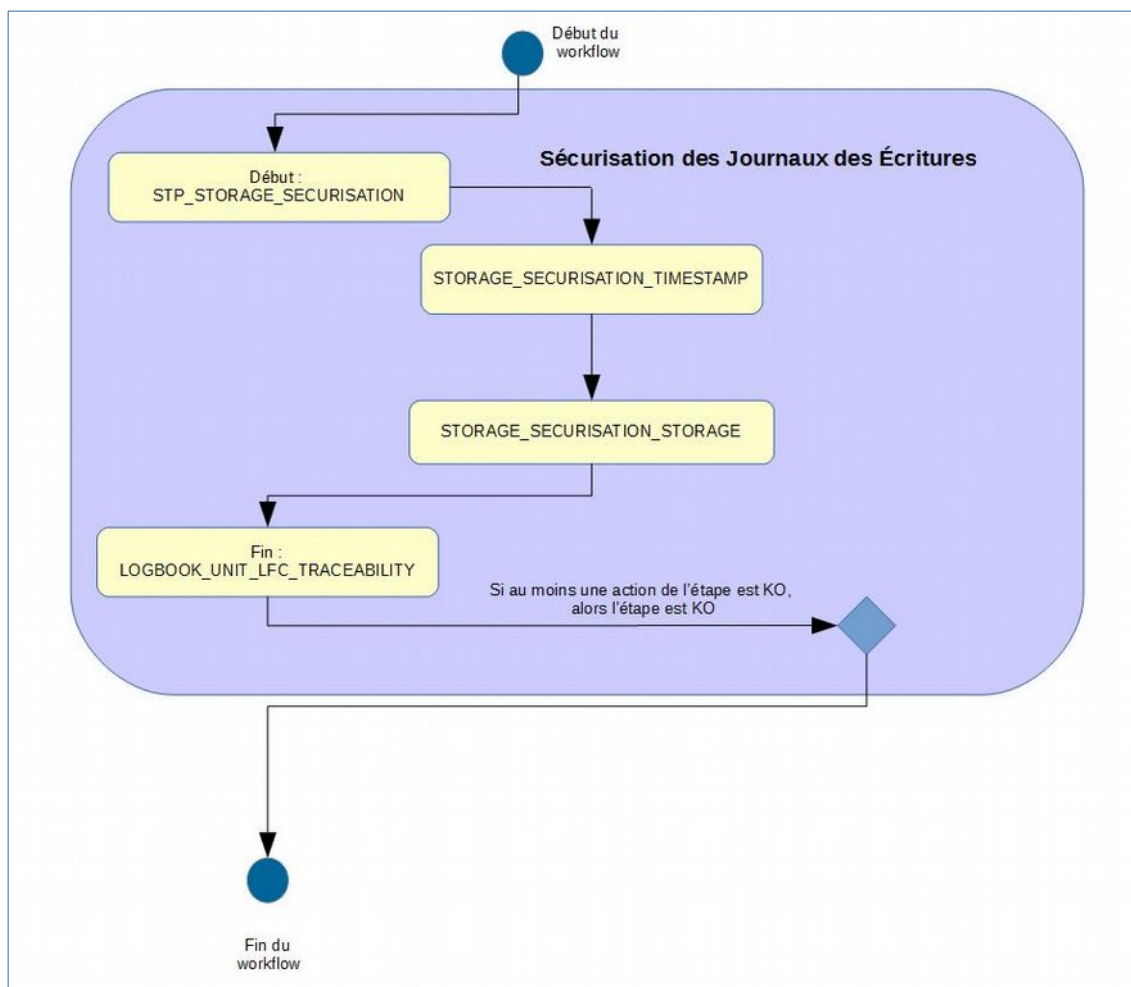
#### 13.4.1.2. Stockage des journaux d'écriture STORAGE\_SECURISATION\_STORAGE

- **Règle** : tâche consistant à écrire les journaux d'écriture sur les offres de stockage, en fonction de la stratégie de stockage

- **Type** : bloquant
- **Statuts** :
  - OK : les journaux d'écriture ont été écrits sur les offres de stockage (STORAGE\_SECURISATION\_STORAGE.OK = Succès du stockage des journaux d'écriture)
  - FATAL : une erreur technique est survenue lors de l'écriture des journaux des écritures sur les offres de stockage (STORAGE\_SECURISATION\_STORAGE.FATAL = Erreur technique lors du stockage des journaux d'écriture)

### 13.4.2. Structure du workflow du processus de sécurisation des journaux des écritures

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



## 13.5. Sauvegarde des journaux des écritures

Cette section décrit la sauvegarde des journaux des écritures. Contrairement aux autres sécurisations de journaux du cycle de vie des groupes d'objets et des unités archivistiques ou du journal des opérations, celle-ci n'est pas utilisée au sein d'un workflow.

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.

### 13.5.1. Processus de sauvegarde des journaux des écritures (STORAGE\_BACKUP)

Le processus de sauvegarde des journaux des écritures consiste en la création de fichier *.log* contenant un certain nombre d'informations.

Ces logs sont un extrait des logs du moteur de stockage, sélectionnés entre deux intervalles de dates.

Lors de la copie du Moteur de stockage vers les Offres, les fichiers sont renommés pour utiliser en date de début la date de début de chaque fichier de log et en date de fin la date du traitement.

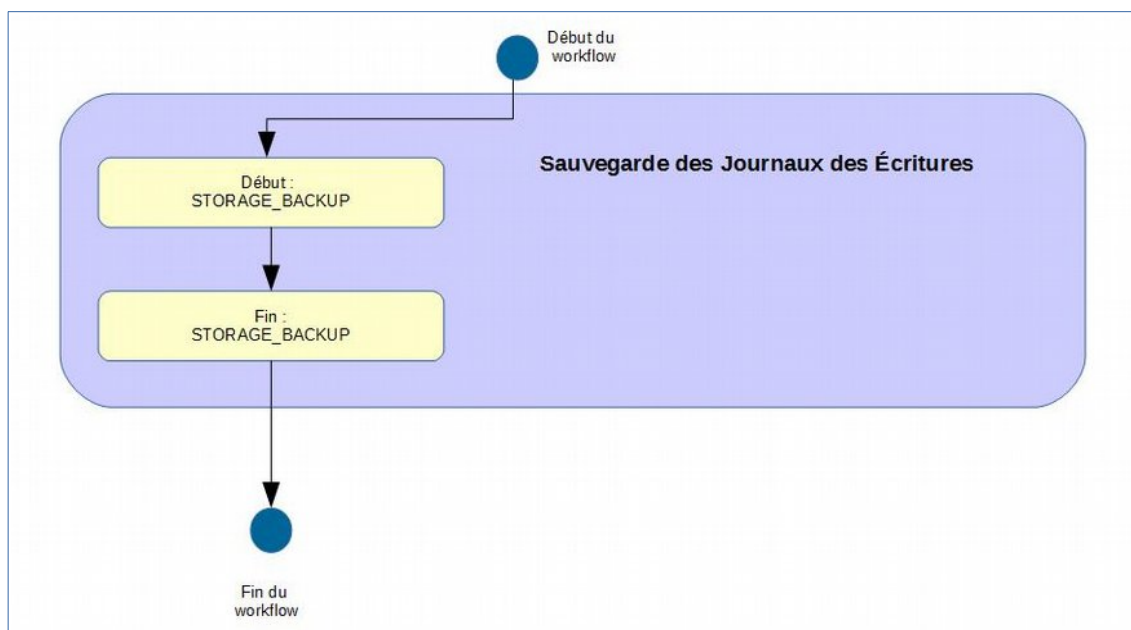
Au niveau du journal des opérations, cette action est entièrement réalisée dans une seule étape

- **Règle** : opération consistant à sauvegarder les journaux des écritures
- **Type** : bloquant
- **Statuts** :
  - OK : la sauvegarde des journaux des écritures réalisée avec succès (STORAGE\_BACKUP.OK = Succès de la sauvegarde des journaux des écritures)
  - WARNING : avertissement lors de la sauvegarde des journaux des écritures (STORAGE\_BACKUP.WARNING = Avertissement lors de la sauvegarde des journaux des écritures)
  - KO : échec de la sauvegarde des journaux des écritures (STORAGE\_ACCESS\_BACKUP.KO = Échec de la sauvegarde des journaux des écritures)
  - FATAL : une erreur technique est survenue lors de la sauvegarde des journaux des écritures (STORAGE\_ACCESS\_BACKUP.FATAL = Erreur technique lors de la sauvegarde des journaux des écritures)



### 13.5.2. Structure du workflow du processus de sauvegarde des journaux des écritures

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



### 13.6. Sauvegarde des logs des accès

Cette section décrit la sauvegarde des logs des accès aux binaires mis en place dans la solution logicielle Vitam. Contrairement aux autres sécurisations de journaux du cycle de vie des groupes d'objets et des unités archivistiques ou du journal des opérations, celle-ci n'est pas utilisée au sein d'un workflow.

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.

#### 13.6.1. Processus de sauvegarde des logs des accès (STORAGE\_ACCESS\_BACKUP)

Le processus de sauvegarde des logs des accès aux objets binaires consiste en la création d'un fichier.log contenant un certain nombre d'informations comme la date des accès, l'identifiant du document récupéré, le contrat utilisé, l'unité archivistique donnant accès à l'objet binaire, et l'identifiant de la requête d'accès.

Ces logs sont un extrait des logs du moteur de stockage, sélectionnés entre deux intervalles de dates.

Lors de la copie du Moteur de stockage vers les Offres, les fichiers sont renommés pour utiliser en date de début la date de début de chaque fichier de log et en date de fin la date du traitement.

Au niveau du journal des opérations, cette action est entièrement réalisée dans une seule étape

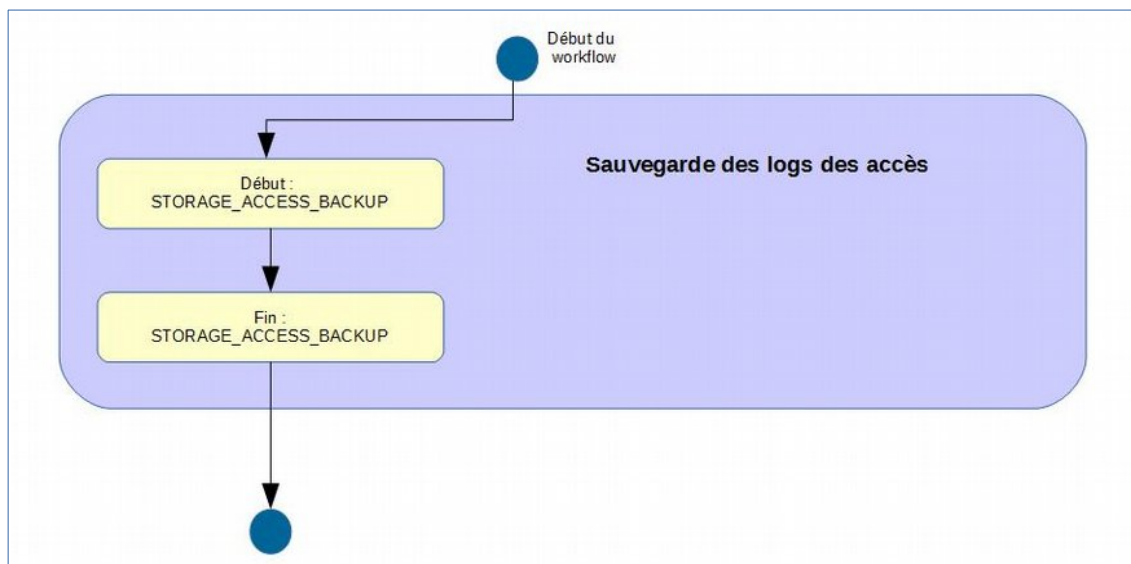
- **Règle** : opération consistant à sauvegarder les logs des accès
- **Type** : bloquant
- **Status** :
  - OK : sauvegarde des logs d'accès avec succès (STORAGE\_ACCESS\_BACKUP = Sauvegarde des logs des accès)
  - WARNING : avertissement lors de la sauvegarde des logs des accès (STORAGE\_ACCESS\_BACKUP.WARNING = Avertissement lors de la sauvegarde des logs des accès)
  - KO : échec de la sauvegarde des logs des accès (STORAGE\_ACCESS\_BACKUP.KO = Échec de la sauvegarde des logs des accès)



- FATAL : une erreur technique est survenue lors de la sauvegarde des logs des accès (STORAGE\_ACCESS\_BACKUP.FATAL = Erreur technique lors de la sauvegarde des logs des accès)

### 13.6.2. Structure du workflow du processus de sauvegarde des logs des accès

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



## 14. AUDIT

### 14.1. Workflow de contrôle d'intégrité d'un journal sécurisé

Cette section décrit le processus (workflow) de contrôle d'intégrité d'un journal sécurisé mis en place dans la solution logicielle Vitam.

Celui-ci est défini dans le fichier *"DefaultCheckTraceability.json"* (situé ici : sources/processing/processing-management/src/main/resources/workflows).

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.

#### 14.1.1. Processus de contrôle d'intégrité d'un journal sécurisé (vision métier)

Le processus de contrôle d'intégrité débute lorsqu'un identifiant d'opération de sécurisation des journaux d'opération, des journaux du cycle de vie ou du journal des écritures est soumis au service de contrôle d'intégrité des journaux sécurisés. Le service permet de récupérer le journal sécurisé, d'extraire son contenu et de valider que son contenu n'a pas été altéré.

Pour cela, le service calcule un arbre de Merkle à partir des journaux d'opérations que contient le journal sécurisé, puis en calcule un second à partir des journaux correspondants disponibles dans la solution logicielle Vitam. Une comparaison est ensuite effectuée entre ces deux arbres et celui contenu dans les métadonnées du journal sécurisé.

Ensuite, dans une dernière étape, le tampon d'horodatage est vérifié et validé.

### 14.1.2. Processus de préparation de la vérification des journaux sécurisés (STP\_PREPARE\_TRACEABILITY\_CHECK)

#### Préparation de la vérification des journaux sécurisés PREPARE\_TRACEABILITY\_CHECK (PrepareTraceabilityCheckProcessActionHandler.java)

- **Règle** : tâche permettant de vérifier que l'opération donnée en entrée est de type TRACEABILITY et à récupérer le fichier au format .zip associé à cette opération et à extraire son contenu
- **Type** : bloquant
- **Statuts** :
  - OK : l'opération donnée en entrée est une opération de type TRACEABILITY, le zip a été trouvé et son contenu extrait (PREPARE\_TRACEABILITY\_CHECK.OK = Succès de la préparation de la vérification des journaux sécurisés)
  - KO : l'opération donnée en entrée n'est pas une opération de type TRACEABILITY (PREPARE\_TRACEABILITY\_CHECK.KO = Échec de la préparation de la vérification des journaux sécurisés)
  - FATAL : une erreur technique est survenue lors du processus de préparation de vérification (PREPARE\_TRACEABILITY\_CHECK.FATAL = Erreur technique lors de la préparation de la vérification des journaux sécurisés)

### 14.1.3. Processus de vérification de l'arbre de Merkle (STP\_MERKLE\_TREE)

#### 14.1.3.1. Vérification de l'arbre de Merkle CHECK\_MERKLE\_TREE (VerifyMerkleTreeActionHandler.java)

- **Règle** : tâche consistant à recalculer l'arbre de Merkle des journaux contenus dans le journal sécurisé, à calculer un autre arbre à partir des journaux indexés correspondants et à vérifier que tous deux correspondent à celui stocké dans les métadonnées du journal sécurisé
- **Type** : bloquant
- **Statuts** :
  - OK : les arbres de Merkle correspondent (CHECK\_MERKLE\_TREE.OK = Succès de la vérification de l'arbre de MERKLE)
  - KO : les arbres de Merkle ne correspondent pas (CHECK\_MERKLE\_TREE.KO = Échec de la vérification de l'arbre de MERKLE)
  - FATAL : une erreur technique est survenue lors de la vérification des arbres de Merkle (CHECK\_MERKLE\_TREE.FATAL = Erreur technique lors de la vérification de l'arbre de MERKLE)

La tâche Check\_Merkle\_Tree contient les traitements suivants :

#### 14.1.3.2. Comparaison de l'arbre de Merkle avec le Hash enregistré CHECK\_MERKLE\_TREE.COMPARE\_MERKLE\_HASH\_WITH\_SAVED\_HASH

- **Règle** : traitement consistant à vérifier que l'arbre de Merkle calculé à partir des journaux contenus dans le journal sécurisé est identique à celui stocké dans les métadonnées du journal sécurisé
- **Type** : bloquant
- **Statuts** :
  - OK : l'arbre de Merkle des journaux contenus dans le journal sécurisé correspond à celui stocké dans les métadonnées du journal sécurisé (CHECK\_MERKLE\_TREE.COMPARE\_MERKLE\_HASH\_WITH\_SAVED\_HASH.OK = Succès de

- la comparaison de l'arbre de MERKLE avec le Hash enregistré)
- KO : l'arbre de Merkle des journaux contenus dans le journal sécurisé ne correspond pas à celui stocké dans les métadonnées du journal sécurisé  
(CHECK\_MERKLE\_TREE.COMPARE\_MERKLE\_HASH\_WITH\_SAVED\_HASH.KO = Échec de la comparaison de l'arbre de MERKLE avec le Hash enregistré)
- FATAL : une erreur technique est survenue lors de la comparaison de l'arbre de MERKLE avec le Hash enregistré  
(CHECK\_MERKLE\_TREE.COMPARE\_MERKLE\_HASH\_WITH\_SAVED\_HASH.FATAL = Erreur technique lors de la comparaison de l'arbre de MERKLE avec le Hash enregistré)

#### 14.1.3.3. Comparaison de l'arbre de Merkle avec le Hash indexé

##### CHECK\_MERKLE\_TREE.COMPARE\_MERKLE\_HASH\_WITH\_INDEXED\_HASH

- **Règle** : traitement consistant à vérifier que l'arbre de Merkle calculé à partir des journaux indexés est identique à celui stocké dans les métadonnées du journal sécurisé
- **Type** : bloquant
- **Statuts** :
  - OK : l'arbre de Merkle des journaux indexés correspond à celui stocké dans les métadonnées du journal sécurisé  
(CHECK\_MERKLE\_TREE.COMPARE\_MERKLE\_HASH\_WITH\_INDEXED\_HASH.OK = Succès de la comparaison de l'arbre de MERKLE avec le Hash indexé)
  - KO : l'arbre de Merkle des journaux indexés ne correspond pas à celui stocké dans les métadonnées du journal sécurisé  
(CHECK\_MERKLE\_TREE.COMPARE\_MERKLE\_HASH\_WITH\_INDEXED\_HASH.KO = Échec de la comparaison de l'arbre de MERKLE avec le Hash indexé)
  - FATAL : une erreur technique est survenue lors de la comparaison de l'arbre de Merkle des journaux indexés avec celui stocké dans les métadonnées du journal sécurisé  
(CHECK\_MERKLE\_TREE.COMPARE\_MERKLE\_HASH\_WITH\_INDEXED\_HASH.FATAL = Erreur technique lors de la comparaison de l'arbre de MERKLE avec le Hash indexé)

#### 14.1.4. Processus de vérification de l'horodatage (STP\_VERIFY\_STAMP)

##### 14.1.4.1. Vérification et validation du tampon d'horodatage VERIFY\_TIMESTAMP (VerifyTimeStampActionHandler.java)

- **Règle** : tâche consistant à vérifier et à valider le tampon d'horodatage
- **Type** : bloquant
- **Statuts** :
  - OK : le tampon d'horodatage est correct (VERIFY\_TIMESTAMP.OK = Succès de la vérification de l'horodatage)
  - KO : le tampon d'horodatage est incorrect (VERIFY\_TIMESTAMP.KO = Échec de la vérification de l'horodatage)
  - FATAL : une erreur technique est survenue lors de la vérification du tampon d'horodatage (VERIFY\_TIMESTAMP.FATAL = Erreur technique lors de la vérification de l'horodatage)

La tâche Verify\_Timestamp contient les traitements suivants :

##### 14.1.4.2. Comparaison du tampon du fichier (token.tsp) par rapport au tampon enregistré dans le logbook VERIFY\_TIMESTAMP.COMPARE\_TOKEN\_TIMESTAMP

- **Règle** : traitement consistant à vérifier que le tampon enregistré dans la collection logbookOperation est le même que celui présent dans le fichier zip généré
- **Type** : bloquant

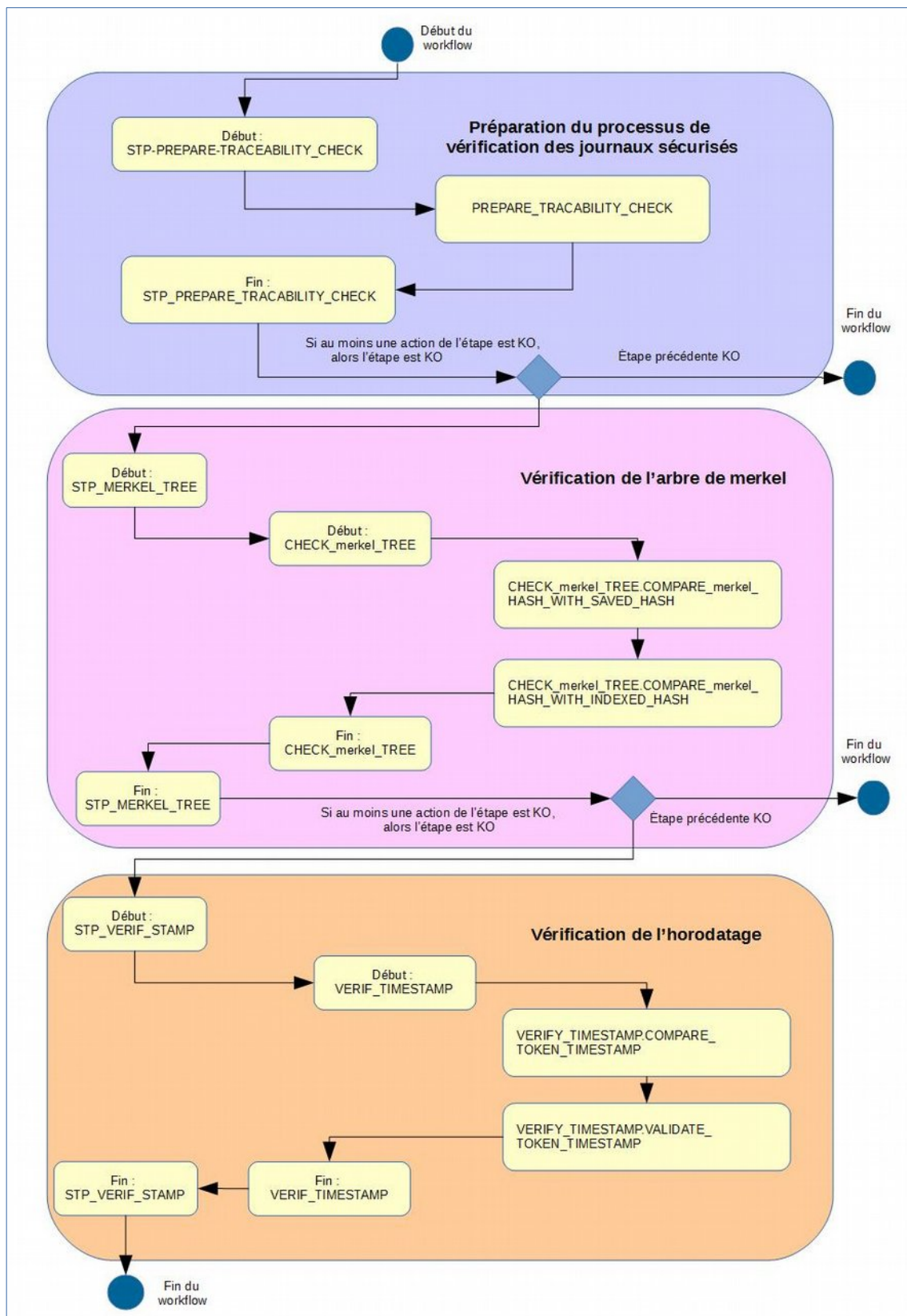
- **Statuts :**
  - OK : les tampons sont identiques (VERIFY\_TIMESTAMP.COMPARE\_TOKEN\_TIMESTAMP.OK = Succès de la comparaison des tampons d’horodatage)
  - KO : les tampons sont différents (VERIFY\_TIMESTAMP.COMPARE\_TOKEN\_TIMESTAMP.KO = Échec de la comparaison des tampons d’horodatage)
  - FATAL : erreur technique lors de la vérification des tampons (VERIFY\_TIMESTAMP.COMPARE\_TOKEN\_TIMESTAMP.FATAL = Erreur technique lors de la comparaison des tampons d’horodatage)

#### 14.1.4.3. Validation du tampon d’horodatage VERIFY\_TIMESTAMP.VALIDATE\_TOKEN\_TIMESTAMP

- **Règle :** traitement consistant à vérifier cryptographiquement le tampon et à vérifier la chaîne de certification
- **Type :** bloquant
- **Statuts :**
  - OK : le tampon est validé (VERIFY\_TIMESTAMP.VALIDATE\_TOKEN\_TIMESTAMP.OK = Succès de la validation du tampon d’horodatage)
  - KO : le tampon est invalidé (VERIFY\_TIMESTAMP.VALIDATE\_TOKEN\_TIMESTAMP.KO = Échec de la validation du tampon d’horodatage)
  - FATAL : erreur technique lors de la validation du tampon d’horodatage (VERIFY\_TIMESTAMP.VALIDATE\_TOKEN\_TIMESTAMP.FATAL = Erreur technique lors de la validation du tampon d’horodatage)

### 14.1.5. Structure du workflow de contrôle d'intégrité d'un journal sécurisé

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



## 14.2. Workflow d'audit de vérification des journaux sécurisés

Cette section décrit le processus (workflow) d'audit de vérification des sécurisations en masse mis en place afin de s'assurer de la non perte et de la validité des données conservées dans la solution logicielle Vitam.

Celui-ci est défini dans le fichier « LinkedCheckTraceability.json » (situé ici : sources/processing/processing-management/src/main/resources/workflows/LinkedCheckTraceability.json)

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.

### 14.2.1. Processus d'audit de vérification des journaux sécurisés (vision métier)

L'audit de sécurisation doit permettre de vérifier que les journaux de sécurisation existent, pour chaque type de journal, sur une période définie, et être effectué sur chaque offre selon la stratégie par défaut.

### 14.2.2. Processus de préparation de la vérification des journaux sécurisés (STP\_PREPARE\_TRACEABILITY\_LINKED\_CHECK)

**Règle :** Étape qui permet de retrouver la liste des journaux de sécurisation tels que définis dans la requête DSL (type de journal et intervalle de temps) et de récupérer la liste des stratégies de stockage.

**Type :** Bloquant

**Statuts :**

- **OK :** Tous les journaux demandés par la requête ont bien été récupérés et les stratégies également (PREPARE\_TRACEABILITY\_CHECK.OK = Succès de la préparation de la vérification des journaux sécurisés)
- **FATAL :** une erreur technique est survenue lors du processus de préparation de vérification (PREPARE\_TRACEABILITY\_CHECK.FATAL = Erreur technique lors de la préparation de la vérification des journaux sécurisés)
- **WARNING :** Avertissement lors du processus de préparation de la vérification des journaux sécurisés (STP\_PREPARE\_TRACEABILITY\_CHECK.WARNING = Avertissement lors du processus de préparation de la vérification des journaux sécurisés)

#### 14.2.2.1. Préparation de la vérification des journaux sécurisés TRACEABILITY\_LINKED\_CHECK\_PREPARE (TraceabilityLinkedCheckPreparePlugin.java)

- **Règle :** Tâche permettant de retrouver la liste des journaux de sécurisation tels que définis dans la requête DSL (type de journal et intervalle de temps).
- **Type :** Bloquant
- **Statuts :**
  - **OK :** Tous les journaux demandés par la requête ont bien été récupérés (TRACEABILITY\_LINKED\_CHECK\_PREPARE.OK= Succès de la préparation de la vérification des journaux sécurisés)

- **FATAL** : Erreur technique dans la récupération des journaux demandés (TRACEABILITY\_LINKED\_CHECK\_PREPARE.FATAL= Erreur technique lors de la préparation de la vérification des journaux sécurisés)
- **WARNING** : Pas de journaux trouvés ou certains journaux ne possède pas de données (TRACEABILITY\_LINKED\_CHECK\_PREPARE.WARNING= Avertissement lors de la préparation de la vérification des journaux sécurisés)

#### 14.2.2.2. Récupération de la liste des stratégies de stockage PREPARE\_STORAGE\_STRATEGIES

##### (PrepareStorageStrategiesPlugin.java)

- **Règle** : Tâche consistant à récupérer la configuration des stratégies de stockage définies pour la plateforme.
- **Type** : Bloquant
- **Statuts** :
  - **OK** : Les données ont été récupérées et enregistrées dans un fichier de travail avec succès (PREPARE\_STORAGE\_STRATEGIES.OK=Succès de la récupération de la liste des stratégies de stockage)
  - **FATAL** : Erreur technique lors de la récupération de la liste des stratégies de stockage (PREPARE\_STORAGE\_STRATEGIES.FATAL=Erreur technique lors de la récupération de la liste des stratégies de stockage)

#### 14.2.3. Processus de la vérification des journaux sécurisés (STP\_TRACEABILITY\_LINKED\_CHECKS)

**Règle** : Étape permettant de vérifier que les journaux de sécurisation sont bien sur toutes les offres attendues et de vérifier leur intégrité.

**Type** : Non bloquant

**Statuts** :

- **OK** : Tous les journaux ont été trouvés sur les offres définies par la stratégie par défaut et leur intégrité est bien vérifiée (STP\_TRACEABILITY\_LINKED\_CHECKS.OK=Succès du processus de la vérification des journaux sécurisés)
- **KO** : Tous les journaux n'ont été trouvés sur les offres définies par la stratégie par défaut OU leur intégrité n'est vérifiée (STP\_TRACEABILITY\_LINKED\_CHECKS.KO=Échec du processus de la vérification des journaux sécurisés)
- **FATAL** : Erreur technique lors de la vérification du stockage et de l'intégrité des journaux de sécurisation (STP\_TRACEABILITY\_LINKED\_CHECKS.FATAL=Erreur technique lors du processus de la vérification des journaux sécurisés)
- **WARNING** : Aucun journal de sécurisation correspondant à la requête (type de journal et intervalle de temps) n'a été trouvé (STP\_TRACEABILITY\_LINKED\_CHECKS.WARNING=Avertissement lors du processus de la vérification des journaux sécurisés)

### 14.2.3.1. Vérification d'existence de fichier de sécurisation dans les offres de stockage et récupération du hash correspondant RETRIEVE\_SECURE\_TRACEABILITY\_DATA\_FILE

(RetrieveSecureTraceabilityDataFilePlugin.java)

- **Règle** : Tâche de vérification de l'existence des fichiers de sécurisation dans les offres de stockage par défaut et récupération du hash correspondant.
- **Type** : Non bloquant
- **Statuts** :
  - OK : Succès de la vérification d'existence de fichier de sécurisation dans les offres de stockage et récupération du hash correspondant (RETRIEVE\_SECURE\_TRACEABILITY\_DATA\_FILE.OK=Succès de la vérification d'existence de fichier de sécurisation dans les offres de stockage et récupération du hash correspondant)
  - KO : Au moins un fichier sécurisation n'a pas été trouvé sur l'une des offres de stockage (RETRIEVE\_SECURE\_TRACEABILITY\_DATA\_FILE.KO=Échec de la vérification d'existence de fichier de sécurisation dans les offres de stockage et récupération du hash correspondant)
  - FATAL : Erreur technique sur au moins une offre au moment de la vérification de l'existence des fichiers de sécurisation (RETRIEVE\_SECURE\_TRACEABILITY\_DATA\_FILE.FATAL=Erreur technique lors de la vérification d'existence de fichier de sécurisation dans les offres de stockage)

### 14.2.3.2. Vérification de l'intégrité du fichier de sécurisation CHECKS\_SECURE\_TRACEABILITY\_DATA\_HASHES

(ChecksSecureTraceabilityDataHashesPlugin.java)

- **Règle** : Comparaison du hash du fichier de sécurisation présent dans l'offre référente avec les hash des fichiers de sécurisation obtenus par la tâche précédente.
- **Type** : Non bloquant
- **Statuts** :
  - OK : L'intégrité du fichier de sécurisation est vérifiée (CHECKS\_SECURE\_TRACEABILITY\_DATA\_HASHES.OK=Succès de la vérification de l'intégrité du fichier de sécurisation)
  - KO : L'intégrité du fichier de sécurisation n'est pas vérifiée (CHECKS\_SECURE\_TRACEABILITY\_DATA\_HASHES.KO=Échec de la vérification de l'intégrité du fichier de sécurisation)
  - FATAL : Erreur technique lors du contrôle de l'intégrité du fichier de sécurisation (CHECKS\_SECURE\_TRACEABILITY\_DATA\_HASHES.FATAL=Erreur technique lors de la vérification de l'intégrité du fichier de sécurisation)



### 14.2.3.3. Récupération du fichier de sécurisation EXTRACT\_SECURE\_TRACEABILITY\_DATA\_FILE

#### (ExtractSecureTraceabilityDataFilePlugin.java)

- **Règle** : Récupération et décompression du contenu du fichier compressé de sécurisation en vue de son traitement.
- **Type** : Non bloquant
- **Statuts** :
  - OK : Succès de la décompression du fichier de sécurisation (EXTRACT\_SECURE\_TRACEABILITY\_DATA\_FILE.OK=Succès de l'extraction du fichier de sécurisation)
  - FATAL : Erreur technique au moment de la décompression du fichier de sécurisation (EXTRACT\_SECURE\_TRACEABILITY\_DATA\_FILE.FATAL=Erreur technique lors de l'extraction du fichier de sécurisation)

### 14.2.3.4. Vérification de l'arbre de MERKLE CHECK\_MERKLE\_TREE

- **Règle** : Tâche consistant à recalculer l'arbre de Merkle des journaux contenus dans le journal sécurisé, à calculer un autre arbre à partir des journaux indexés correspondants et à vérifier que tous deux correspondent à celui stocké dans les métadonnées du journal sécurisé.
- **Type** : Non bloquant
- **Statuts** :
  - OK : les arbres de Merkle correspondent (CHECK\_MERKLE\_TREE.OK=Succès de la vérification de l'arbre de MERKLE)
  - KO : les arbres de Merkle ne correspondent pas (CHECK\_MERKLE\_TREE.KO=Échec de la vérification de l'arbre de MERKLE)
  - FATAL : une erreur technique est survenue lors de la vérification des arbres de Merkle (CHECK\_MERKLE\_TREE.FATAL=Erreur technique lors de la vérification de l'arbre de MERKLE)

Cette tâche contient les traitements suivants :

#### 14.2.3.5. 1. Comparaison de l'arbre de Merkle avec le hash enregistré

##### CHECK\_MERKLE\_TREE.COMPARE\_MERKLE\_HASH\_WITH\_SAVED\_HASH

- **Règle** : Traitement consistant à vérifier que l'arbre de Merkle calculé à partir des journaux contenus dans le journal sécurisé est identique à celui stocké dans le fichier de sécurisation.
- **Type** : Non bloquant

- Statuts :
  - OK : L'arbre de Merkle des journaux contenus dans le journal sécurisé correspond à celui stocké dans le fichier de sécurisation. (CHECK\_MERKLE\_TREE.COMPARE\_MERKLE\_HASH\_WITH\_SAVED\_HASH.OK=Succès de la comparaison de l'arbre de MERKLE avec le Hash enregistré)
  - KO : L'arbre de Merkle des journaux contenus dans le journal sécurisé ne correspond pas à celui stocké dans le fichier de sécurisation. (CHECK\_MERKLE\_TREE.COMPARE\_MERKLE\_HASH\_WITH\_SAVED\_HASH.KO=Échec de la comparaison de l'arbre de MERKLE avec le Hash enregistré)
  - FATAL : Une erreur technique est survenue lors de la comparaison de l'arbre de MERKLE avec le hash enregistré (CHECK\_MERKLE\_TREE.COMPARE\_MERKLE\_HASH\_WITH\_SAVED\_HASH.FATAL=Erreur technique lors de la comparaison de l'arbre de MERKLE avec le Hash enregistré)

#### 14.2.3.6. 2. Comparaison de l'arbre de Merkle avec le hash indexé

##### CHECK\_MERKLE\_TREE.COMPARE\_MERKLE\_HASH\_WITH\_INDEXED\_HASH

- **Règle** : Traitement consistant à vérifier que l'arbre de Merkle calculé à partir des journaux contenus dans le journal sécurisé est identique à celui stocké dans les métadonnées du journal sécurisé.
- **Type** : Non bloquant
- **Statuts** :
  - OK : l'arbre de Merkle des journaux indexés correspond à celui stocké dans les métadonnées du journal sécurisé (CHECK\_MERKLE\_TREE.COMPARE\_MERKLE\_HASH\_WITH\_INDEXED\_HASH.OK =Succès de la comparaison de l'arbre de MERKLE avec le Hash enregistré)
  - KO : l'arbre de Merkle des journaux indexés correspond à celui stocké dans les métadonnées du journal sécurisé (CHECK\_MERKLE\_TREE.COMPARE\_MERKLE\_HASH\_WITH\_INDEXED\_HASH.KO =Échec de la comparaison de l'arbre de MERKLE avec le Hash enregistré)
  - FATAL : une erreur technique est survenue lors de la comparaison de l'arbre de Merkle des journaux indexés avec celui stocké dans les métadonnées du journal sécurisé (CHECK\_MERKLE\_TREE.COMPARE\_MERKLE\_HASH\_WITH\_INDEXED\_HASH.FATAL =Erreur technique lors de la comparaison de l'arbre de MERKLE avec le Hash enregistré)

#### 14.2.3.7. Vérification des journaux des écritures (journal des offres de stockage) CHECKS\_SECURE\_TRACEABILITY\_DATA\_STORAGELOG

- **Règle** : Ce traitement vérifie que le (ou les) journaux des écritures sur les offres de stockage existent et sont intègres.
- **Type** : Bloquant
- **Statuts** :
  - OK : Le ou les journaux des offres de stockage existent et leur intégrité est vérifiée

(CHECKS\_SECURE\_TRACEABILITY\_DATA\_STORAGELOG.OK=Succès de la vérification des journaux des écritures)

- KO : Le ou les journaux des offres de stockage n'existent pas et/ou leur intégrité est n'est pas vérifiée (CHECKS\_SECURE\_TRACEABILITY\_DATA\_STORAGELOG.KO=Échec de la vérification des journaux des écritures)
- FATAL : Erreur technique lors de la vérification des journaux des écritures sur les offres de stockage (CHECKS\_SECURE\_TRACEABILITY\_DATA\_STORAGELOG.FATAL=Erreur technique lors de la vérification des journaux des écritures)

#### 14.2.4. Processus de la finalisation de la vérification des journaux sécurisés STP\_FINALIZE\_TRACEABILITY\_LINKED\_CHECKS

**Règle :** Étape d'utilisation des résultats d'analyse et création du rapport d'audit des sécurisations

**Type :** Bloquant

**Statuts :**

- OK : Le rapport d'audit des sécurisations a été généré avec succès (STP\_FINALIZE\_TRACEABILITY\_LINKED\_CHECKS.KO=Échec du processus de la finalisation de la vérification des journaux sécurisés)
- FATAL : Erreur technique lors de la génération du rapport d'audit des sécurisations (STP\_FINALIZE\_TRACEABILITY\_LINKED\_CHECKS.FATAL=Erreur technique lors du processus de la finalisation de la vérification des journaux sécurisés)

##### 14.2.4.1. Finalisation de la vérification des journaux sécurisés

#### TRACEABILITY\_FINALIZATION

(TraceabilityFinalizationPlugin.java)

- **Règle :** Utilisation des résultats d'analyse et création du rapport d'audit des sécurisations.
- **Type :** Bloquant
- **Statuts :**
  - OK : Le rapport d'audit des sécurisations a été généré avec succès (TRACEABILITY\_FINALIZATION.OK=Succès de la finalisation de la vérification des journaux sécurisés)
  - FATAL : Erreur technique lors de la génération du rapport d'audit des sécurisations (TRACEABILITY\_FINALIZATION.FATAL=Erreur technique lors de la finalisation de la vérification des journaux sécurisés)

### 14.2.5. Audit de la vérification des journaux sécurisés LINKED\_CHECK\_SECURISATION

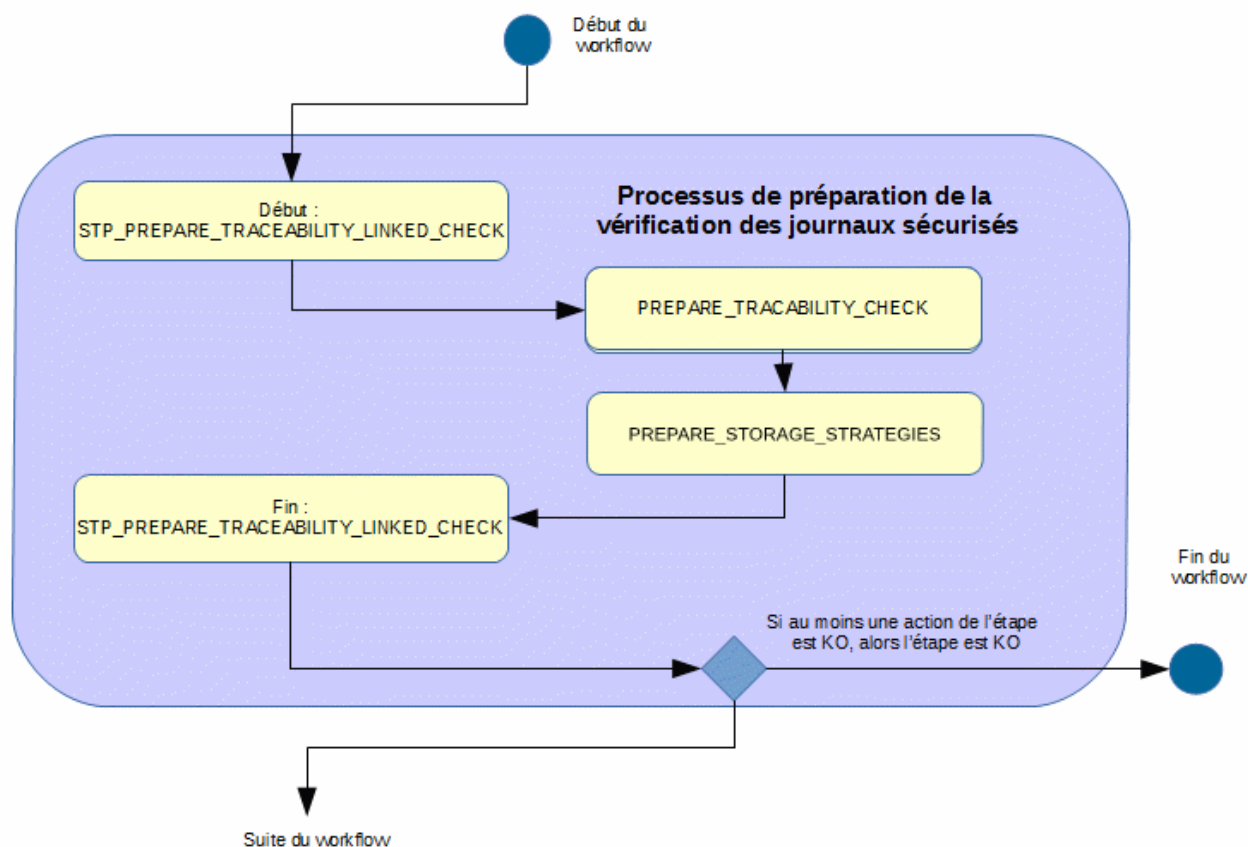
**Règle :** Affichage du résultat du workflow d’audit des sécurisations dans l’IHM démo

**Type :** N/A

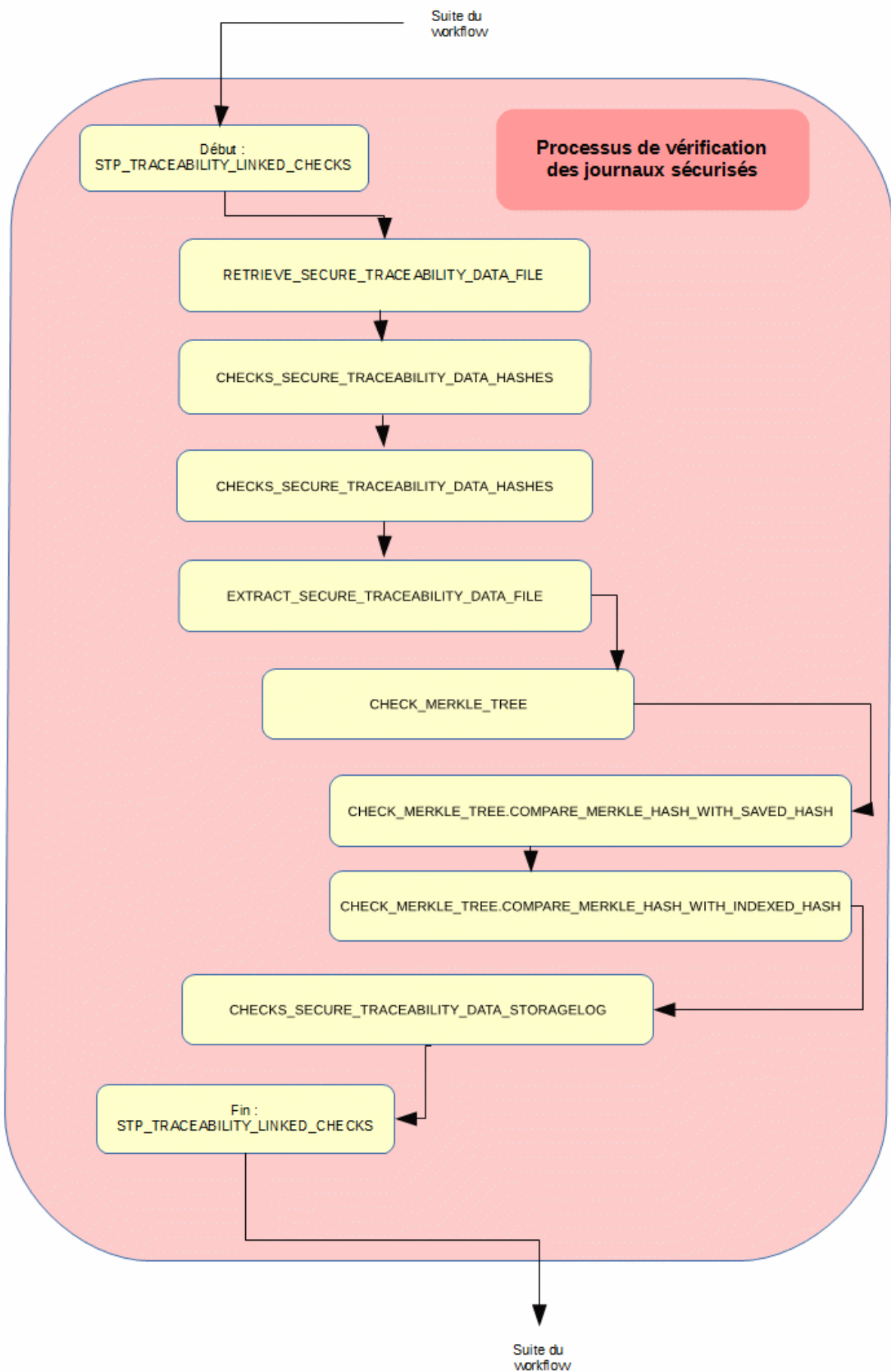
**Statuts :**

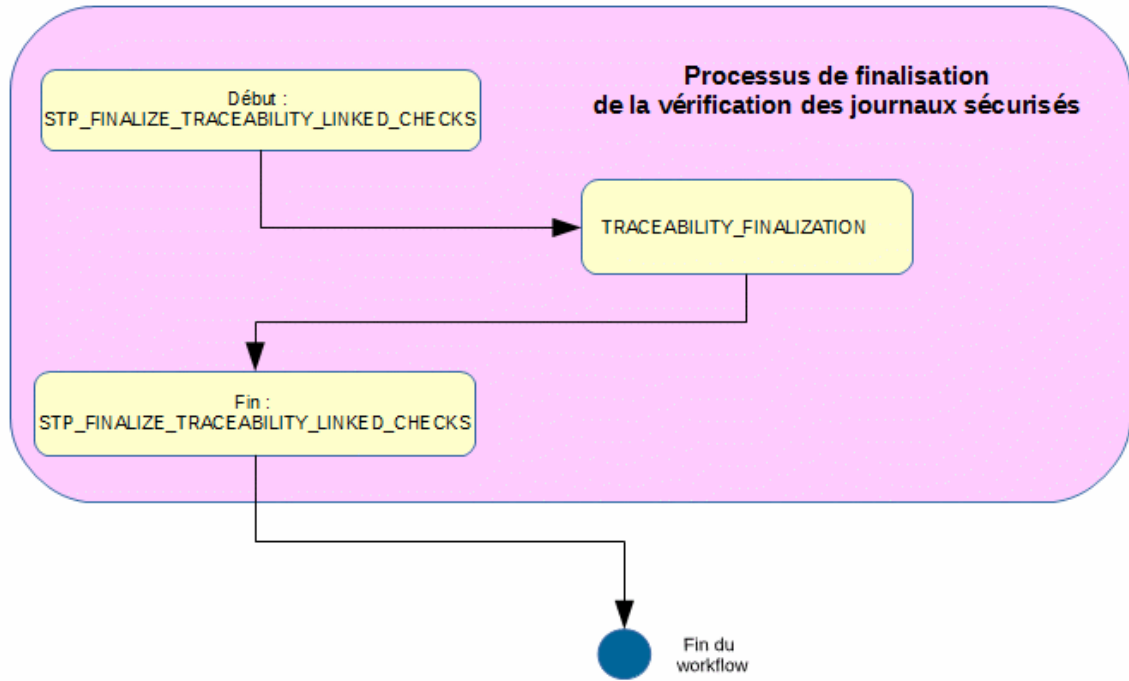
- OK : Le résultat global de l’audit des sécurisations est conforme à l’attendu (LINKED\_CHECK\_SECURISATION.OK=Succès d’audit de la vérification des journaux sécurisés)
- KO : Le résultat global de l’audit des sécurisations n’est pas conforme à l’attendu (LINKED\_CHECK\_SECURISATION.KO=Échec d’audit de la vérification des journaux sécurisés)
- FATAL : Erreur fatal lors de l’exécution du workflow d’audit des sécurisations (LINKED\_CHECK\_SECURISATION.FATAL=Erreur technique lors d’audit de la vérification des journaux sécurisés)
- WARNING : Le résultat du workflow d’audit des sécurisations est en avertissement (LINKED\_CHECK\_SECURISATION.WARNING=Avertissement lors d’audit de la vérification des journaux sécurisés)

### 14.2.6. Structure du workflow



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





## 14.3. Workflow de l'audit de l'existence et de l'intégrité des fichiers

Cette section décrit le processus (workflow) d'audit de l'existence et de l'intégrité des fichiers mis en place dans la solution logicielle Vitam.

Celui-ci est défini dans le fichier « *DefaultAuditWorkflow.json* » (situé ici : sources/processing/processing-management/src/main/resources/workflows).

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.

Le processus d'audit prend comme point d'entrée l'identifiant d'un tenant, l'identifiant d'un service producteur ou une requête DSL sur des Units. Il est possible de lancer un audit de l'existence des fichiers uniquement, ou de lancer un audit vérifiant l'existence et l'intégrité des fichiers en même temps.

### 14.3.1. Processus d'audit d'existence des fichiers (vision métier)

Pour chaque objet du tenant choisi ou chaque objet appartenant au service producteur, l'audit va vérifier :

- Que la stratégie de stockage définie dans le groupe d'objet existe bien sur la plateforme
- Que tous les fichiers correspondant aux objets existent sur les offres déclarées dans la stratégie, dans un nombre de copie spécifié via la stratégie de stockage

### 14.3.2. Processus d'audit d'intégrité des fichiers (vision métier)

Si l'audit d'intégrité des objets est lancé, le processus va également vérifier que les empreintes des objets stockés en base de données sont bien les mêmes que les empreintes fournies par les espaces de stockage, alors recalculées à la demande de l'audit.

Dans une première étape technique, le processus prépare la liste des groupes d'objets à auditer afin de paralléliser la tâche. Dans un second temps, il effectue la vérification elle-même. A la fin du processus d'audit, un rapport est généré.

### 14.3.3. Processus de préparation de l'audit (STP\_PREPARE\_AUDIT)

#### 14.3.3.1. Vérification des seuils de limitation de traitement des unités archivistiques CHECK\_DISTRIBUTION\_THRESHOLD

- **Règle** : tâche consistant à vérifier les seuils de limitation de traitement des unités archivistiques
- **Type** : bloquant
- **Statuts** :
  - OK : la vérification des seuils de limitation de traitement des unités archivistiques a bien été effectuée (CHECK\_DISTRIBUTION\_THRESHOLD.OK = Succès de la vérification des seuils de limitation de traitement des unités archivistiques)
  - KO : une incohérence a été détectée entre le seuil et le nombre d'unités archivistiques à traiter. (CHECK\_DISTRIBUTION\_THRESHOLD.KO = Échec de la vérification des seuils de limitation de traitement des unités archivistiques)
  - FATAL : une erreur technique est survenue lors de la vérification des seuils de limitation de traitement des unités archivistiques (CHECK\_DISTRIBUTION\_THRESHOLD.FATAL = Erreur technique lors de la vérification des seuils de limitation de traitement des unités archivistiques)

### 14.3.3.2. Récupération de la configuration des stratégie de stockage de la plateforme PREPARE\_STORAGE\_STRATEGIES (PrepareStorageStrategiesPlugin.java)

- **Règle** : tâche consistant à charger la configuration des stratégies de la plateforme
- **Type** : bloquant
- **Statuts** :
  - OK : les données ont été récupérées et enregistrées dans un fichier de travail avec succès (PREPARE\_STORAGE\_STRATEGIES.OK = Succès de la récupération de la liste des stratégies de stockage)
  - FATAL : une erreur technique est survenue lors de la création de la liste (PREPARE\_STORAGE\_STRATEGIE.FATAL = Erreur technique lors de la récupération de la liste des stratégies de stockage)

### 14.3.3.3. Création de la liste des groupes d'objets LIST\_OBJECTGROUP\_ID (PrepareAuditActionHandler.java)

- **Règle** : tâche consistant à créer la liste des groupes d'objets à auditer
- **Type** : bloquant
- **Statuts** :
  - OK : la liste a été créée avec succès (LIST\_OBJECTGROUP\_ID.OK = Succès de la création de la liste des groupes d'objets à auditer)
  - FATAL : une erreur technique est survenue lors de la création de la liste (LIST\_OBJECTGROUP\_ID.FATAL = Erreur technique lors de la création de la liste des groupes d'objets à auditer)

## 14.3.4. Processus d'exécution de l'audit (STP\_AUDIT)

### 14.3.4.1. Audit de la vérification des objets AUDIT\_CHECK\_OBJECT (AuditCheckObjectPlugin.java)

- **Règle** : tâche consistant à organiser et lancer l'action d'audit de la vérification des objets.
- **Type** : bloquant
- **Statuts** :
  - OK : l'action d'audit de la vérification des objets s'est terminée en OK (AUDIT\_CHECK\_OBJECT.OK = Succès de l'audit de la vérification des objets)
  - KO : l'action d'audit de la vérification des objets s'est terminée en KO (AUDIT\_CHECK\_OBJECT.KO = Échec de l'audit de la vérification des objets : au moins un objet demandé n'existe pas ou des stratégies de stockage n'existent pas sur la plateforme)
  - FATAL : une erreur technique est survenue lors du lancement de l'action d'audit (AUDIT\_CHECK\_OBJECT.FATAL = Erreur technique lors de l'audit de la vérification des objets)

La tâche Audit\_Check\_Object contient le traitement suivant :

### 14.3.4.2. Audit de l'existence et de l'intégrité des objets AUDIT\_CHECK\_OBJECT.AUDIT\_CHECK\_OBJECT

- **Règle** : traitement consistant à auditer l'existence et l'intégrité des objets
  - La stratégie de stockage du groupe d'objets est conforme à celle du moteur de stockage
  - Les fichiers correspondant aux objets, déclarés dans le groupe d'objets, existent bien sous le même nom dans les offres de stockage
- **Type** : bloquant



- **Statuts :**
  - OK : tous les objets de tous les groupes d'objet audités existent bien sur les offres de stockage et leurs empreintes sont identiques entre celles enregistrées en base de données et celles recalculées par les offres de stockage (AUDIT\_CHECK\_OBJECT.AUDIT\_CHECK\_OBJECT.OK = Succès de l'audit de l'existence et de l'intégrité des objets)
  - KO : au moins un objet n'est pas intègre pour les groupes d'objets, audités (AUDIT\_CHECK\_OBJECT.AUDIT\_CHECK\_OBJECT.KO = Échec de l'audit de l'existence de fichiers)
  - Warning : il n'y a aucun objet à auditer (cas par exemple d'un producteur sans objets) (AUDIT\_CHECK\_OBJECT.AUDIT\_CHECK\_OBJECT.WARNING = Avertissement lors de l'audit de l'existence et de l'intégrité des objets : au moins un élément n'a pas d'objet binaire à vérifier)
  - FATAL : une erreur technique est survenue lors de l'audit de l'existence et de l'intégrité des objets (AUDIT\_CHECK\_OBJECT.AUDIT\_FILE\_EXISTING.FATAL = Erreur technique lors de l'audit de l'existence et de l'intégrité des objets)

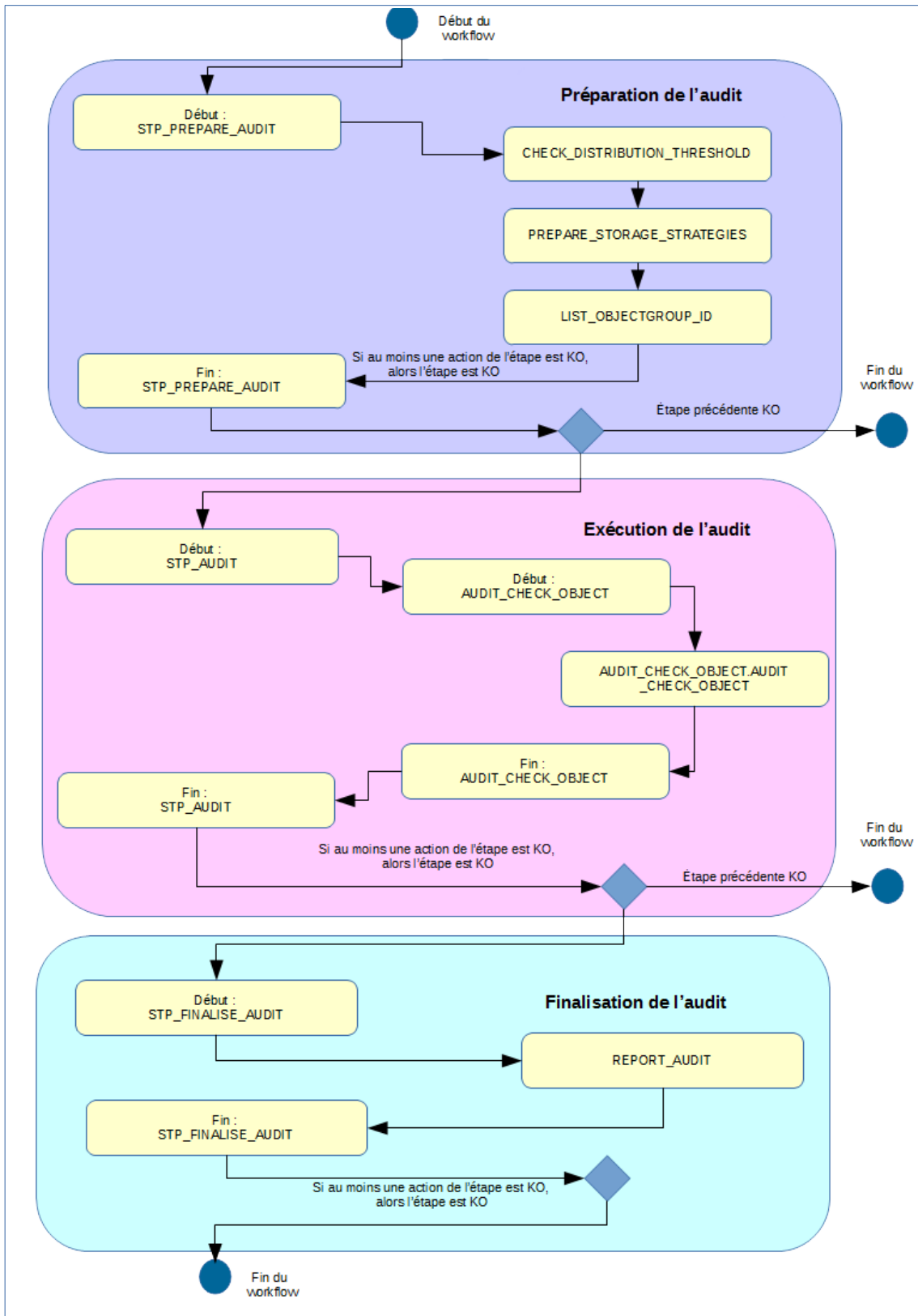
#### 14.3.5.Processus de finalisation de l'audit (STP\_FINALISE\_AUDIT)

##### **Notification de la fin d'audit REPORT\_AUDIT (GenerateAuditReportActionHandler.java)**

- **Règle :** tâche consistant à générer le rapport d'audit
- **Type :** bloquant
- **Statuts :**
  - OK : le rapport a été créé avec succès (REPORT\_AUDIT.OK = Succès de la notification de la fin de l'audit)
  - FATAL : une erreur technique est survenue lors de la création du rapport d'audit (REPORT\_AUDIT.OK.FATAL = Erreur technique lors de la notification de la fin de l'audit)

### 14.3.6. Structure du workflow de l'audit de l'existence et de l'intégrité des fichiers

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



### **14.3.7. Rapport d'audit**

Le rapport d'audit est un fichier JSON généré par la solution logicielle Vitam lorsqu'une opération d'audit se termine. Cette section décrit la manière dont ce rapport est structuré.

### **14.3.7.1. Exemple de JSON : rapport d'audit d'intégrité et d'existence KO**

14.3.7.2.

```

"tenant": 8,
"evId": "aeaaaaabchemhquaajkyaloieebysqaaaq",
"evType": "PROCESS_AUDIT",
"outcome": "KO",
"outDetail": "AUDIT_CHECK_OBJECT.AUDIT_CHECK_OBJECT_KO",
"outMsg": "Échec de l'audit de l'existence et de l'intégrité des objets Detail= OK:3 KO:1",
"rightsStatementIdentifier": {
  "AccessContract": "AC-000006"
},
"evDetData": null
}

```

### Operation Summary

```

"evStartDateTime": "2019-11-06T14:04:41.360",
"evEndDateTime": "2019-11-06T14:04:44.841",
"reportType": "AUDIT",
"vitamResults": {
  "OK": 3,
  "KO": 1,
  "WARNING": 0,
  "total": 4
},
"extendedInfo": {
  "nbObjectGroups": 4,
  "nbObjects": 4,
  "spis": ["aeaaaaabchemhquaaaealoidfgj4iaaaaq"],
  "globalResults": {
    "objectGroupsCount": {
      "OK": 3,
      "WARNING": 0,
      "KO": 1
    },
    "objectsCount": {
      "OK": 3,
      "WARNING": 0,
      "KO": 1
    }
  },
  "originatingAgencyResults": {
    "RATP": {
      "objectGroupsCount": {
        "OK": 3,
        "WARNING": 0,
        "KO": 1
      },
      "objectsCount": {
        "OK": 3,
        "WARNING": 0,
        "KO": 1
      }
    }
  }
}

```

### Report Summary





- « evType » : code du type de l'opération
- « outcome »: statut de l'événement
- « outcomeMsg » : détail du résultat de l'événement.
- « rightsStatementIdentifiant »:identifiant des données référentielles en vertu desquelles l'opération peut s'exécuter.
- « evDetData » : détails des données l'événement

La partie « ReportSummary », c'est-à-dire le bloc situé sous la racine du rapport et correspondant au résumé du rapport est composée des champs suivants :

- « evStartDateTime » : date du début de l'opération (evDateTime de l'event master de l'opération dans le Journal des Opérations)
- « evEndDateTime » : date de fin d'opération (dernier evDateTime dans l'opération dans le Journal des Opérations)
- « reportType »: correspond au modèle du rapport. Chaque opération de masse (audit, élimination...) aura un « reportType » associé
- « vitamResults » : est le nombre de OK, KO, WARNING de l'opération ainsi que le total de ces 3 statuts.
- « extendedInfo » : partie libre où chaque type de rapport pourra ajouter des informations qui lui est propre. Ici on y trouve :
  - le nombre total de groupes d'objets et d'objets audités pour l'ensemble de l'opération (« nbObjectGroups », « nbObjects »)
  - les identifiants des opérations d'entrée concernant ces groupes d'objets et objets (« opis »)
  - un résultat global (« globalResults ») remontant le nombre de groupes d'objets et d'objets aux statuts « OK », « KO » et « WARNING » (« objectGroupsCount » et « objectsCount »)
  - Un résultat par service producteur (« originatingAgencyResultats ») remontant le nombre de groupes d'objets et d'objets aux statuts « OK », « KO » et « WARNING » (« objectGroupsCount » et « objectsCount »).

La partie « Context » correspond à la requête DSL utilisée pour créer la distribution sur chaque unité archivistique. On y retrouve outre la requête elle-même le type d'audit réalisée (« auditActions », ici AUDIT\_FILE\_INTEGRITY), l'élément sur lequel l'audit a été lancé. Celui-ci peut-être par « tenant », ou par « originatingagency » (« auditType », ici ORIGINATINGAGENCY) et enfin, identifiant de l'élément (tenant ou service producteur), (« objectId », ici « RATP »).

La partie « ReportDetail » contient les détails pour chaque groupe d'objets et objets audités dont le statut est « KO »

La liste des éléments singuliers générant un avertissement « auditWarning » décrit les identifiants des services producteurs ayant généré un avertissement. Dans le cas de l'audit de l'existence des fichiers, une alerte correspond au fait qu'un service producteur n'a aucun objet à auditer. Cette liste est donc l'ensemble des services producteurs concernés par l'audit mais dont il n'existe aucun objet à auditer.



## 14.4.Workflow d'audit de cohérence des fichiers

Cette section décrit le processus (workflow) d'audit de cohérence des fichiers mis en place dans la solution logicielle Vitam.

Celui-ci est défini dans le fichier « *EvidenceAuditWorkflow.json* » (situé ici : sources/processing/processing-management/src/main/resources/workflows).

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.

### 14.4.1.Processus d'audit de cohérence des fichiers (vision métier)

Le processus d'audit de cohérence permet de vérifier la cohérence entre les signatures calculées pour chaque élément audité, en comparant celle présente dans le journal sécurisé, avec celle présente dans la base de donnée, et celle de l'offre de stockage.

L'audit s'applique au niveau des unités archivistiques, des objets et des groupes d'objets.

### 14.4.2.Processus de préparation d'audit (STP\_EVIDENCE\_AUDIT\_PREPARE)

#### 14.4.2.1. Vérification des seuils de limitation de traitement des unités archivistiques CHECK\_DISTRIBUTION\_THRESHOLD

- **Règle** : tâche consistant à vérifier les seuils de limitation de traitement des unités archivistiques
- **Type** : bloquant
- **Statuts** :
  - OK : la vérification des seuils de limitation de traitement des unités archivistiques a bien été effectuée (CHECK\_DISTRIBUTION\_THRESHOLD.OK = Succès de la vérification des seuils de limitation de traitement des unités archivistiques)
  - KO : une incohérence a été détectée entre le seuil et le nombre d'unités archivistiques à traiter. (CHECK\_DISTRIBUTION\_THRESHOLD.KO = Échec de la vérification des seuils de limitation de traitement des unités archivistiques)
  - FATAL : une erreur technique est survenue lors de la vérification des seuils de limitation de traitement des unités archivistiques (CHECK\_DISTRIBUTION\_THRESHOLD.FATAL = Erreur technique lors de la vérification des seuils de limitation de traitement des unités archivistiques)

#### 14.4.2.2. Récupération de la configuration des stratégies de stockage de la plateforme PREPARE\_STORAGE\_STRATEGIES (PrepareStorageStrategiesPlugin.java)

- **Règle** : tâche consistant à charger la configuration des stratégies de la plateforme
- **Type** : bloquant
- **Statuts** :
  - OK : les données ont été récupérées et enregistrées dans un fichier de travail avec succès (PREPARE\_STORAGE\_STRATEGIES.OK = Succès de la récupération de la liste des stratégies de stockage)
  - FATAL : une erreur technique est survenue lors de la création de la liste (PREPARE\_STORAGE\_STRATEGIE.FATAL = Erreur technique lors de la récupération de la liste des stratégies de stockage)

#### 14.4.2.3. Création de la liste à auditer EVIDENCE\_AUDIT\_LIST\_OBJECT (EvidenceAuditPrepare.java)

- **Règle** : tâche consistant à établir la liste des objets à auditer
- **Type** : bloquant
- **Statuts** :
  - OK : la liste a été créée avec succès (EVIDENCE\_AUDIT\_LIST\_OBJECT.OK = Succès de la création de la liste à auditer)
  - KO : échec de la création de la liste à auditer (EVIDENCE\_AUDIT\_LIST\_OBJECT.KO = Échec de la création de la liste à auditer)
  - FATAL : une erreur technique est survenue lors de la création de la liste (EVIDENCE\_AUDIT\_LIST\_OBJECT.FATAL = Erreur technique lors de la création de la liste à auditer)

#### 14.4.3. Processus de récupération des données de la base (STP\_EVIDENCE\_AUDIT\_CHECK\_DATABASE)

##### 14.4.3.1. Récupération des données dans la base de données EVIDENCE\_AUDIT\_CHECK\_DATABASE (EvidenceAuditDatabaseCheck.java)

- **Règle** : tâche consistant à récupérer les informations nécessaires à l'audit dans la base de données
- **Type** : bloquant
- **Statuts** :
  - OK : la récupération des données dans la base de données est un succès (EVIDENCE\_AUDIT\_CHECK\_DATABASE.OK = Succès de la récupération des données dans la base de données)
  - KO : la récupération des données dans la base de donnée est un échec (EVIDENCE\_AUDIT\_CHECK\_DATABASE.KO = Échec de la récupération des données dans la base de données)
  - FATAL : une erreur technique est survenue dans la récupération des données dans la base de données (EVIDENCE\_AUDIT\_CHECK\_DATABASE.FATAL = Erreur technique lors de la récupération des données dans la base de données)
  - WARNING : avertissement lors de la récupération des données dans la base de données (EVIDENCE\_AUDIT\_CHECK\_DATABASE.WARNING = Avertissement lors de la récupération des données dans la base de données)

#### 14.4.4. Processus de préparation des signatures à partir des fichiers sécurisés (STP\_EVIDENCE\_AUDIT\_LIST\_SECURED\_FILES\_TO\_DOWNLOAD)

##### 14.4.4.1. Préparation de la liste des signatures dans les fichiers sécurisés EVIDENCE\_AUDIT\_LIST\_SECURED\_FILES\_TO\_DOWNLOAD (EvidenceAuditListSecuredFiles.java)

- **Règle** : tâche consistant à préparer la liste des signatures des objets, groupes d'objets ou unités archivistiques archivées, dans les fichiers sécurisés
- **Type** : bloquant
- **Statuts** :
  - OK : la préparation de la liste des signatures dans les fichiers sécurisés est un succès (EVIDENCE\_AUDIT\_LIST\_SECURED\_FILES\_TO\_DOWNLOAD.OK = Succès de la préparation)

- de la liste des signatures dans les fichiers sécurisés)
- KO : la préparation de la liste des signatures dans les fichiers sécurisés est un échec (EVIDENCE\_AUDIT\_LIST\_SECURED\_FILES\_TO\_DOWNLOAD.KO = Échec de la préparation de la liste des signatures dans les fichiers sécurisés)
- WARNING : avertissement lors de la préparation de la liste des signatures (EVIDENCE\_AUDIT\_LIST\_SECURED\_FILES\_TO\_DOWNLOAD.WARNING = Avertissement lors de la préparation de la liste des signatures dans les fichiers sécurisés)
- FATAL : une erreur technique est survenue lors de la préparation de la liste des signatures dans les fichiers sécurisés (EVIDENCE\_AUDIT\_LIST\_SECURED\_FILES\_TO\_DOWNLOAD.FATAL = Erreur technique lors de la préparation de la liste des signatures dans les fichiers sécurisés)

#### 14.4.5. Processus d'extraction des signatures à partir des fichiers sécurisés (STP\_EVIDENCE\_AUDIT\_EXTRACT\_ZIP\_FILE)

##### 14.4.5.1. Extraction des signatures à partir des fichiers sécurisés EVIDENCE\_AUDIT\_EXTRACT\_ZIP\_FILE (EvidenceAuditExtractFromZip.java)

- **Règle** : tâche consistant à extraire les signatures des objets, groupes d'objets ou unités archivistiques archivées, dans les fichiers sécurisés
- **Type** : bloquant
- **Statuts** :
  - OK : l'extraction des signatures à partir des fichiers sécurisés est un succès (EVIDENCE\_AUDIT\_EXTRACT\_ZIP\_FILE.OK = Succès de l'extraction des signatures à partir des fichiers sécurisés)
  - KO : l'extraction des signatures à partir des fichiers sécurisés est un échec (EVIDENCE\_AUDIT\_EXTRACT\_ZIP\_FILE.KO = Échec de l'extraction des signatures à partir des fichiers sécurisés)
  - WARNING : avertissement lors de l'extraction des signatures à partir des fichiers sécurisés (STP\_EVIDENCE\_AUDIT\_EXTRACT\_ZIP\_FILE.WARNING = Avertissement lors de l'extraction des signatures à partir des fichiers sécurisés)
  - FATAL : une erreur technique est survenue lors de l'extraction des signatures à partir des fichiers sécurisés (EVIDENCE\_AUDIT\_EXTRACT\_ZIP\_FILE.FATAL = Erreur technique lors de l'extraction des signatures à partir des fichiers sécurisés)

#### 14.4.6. Processus de préparation des rapports pour chaque objet, groupe d'objets ou unité audité (STP\_EVIDENCE\_AUDIT\_PREPARE\_GENERATE\_REPORTS)

##### 14.4.6.1. Création du rapport pour chaque unité archivistique ou objet ou groupe d'objets EVIDENCE\_AUDIT\_PREPARE\_GENERATE\_REPORTS (EvidenceAuditGenerateReports.java)

- **Règle** : tâche consistant à créer le rapport pour chaque unité archivistique, objet ou groupe d'objets audité
- **Type** : bloquant
- **Statuts** :
  - OK : La création du rapport pour chaque unité archivistique ou objet ou groupe d'objets est un succès (EVIDENCE\_AUDIT\_PREPARE\_GENERATE\_REPORTS.OK = Succès de la création du rapport pour chaque unité archivistique ou objet ou groupe d'objets)
  - KO : la création du rapport pour chaque unité archivistique ou objet ou groupe d'objets est un échec (EVIDENCE\_AUDIT\_PREPARE\_GENERATE\_REPORTS.KO = Échec de la création

du rapport pour chaque unité archivistique ou objet ou groupe d'objets)

- FATAL : une erreur technique est survenue de la création du rapport pour chaque unité archivistique ou objet ou groupe d'objets (EVIDENCE\_AUDIT\_PREPARE\_GENERATE\_REPORTS.FATAL = Erreur technique lors de la création du rapport pour chaque unité archivistique ou objet ou groupe d'objets)
- WARNING : avertissement lors de la création du rapport pour chaque unité archivistique ou objet ou groupe d'objets (EVIDENCE\_AUDIT\_PREPARE\_GENERATE\_REPORTS.WARNING = Avertissement lors de la création du rapport pour chaque unité archivistique ou objet ou groupe d'objets)

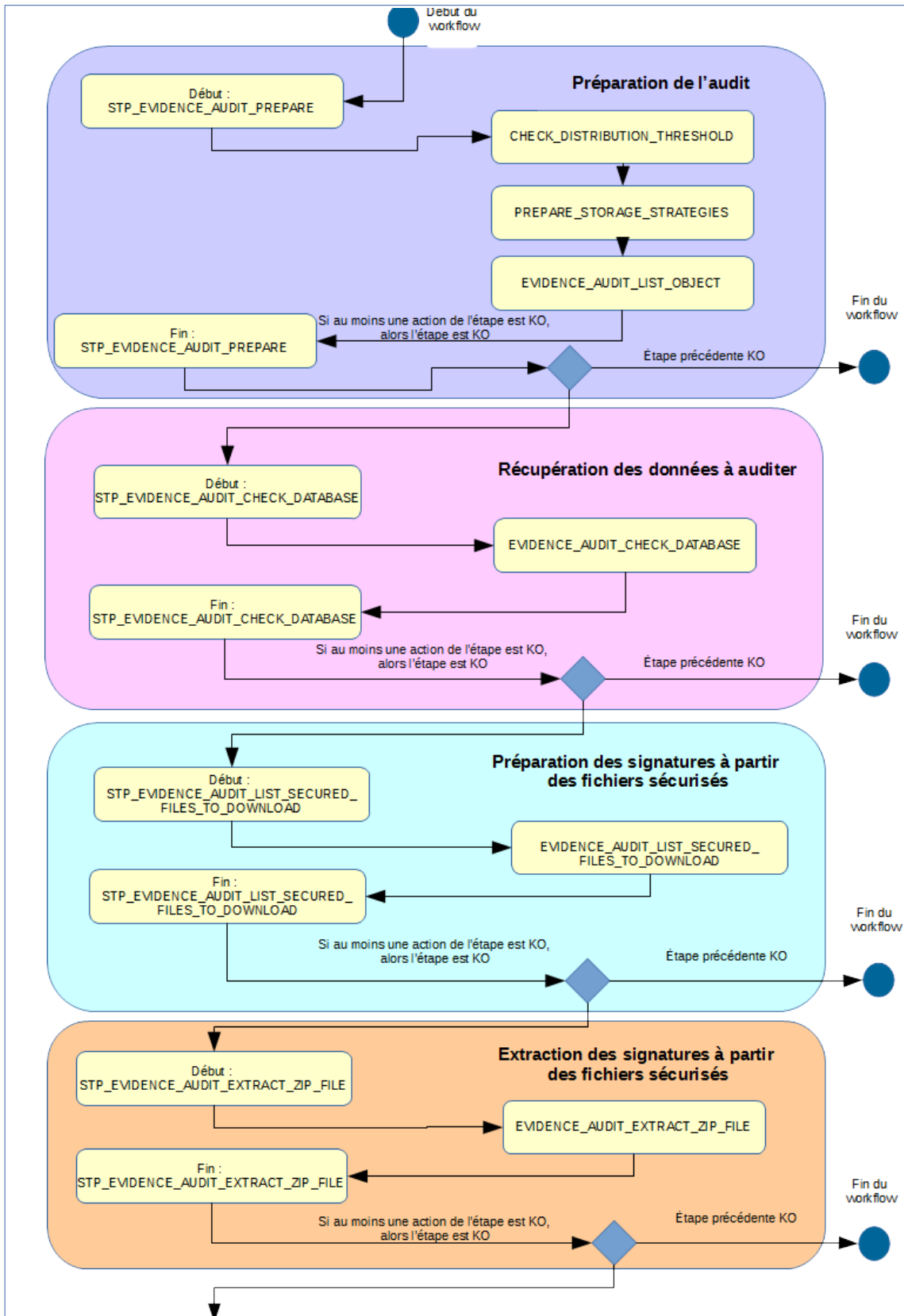
#### 14.4.7. Processus de finalisation d'audit et génération du rapport final (STP\_EVIDENCE\_AUDIT\_FINALIZE)

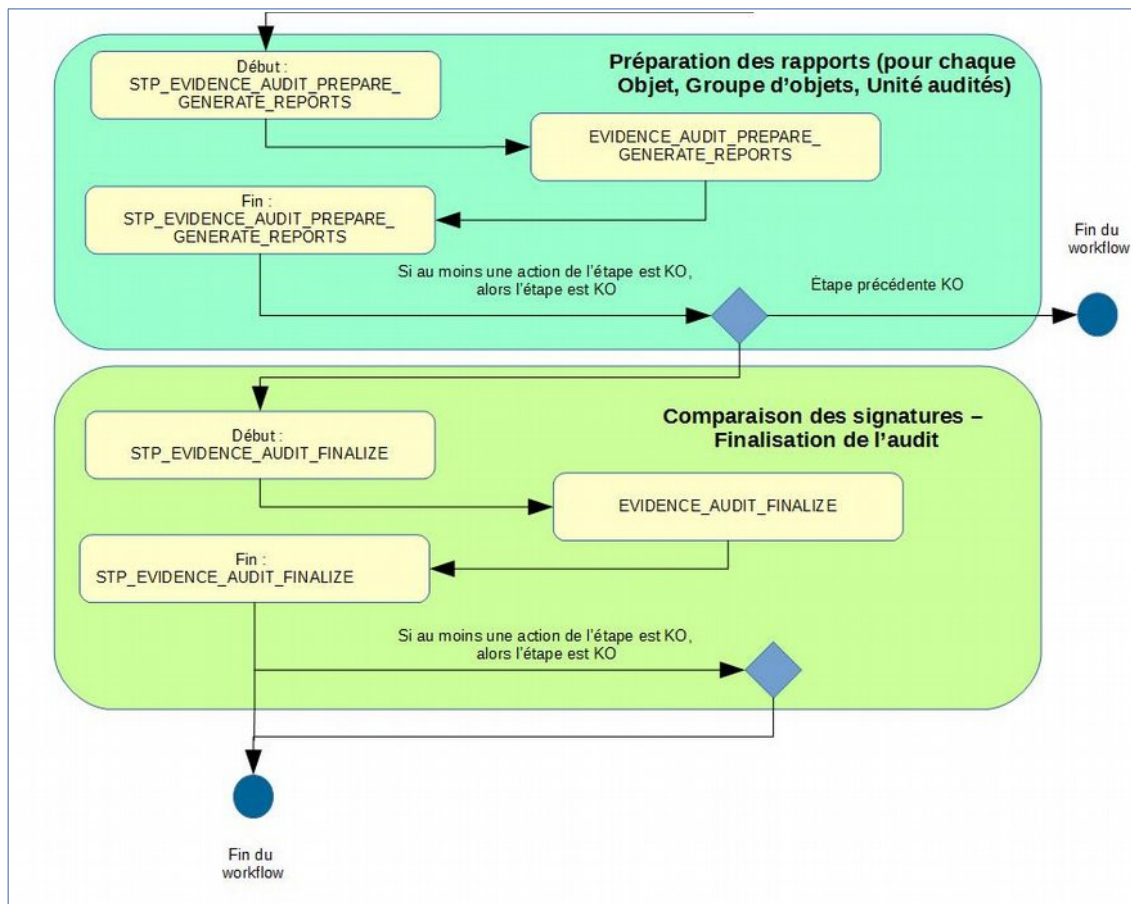
##### 14.4.7.1. Création du rapport d'audit de cohérence EVIDENCE\_AUDIT\_FINALIZE (EvidenceAuditFinalize.java)

- **Règle** : tâche consistant à créer le rapport permettant de comparer les signatures extraites des fichiers sécurisés avec les données de la base de données et de l'offre de stockage
- **Type** : bloquant
- **Statuts** :
  - OK : la création du rapport d'audit de cohérence est un succès (EVIDENCE\_AUDIT\_FINALIZE.OK = Succès de la création du rapport d'audit de cohérence)
  - KO : la création du rapport d'audit de cohérence est un échec (EVIDENCE\_AUDIT\_FINALIZE.KO = Échec de la création du rapport d'audit de cohérence)
  - FATAL : une erreur technique est survenue lors de la création du rapport d'audit de cohérence (EVIDENCE\_AUDIT\_FINALIZE.FATAL = Erreur technique lors de la création du rapport d'audit de cohérence)

### 14.4.8. Structure du workflow d'audit de cohérence des fichiers

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









#### 14.4.9.2. Détail du rapport

La partie « OperationSummary », c’est-à-dire le bloc à la racine du rapport et correspondant au résumé de l’opération (sans indentation) est composée des champs suivants :

- « tenant » : tenant sur lequel l’opération d’audit a été lancée
- « evId » : identifiant de l’événement
- « evType » : code du type de l’opération
- « outcome »: statut de l’événement
- « outcomeMsg » : détail du résultat de l’événement.
- « rightsStatementIdentifier »:identifiant des données référentielles en vertu desquelles l’opération peut s’exécuter.
- « evDetData » : détails des données l’événement

La partie « ReportSummary », c’est-à-dire le bloc situé sous la racine du rapport et correspondant au résumé du rapport est composée des champs suivants :

- « evStartDateTime » : date du début de l’opération (evDateTime de l’event master de l’opération dans le journal des opérations) ;
- « evEndDateTime » : date de fin d’opération (dernier evDateTime dans l’opération dans le journal des opérations) ;
- « reportType »: correspond au modèle du rapport. Chaque opération de masse (audit, élimination...) a un « reportType » associé. Dans le cas d’un audit de cohérence, la valeur du champ est égale à « EVIDENCE\_AUDIT » ;
- « vitamResults » : est le nombre de OK, KO, WARNING de l’opération ainsi que le total de ces 3 statuts ;
- « extendedInfo » : partie libre où chaque type de rapport contient des informations qui lui est



propre. Ici on y trouve :

- le nombre total d'unités archivistiques auditées pour l'ensemble de l'opération (« nbArchiveUnits »),
- le nombre total de groupes d'objets audités pour l'ensemble de l'opération (« nbObjectGroups »),
- le nombre total d'objets audités pour l'ensemble de l'opération (« nbObjects »),
- un résultat global (« globalResults ») remontant le nombre d'unités archivistiques, de groupes d'objets et d'objets aux statuts « OK », « KO » et « WARNING » (« objectGroupsCount », « archiveUnitsCount » et « objectsCount »)

La partie « Context » correspond à la requête DSL utilisée pour créer la distribution sur chaque unité archivistique. Dans le rapport d'un audit de cohérence, on y retrouve uniquement la requête.

La partie « ReportDetail » contient les détails de l'opération d'audit de cohérence effectuée sur chacune des unités archivistiques, des groupes d'objets techniques et/ou des objets techniques, uniquement quand ces derniers sont aux statuts « KO » ou « WARNING ».

- « identifier » : identifiant de l'objet, groupe d'objets ou unité archivistique audité ;
- « status » : résultat de l'opération pour le groupe d'objets, l'objet ou l'unité archivistique concerné, pouvant correspondre aux valeurs suivantes : « KO » ou « WARNING » ;
- « objectType » : type de l'objet audité, pouvant correspondre aux valeurs suivantes : « UNIT », « OBJECTGROUP » ou « OBJECT » ;
- « message » : message qui signale une incohérence entre les signatures des fichiers sécurisés, des offres de stockage et de la base de données ;
- « securedHash » : empreinte du journal sécurisé de l'unité archivistique, objet ou groupe d'objets ;
- « offersHashes » : signatures de l'élément audité (unité archivistique ou groupes d'objets techniques) dans les différentes offres de stockage ;
- « strategyId » : identifiant de la stratégie de stockage.

## 14.5. Workflow de l'audit correctif

Cette section décrit le processus (workflow) de l'audit correctif des fichiers mis en place dans la solution logicielle Vitam. Cette opération peut être effectuée suite à l'audit de cohérence. Cette action est effectuée via API. Cependant, dans un souci de démonstration et à titre d'écran expérimental un lancement de l'audit correctif peut-être effectué à partir de l'IHM recette (cf 2.4 Test Audit correctif).

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.

### 14.5.1. Processus d'audit correctif des fichiers (vision métier)

Cette action a pour but de corriger des objets défectueux. Cette correction fait suite à l'audit de cohérence. L'audit correctif veut résoudre une situation anormale : l'empreinte des fichiers ne correspond plus à l'empreinte enregistrée en base et sur les espaces de stockage. Afin de corriger et de rectifier ces erreurs, l'audit correctif va supprimer la copie défectueuse (si cela est possible) et restaurer à partir de la stratégie de stockage une copie saine (copie dont l'empreinte est OK). Les premières parties du workflow de correction de l'audit suivent les mêmes étapes que l'audit de cohérence, le traitement de correction et de récupération des données dans les offres de stockage ou la base de données intervient dans la seconde partie du workflow.

### 14.5.2. Processus de préparation d'audit (STP\_EVIDENCE\_AUDIT\_PREPARE)

#### 14.5.2.1. Vérification des seuils de limitation de traitement des unités archivistiques CHECK\_DISTRIBUTION\_THRESHOLD

- **Règle** : étape consistant à vérifier les seuils de limitation de traitement des unités archivistiques
- **Type** : bloquant
- **Statuts** :
  - OK : la vérification des seuils de limitation de traitement des unités archivistiques a bien été effectuée (CHECK\_DISTRIBUTION\_THRESHOLD.OK = Succès de la vérification des seuils de limitation de traitement des unités archivistiques)
  - KO : une incohérence a été détectée entre le seuil et le nombre d'unités archivistiques à traiter (CHECK\_DISTRIBUTION\_THRESHOLD.KO = Échec de la vérification des seuils de limitation de traitement des unités archivistiques)
  - FATAL : une erreur technique est survenue lors de la vérification des seuils de limitation de traitement des unités archivistiques (CHECK\_DISTRIBUTION\_THRESHOLD.FATAL = Erreur technique lors de la vérification des seuils de limitation de traitement des unités archivistiques)

#### 14.5.2.2. Récupération de la configuration des stratégies de stockage de la plateforme PREPARE\_STORAGE\_STRATEGIES (PrepareStorageStrategiesPlugin.java)

- **Règle** : tâche consistant à charger la configuration des stratégies de la plateforme
- **Type** : bloquant
- **Statuts** :
  - OK : les données ont été récupérées et enregistrées dans un fichier de travail avec succès (PREPARE\_STORAGE\_STRATEGIES.OK = Succès de la récupération de la liste des stratégies de stockage)
  - FATAL : une erreur technique est survenue lors de la création de la liste (PREPARE\_STORAGE\_STRATEGIES.FATAL = Erreur technique lors de la récupération de la liste des stratégies de stockage)

### 14.5.2.3. Création de la liste à auditer EVIDENCE\_AUDIT\_LIST\_OBJECT (EvidenceAuditPrepare.java)

- **Règle** : tâche consistant à établir la liste des objets à auditer
- **Type** : bloquant
- **Statuts** :
  - OK : la liste a été créée avec succès (EVIDENCE\_AUDIT\_LIST\_OBJECT.OK = Création de la liste à auditer)
  - KO : échec de la création de la liste à auditer (EVIDENCE\_AUDIT\_LIST\_OBJECT.KO = Échec lors de la création de la liste à auditer)
  - FATAL : une erreur technique est survenue lors de la création de la liste (EVIDENCE\_AUDIT\_LIST\_OBJECT.FATAL = Erreur technique lors de la création de la liste à auditer)

### 14.5.3. Processus de récupération des données de la base (STP\_EVIDENCE\_AUDIT\_CHECK\_DATABASE)

#### 14.5.3.1. Récupération des données dans la base de donnée EVIDENCE\_AUDIT\_CHECK\_DATABASE (EvidenceAuditDatabaseCheck.java)

- **Règle** : tâche consistant à récupérer les informations nécessaires à l'audit dans la base de données
- **Type** : bloquant
- **Statuts** :
  - OK : la récupération des données dans la base de données est un succès (EVIDENCE\_AUDIT\_CHECK\_DATABASE.OK = Succès de la récupération des données dans la base de données)
  - KO : la récupération des données dans la base de données est un échec (EVIDENCE\_AUDIT\_CHECK\_DATABASE.KO = Échec de la récupération des données dans la base de données)
  - FATAL : une erreur technique est survenue dans la récupération des données dans la base de données (EVIDENCE\_AUDIT\_CHECK\_DATABASE.FATAL = Erreur technique lors de la récupération des données dans la base de données)
  - WARNING : avertissement lors de la récupération des données dans la base de données (EVIDENCE\_AUDIT\_CHECK\_DATABASE.WARNING = Avertissement lors de la récupération des données dans la base de données)

### 14.5.4. Processus de préparation des signatures à partir des fichiers sécurisés (STP\_EVIDENCE\_AUDIT\_LIST\_SECURED\_FILES\_TO\_DOWNLOAD)

#### 14.5.4.1. Préparation de la liste des signatures dans les fichiers sécurisés EVIDENCE\_AUDIT\_LIST\_SECURED\_FILES\_TO\_DOWNLOAD (EvidenceAuditListSecuredFiles.java)

- **Règle** : tâche consistant à préparer la liste des signatures des objets, groupes d'objets ou unités archivistiques archivées, dans les fichiers sécurisés
- **Type** : bloquant
- **Statuts** :
  - OK : la préparation de la liste des signatures dans les fichiers sécurisés est un succès (EVIDENCE\_AUDIT\_LIST\_SECURED\_FILES\_TO\_DOWNLOAD.OK = Succès de la préparation)

- de la liste des signatures dans les fichiers sécurisés)
- KO : la préparation de la liste des signatures dans les fichiers sécurisés est un échec (EVIDENCE\_AUDIT\_LIST\_SECURED\_FILES\_TO\_DOWNLOAD.KO = Échec de la préparation de la liste des signatures dans les fichiers sécurisés)
- WARNING : avertissement lors de la préparation de la liste des signatures (EVIDENCE\_AUDIT\_LIST\_SECURED\_FILES\_TO\_DOWNLOAD.WARNING = Avertissement lors de la préparation de la liste des signatures dans les fichiers sécurisés)
- FATAL : une erreur technique est survenue lors de la préparation de la liste des signatures dans les fichiers sécurisés (EVIDENCE\_AUDIT\_LIST\_SECURED\_FILES\_TO\_DOWNLOAD.FATAL = Erreur technique lors de la préparation de la liste des signatures dans les fichiers sécurisés)

#### 14.5.5. Processus d'extraction des signatures à partir des fichiers sécurisés (STP\_EVIDENCE\_AUDIT\_EXTRACT\_ZIP\_FILE)

##### 14.5.5.1. Extraction des signatures à partir des fichiers sécurisés EVIDENCE\_AUDIT\_EXTRACT\_ZIP\_FILE (EvidenceAuditExtractFromZip.java)

- **Règle** : tâche consistant à extraire les signatures des objets, groupes d'objets ou unités archivistiques archivées, dans les fichiers sécurisés
- **Type** : bloquant
- **Statuts** :
  - OK : l'extraction des signatures à partir des fichiers sécurisés est un succès (EVIDENCE\_AUDIT\_EXTRACT\_ZIP\_FILE.OK = Succès de l'extraction des signatures à partir des fichiers sécurisés)
  - KO : l'extraction des signatures à partir des fichiers sécurisés est un échec (EVIDENCE\_AUDIT\_EXTRACT\_ZIP\_FILE.KO = Échec de l'extraction des signatures à partir des fichiers sécurisés)
  - WARNING : avertissement lors de l'extraction des signatures à partir des fichiers sécurisés (STP\_EVIDENCE\_AUDIT\_EXTRACT\_ZIP\_FILE.WARNING = Avertissement lors de l'extraction des signatures à partir des fichiers sécurisés )
  - FATAL : Une erreur technique est survenue lors de l'extraction des signatures à partir des fichiers sécurisés (EVIDENCE\_AUDIT\_EXTRACT\_ZIP\_FILE.FATAL = Erreur technique lors de l'extraction des signatures à partir des fichiers sécurisés)

#### 14.5.6. Processus de préparation des rapports pour chaque objet, groupe d'objets ou unité audité (STP\_EVIDENCE\_AUDIT\_PREPARE\_GENERATE\_REPORTS)

##### 14.5.6.1. Création du rapport pour chaque unité archivistique ou objet ou groupe d'objets EVIDENCE\_AUDIT\_PREPARE\_GENERATE\_REPORTS (EvidenceAuditGenerateReports.java)

- **Règle** : tâche consistant à créer le rapport pour chaque unité archivistique, objet ou groupe d'objets audité
- **Type** : bloquant
- **Statuts** :
  - OK : la création du rapport pour chaque unité archivistique ou objet ou groupe d'objets est un succès (EVIDENCE\_AUDIT\_PREPARE\_GENERATE\_REPORTS.OK = Succès de la création du rapport pour chaque unité archivistique ou objet ou groupe d'objets)
  - KO : la création du rapport pour chaque unité archivistique ou objet ou groupe d'objets est un échec (EVIDENCE\_AUDIT\_PREPARE\_GENERATE\_REPORTS.KO = Échec de la création du rapport)

- pour chaque unité archivistique ou objet ou groupe d'objets)
- FATAL : une erreur technique est survenue de la création du rapport pour chaque unité archivistique ou objet ou groupe d'objets (EVIDENCE\_AUDIT\_PREPARE\_GENERATE\_REPORTS.FATAL = Erreur technique lors de la création du rapport pour chaque unité archivistique ou objet ou groupe d'objets)
- WARNING : avertissement lors de la création du rapport pour chaque unité archivistique ou objet ou groupe d'objets (EVIDENCE\_AUDIT\_PREPARE\_GENERATE\_REPORTS.WARNING = Avertissement lors de la création du rapport pour chaque unité archivistique ou objet ou groupe d'objets)

#### 14.5.7. Processus de correction des signatures pour chaque objet, groupe d'objets ou unité auditée défailante (STP\_CORRECTIVE\_AUDIT)

##### 14.5.7.1. Vérification de la disponibilité des ressources

###### **CORRECTIVE\_AUDIT\_CHECK\_RESOURCE\_AVAILABILITY** (DataRectificationCheckResourceAvailability.java)

- **Règle** : tâche consistant à vérifier la disponibilité synchrone en lecture des objets, groupe d'objets ou unité auditée défailante. En cas d'indisponibilité d'un ou plusieurs objets l'exécution de l'étape courante est renvoyée au moteur de stockage pour ré-exécution ultérieure quand les données seront disponibles.
- **Type** : bloquant
- **Statuts** :
  - OK : les objets sont disponibles (CORRECTIVE\_AUDIT\_CHECK\_RESOURCE\_AVAILABILITY.OK = Succès de la vérification de la disponibilité des objets sur l'offre de stockage)
  - FATAL : une erreur technique est survenue lors de la vérification de la disponibilité des objets (CORRECTIVE\_AUDIT\_CHECK\_RESOURCE\_AVAILABILITY.FATAL = Erreur technique lors de la vérification de la disponibilité des objets sur l'offre de stockage)

##### 14.5.7.2. Correction des signatures pour chaque objet, groupe d'objets ou unité auditée défailante (CORRECTIVE\_AUDIT)

- **Règle** : tâche consistant à récupérer les données du/des fichiers corrompus auprès de la base de données ou d'une offre de stockage valide
- **Type** : bloquant
- **Statuts** :
  - OK : La correction des signatures de chaque unité archivistique ou objet ou groupe d'objets corrompue a bien été effectuée (CORRECTIVE\_AUDIT.OK = Succès de la correction des signatures de chaque unité archivistique ou objet ou groupe d'objets corrompue)
  - KO : la correction des signatures de chaque unité archivistique ou objet ou groupe d'objets corrompue n'a pas été effectuée (CORRECTIVE\_AUDIT.KO = Échec de la correction des signatures de chaque unité archivistique ou objet ou groupe d'objets corrompue)
  - FATAL : une erreur technique est survenue lors de la correction des signatures de chaque unité archivistique ou objet ou groupe d'objets corrompu (CORRECTIVE\_AUDIT.FATAL = Erreur technique lors de la correction des signatures de chaque unité archivistique ou objet ou groupe d'objets corrompue)
  - WARNING : avertissement lors de la correction des signatures de chaque unité archivistique ou objet ou groupe d'objets corrompu (CORRECTIVE\_AUDIT.WARNING = Avertissement lors de la correction des signatures de chaque unité archivistique ou objet ou groupe d'objets corrompue)

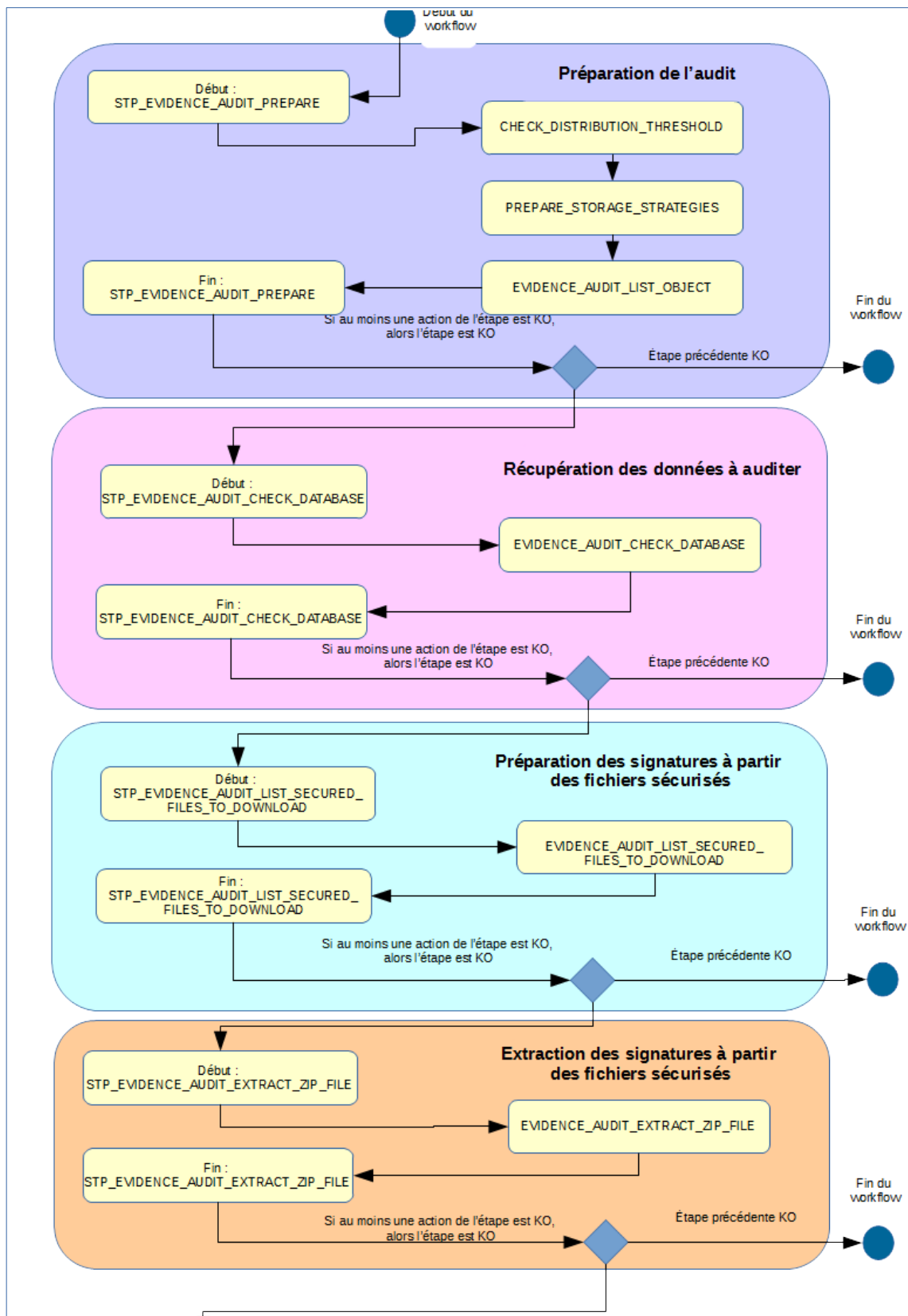
## 14.5.8. Processus de finalisation des corrections des signatures pour chaque objet, groupe d'objets ou unité auditée défailante (STP\_CORRECTION\_FINALIZE)

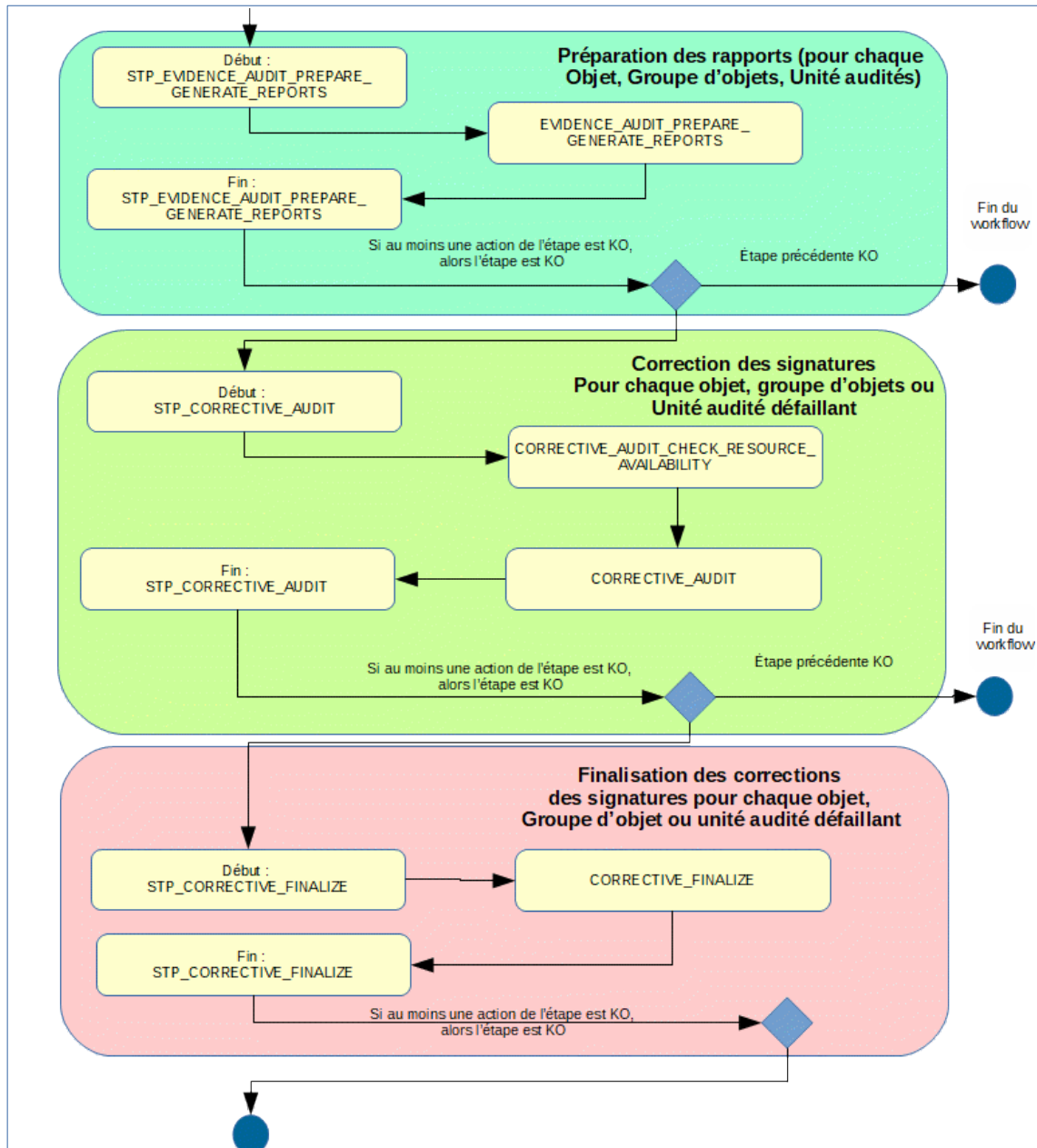
### 14.5.8.1. Finalisation des corrections des signatures pour chaque objet, groupe d'objets ou unité auditée, défailante (CORRECTION\_FINALIZE)

- **Règle** : tâche consistant à finaliser la correction des signatures sur les données du/des fichiers corrompus auprès de la base de données ou d'une offre de stockage valide
- **Type** : bloquant
- **Statuts** :
  - OK : la finalisation des corrections des signatures pour chaque objet, groupe d'objets ou unité auditée défailante a bien été effectuée (CORRECTION\_FINALIZE.OK = Succès du processus de finalisation des corrections des signatures pour chaque objet, groupe d'objets ou unité auditée défailante)
  - KO : la finalisation des corrections des signatures pour chaque objet, groupe d'objets ou unité auditée défailante n'a pas été effectuée (CORRECTION\_FINALIZE.KO = Échec du processus de finalisation des corrections des signatures pour chaque objet, groupe d'objets ou unité auditée défailante)
  - FATAL : une erreur technique est survenue lors de la finalisation des corrections des signatures pour chaque objet, groupe d'objets ou unité auditée défailante (CORRECTION\_FINALIZE.FATAL = erreur technique est survenue lors du processus de finalisation des corrections des signatures pour chaque objet, groupe d'objets ou unité auditée défailante)
  - WARNING : avertissement lors de la finalisation des corrections des signatures pour chaque objet, groupe d'objets ou unité auditée défailante (CORRECTION\_FINALIZE.WARNING = Avertissement lors du processus de finalisation des corrections des signatures pour chaque objet, groupe d'objets ou unité auditée défailante)

### 14.5.9. Structure du workflow de correction suite à audit

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







#### 14.5.10. Rapport d'audit correctif

Le rapport d'audit correctif est un fichier JSON généré par la solution logicielle Vitam lorsqu'une opération d'audit se termine. En cas de succès (RECTIFICATION\_AUDIT.OK), le rapport précise uniquement l'identifiant des unités archivistiques qui ont fait l'objet de l'audit.

##### 14.5.10.1. Exemple de JSON : rapport d'audit correctif

```
[{"Id":"aeaqaahgynv2aamukalojeze6iiaaaq","Type":"UNIT"}]
```

## 14.6. Workflow de relevé de valeur probante

Cette section décrit le processus (workflow) de relevé de valeur probante. L'objectif est de rendre prouvable toute opération effectuée sur toute unité archivistique ou tout objet qui lui est associé. Ce relevé de valeur probante réunit les éléments permettant de fournir à un auditeur externe une présomption de confiance dans ce qui lui est communiqué.

Toutes les étapes, tâches et traitement 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.

### 14.6.1. Processus de préparation du relevé de valeur probante (STP\_PROBATIVE\_VALUE\_PREPARE)

#### 14.6.1.1. Vérification des seuils de limitation de traitement des unités archivistiques CHECK\_DISTRIBUTION\_THRESHOLD

- **Règle** : étape consistant à vérifier les seuils de limitation de traitement des unités archivistiques
- **Type** : bloquant
- **Statuts** :
  - OK : la vérification des seuils de limitation de traitement des unités archivistiques a bien été effectuée (CHECK\_DISTRIBUTION\_THRESHOLD.OK = Succès de la vérification des seuils de limitation de traitement des unités archivistiques)
  - KO : une incohérence a été détectée entre le seuil et le nombre d'unités archivistiques à traiter. (CHECK\_DISTRIBUTION\_THRESHOLD.KO = Échec de la vérification des seuils de limitation de traitement des unités archivistiques)
  - FATAL : une erreur technique est survenue lors de la vérification des seuils de limitation de traitement des unités archivistiques (CHECK\_DISTRIBUTION\_THRESHOLD.FATAL = Erreur technique lors de la vérification des seuils de limitation de traitement des unités archivistiques)

#### 14.6.1.2. Création du fichier de distribution permettant de réaliser le relevé de valeur probante (PROBATIVE\_VALUE\_\_CREATE\_DISTRIBUTION\_FILE)

- **Règle** : étape consistant à créer un fichier de distribution permettant de réaliser le relevé de valeur probante
- **Type** : bloquant
- **Statuts** :
  - OK : la création du fichier de distribution a bien été effectuée (PROBATIVE\_VALUE\_CREATE\_DISTRIBUTION\_FILE.OK = Succès de la création du fichier de distribution permettant de réaliser le relevé de valeur probante)
  - KO : la création du fichier de distribution n'a pas été effectuée (PROBATIVE\_VALUE\_CREATE\_DISTRIBUTION\_FILE.KO = Échec lors de la création du fichier de distribution permettant de réaliser le relevé de valeur probante)
  - FATAL : une erreur technique est survenue lors de la création du fichier de distribution (PROBATIVE\_VALUE\_CREATE\_DISTRIBUTION\_FILE.FATAL = Une erreur technique est survenue lors de la création du fichier de distribution permettant de réaliser le relevé de valeur probante)

## 14.6.2. Processus d'alimentation du relevé de valeur probante (STP\_PROBATIVE\_VALUE\_ACTION)

### 14.6.2.1. Ajout d'une entrée dans le relevé de valeur probante (PROBATIVE\_VALUE\_CREATE\_PROBATIVE\_REPORT\_ENTRY)

- **Règle** : tâche consistant à créer une entrée dans le relevé de valeur probante
- **Type** : bloquant
- **Statuts** :
  - OK : la création d'une entrée dans le relevé de valeur probante à bien été effectué (PROBATIVE\_VALUE\_CREATE\_PROBATIVE\_REPORT\_ENTRY.OK = Succès de la création d'une entrée dans le relevé de valeur probante)
  - KO : la création d'une entrée dans le relevé de valeur probante n'a pas été effectué (PROBATIVE\_VALUE\_CHECK\_OBJECT\_GROUP.KO = Échec de la création d'une entrée dans le relevé de valeur probante)
  - WARNING : avertissement lors de la création d'une entrée dans le relevé de valeur probante (PROBATIVE\_VALUE\_CHECK\_OBJECT\_GROUP.WARNING = Avertissement lors de la création d'une entrée dans le relevé de valeur probante)
  - FATAL : une erreur technique est survenue lors de la création d'une entrée dans le relevé de valeur probante (PROBATIVE\_VALUE\_CHECK\_OBJECT\_GROUP.FATAL = Erreur technique lors de la création d'une entrée dans le relevé de valeur probante)

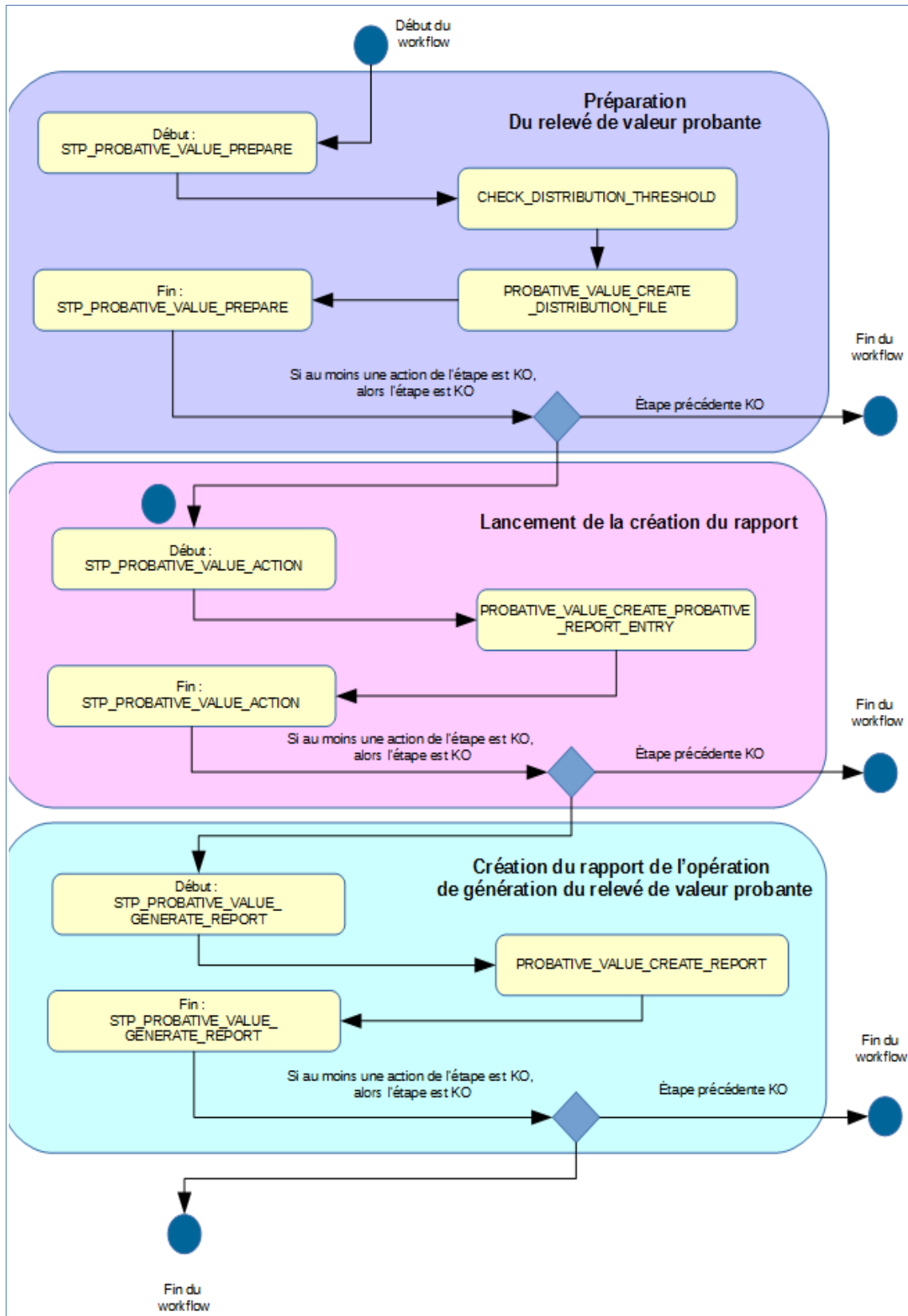
## 14.6.3. Processus de finalisation du relevé de valeur probante (STP\_PROBATIVE\_VALUE\_GENERATE\_REPORT)

### 14.6.3.1. Finalisation du relevé de valeur probante (PROBATIVE\_VALUE\_CREATE\_REPORT)

- **Règle** : tâche consistant à finaliser le relevé de valeur probante
- **Type** : bloquant
- **Statuts** :
  - OK : Finalisation du relevé de valeur probante à été effectué avec succès. (PROBATIVE\_VALUE\_CREATE\_REPORT.OK = Succès de la création du relevé de valeur probante pour chaque unité archivistique, objet ou groupe d'objets audité)
  - KO : Finalisation du relevé de valeur probante n'a pas été effectué. (PROBATIVE\_VALUE\_CREATE\_REPORT.KO = Échec de la création du relevé de valeur probante pour chaque unité archivistique, objet ou groupe d'objets audité)
  - WARNING : avertissement lors de la finalisation du relevé de valeur probante (PROBATIVE\_VALUE\_CREATE\_REPORT.WARNING = Avertissement lors de la création du relevé de valeur probante pour chaque unité archivistique, objet ou groupe d'objets audité)
  - FATAL : une erreur technique est survenue lors de la finalisation du relevé de valeur probante (PROBATIVE\_VALUE\_CREATE\_REPORT.FATAL = Erreur technique lors de la création du relevé de valeur probante pour chaque unité archivistique, objet ou groupe d'objets audité)

### 14.6.4. Structure de workflow du relevé de valeur probante

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



## 14.7. Rapport du relevé de valeur probante

Le relevé de valeur probante est un fichier JSON généré par la solution logicielle Vitam. Le relevé de valeur probante réunit les éléments permettant de fournir à un auditeur externe une présomption de confiance dans ce qui lui est communiqué.

### 14.7.1. Exemple de JSON : rapport de valeur probante

```

{
  "operationSummary" : {
    "tenant" : 0,
    "evId" : "aeaaaaaachemhquabmugaloiuua7xaaaaaq",
    "evType" : "EXPORT_PROBATIVE_VALUE",
    "outcome" : "OK",
    "outDetail" : "EXPORT_PROBATIVE_VALUE.OK",
    "outMsg" : "Succès lors du processus d'export du relevé de valeur probante",
    "rightsStatementIdentifier" : {
      "accessContract" : "CONTRACTINK"
    }
  },
  "reportSummary" : {
    "evStartDateTime" : "2019-11-07T09:18:05.178",
    "evEndDateTime" : "2019-11-07T09:18:06.640",
    "reportType" : "PROBATIVE_VALUE",
    "vitamResults" : {
      "OK" : 1,
      "KO" : 0,
      "WARNING" : 0,
      "total" : 1
    }
  },
  "context" : {
    "dslQuery" : {
      "$query" : [ {
        "$or" : [ {
          "$in" : {
            "#id" : [ "aeaqaahgynv2aa3oealoiqpvbraaaaba" ]
          }
        }
      ]
    }
  }
}

```

**OperationSummary**

**ReportSummary**

**Context**

```

    }, {
      "$in" : {
        "#allunitups" : [ ]
      }
    } ]
  } ],

```

```

"$filter" : { },
"$projection" : { }
},
"usage" : "BinaryMaster",

```

```

"version" : "1"

```

```

"reportEntries" : [ {
  "unitIds" : [ "aeaqaahgynv2aa3oealoiqpvbraaaaba" ],
  "objectGroupId" : "aebaaaaahgynv2aa3oealoiqpva2qaaaaq",
  "objectId" : "aeaaaaahgynv2aa3oealoiqpvazyaaaaq",
  "usageVersion" : "BinaryMaster_1",
  "operations" : [ {
    "id" : "aecaaaaachnaqi7abbwqaloirpkzpiaaaaq",
    "evTypeProc" : "STP_OP_SECURISATION",
    "evDateTime" : "2019-11-07T05:38:05.629"
  }, {
    "id" : "aecaaaaachnaqi7abbwqaloirpsv3qaaaaq",
    "evTypeProc" : "LOGBOOK_OBJECTGROUP_LFC_TRACEABILITY",
    "evDateTime" : "2019-11-07T05:38:37.934"
  }, {
    "id" : "aeaaaaachemhquabwdcaloiqpuvhiaaaaaq",
    "evTypeProc" : "PROCESS_SIP_UNITARY",
    "rightsStatementIdentifier" : "{\"ArchivalAgreement\":\"ArchivalAgreement0\"}",
    "agIdApp" : "CT-000001",
    "evDateTime" : "2019-11-07T04:28:51.829"
  } ],
  "checks" : [ {
    "name" : "TIMESTAMP_OPERATION_DATABASE_TRACEABILITY_VALIDATION",
    "details" : "Validating timestamp operation found in database 'AND' timestamp operation found in traceability secured file.",
    "type" : "TIMESTAMP_CHECKING",

```

### ReportDetail

```
"source" : "DATABASE",  
  
"destination" : "TRACEABILITY_FILE",  
  
"sourceComparable" :  
"MIIILITAVAgEAMBAMdk9wZXJhdGlvbiBPa2F5MIIlBgyJKoZIHvcNAQcCoIIK9zCCCvMCAQMxDzANBglghkgBZQM  
EAgMFADCBgAYLkoZIHvcNAQkQAQSGcQRvMG0CAQEGASkwUTANBglghkgBZQMEAgMFAARA4fpDituDLas2w6jnrd3  
1fi/  
4vyJQ9uzEyVvngUM2IybKkVjZzTNLwnO3lZb+ZhrjvYoc2+2gzJ8Rw93deFzAtgIBARgPMjAxOTExMDcWNTM4MDD  
aoIGhzCCBoMwggRroAMCAQICAgDUMA0GCSqGSIB3DQEBcWUAMHgxCzAJBgNVBAYTAmZyMQwwCgYDVQQIDANpZGY  
xDjAMBgNVBAcMBXBhcm1lZmQ4wDAYDVQKDAV2aXRhbTEUMBIGA1UECwwLYXV0aG9yaXRpZXMxJTAjBgNVBAMMHGN  
hX2ludGVyYbWVkaWF0ZV90aW1lc3RhbXBpbmcwHhcNMTkxMTA3MDIzMzA1WhcNMjIxMTA2MDIzMzA1WjBUMQswCQY  
DVQQGEWJmcjEMMAoGA1UECAwDaWRmMQ4wDAYDVQQHDAVwYXJpczEOMAwGA1UECgwFdm10YW0xZmZAVBgNVBAMMDnN  
lY3VyZS1sb2dib29rMIICiANBgkqhkiG9w0BAQEFAAOCAg8AMIICCgKCAgEA09lqSj5nvYlemZC4CHCzTphNgK1  
WzA/ILRjVL4ho2vft6yx9TstAQqsVARrsknmAGJ2KMillffIOEZ00ph3H0oo7auWOUgDytsfTeRmDFSVoxUfM/  
1jBUXgqhg6VSHVka1ueAu0kg4g0ADvWrJbtVKHxps9vrEA+H4tEiTT3mfKEdWBAyLz4cG0ln4ip3s1GMSjmVgDwV  
gbzPnHvD1VUDLi6dYHbkaQaLSRh9h147dPVbzOelqZAUPDML6kVUI6fM9jxAzQJqub0jL+vEGvMUMJTqStnFXAU  
C5Gof3LU5iV/cwyGUWY0rz5kKS1XLT9jvYteijxYe4nf0y/V5Yzp/6F/  
52esT4rqI5x9QGfV9TxSHiymcu316aul4uFw+M+qLy4khC1BhNbsHojjnGhR33KuRa/  
QAZltPJ3KyFC3NLSfGrJSi+JbZV5yS5yYbkd1F25TdyUSfxk+0lQEvmL6ngsjKQHqSNdsTWMAhDbuE614ErrySxG  
wjnFBMOfsPK2HKe3dc7iZUZyHPPUyBq8RVWdvHVFtQuXBCKCMSVnk6rVjmYmuSiI8oAy0TK7ixfAqMmE2yZwoJt7  
hMYbo0Adrfjg7Bf95uP2LDc8LkWsAlhIJDFOJT5ANXsWxGM0LmFmGTRoPF0b2E0G/  
QQKAc0GEGRiG4gGihCLiZSGJ81CYw0CAwEAaOCATkwgE1MUCUGCWGSAGG+EIBDQQYfHZDXJ0awZpY2F0IFN1c  
nZldXIGu1NMMB0GA1UdDgQWBRRpel2gJ6UBuGBR8LG5d7Sa8kfmazCBmYDVR0jBIGTMIGQgBQhZkzBQYoL+UCSH  
cYIsFk+rST006F0PHIwcDELMAkGA1UEBhMCZnIxDDAKBGNVBAGMA21kZjEOMAwGA1UEBwwFcgFyaXNlZmZAVBgNVB  
AoMBXZpdGFtMRQwEgYDVQQLDAtHdXRob3JpdG1lcEdEMBsGA1UEAwUY2Ffcm9vdF90aW1lc3RhbXBpbmeCAgDTM  
AkGA1UdEgQCAAwDAYDVR0TAQH/BAIwADALBgNVHQ8EBAMCBsAwEQYJYIZIAyb4QgEBBAQDAgZAMBYGA1UdJQEB/  
wQMAoGCCsGAQFwMIMA0GCSqGSIB3DQEBcWUAA4ICAQAwjDb7PvGAMqgW62+mXQHHo6hh8ikJyyz6kT10jLaRV  
Uou77ZMoUqF5GLS6d+tsPvGL5cdrs2Z5WqZPny3DyOxBSsGnIY9FM0herQAr6mY6xyTB+TrQDfQZeaqL0m1I/  
PAWiQNWjDUJCDGEhvaom046fJNKcXrPjb9cQ/  
CESWdOJ4ERBznUV8nOyVjCPQeW2yrY1hZrtYrFpTG17thy0INGBz/zySH15ArF11bd75Z3o9ABMrtTqPjJ5R/  
FeEzOYXZdpdc5C7UqQwyZxcQpRQlYJvedBDrBqGQ5AZ9KMNqEmQEHOElzyXwoxpo2I1C/G/X/  
nvBiIqgVUGsxTdh/btOIQPvKS/PuvmuqUeCb1KKWs7txgY4g5387q8NuV  
  
MrhUc6iux3JlmcX7ismaAk/1AZPojzbNNxFwW5N/  
4+q2TM8Q+0q3lx09wyiGQqoCuP76Zv00REukpFgUsBf+Yot1qyTxFYxpwgW1hX9ed8VsWJPHUNOy97Cd8MOPfgj+  
CefBemvMa8zQH958KWtkjLlYKAhQuXHM5E6CuDUPe98090+ulNp4g71SSJbaLOCMZwpljioSyFRIGuirNloDruhi  
lX/NOuGvBk5AUtvVvsbfFm3gOCpx+ijUMLcaBd5e6J0+xyvK2v0a+/x25hPjuLp/  
weXmiMGtANyzbgmvJrNkgUDGCA80wggPJAgeEBMH4weDELMAkGA1UEBhMCZnIxDDAKBGNVBAGMA21kZjEOMAwGA1U  
EBwwFcgFyaXNlZmZAVBgNVBAAoMBXZpdGFtMRQwEgYDVQQLDAtHdXRob3JpdG1lcEdEMCMGA1UEAwY2FfaW50ZXJz  
tZWRRpYXRlX3RpbW5udGFTcDGLuZwICANQWdYJYIZIAWUDBAIDBQCGggEgMBoGCSqGSIB3DQEAJAZENBgsqhkig9w0  
BCRABBDACBgkqhkiG9w0BCQUxXcNMTkxMTA3MDUzODAZwAtBgkqhkiG9w0BCTQxIDAEMA0GCWGSAGFAwQCAwU  
AoQ0GCSqGSIB3DQEBDQUAME8GCSqGSIB3DQEJBDFCBEZFLQlPJunF7t9ueEP12tiDj31dsEAWDnL/  
6UUUi+duet24AvfKjt00lwlEeJrR0oHeYnUadXZwGQo7/+Qs5a0MGQGCyqGSIB3DQEJEAIVMVUwUzBRME8wCwYJY  
IZIAWUDBAIDBEaduVLSR3MT88jYzAMR8nbaiHoZPSUfMPxK/  
1Ekh3DXK7z46X0bKI87kWN8Rjacz7QuF7nDQmPRcETZHzjUjoma0GCSqGSIB3DQEBDQUABIIICAF6U41AI1cQTEJG  
OXIQinnZ4clRer9S9w91s9WDWzCxyOq4d60vunq3rTgKdNiIYBeLomRujr3DjUI6Byt57mHJe/  
AWqyAb+1BBhUWkPKjWadkXPyeMxMTHhgUUKd4QBMMBITK0g9WWR7Vz5ouB2A0V2XMcKce5B17N5R1v5mH5J9EgH  
Ffp+osLE2GM5jtw5L8nOxA+U9056vxXvKRMNCKk8X6cKGJCL9uUkq+UiZQu1sQEALQC+7x6YmvfL51RXy+orZD5W  
85InR5IqGFjjdazD1wbONBODrVFn4t0PI6PqW56KFsJDYXhP4xiyY9s+0857b1P3rMfiDrPDCEcJzTc6ChOmn2XD  
/5F6958w9x/3LQOCI1c7grLLuChk28ndgiS5nuQ4k/  
yOEnLbGh1ZuFKDxpAZctRCS3gDz4hLcbJmiCns0trQ29BybHddwzqUQf2LHuzrciZgWjpb+MYOBTfUu+ifau8pSB  
lwctAe6GSczKdYS/E0tLd9NFzXayxDwgs1qvxv6FdstLg6LHe8xFNf/KfNzgzzeQN8K6xE76qc3/  
dvqmIKUWqzndBjc/  
6t13zDxOBnyFITcnCWJZFTLbCPA+ZUaw4yiWC26Gz7BxOutTvurZoyB5VkcMhB4IQdrR1GhG1hRH9v7ES8dBHCz  
dHnflX5igjqPjEpZVDezfM",  
  
"destinationComparable" :  
"MIIILITAVAgEAMBAMdk9wZXJhdGlvbiBPa2F5MIIlBgyJKoZIHvcNAQcCoIIK9zCCCvMCAQMxDzANBglghkgBZQM  
EAgMFADCBgAYLkoZIHvcNAQkQAQSGcQRvMG0CAQEGASkwUTANBglghkgBZQMEAgMFAARA4fpDituDLas2w6jnrd3  
1fi/  
4vyJQ9uzEyVvngUM2IybKkVjZzTNLwnO3lZb+ZhrjvYoc2+2gzJ8Rw93deFzAtgIBARgPMjAxOTExMDcWNTM4MDD  
aoIGhzCCBoMwggRroAMCAQICAgDUMA0GCSqGSIB3DQEBcWUAMHgxCzAJBgNVBAYTAmZyMQwwCgYDVQQIDANpZGY  
xDjAMBgNVBAcMBXBhcm1lZmQ4wDAYDVQKDAV2aXRhbTEUMBIGA1UECwwLYXV0aG9yaXRpZXMxJTAjBgNVBAMMHGN  
hX2ludGVyYbWVkaWF0ZV90aW1lc3RhbXBpbmcwHhcNMTkxMTA3MDIzMzA1WhcNMjIxMTA2MDIzMzA1WjBUMQswCQY  
DVQQGEWJmcjEMMAoGA1UECAwDaWRmMQ4wDAYDVQQHDAVwYXJpczEOMAwGA1UECgwFdm10YW0xZmZAVBgNVBAMMDnN  
lY3VyZS1sb2dib29rMIICiANBgkqhkiG9w0BAQEFAAOCAg8AMIICCgKCAgEA09lqSj5nvYlemZC4CHCzTphNgK1  
WzA/ILRjVL4ho2vft6yx9TstAQqsVARrsknmAGJ2KMillffIOEZ00ph3H0oo7auWOUgDytsfTeRmDFSVoxUfM/  
1jBUXgqhg6VSHVka1ueAu0kg4g0ADvWrJbtVKHxps9vrEA+H4tEiTT3mfKEdWBAyLz4cG0ln4ip3s1GMSjmVgDwV  
gbzPnHvD1VUDLi6dYHbkaQaLSRh9h147dPVbzOelqZAUPDML6kVUI6fM9jxAzQJqub0jL+vEGvMUMJTqStnFXAU  
C5Gof3LU5iV/cwyGUWY0rz5kKS1XLT9jvYteijxYe4nf0y/V5Yzp/6F/  
52esT4rqI5x9QGfV9TxSHiymcu316aul4uFw+M+qLy4khC1BhNbsHojjnGhR33KuRa/  
QAZltPJ3KyFC3NLSfGrJSi+JbZV5yS5yYbkd1F25TdyUSfxk+0lQEvmL6ngsjKQHqSNdsTWMAhDbuE614ErrySxG  
wjnFBMOfsPK2HKe3dc7iZUZyHPPUyBq8RVWdvHVFtQuXBCKCMSVnk6rVjmYmuSiI8oAy0TK7ixfAqMmE2yZwoJt7  
hMYbo0Adrfjg7Bf95uP2LDc8LkWsAlhIJDFOJT5ANXsWxGM0LmFmGTRoPF0b2E0G/  
QQKAc0GEGRiG4gGihCLiZSGJ81CYw0CAwEAaOCATkwgE1MUCUGCWGSAGG+EIBDQQYfHZDXJ0awZpY2F0IFN1c  
nZldXIGu1NMMB0GA1UdDgQWBRRpel2gJ6UBuGBR8LG5d7Sa8kfmazCBmYDVR0jBIGTMIGQgBQhZkzBQYoL+UCSH  
cYIsFk+rST006F0PHIwcDELMAkGA1UEBhMCZnIxDDAKBGNVBAGMA21kZjEOMAwGA1UEBwwFcgFyaXNlZmZAVBgNVB  
AoMBXZpdGFtMRQwEgYDVQQLDAtHdXRob3JpdG1lcEdEMBsGA1UEAwUY2Ffcm9vdF90aW1lc3RhbXBpbmeCAgDTM  
AkGA1UdEgQCAAwDAYDVR0TAQH/BAIwADALBgNVHQ8EBAMCBsAwEQYJYIZIAyb4QgEBBAQDAgZAMBYGA1UdJQEB/  
wQMAoGCCsGAQFwMIMA0GCSqGSIB3DQEBcWUAA4ICAQAwjDb7PvGAMqgW62+mXQHHo6hh8ikJyyz6kT10jLaRV  
Uou77ZMoUqF5GLS6d+tsPvGL5cdrs2Z5WqZPny3DyOxBSsGnIY9FM0herQAr6mY6xyTB+TrQDfQZeaqL0m1I/  
PAWiQNWjDUJCDGEhvaom046fJNKcXrPjb9cQ/  
CESWdOJ4ERBznUV8nOyVjCPQeW2yrY1hZrtYrFpTG17thy0INGBz/zySH15ArF11bd75Z3o9ABMrtTqPjJ5R/  
FeEzOYXZdpdc5C7UqQwyZxcQpRQlYJvedBDrBqGQ5AZ9KMNqEmQEHOElzyXwoxpo2I1C/G/X/  
nvBiIqgVUGsxTdh/btOIQPvKS/PuvmuqUeCb1KKWs7txgY4g5387q8NuV
```







1jBUXgqhg6VSHVka1ueAu0kg4g0ADvWrJbtVKHxps9vrEA+H4tEiTT3mfKEdWBAyLz4cG0ln4ip3s1GMSjmVgDwV  
gbzPnHvD1VUDLij6dYHbkAqaLSR9h147dPVbzOelqZAUPDML6kVUI6fM9jxAzQJqub0jL+vEGvMUMJTqStnFXAU  
C5Gof3LU5iV/cwyGUWY0rz5kKS1XLT9jvYteijxYe4nf0y/V5Yzpz/6F/  
52esT4rqI5x9QGfv9TxSHiymcu316au14uFw+M+qLy4khC1BhNbsHojjnGhR33KuRa/  
QAZ1tPJ3KyFC3NLSfGrJSi+JbZV5yS5yybkdlF25TdyUSfxk+01QEvmL6ngsjKQHqSNdsTWmAHdBuE614ErrySxG  
wjnFBMOfsPK2HKe3dc7iZUZyHPPUyBq8RVWDvHVfTQuXBCKCMSVnk6rVjmYMuSiI8oAy0TK7ixfAqMmE2yZwoJt7  
hMYbo0Adrffjg7Bf95uP2LDC8LkWsALhIJDFOJT5ANXsWxGM0LmFmGTRoPF0b2E0G/  
QQKAc0GEgrIG4gGihCLiZSGJ81CYw0CAwEAAaOCATkwggE1MCUGCWCWSAGG+EIBDQQYFhZDXJ0aWzP2F0IFN1c  
nZ1dXIGU1NMBOGA1UDdGQWBBRpel2gJ6UBUGBr8LG5d7Sa8kfmazCBmwYDVR0jBIGTMIQGgBQHxKjBQYoL+UCSH  
cYIsFk+rST006F0PHIwcDELMAkGA1UEBhMCZnIxDDAKBgNVBAgMA2lkZjEOMAwGA1UEBwWFcGFyaXMxZjAMBGNVB  
AoMBXZpdGFtMRQwEgYDVQQLDAtHdXR0b3JpdGllczEdMBSGA1UEAwUY2Ffcm9vdF90aW1lc3RhbXBpbmeCAGDTM  
AkGA1UdEgQCAAwDAYDVR0TAQH/BAIwADALBgNVHQ8EBAMCBsAwEQYJYIZIAyb4QgEBBAQDAGZAMBGA1UdJQEB/  
wQMAAGCCSGAQUFBwMIMA0GCSqGSIb3DQEBCwUAA4ICAQAwjDb7PvGAmqgW62+mXQHh06hh8ikJyyz6kT10jLaRV  
Uou77ZMoUqF5GLS6d+tSpvGL5cdrs2Z5WqZPny3DyOxBBsGnIY9FM0herQAr6mY6xyTB+TrQDfQZeaQ0m1I/  
PAWiQNWjDUJCDGEhva0m046fJNKcXrPjB9cQ/  
CESWdOJ4ERBznUV8nOyVjCPQeW2yrY1hZrtYrYfpTG17thy0INGBz/zySH15ArF11bd75Z3o9ABMtRTqPjJ5R/  
FeZ0YXZdpdc5C7UqQWYzXcQpRQ1YJvedBDRBqGQ5AZ9KMNQEMQEHOElzyXwoXpo2i1C/G/X/  
nvBIiqgVUGsxTdh/btOIQPVKS/PuvmuqUeCb1KKwS7txgY4g5387q8NuVMrhUc6iux3J1mcX7ismaAk/  
1AZPoJzbNNxFwW5N/  
4+q2TM8Q+0q31x09wyiGQqoCuP76Zv00REukpFgUsBf+Yot1qyTxFYxpwgW1hX9ed8VsWJPHUNOy97Cd8MOPfgj+  
CeFBemvMa8zQH958KWtkjLlYKAhQuXHM5E6CuDUPE98090+ulNp4g71SSJbaLOCMZWP1j0syFRIGuirNloDruhi  
lX/NOuGvBk5AUtvVvsbFm3gOCpx+ijUMLcaBd5e6J0+xyvK2v0a+/x25hPjuLp/  
weXmiMGtAnyzbgmvJrNkgUDGCA80wggPJAqEBMH4weDELMAkGA1UEBhMCZnIxDDAKBgNVBAgMA2lkZjEOMAwGA1U  
EBWwFcgGFyaXMxZjAMBGNVBAoMBXZpdGFtMRQwEgYDVQQLDAtHdXR0b3JpdGllczEdMBSGA1UEAwUY2Ffcm9vdF90aW1lc3RhbXBpbmeCAGDTM  
tZWRpYXRlX3RpbWVzdGFtcGluZwICANQwDQYJYIZIAWUDBAIDBQCgggEgMB0GCSqGSIb3DQEJAzENBgsqhkiG9w0  
BCRABBDACBgkqhkiG9w0BCQUxDxcNMTkxMjA3MDUzODA3WjAtBgkqhkiG9w0BCTQxIDAEMA0GCWCSAF1AwQCAWU  
AoQOGCSqGSIb3DQEEDQUAME8GCSqGSIb3DQEBJDFCBECZFLQ1PJunF7t9ueEP12tiDJ31dsEAWDnL/  
6UUUi+duet24AvFKjt00lwlEeJrR0oHeYnUadXZwGQo7/+Qs5a0MGQGCyqGSib3DQEEJEAiVMVUwUzBRME8wCwYJY  
IZIAWUDBAIDBEAeduVLSR3MT88jYzZAMR8nbaiHoZPSUFMPxK/  
1EKH3DXK7z46X0bkI5s87kWN8Rjacz7QuF7nDQmPRcETZjHjUjoma0GCSqGSIb3DQEEDQUABIICAF6U41A1lcQTEJG  
OXEQinnZ4clRer9SW91s9WDWzCxyOq4d60vnc3rTgKdNiIYBELOmRujr3DjUI6Byt57mHJe/  
AWqvAb+1BBhUwKPKjbWadkXPYeMxMTHhgUUKd4OBMMBITK0g9WWR7Vz5ouB2AOV2XmCkce5B17N5R1v5mH5J9EgH  
Ffp+osLE2GM5jtW5L8nOxA+U9056vxXvKRMNCKk8X6cKGJCL9uUkq+UiZQu1sQEALQC+7x6YMvfl51RXy+orZD5W  
85InR5IqGFjjdzD1wbONBODrVFn4t0PI6PqW56KfSJDYXhP4xiyY9s+0857b1P3rMfiDrPDCEcJzTc6ChOmn2XD  
/5F6958w9x/3LqQC1lc7grLLuCK28ndgiS5nuQ4k/  
yOEnLbGhlZuFKDxpAZctRCS3gDz4hLcbJmiCns0trQ29BybHddwzqUQf2LHuzrciZgWjpb+MYOBTfUu+ifau8pSB  
lwctAe6GSczKDYs/E0tLd9NFzXayxDwgs1qvxv6FdslG6LHe8xFNF/KfNzgzQn8K6xE76qc3/  
dvqmIKUWqzndBjc/  
6t13zDxOBnyFITcnCWJZFTLbCPA+ZUaw4yiWC26Gz7BxOuTtvurZoyB5VkcDMhb4IQdrR1GhG1hRH9v7ES8dBHCz  
dHnflX5igjqPJRzVDezfM",

"destinationComparable" :

"MIIILITAVAgEAMBAMdK9wZXhdGlvbiBPa2F5MIIlBgyJKoZIHvcNAQCcoI1K9zCCCvMCAQMxDzANBglghkgBZQOM  
EAqMFADCBgAYLkoZIHvcNAQkQAQSGcQRvMG0CAQEAGASkUTANBglghkgBZQMEAgMFAARA4fpDituDLas2w6jnrd3  
1fi/  
4VyJQ9uzEyVvngUM2IybKkVjZzTNLwnO3lZb+ZHrjvYoc2+2gzJ8Rw93deFzAtgIBARgPMjAxOTExMDcwNTM4MDD  
aoIIGhzCCBoMwggRroAMCAQICAgDUMA0GCSqGSIb3DQEBwUAMHgxCzAJBgNVBAYTAmZyMQwwCgYDVQQIDANpZGY  
xDjAMBGNVBAcMBXBhcm1zMzQ4wDAYDVQKDAV2aXRhbTEUMBIGA1UECwwLYXV0aG9yaXRpZXMxZTAjBgNVBAMMHGQ  
hX21udGVyYjVwKwF0ZV90aW1lc3RhbXBpbmcwHhcNMTkxMjA3MDUzODA3WjAtBgkqhkiG9w0BAQEFAAOCAg8AMI  
CCGKCAgEA091qSj5nvYlemZC4CHCzTzPHNgK1  
WzA/ILRjVL4ho2vft6yx9TstAQqsVARrsknmAGJ2KMllffIOEZ00ph3H0oo7auWOUgDytsfTerMdfSVoXUfM/  
1jBUXgqhg6VSHVka1ueAu0kg4g0ADvWrJbtVKHxps9vrEA+H4tEiTT3mfKEdWBAyLz4cG0ln4ip3s1GMSjmVgD

wVgbzPnHvD1VUDLij6dYHbkAqaLSR9h147dPVbzOelqZAUPDML6kVUI6fM9jxAzQJqub0jL+vEGvMUMJTqStnFX  
AUC5Gof3LU5iV/cwyGUWY0rz5kKS1XLT9jvYteijxYe4nf0y/V5Yzpz/6F/  
52esT4rqI5x9QGfv9TxSHiymcu316au14uFw+M+qLy4khC1BhNbsHojjnGhR33KuRa/  
QAZ1tPJ3KyFC3NLSfGrJSi+JbZV5yS5yybkdlF25TdyUSfxk+01QEvmL6ngsjKQHqSNdsTWmAHdBuE614ErrySxG  
wjnFBMOfsPK2HKe3dc7iZUZyHPPUyBq8RVWDvHVfTQuXBCKCMSVnk6rVjmYMuSiI8oAy0TK7ixfAqMmE2yZwoJt7  
hMYbo0Adrffjg7Bf95uP2LDC8LkWsALhIJDFOJT5ANXsWxGM0LmFmGTRoPF0b2E0G/  
QQKAc0GEgrIG4gGihCLiZSGJ81CYw0CAwEAAaOCATkwggE1MCUGCWCWSAGG+EIBDQQYFhZDXJ0aWzP2F0IFN1c  
nZ1dXIGU1NMBOGA1UDdGQWBBRpel2gJ6UBUGBr8LG5d7Sa8kfmazCBmwYDVR0jBIGTMIQGgBQHxKjBQYoL+UCSH  
cYIsFk+rST006F0PHIwcDELMAkGA1UEBhMCZnIxDDAKBgNVBAgMA2lkZjEOMAwGA1UEBwWFcGFyaXMxZjAMBGNVB  
AoMBXZpdGFtMRQwEgYDVQQLDAtHdXR0b3JpdGllczEdMBSGA1UEAwUY2Ffcm9vdF90aW1lc3RhbXBpbmeCAGDTM  
AkGA1UdEgQCAAwDAYDVR0TAQH/BAIwADALBgNVHQ8EBAMCBsAwEQYJYIZIAyb4QgEBBAQDAGZAMBGA1UdJQEB/  
wQMAAGCCSGAQUFBwMIMA0GCSqGSIb3DQEBCwUAA4ICAQAwjDb7PvGAmqgW62+mXQHh06hh8ikJyyz6kT10jLaRV  
Uou77ZMoUqF5GLS6d+tSpvGL5cdrs2Z5WqZPny3DyOxBBsGnIY9FM0herQAr6mY6xyTB+TrQDfQZeaQ0m1I/  
PAWiQNWjDUJCDGEhva0m046fJNKcXrPjB9cQ/  
CESWdOJ4ERBznUV8nOyVjCPQeW2yrY1hZrtYrYfpTG17thy0INGBz/zySH15ArF11bd75Z3o9ABMtRTqPjJ5R/  
FeZ0YXZdpdc5C7UqQWYzXcQpRQ1YJvedBDRBqGQ5AZ9KMNQEMQEHOElzyXwoXpo2i1C/G/X/  
nvBIiqgVUGsxTdh/btOIQPVKS/PuvmuqUeCb1KKwS7txgY4g5387q8NuVMrhUc6iux3J1mcX7ismaAk/  
1AZPoJzbNNxFwW5N/  
4+q2TM8Q+0q31x09wyiGQqoCuP76Zv00REukpFgUsBf+Yot1qyTxFYxpwgW1hX9ed8VsWJPHUNOy97Cd8MOPfgj+  
CeFBemvMa8zQH958KWtkjLlYKAhQuXHM5E6CuDUPE98090+ulNp4g71SSJbaLOCMZWP1j0syFRIGuirNloDruhi  
lX/NOuGvBk5AUtvVvsbFm3gOCpx+ijUMLcaBd5e6J0+xyvK2v0a+/x25hPjuLp/  
weXmiMGtAnyzbgmvJrNkgUDGCA80wggPJAqEBMH4weDELMAkGA1UEBhMCZnIxDDAKBgNVBAgMA2lkZjEOMAwGA1U  
EBWwFcgGFyaXMxZjAMBGNVBAoMBXZpdGFtMRQwEgYDVQQLDAtHdXR0b3JpdGllczEdMBSGA1UEAwUY2Ffcm9vdF90aW1lc3RhbXBpbmeCAGDTM  
tZWRpYXRlX3RpbWVzdGFtcGluZwICANQwDQYJYIZIAWUDBAIDBQCgggEgMB0GCSqGSIb3DQEJAzENBgsqhkiG9w0  
BCRABBDACBgkqhkiG9w0BCQUxDxcNMTkxMjA3MDUzODA3WjAtBgkqhkiG9w0BCTQxIDAEMA0GCWCSAF1AwQCAWU  
AoQOGCSqGSIb3DQEEDQUAME8GCSqGSIb3DQEBJDFCBECZFLQ1PJunF7t9ueEP12tiDJ31dsEAWDnL/  
6UUUi+duet24AvFKjt00lwlEeJrR0oHeYnUadXZwGQo7/+Qs5a0MGQGCyqGSib3DQEEJEAiVMVUwUzBRME8wCwYJY  
IZIAWUDBAIDBEAeduVLSR3MT88jYzZAMR8nbaiHoZPSUFMPxK/  
1EKH3DXK7z46X0bkI5s87kWN8Rjacz7QuF7nDQmPRcETZjHjUjoma0GCSqGSIb3DQEEDQUABIICAF6U41A1lcQTEJG  
OXEQinnZ4clRer9SW91s9WDWzCxyOq4d60vnc3rTgKdNiIYBELOmRujr3DjUI6Byt57mHJe/  
AWqvAb+1BBhUwKPKjbWadkXPYeMxMTHhgUUKd4OBMMBITK0g9WWR7Vz5ouB2AOV2XmCkce5B17N5R1v5mH5J9EgH  
Ffp+osLE2GM5jtW5L8nOxA+U9056vxXvKRMNCKk8X6cKGJCL9uUkq+UiZQu1sQEALQC+7x6YMvfl51RXy+orZD5W  
85InR5IqGFjjdzD1wbONBODrVFn4t0PI6PqW56KfSJDYXhP4xiyY9s+0857b1P3rMfiDrPDCEcJzTc6ChOmn2XD  
/5F6958w9x/3LqQC1lc7grLLuCK28ndgiS5nuQ4k/  
yOEnLbGhlZuFKDxpAZctRCS3gDz4hLcbJmiCns0trQ29BybHddwzqUQf2LHuzrciZgWjpb+MYOBTfUu+ifau8pSB  
lwctAe6GSczKDYs/E0tLd9NFzXayxDwgs1qvxv6FdslG6LHe8xFNF/KfNzgzQn8K6xE76qc3/  
dvqmIKUWqzndBjc/  
6t13zDxOBnyFITcnCWJZFTLbCPA+ZUaw4yiWC26Gz7BxOuTtvurZoyB5VkcDMhb4IQdrR1GhG1hRH9v7ES8dBHCz  
dHnflX5igjqPJRzVDezfM",

```

FeEzOYXZdpdc5C7UqQwyZXCqPRL1YJvedBDRBqGQ5AZ9KMnQEmQEHoE1zyxWoxpo2I1C/G/X/
nvBIiqgVUGsxTdh/btOIQPvKS/PuvmuqUeCb1KKWs7txgY4g5387q8NuVMrhUc6iux3J1mcX7ismaAk/
1AZPojzbNNxFwW5N/
4+q2TM8Q+0q31x09wYiGQqoCuP76Zv00REukpFgUsBf+Yot1qyTxFYxpwgW1hX9ed8VsWJPHUNOy97Cd8MOPfgj+
CefBemvMa8zQH958KwtkjLlYKAhQuXHM5E6CuDUpE98090+ulNp4g71SSJBaLOCMZWplji0syFRIGuirNloDruhi
lX/NOuGvBk5AUtvVvsbfFm3gOCpx+ijUMLcaBd5e6J0+xyvK2v0a+/x25hPjuLp/
weXmiMGtANyZbgmvJrNkgUDGCA80wggPJAgEBMH4weDELMaKGA1UEBhMCZnIxDDAKBgNVBAGMA2lkZjEOMAwGA1U
EBwwFcgGFvaXMxZjAMBQNVBAoMBXZpdGFtMROwEgYDVOOLDAthdXRob3JpdGllczElMCMGA1UEAwY2FfaW50ZXJ
tZWVpYXRlX3RpbWVzdGFTcGluZwICANQwDQYJYIZIAWUDBAIDBQCgggEgMBoGCSqGSIB3DQEJAzENBgsqhkiG9w0
BCRABBDACBgkqhkiG9w0BCQUxDxcnMTA3MDUzODA3WjAtBgkqhkiG9w0BCTQxIDAEMA0GCWCGSAF1AwQCAwU
AoQ0GCSqGSIB3DQEJBDQUAME8GCSqGSIB3DQEBDFCBECZFLQlPJunF7t9ueEP12tiDJ31dsEAWDnL/
6UUUi+duet24AvfKjt00lwlEeJrR0oHeYnUadXZwGQo7/+Qs5a0MGQGCyqGSIB3DQEJEAIVMVUwUzBRME8wCwYJY
IZIAWUDBAIDBEAeduVLSR3MT88jYzzAMR8nbaiHoZPSUfMPxK/
1Ekh3DXK7z46X0bKIs87kWN8RjacZ7QuF7nDQmPRcETZjUjoma0GCSqGSIB3DQEBDQUABIIICAF6U41A1lcQTEJG
OXIQinnZ4clRer9SW91s9WDWzCxyOq4d60vnq3rTgKdNiYBeLomRujr3DjUI6Byt57mHJe/
AWqvAb+1BBhUWkPKjbWadkXPyeMxMTHhgUUKd4OBMMBITK0g9WWR7Vz5ouB2AOV2XMcKce5B17N5Rlv5mH5J9EgH
Ffp+osLE2GM5jtw5L8nOxA+U9056vxXvKRMNCKk8X6cKGJCL9uUkq+UiZQu1sQEALQC+7x6YMvfl51RXy+orZD5W
85InR5IqGFjjdazD1wbONBODrVFn4t0PI6PqW56KFsJDYXhP4xiyY9s+O857b1P3rMfiDrPDCEcJzTc6ChOmn2XD
/5F6958w9x/3LQOCi1c7grLLuCK28ndgis5nuQ4k/
yOEnLbGhlZuFKDxpAZctRCS3gDz4hLcbJmiCns0trQ29BybHddwzqUQf2LHzrciZgWjpb+MYOBTfUu+ifau8pSB
lwctAe6GSczKDYs/E0tLd9NFzXayxDwgs1qvxdV6FdstLgB6LHe8xFNf/KfNzgzeQN8K6xe76qc3/
dvqmIKUWqzndBjc/
6t13zDxOBnyFITcnCWJZFTLbCPA+ZUaw4yiWC26Gz7BxOutTvurZoyB5VkcDMhb4IQdrR1GhG1hRH9v7ES8dBHcZ
dHnflX5igjqPJRjEpZVDezfM",

    "action" : "COMPARISON",

    "item" : "TIMESTAMP_OPERATION",

    "status" : "OK"

  }, {

    "name" : "MERKLE_OPERATION_DIGEST_DATABASE_TRACEABILITY_COMPARISON",

    "details" : "Comparing operation merkle digest found in database with operation
merkle digest found in traceability secured file.",

    "type" : "MERKLE_INTEGRITY",

    "source" : "DATABASE",

    "destination" : "TRACEABILITY_FILE",

    "sourceComparable" :
"JI89QHGLjTANLgYByeJ2NY21JOQTMnL3UOmVrICW7Oe8e+0fXSK6CPyLDXXnt1/9KyphZaKXODZvgnrV+PsJww=
=",

    "destinationComparable" :
"JI89QHGLjTANLgYByeJ2NY21JOQTMnL3UOmVrICW7Oe8e+0fXSK6CPyLDXXnt1/9KyphZaKXODZvgnrV+PsJww=
=",

    "action" : "COMPARISON",

    "item" : "MERKLE_TREE_ROOT_OPERATION_DIGEST",

    "status" : "OK"

  }, {

    "name" : "MERKLE_OPERATION_DIGEST_COMPUTATION_TRACEABILITY_COMPARISON",

    "details" : "Comparing operation merkle digest computed from secured data with
operation merkle digest found in traceability secured file.",

    "type" : "MERKLE_INTEGRITY",

    "source" : "COMPUTATION",

    "destination" : "TRACEABILITY_FILE",

    "sourceComparable" :

```







```

lX/NOuGvBk5AUtvVvsbfFm3gOCpx+ijUMLcaBd5e6J0+xyvK2v0a+/x25hPjuLp/
weXmiMGtANyzbgmvJrNkgUDGCA80wggPJAgEBMH4weDELMaKGA1UEBhMCZnInxDDAKBgNVBAGMA2lkZjEOMAwGA1U
EBWwFwCFYaxMxDjAMBgNVBAoMBXZpdGFtMRQwEgYDVQQLDAtHdXRob3JpdGllczElMCMGA1UEAwcY2FfaW50ZXJt
tZWRpYXRlX3RpbWVzdGFTcGluZwICANQwDQYJYiZiAWUDBAIDBQCGggEgMBoGCSqGSIB3DQEJAzENBgsqhkiG9w0
BCRABBDACBgkqhkiG9w0BCQUxDxcNMTkxMTA3MDQwMDA5WjAtBgkqhkiG9w0BCTQxIDAEMA0GCWCGSAFlAwQCAwU
AoQOGCSqGSIB3DQEBDQUAME8GCSqGSIB3DQEJBDFCBECaBl12eV9W7T0Y2Qbm43JWJDlmeXFeRgiwCTSfh91uIPd
xavc8huFSy5UuP9xeJ8gX7/
8nbaiHoZPSUfMPxK/
1Ekh3DXK7z46X0bKIs87kWN8Rjacz7QuF7nDQmPRcETZHjUjoma0GCSqGSIB3DQEBDQUABIIICALZUqEjXV7rYVWA
gJ31YxJHADf3mY0v3aehjQlGMIuCzmdzgw9V3FagQvQB60rJg/sApdwgvRM4uAX9TG0sVKigsZ/
sxhnhEEg5CEDp93R16y47/
TZYXs37Vqa3Qycd5xiEvL4zmJU12ZA2SKptweiqt3VmWID4czkYonTgwTgFIwev3YpDXd/
lDylzV5AnAs65b037jVqxjVRUA2PUwxAvvOxG6uQLXj7G1bncTaQpSYhMERr1H01pNvQ1F3D30/
RpVSWnVjbToLTeADDsk4fJjsIU1fZec3Im/
OZwlrKVzEDZdtuRvCr7LIst1JtxEcZTHDgiKmQXN4t39h+S49nB0ldFg/
V43IdqMT+bjKxQLOG0x4dtnIvfDnH0eu93KuHr9ivGDp3GnDBLzePTX99kzM2Ky1vBeVL2+MzN27CidceuZlQ1zp
ahXDgi+iOERhIf1GS+P2iR4vF4GKbUvsYQURB2WNNaG/
naffGFDPVcd4tB6rJDOBR1TzgwaseUuamLvebSkzBgvDIAIzb12Cuefhy+bMUPGbtQlWrrr6HizPTL2/
KrPakCoZk5nw9/
MICO+20eU6gP+s69TsV32CCTIpKxK87QD0NHz2ZBj5Ejte4h1sYxtr+3HvQYITaH1YxHMvy0KZVytMqCMVM/
Fyf2LGN9XjPd+ddDvyA7htRT",

    "action" : "VALIDATION",

    "item" : "PREVIOUS_TIMESTAMP_OPERATION",

    "status" : "OK"

}, {

    "name" : "TIMESTAMP_OPERATION_COMPUTATION_TRACEABILITY_COMPARISON",

    "details" : "Comparing timestamp operation computed from computing information
traceability file with timestamp operation found in traceability secured file.",

    "type" : "TIMESTAMP_CHECKING",

    "source" : "COMPUTATION",

    "destination" : "TRACEABILITY_FILE",

    "sourceComparable" :
"4fpDituDLas2w6jnrd31fi/4VyJQ9uzEyVvngUM2IybKkVjZzTNLwnO3lZb+ZhrjvYoc2+2gzJ8Rw93deFzAtg=
=",

    "destinationComparable" :
"4fpDituDLas2w6jnrd31fi/4VyJQ9uzEyVvngUM2IybKkVjZzTNLwnO3lZb+ZhrjvYoc2+2gzJ8Rw93deFzAtg=
=",

    "action" : "COMPARISON",

    "item" : "TIMESTAMP_OPERATION",

    "status" : "OK"

}, {

    "name" : "PREVIOUS_TIMESTAMP_OPERATION_DATABASE_TRACEABILITY_COMPARISON",

    "details" : "Comparing previous timestamp operation found in database with
previous timestamp operation found in traceability secured file.",

    "type" : "CHAIN",

    "source" : "DATABASE",

    "destination" : "TRACEABILITY_FILE",

    "sourceComparable" :
"MIILITAVAgEAMBAMdk9wZXJhdGlvbiBPa2F5MIILBgyJKoZiHvcNAQcCoIiK9zCCCvMCAQMxDzANBglghkgBZQM
EAgMFADCBgAYLkoZiHvcNAQcQAQSGcQRvMG0CAQEgASkwUTANBglghkgBZQMEAgMFAARACPakQZakyhRDnsAhDW

```

afWC2tL1A1Wg0AZIk2ycySfEFppfp1hY6Zpy0c82fZ7srdKPl+uWyqxH6a5q5lDb15AIBARgPMjAxOTEExMdcwNDA  
wMDlaoIIGhzCCBoMwggRroAMCAQICAgDUMA0GCSqGSIB3DQEBcWUAMHgxCzAJBgNVBAYTAxZyMQwwCgYDVQQIDAN  
pZGYxZjAMBGNVBAcMBXBhcm1zMQ4wDAYDVQQKDAV2aXRhbTEUMBIGA1UECwwLYXV0aG9yaXRpZXMxJTAjBgNVBAM  
MHGNhX2ludGVybWVkaWF0ZV90aW1lc3RhbXBpbmcwHhcNMTkxMTA3MDIzMzA1WWhcNMjIxMTA2MDIzMzA1WjBUMQs  
wCQYDVQGEwJmcjEMMAoGA1UECAwDaWRmMQ4wDAYDVQQHDAVwYXJpczEOMAwGA1UECgwFdm10YW0xZmZAVBGNVBA  
MDnNlY3VyZS1sb2dib29rMIIC1jANBgkqhkiG9w0BAQEFAAOCAg8AMIICCGKCAgEA091qSj5nvYlemZC4CHCzTpH  
NgK1WzA/  
ILRjVL4ho2vfT6yx9TstAQqsVARrsknAGJ2KMillffIOEZ00ph3H0oo7auWOUgDytsfTeRmDFSVOxUFM/  
1jBUXgqhg6VSHVka1ueAu0kg4g0ADvWrJbtVKHxps9vrEA+H4tEiTT3mfKEdWBAyLz4cG0ln4ip3s1GMSjmVgDwV

gbzPnHvD1VUDLij6dYHbkaQaLSRh9h147dPVbzOelqZAUPL6kVUI6fM9jxAzQJqub0jL+vEGvMUUMJTqStnFXAU  
C5Gof3LU5iV/cwyGUWY0rz5kKS1XLT9jvYteijxYe4nf0y/V5Yzp/6F/  
52esT4rqI5x9QGfv9TxShiymcu316aul4uFw+M+qLy4khC1BhNbsHojjnGhR33KuRa/  
QAZ1tPj3KyFC3NLSfGrJSi+JbZV5yS5yybkdlF25TdyUSFxxk+01QEvml6ngsjKQHqSNdsTWmAHdBuE614ErrySxG  
wjnFBMOfsPK2HKe3dc7iZUZyHPPUyBq8RVWdvHVfTQuXBCKCMSVnk6rVjmYMuSiI8oAy0TK7ixfAqMmE2yZwoJt7  
hMYbo0Adrfjg7Bf95uP2LDC8LkWsAlhIJDFOJT5ANXsWxGM0LmFmGTRoPFob2E0G/  
QQKAc0GEgRiG4qGihCLiZSGJ81CYw0CAwEAAOCATkwwgE1MCUGCWCsAGG+EIBDQYFhZDZXJ0awZpY2F0IFNlc  
nZldXIGU1NMB0GA1UdDQWBBRpel2gJ6UBuGBR8LG5d7Sa8kfmazCbmwYDVR0jBIGTMIQGQhZKjBQYoL+UCSH  
cYIsFk+rST006F0pHIwcDELMAKGA1UEBhMCZnIxDDAKBgNVBAGMA2lkZjEOMAwGA1UEBwwFcgFyaXMsDjAMBGNVBA  
AoMBXZpdGFtMRQwEgYDVQQLDAthdXRob3JpdGllczEdMBSGA1UEAwUY2Ffc9vdf90aW1lc3RhbXBpmeCAGDTM  
AkGA1UdEgQCAAwDAYDVR0TAQH/BAIwADALBgNVHQ8EBAMCBsAwEQYJYIZIAyB4QgEBBAQDAgZAMBGA1UdJQEB/  
wQMAoGCCsGAQUFBwMIMA0GCSqGSIB3DQEBcWUAA4ICAQAwjDb7PvGAmqgW62+mXQHh06hh8ikJyyz6kT10jLaRV  
Uou77ZMoUqF5GLS6d+tsPvGL5cdrs2Z5WqZPny3DyOxBBsGnIY9FM0herQAR6mY6xyTB+TrQDfQZeaQL0m1I/  
PAWiQNWjDUJCDGEhva0m046fJNKcXrPjb9cQ/  
CEsWd0J4ERBznUV8n0yVjCPQeW2yrY1hZrtYrfTgl7thy0INGBz/zySH15ArF11bd75Z3o9ABMrtTqPjJ5R/  
FeEzOYXZdpdc5C7UqQwyZxcQpRQlYJvedBDRBqGQ5AZ9KMNQEMQEHOe1zyXwoxpo2I1C/G/X/  
nvBIiqgVUGsxTdh/btOIQPvKS/PuvmuqUeCblKKW57txgY4g5387q8NuVMrhUc6iux3JlmcX7ismaAk/  
lAZPojzbNNxFw5N/  
4+q2TM8Q+0q31x09wyiGQqoCuP76Zv00REukpFgUsBf+Yot1qyTxFYxpwgW1hX9ed8VsJPHUNOy97Cd8MOPfgj+  
CefBemvMa8zQH958KWtkjLlYKAhQuXHM5E6CuDUpe98090+ulNp4g7lSSJBALOCMZWP1j0syFRIGuirNloDruhi  
lX/NOuGvBk5AUtVvsbFm3gOCpx+ijUMLcaBd5e6J0+xyvK2v0a+/x25hPjuLp/  
weXniMGtANyzbmvJrNkgUDGCA80wggPJAqEBMH4wDELMAKGA1UEBhMCZnIxDDAKBgNVBAGMA2lkZjEOMAwGA1U  
EBwwFcgFyaXMsDjAMBGNVBAoMBXZpdGFtMRQwEgYDVQQLDAthdXRob3JpdGllczElMCMGA1UEAwcY2FfaW50ZXJ  
tZWRRpYXRlX3RpbWVzdGFtcGUuZwICANQwDQYJYIZIAWUDBAIDBQCGggEgMBoGCSqGSIB3DQEJAzENBgsqhkiG9w0  
BCRABBDacBgkqhkiG9w0BCQUxDxcNMTkxMTA3MDQwMDA5WjAtBgkqhkiG9w0BCTQxIDAEMA0GCSqGSAFlAwQCAwU  
AoQ0GCSqGSIB3DQEBDQUAME8GCSqGSIB3DQEJBDfCBCEaBL12eV9W7T0Y2Qbm43JWJDlmeXFeRgiwCTSfh91uIPd  
xavc8huF5y5UuP9xeJ8gX7/NQ5ftuNy4SHxRM60YyMGQGCyqGSIB3DQEJEAIVMVUwZBRME8

wCwYJYIZIAWUDBAIDBEAeduVLSR3MT88jYzZAMR8nbaiHoZPSUFMPxK/1Ekh3DXK7z46X0bKIs87kWN8Rjacz7Qu  
F7nDQmPRcETZHzUjoma0GCSqGSIB3DQEBDQUABIIICALZUqEjXV7rYVWAj31YxJHADf3mYOv3aehjQ1GMIUCzzmd  
zGw9V3FagQvQB60rJg/sApdwgvrRM4uAX9TG0sVKigsZ/sxhnhEEG5CEDp93R16y47/  
TZYXs37Vqa3Qycd5xiEvL4zmJU12ZA2SKptweiqt3VmWID4cZkYonTgWtGF1Iew3YpDXd/  
1Dy1zV5AnAs65b037jVqxjVRUA2PUwxAvvOxG6uQLXj7G1bncZaQpSYhMER1R10lpNvQ1F3D30/  
RpVSWnVjbToLTeADDsk4fJjsIU1fZec3Im/  
OZw1rKVzEDZdtuRvCr7L1st1JtxEcZTHDgiKmQXN4t39h+S49nB0ldFq/  
V43IdqMT+bjKxQ1OG0x4dtnIvfdnH0eu93KuHr9ivGDp3GnDBLzePTX99kz2M2Ky1vBeVL2+MzN27CIdceuZlQ1zp  
ahXDgi+ioERHif1GS+P2iR4vF4GKbUvsYQURB2WNNaG/  
naFFGDPVcd4tB6rJDOBR1TzgwaseWUamlVebSkZBgvDialzB12Cuefhy+bMUPGbtQ1Wrrr6HizPTL2/  
KrPakCoZk5nw9/  
MICO+20eU6gP+s69TsV32CCTIpKxK87QD0NHZ2ZBj5Ejte4h1sYxtR+3HvQYITaH1YxHMvY0KZVytMqCMVM/  
Fyf2LGN9XjPd+ddDvyA7htRT",

"destinationComparable" :

"MIILITAVAgEAMBAMdk9wZXJhdGlubiBPa2F5MIILBgYJKoZihvcNAQcCoIiK9zCCCvMCAQMxDzANBg1ghkgBZQM  
EAgMFADCBgAYLkoZihvcNAQkQAQSGcQRvMG0CAQEGASkwUTANBg1ghkgBZQMEAgMFAARACPaKQZakyhRDnsAhDW  
afWC2tL1A1Wg0AZIk2ycySfEFppfp1hY6Zpy0c82fZ7srdKPl+uWyqxH6a5q5lDb15AIBARgPMjAxOTEExMdcwNDA  
wMDlaoIIGhzCCBoMwggRroAMCAQICAgDUMA0GCSqGSIB3DQEBcWUAMHgxCzAJBgNVBAYTAxZyMQwwCgYDVQQIDAN  
pZGYxZjAMBGNVBAcMBXBhcm1zMQ4wDAYDVQQKDAV2aXRhbTEUMBIGA1UECwwLYXV0aG9yaXRpZXMxJTAjBgNVBAM  
MHGNhX2ludGVybWVkaWF0ZV90aW1lc3RhbXBpbmcwHhcNMTkxMTA3MDIzMzA1WWhcNMjIxMTA2MDIzMzA1WjBUMQs  
wCQYDVQGEwJmcjEMMAoGA1UECAwDaWRmMQ4wDAYDVQQHDAVwYXJpczEOMAwGA1UECgwFdm10YW0xZmZAVBGNVBA  
MDnNlY3VyZS1sb2dib29rMIIC1jANBgkqhkiG9w0BAQEFAAOCAg8AMIICCGKCAgEA091qSj5nvYlemZC4CHCzTpH  
NgK1WzA/  
ILRjVL4ho2vfT6yx9TstAQqsVARrsknAGJ2KMillffIOEZ00ph3H0oo7auWOUgDytsfTeRmDFSVOxUFM/  
1jBUXgqhg6VSHVka1ueAu0kg4g0ADvWrJbtVKHxps9vrEA+H4tEiTT3mfKEdWBAyLz4cG0ln4ip3s1GMSjmVgDwV

gbzPnHvD1VUDLij6dYHbkaQaLSRh9h147dPVbzOelqZAUPL6kVUI6fM9jxAzQJqub0jL+vEGvMUUMJTqStnFXAU  
C5Gof3LU5iV/cwyGUWY0rz5kKS1XLT9jvYteijxYe4nf0y/V5Yzp/6F/  
52esT4rqI5x9QGfv9TxShiymcu316aul4uFw+M+qLy4khC1BhNbsHojjnGhR33KuRa/  
QAZ1tPj3KyFC3NLSfGrJSi+JbZV5yS5yybkdlF25TdyUSFxxk+01QEvml6ngsjKQHqSNdsTWmAHdBuE614ErrySxG  
wjnFBMOfsPK2HKe3dc7iZUZyHPPUyBq8RVWdvHVfTQuXBCKCMSVnk6rVjmYMuSiI8oAy0TK7ixfAqMmE2yZwoJt7

```

hMYbo0Adrfjg7Bf95uP2LDC8LkWsALhIJDFOJT5ANXsWxGM0LmFmGTRoPFOb2E0G/
QQKAc0GEGRiG4qGihCLiZSGJ81CYw0CAwEAAaOCATkwggE1MCUGCWCsSAGG+EIBDQQYFhZDXJ0aWZpY2F0IFN1c
nZldXIGu1NMMB0GAlUdDgQWBBRpel2gJ6UBuGBr8LG5d7Sa8kfmazCBmwYDVR0jBIGTMIGQgBQHkzKjBQYoL+UCSH
cYIsFk+rST006F0PHIwCDELMAkGAlUEBhMCZnIxDDAKBGNVBAgMA21kZjEOMAwGAlUEBwFcgFyaXmxDjAMBGNVB
AoMBXZpdGfTMRQwEgYDVQQLDAtHdXR0b3JpdGllczEdMBSGA1UEAwWUY2Ffcm9vdF90aW1lc3RhbXBpbmeCAGDTM
AkGAlUdEgQCMAAwDAYDVR0TAQH/BAIwADALBgNVHQ8EBAMCBsAwEQYJYIZIAyb4QgEBBAQDAGZAMBGA1UdJQEB/
wQMMaOGCCsGAQUFBwMIMA0GCSqGSIb3DQEBCwUAA4ICAQAwjDb7PvGAMggW62+mXQHh06hh8ikJyyz6kT10jLaRV
8ca77BMcUqF5C10Cd+L0p+CL5ed1s2E5Wq2Fny3DjOnDB08LTV9FM0h0zQmGmY6nyTB+T+QDfGqEcaqL0m1I/
PAWiQNwjDUJCDGEhvaoOm046fJNKcXrPjb9cQ/
CESWdOJ4ERBznUV8nOyVjCPQeW2yrY1hZrtYrpfTG17thy0INGBz/zySH15ArF11bd75Z3o9ABMtRtqPjJ5R/
FeEzOYXZdpdc5C7UqQwyZxcPqRQlYJvedBDRBqGQ5AZ9KMnQEmQEHOElzyxWoxpo2I1C/G/X/
nvBiIqgVUgsxTdh/btOIQPvKs/PuvmuqUeCb1KKWs7txgY4g5387q8NuVMrhUc6iux3J1mcX7ismaAk/
1AZPojzbNNxFwW5N/
4+q2TM8Q+0q3lx09wYiGQqoCuP76Zv00REukpFgUsBf+Yot1qyTxFYxpwgW1hX9ed8VsWJPHUNOy97Cd8MOPfgj+
CefBemvMa8zQH958KWtkjLlYKAhQuXHM5E6CuDUPE98090+ulNp4g71SSJBaLOCMZwPlji0syFRIGuirNloDruhi
lX/NOuGvBk5AUtvVvsbfFm3gOCpx+ijUMLcaBd5e6J0+xyvK2v0a+/x25hPjuLp/
weXmiMGtANyZbgmvJrNkgUDGCA80wggPJAgeBmH4weDELMAkGAlUEBhMCZnIxDDAKBGNVBAgMA21kZjEOMAwGAlU
EBwFcgFyaXmxDjAMBGNVBAoMBXZpdGfTMRQwEgYDVQQLDAtHdXR0b3JpdGllczEdMCMGA1UEAwW50ZXXJ
tZWVpYXRlX3RpbWVzdGFTcGluZwICANQwDQYJYIZIAWUDBAIDBQCgggEgMBoGCSqGSIb3DQEJAzENBgsqhkiG9w0
BCRABBDACBgkqhkiG9w0BCQUxXcNMTkxMTA3MDQwMDA5WjAtBgkqhkiG9w0BCTQxIDAEMA0GCWCGSAFlAwQCAwU
AoQ0GCSqGSIb3DQEBDQUAME8GCSqGSIb3DQEJBDfCBECaBl12eV9W7T0Y2Qbm43JWJDlmeXFeRgiwCTsfh91uIPd
xavc8huFSy5UuP9xeJ8gX7/
NQ5ftuNy4SHxRM60YyMGQGCyqGSIb3DQEJEAIVMVUwUzBRME8wCwYJYIZIAWUDBAIDBEAeduVLSR3MT88jYzzAMR
8nbaiHoZPSUfMPxK/
1Ekh3DXK7z46X0bKIs87kWN8Rjacz7QuF7nDQmPRcETZHzUjoma0GCSqGSIb3DQEBDQUABIIICALZUqEjXV7rYVWA
gJ31YxJHADf3mYov3aehjQlGMIuCzZmdzgW9V3FagQvQB60rJg/sApdwgvRM4uAX9TG0sVKigsZ/
sxhnhEEg5CEDp93R16y47/
TZYXs37Vqa3Qycd5xiEvL4zmJU12ZA2SKptweiqt3VmWID4cZkYonTgwtGFIwev3YpDXd/
1DylzV5AnAs65b037jVqxjVRUA2PUwxAvvOxG6uQLXj7GlbncTaQpSYhMERr1H01pNvQ1F3D30/
RpVSwNvjbToLTeADDsk4fJjsIU1fZec3Im/
OZw1rKVzEDZdtuRvCr7LIst1JtxEcZTHDgiKmQXN4t39h+S49nB0ldFq/
V43IdqMT+bjKxQ10G0x4dtnIvfDnH0eu93KuHr9ivGDp3GnDBLzePTX99kzM2Ky1vBeVL2+MzN27CIdceuZlQ1zp
ahXDgi+ioERHif1GS+P2iR4vF4GkbUvsYQURB2WNNaG/
naFFGfDPVcd4tB6rJDOBR1TzgwaseWUamlVebSkZBgvDIAIzB12CuefhY+bMUPGbtQlWrrr6HizPTL2/
KrPakCoZk5nw9/
MICO+20eU6gp+S69TsV32CCTIpKxK87QD0NHZ2ZBj5Ejte4h1sYxtr+3HvQYITaHlYxHMvy0KZVytMqCMVM/
Fyf2LGN9XjPd+ddDvyA7htRT",

```

```

    "action" : "COMPARISON",
    "item" : "PREVIOUS_TIMESTAMP_OPERATION",
    "status" : "OK"
}, {
    "name" : "EVENTS_OPERATION_DATABASE_TRACEABILITY_COMPARISON",
    "details" : "Comparing operation ID found in database with operation ID found in
traceability secured file.",
    "type" : "LOCAL_INTEGRITY",
    "source" : "DATABASE",
    "destination" : "TRACEABILITY_FILE",
    "sourceComparable" : "aeaaaaaachemhquabwdcaloiqpuvhiaaaaq",
    "destinationComparable" : "aeaaaaaachemhquabwdcaloiqpuvhiaaaaq",
    "action" : "COMPARISON",
    "item" : "EVENT_OPERATION",
    "status" : "OK"
}, {
    "name" : "TIMESTAMP_OBJECT_GROUP_DATABASE_TRACEABILITY_VALIDATION",

```





```
MHGNhX2ludGVybyWVkaWF0ZV90aW1lc3RhbXBpbmcwHhcNMTkxMTA3MDIzMzA1WhcNMjIxMTA2MDIzMzA1WjBUMQs
wCQYDVQQGEwJmcjEMMAoGA1UECAwDaWRmMQ4wDAYDVQQHDAVwYXJpczEOMAwGA1UECgwFdm10YW0xZmZAVBGNVBAW
MDnNlY3VyZS1sb2dib29rMIICIjANBgkqhkiG9w0BAQEFAAOCAg8AMIICCgKCAgEA09lqSj5nvYlemZC4CHCzTph
NgK1WzA/
ILRjVL4ho2vfT6yx9TstAQqsVARrsknAGJ2KMillffIOEZ00ph3H0oo7auWouGDytsfteRmdFSVOxUfM/
1jBUXgqhg6VSHVka1ueAu0kg4g0ADvWrJbtVKHxps9vrEA+H4tEiTT3mfKEdWBAYLz4cGoln4ip3slGMSjmVgDwV
gbzPnHvD1VUDLij6dYHbkaQaLSRh9h147dPVbzOelqZAUPLML6kVUI6fM9jxAzQJqub0jL+vEGvMUmJTqSTnFXAU
C5Gof3LU5iV/cwyGUWY0rz5kKS1XLT9jvYteiJxYe4nf0y/V5Yzp/6F/
52esT4rqI5x9QGfv9TxShiymcu316au14uFw+M+qLy4khC1BhNbsHojjnGhR33KuRa/
QAZltPJ3KyFC3NLSfGrJSi+JbZV5yS5yybkdlF25TdyUSFkk+0lQEvml6ngsjKQHqSNdsTWmAHdBuE614ErrySxG
wjnFBMofspk2HKe3dc7iZUZyHPuYBq8RVWdVHVftQuXBCkCMSVnk6rVjmYMuSiI8oAy0TK7ixfAqMmE2yZwoJt7
hMYbo0AdrFjg7Bf95uP2LDC8LkWsAlhIJDFOJT5ANXsWxGM0LmFmGTRoPF0b2E0G/
QQKAc0GEgRiG4qGihCLiZSGJ81CYw0CAwEAAOCATkwgE1MUCGWCWSAGG+EIBDQQYFhZDZXJ0aWZpY2F0IFNlc
nZlXIGU1NMMB0GA1UdDgQWBRRpel2gJ6UBuGBR8LG5d7Sa8kfmazCBmwYDVR0jBIGTMIGQgBQHzKjBQYoL+UCSH
cYIsFk+rST006F0PHIwcDELMAkGA1UEBhMCZnIxDDAKBgNVBAGMA2lkZjEOMAwGA1UEBwwFcGFyaXNlZjEOMAwGA1U
AoMBXZpdGFtMRQwEgYDVQQLDAthdXRob3JpdGllczEdMBSGA1UEAwUY2Ffcm9vdF90aW1lc3RhbXBpbmeCAgDTM
AKGA1UdEgQCAAwDAYDVROTAQH/BAIwADALBgNVHQ8EBAMCbsAwEQYJYIZIAyb4QgEBBAQDAGZAMBGA1UdJQEB/
wQMMaOGCCsGAQUFBwMIMA0GCSqGSIb3DQEBcUUA4ICAQAwjDb7PvGAmqgW62+mXQHHo6hh8ikJyyz6kT10jLaRV
Uou77ZMoUqF5GLS6d+tSpvGL5cdrs2Z5WqZPny3DyOxBBsGnIY9FM0herQAr6mY6xyTB+TrQDfqZeaL0mlI/
PAWlQNWjDUJCDGEhvaom046fJNKcXrPjb9cQ/
CESwdOJ4ERBznUv8nOyVjCPQeW2yrY1hZrtYrfpTGL7thy0INGBz/zySH15ArF11bd75Z3o9ABMtrTqPjJ5R/
FeEzOYXZdpdc5C7UqQwyZxcQpRQlYJvedBDRBqGQ5AZ9KMNqEmQEHoElzyxWoxpo2I1C/G/X/
nvBIiqgVUGsxTdh/btOIQPvKS/PuvmuqUeCb1KKW57txgY4g5387q8NuVMrhUc6iux3J1mcX7ismaK/
1AZPojzbNNxFwW5N/
4+q2TM8Q+0q3lx09wyiGQqoCuP76Zv00REukpFgUsBf+Yot1qyTxFYxpwgW1hX9ed8VsWJPHUNOy97Cd8MOPfgj+
CefBemvMa8zQH958KwtKjLlYKAhQuXHM5E6CuDUPe98090+ulNp4g7lSSJBALOCMZWplji0syFRIGuirNloDruhi
lX/NOuGvBk5AUtvVvsbFm3gOCpx+ijUMLcaBd5e6J0+xyvK2v0a+/x25hPjuLp/
weXmiMGtANyZbgmvJrNkgUDGCA80wggPJAqEBMH4weDELMAkGA1UEBhMCZnIxDDAKBgNVBAGMA2lkZjEOMAwGA1U
EBwwFcGFyaXNlZjEOMAwGA1UEBwwFcGFyaXNlZjEOMAwGA1UEBwwFcGFyaXNlZjEOMAwGA1UEBwwFcGFyaXNlZjE
OAwGA1UdEgQCAAwDAYDVROTAQH/BAIwADALBgNVHQ8EBAMCbsAwEQYJYIZIAyb4QgEBBAQDAGZAMBGA1UdJQEB/
wQMMaOGCCsGAQUFBwMIMA0GCSqGSIb3DQEBcUUA4ICAQAwjDb7PvGAmqgW62+mXQHHo6hh8ikJyyz6kT10jLaRV
Uou77ZMoUqF5GLS6d+tSpvGL5cdrs2Z5WqZPny3DyOxBBsGnIY9FM0herQAr6mY6xyTB+TrQDfqZeaL0mlI/
DBEAeduVLSR3MT88jYzzAMR8nbaiHoZPSUfMPxK/
1Ekh3DXK7z46X0bKIs87kWN8Rjacz7QuF7nDQmPRcETZjUjoma0GCSqGSIb3DQEBDQUABIIICAC73jsTyXRCDvxU
9EV/0U4CrLR/y04wnlF3m+/LAFuJNXANo7OVTjyj8zEyG+s2X4/
uyQ6pSTI67yTAEcFT6CnUqtaX2tGgDJHEvvl6qMNsErEwKNf8PnjiEjlmZf5E715hTOP+L0B/
EFZw7wr2yX5yVAECYBf2o/row9lS1MK590uMbkMcM6Eu/
yU6nOwY58ehiO2tAerR6cF9Io6EeVoqQNfMUPCSEVAT3f6+CigON30nhK3bCuX7w9TG287IKE57CWmQcwyPi5N9
A5WBBIEzITxOstEeullHaq3GE0yjo/
vv4dQeDrtfwi7Uyvw1irxpaRzvel8UANM9peZxJ4an0ev4QbY43XZ1FsnOtTNBhNGGhsnz1HED6r+CX59kjL8uTO
4dCljvcrGyrDw4g5i90SINjxtjtzeZ6746y2lXGnuOrdOTPRkQy22Bwdf63eP4m5IvkCT4vK1dZbOmPklhDMSLvKb
pfqsJ14e5o5a7osNj3XYR+HklmajZFzD9Q6OvFgMtLcY4Bpob7pDKqim4XBdb1AasDYES1eAhI4X8Oivx5Mh0iD2
vfBrInbLAAFn6UphpjOaWtVzgeK5NDwpTeOERop2fqEnS9Zak9zIwtI8/
bLXtVnnNxESX+h6MovQ8QW8cK8vvcSQQbt9IyxLvXAcUD8gjOUoVwzkzmm",
"action" : "VALIDATION",
"item" : "TIMESTAMP_OBJECT_GROUP",
"status" : "OK"
}, {
"name" : "TIMESTAMP_OBJECT_GROUP_DATABASE_TRACEABILITY_COMPARISON",
"details" : "Comparing timestamp object group found in database with timestamp
object group found in secured file.",
"type" : "TIMESTAMP_CHECKING",
"source" : "DATABASE",
"destination" : "TRACEABILITY_FILE",
"sourceComparable" :
"MIILITAVAgEAMBAMdk9wZXJhdGlvbiBPa2F5MIILBgyJKoZIHvcNAQcCoIiK9zCCCvMCAQMxDzANBg1ghkgBZQM
EAgMFADCBgAYLkoZIHvcNAQcCoIiK9zCCCvMCAQMxDzANBg1ghkgBZQMCAQEGASkwUTANBg1ghkgBZQMCAQEFARAIhHQjJ8lc0AFrgQw145
Wkds5E2849TF7x0QrgWS3Z/Q6l1p7d0HIYU/
N0rAIGzUdqfaBONivAjk99UL+XVol8QIBARgPMjAxOTExMDcwNTM4NDBoIIGhzCCBoMwggRroAMCAQICAgDUMA0
```



```

hMYbo0Adrfjg7Bf95uP2LDC8LkWsALhIJDFOJT5ANXsWxGM0LmFmGTRoPFOb2E0G/
QQKAc0GEGRiG4gGihCLiZSGJ81CYw0CAwEAAaOCATkwggE1MCUGCWCsSAGG+EIBDQQYFhZDXJ0aWZpY2F0IFN1c
nZldXIGuS1NMMB0GAlUdDgQWBBRpel2gJ6UBuGBr8LG5d7Sa8kfmazCBmwYDVR0jBIGTMIGQGqBQHzKjBQYoL+UCSH
cYIsFk+rST006F0PHIwCDELMAkGAlUEBhMCznIXDDAKBgNVBAgMA2lkZjEOMAwGA1UEBwwFcGFyaXNkZjAMBGNVB
AOMBXZpdGFTMRQwEgYDVQQLDAtHdXRob3JpdGllczEdMBSGA1UEAwUY2Ffcm9vdF90aW1lc3RhbXBpbmeCAGDTM
AkGAlUdEgQCMAAwDAYDVR0TAQH/BAIwADALBgNVHQ8EBAMCBsAwEQYJYIZIAyb4QgEBBAQDAGZAMBGA1UdJQEB/
wQMMaOGCCsGAQUFBwMIMA0GCSqGSIsb3DQEBwUAA4ICAQAwjDb7PvGAMggW62+mXQHHo6hh8ikJyyz6kT10jLaRV
Uou77ZMoUqF5G1S6d+tSpvGL5cdrs2Z5WqZPny3DyOxBBSGnIY9FM0herQAr6mY6xyTB+TrQDfQZeaqL0m1I/
PAW1QNwJDUUCDGenVaoOmU46tJNkcXrPjpb9cQ/
CESWdOJ4ERBznUV8nOyVjCPQeW2yrY1hZrtYrFpTG17thy0INGBz/zySH15ArF11bd75Z3o9ABMtrTqPjJ5R/
FeEzOYXZdpdc5C7UqQwyZXcQpRQlYJvedBDRbqGQ5AZ9KMnQEmQEHOElzyxWoxpo2I1C/G/X/
nvBIiqgVUgxsTdh/btOIQPvKs/PuvmuqUeCb1KKWs7txgY4g5387q8NuVMrhUc6iux3J1mcX7ismaAk/
1AZPojzbNNxFwW5N/
4+q2TM8Q+0q3lx09wYiGQqoCuP76Zv00REukpFgUsBf+Yot1qyTxFYxpwgW1hX9ed8VsWJPHUNOy97Cd8MOPfgj+
CefBemvMa8zQH958KWtkjLlYKAhQuXHM5E6CuDUPE98090+ulNp4g71SSJBaLOCMZwPlji0syFRIGuirNloDruhi
lX/NOuGvBk5AUtvVvsbfFm3gOCpx+ijUMLcaBd5e6J0+xyvK2v0a+/x25hPjuLp/
weXmiMGtANyZbgmvJrNkgUDGCA80wggPJAqEBMH4weDELMAkGAlUEBhMCznIXDDAKBgNVBAgMA2lkZjEOMAwGA1U
EBwwFcGFyaXNkZjAMBGNVBaOMBXZpdGFTMRQwEgYDVQQLDAtHdXRob3JpdGllczE1MCMGA1UEAwY2FfaW50ZXJ
tZWVpYXRlX3RpbWVzdGFTcGluZwICANQwDQYJYIZIAWUDBAIDBQCgggEgMB0GCSqGSIsb3DQE
JAzENBgsqhkig9w0BCRABBDacBgkqhkiG9w0BCQUxDxcNMTkxMTA3MDUzODQwWjAtBgkqhkiG9w0BCTQxIDAEMA0
GCWCGSAlAwQCAwUAoQ0GCSqGSIsb3DQEBDQUAME8GCSqGSIsb3DQEBJDFCBEA8CxI4X4tNwBuejeHjuy8XzJ9Bnty
IAJU9hqh+JySsyL5qsU89Z0xMZhgTEfWEeVGMcvhPqiSdWn7dRZvzRNnfmGQGCyqGSIsb3DQEBJEAIVMVUwUzBRME8
wCwYJYIZIAWUDBAIDBEAeduVLSR3MT88jYzzAMR8nbaiHoZPSUFMPxK/
1Ekh3DXK7z46X0bKI87kWN8Rjacz7QuF7nDQmPRcETZJHjUjoma0GCSqGSIsb3DQEBDQUABIICAC73jsTyXRCDvxU
9EV/0U4CrLR/y04wN1F3m+/LAFuJNXAN
70VTjyj8zEyG+s2X4/
uyQ6pSTI67yTAEcFT6CnUqtaX2tGgDJHEvv16qMNsRwKNFb8PnjieJlmZf5E715hTOP+L0B/
EFZw7wr2yX5YvAECYBf2o/row91S1MK590uMbkMcM6Eu/
yU6nOwY58ehiO2tAeRR6cf9Io6EeVoqQNfMUPCSEVAT3f6+CiGON30nhK3bCuX7w9TG287IKE57CWmQcwYpi5N9
A5WBBIEzITxOstEeullHaq3GE0yjo/
vv4dQeDrtfwi7Uyvv1irxpaRzvel8UANM9peZxJ4an0ev4QbY43XZ1FsntOTNBHnGGHsnZ1HED6r+CX59kjL8uTO
4dCljvcrGyrDw4g5i90SINjxtjteZ6746y21XgnuOrdOTPRkQy22BWdf63eP4m5IvkCT4vK1dZbOmPklhDMSLvKb
pfqsJ14e5o5a7osNj3XYR+HklmajZFzD9Q6OvFgMtLcY4Bpob7pDKqim4XBdb1AasDYESleAhI4X80ivx5Mh0iD2
vfBrInbLAAFn6UphpjOaWtVZgeK5NDwpTe0E9Rop2fqEnS9Zak9zIwtI8/
bLXtVnnNxE3SX+h6MovQ8QW8cK8vvcSQQt9IyxLvXAcUD8gjOUoVwzkzmm",
    "action" : "COMPARISON",
    "item" : "TIMESTAMP_OBJECT_GROUP",
    "status" : "OK"
}, {
    "name" : "MERKLE_OBJECT_GROUP_DIGEST_DATABASE_TRACEABILITY_COMPARISON",
    "details" : "Comparing object group merkle digest found in database with object
group merkle digest found in traceability secured file.",
    "type" : "MERKLE_INTEGRITY",
    "source" : "DATABASE",
    "destination" : "TRACEABILITY_FILE",
    "sourceComparable" :
"CK8TOSHFhaw20w9veL19mg/6If9ogkJayMrWP11yY9u157YseSO97wnpJA3MaW6eoV2Mn9LjO6LxY86Gv1Roiw=
=",
    "destinationComparable" :
"CK8TOSHFhaw20w9veL19mg/6If9ogkJayMrWP11yY9u157YseSO97wnpJA3MaW6eoV2Mn9LjO6LxY86Gv1Roiw=
=",
    "action" : "COMPARISON",
    "item" : "MERKLE_TREE_ROOT_OBJECT_GROUP_DIGEST",
    "status" : "OK"

```

```

    }, {
      "name" : "MERKLE_OBJECT_GROUP_DIGEST_COMPUTATION_TRACEABILITY_COMPARISON",
      "details" : "Comparing object group merkle digest computed from secured data with
object group merkle digest found in traceability secured file.",
      "type" : "MERKLE_INTEGRITY",
      "source" : "COMPUTATION",
      "destination" : "TRACEABILITY_FILE",
      "sourceComparable" :
"CK8TOSHFhaw20w9veL19mg/6If9ogkJayMrWP11yY9u157YseSO97wnpJA3MaW6eoV2Mn9LjO6LxY86Gv1Roiw=
=",
      "destinationComparable" :
"CK8TOSHFhaw20w9veL19mg/6If9ogkJayMrWP11yY9u157YseSO97wnpJA3MaW6eoV2Mn9LjO6LxY86Gv1Roiw=
=",
      "action" : "COMPARISON",
      "item" : "MERKLE_TREE_ROOT_OBJECT_GROUP_DIGEST",
      "status" : "OK"
    }, {
      "name" :
"MERKLE_OBJECT_GROUP_DIGEST_COMPUTATION_ADDITIONAL_TRACEABILITY_COMPARISON",
      "details" : "Comparing object group merkle digest computed from secured data with
object group merkle digest found in additional traceability secured file.",
      "type" : "MERKLE_INTEGRITY",
      "source" : "COMPUTATION",
      "destination" : "ADDITIONAL_TRACEABILITY",
      "sourceComparable" :
"CK8TOSHFhaw20w9veL19mg/6If9ogkJayMrWP11yY9u157YseSO97wnpJA3MaW6eoV2Mn9LjO6LxY86Gv1Roiw=
=",
      "destinationComparable" :
"CK8TOSHFhaw20w9veL19mg/6If9ogkJayMrWP11yY9u157YseSO97wnpJA3MaW6eoV2Mn9LjO6LxY86Gv1Roiw=
=",
      "action" : "COMPARISON",
      "item" : "MERKLE_TREE_ROOT_OBJECT_GROUP_DIGEST",
      "status" : "OK"
    }, {
      "name" : "TIMESTAMP_OBJECT_GROUP_COMPUTATION_TRACEABILITY_COMPARISON",
      "details" : "Comparing timestamp object group computed from computing information
traceability file with timestamp object group found in secured file.",
      "type" : "TIMESTAMP_CHECKING",
      "source" : "COMPUTATION",
      "destination" : "TRACEABILITY_FILE",
      "sourceComparable" :
"ihHQjJ81c0AFrgQw145Wkds5E2849TF7x0QrgWS3Z/Q611p7d0HIYU/N0rAIGzUdqfaBONIVAjK99UL+XV0l8Q=
=",
      "destinationComparable" : "ihHQjJ81c0AFrgQw145Wkds5E2849TF7x0QrgWS3Z/Q611p7d0HIYU/

```

```

N0rAIGzUdqfaBONivAjk99UL+XVol8Q==" ,
    "action" : "COMPARISON",
    "item" : "TIMESTAMP_OBJECT_GROUP",
    "status" : "OK"
}, {
    "name" : "PREVIOUS_TIMESTAMP_OBJECT_GROUP_DATABASE_TRACEABILITY_VALIDATION",
    "details" : "Validating previous timestamp object group found in database 'AND'
previous timestamp object group found in traceability secured file.",
    "type" : "CHAIN",
    "source" : "DATABASE",
    "destination" : "TRACEABILITY_FILE",
    "sourceComparable" :
"ihHQjJ81c0AFrgQw145Wkds5E2849TF7x0QrgWS3Z/Q6lFp7d0HIYU/N0rAIGzUdqfaBONivAjk99UL+XVol8F=
=",
    "destinationComparable" : "ihHQjJ81c0AFrgQw145Wkds5E2849TF7x0QrgWS3Z/Q6lFp7d0HIYU/
N0rAIGzUdqfaBONivAjk99UL+XVol8F==",
    "action" : "VALIDATION",
    "item" : "PREVIOUS_TIMESTAMP_OBJECT_GROUP",
    "status" : "OK"
}, {
    "name" : "PREVIOUS_TIMESTAMP_OBJECT_GROUP_DATABASE_TRACEABILITY_COMPARISON",
    "details" : "Comparing previous timestamp object group found in database with
previous timestamp object group found in traceability secured file.",
    "type" : "CHAIN",
    "source" : "DATABASE",
    "destination" : "TRACEABILITY_FILE",
    "sourceComparable" :
"ihHQjJ81c0AFrgQw145Wkds5E2849TF7x0QrgWS3Z/Q6lFp7d0HIYU/N0rAIGzUdqfaBONivAjk99UL+XVol8Q=
=",
    "destinationComparable" :
"ihHQjJ81c0AFrgQw145Wkds5E2849TF7x0QrgWS3Z/Q6lFp7d0HIYU/
N0rAIGzUdqfaBONivAjk99UL+XVol8Q==" ,
    "action" : "COMPARISON",
    "item" : "PREVIOUS_TIMESTAMP_OBJECT_GROUP",
    "status" : "OK"
}, {
    "name" : "FILE_DIGEST_DATABASE_TRACEABILITY_COMPARISON",
    "details" : "Comparing binary file digest found in object group database with
binary file digest found in traceability secured file.",
    "type" : "LOCAL_INTEGRITY",
    "source" : "DATABASE",

```

```

        "destination" : "TRACEABILITY_FILE",

        "sourceComparable" :
"9fe79f5153bd0b6996a23de06544d2af966dcc9e39cc46b49ae05bf6429bb88ff8d34b9bd0e6fd23a1b0681
136a9767976b6fb65cea41393e54a167e0752d620",

        "destinationComparable" :
"9fe79f5153bd0b6996a23de06544d2af966dcc9e39cc46b49ae05bf6429bb88ff8d34b9bd0e6fd23a1b0681
136a9767976b6fb65cea41393e54a167e0752d620",

        "action" : "COMPARISON",

        "item" : "FILE_DIGEST",

        "status" : "OK"
    }, {

        "name" : "EVENTS_OBJECT_GROUP_DIGEST_DATABASE_TRACEABILITY_COMPARISON",

        "details" : "Comparing object group lfc digest computed from logbook lifecycle
object group database with object group lfc digest found in traceability secured file.",

        "type" : "LOCAL_INTEGRITY",

        "source" : "DATABASE",

        "destination" : "TRACEABILITY_FILE",

        "sourceComparable" :
"UqMkFb5S2HCa/DJ92kadPX4ertC0IssWStvpA18UKEwYyYekTzFAeqIJPETGDeh6VBHLJsXWYvAGfwnqA36qCw=
=",

        "destinationComparable" :
"UqMkFb5S2HCa/DJ92kadPX4ertC0IssWStvpA18UKEwYyYekTzFAeqIJPETGDeh6VBHLJsXWYvAGfwnqA36qCw=
=",

        "action" : "COMPARISON",

        "item" : "EVENT_OBJECT_GROUP",

        "status" : "OK"
    }, {

        "name" : "FILE_DIGEST_OFFER_DATABASE_COMPARISON",

        "details" : "Comparing binary file digest stored in offers with binary file digest
found in object group database.",

        "type" : "LOCAL_INTEGRITY",

        "source" : "OFFER",

        "destination" : "DATABASE",

        "sourceComparable" :
"9fe79f5153bd0b6996a23de06544d2af966dcc9e39cc46b49ae05bf6429bb88ff8d34b9bd0e6fd23a1b0681
136a9767976b6fb65cea41393e54a167e0752d620",

        "destinationComparable" :
"9fe79f5153bd0b6996a23de06544d2af966dcc9e39cc46b49ae05bf6429bb88ff8d34b9bd0e6fd23a1b0681
136a9767976b6fb65cea41393e54a167e0752d620",

        "action" : "COMPARISON",

        "item" : "FILE_DIGEST",

        "status" : "OK"
    }, {

        "name" : "FILE_DIGEST_LFC_DATABASE_COMPARISON",
    
```



```
    "details" : "Comparing binary file digest found in object group database with  
binary file digest found in logbook lifecycle object group database.",  
  
    "type" : "LOCAL_INTEGRITY",  
    "source" : "DATABASE",  
    "destination" : "DATABASE",  
    "sourceComparable" :  
"9fe79f5153bd0b6996a23de06544d2af966dcc9e39cc46b49ae05bf6429bb88ff8d34b9bd0e6fd23a1b0681  
136a9767976b6fb65cea41393e54a167e0752d620",  
    "destinationComparable" :  
"9fe79f5153bd0b6996a23de06544d2af966dcc9e39cc46b49ae05bf6429bb88ff8d34b9bd0e6fd23a1b0681  
136a9767976b6fb65cea41393e54a167e0752d620",  
    "action" : "COMPARISON",  
    "item" : "FILE_DIGEST",  
    "status" : "OK"  
  } ],  
  "evStartDateTime" : "2019-11-07T09:18:05.178",  
  "evEndDateTime" : "2019-11-07T09:18:06.640",  
  "status" : "OK"  
} ],  
"ReportVersion" : 2  
}
```

## 14.7.2.Détails du rapport

La première partie du rapport « operationSummary » fait état des informations essentielles sur l'opération :

- « tenant » : le tenant sur lequel a été lancée l'opération consistant à générer le relevé de valeur probant
- « evId » : l'identifiant unique (GUID) de l'opération ;
- « evType » : le type de l'opération, ici EXPORT\_PROBATIVE\_VALUE ;
- « outcome » : le code de résultat global de l'opération (OK, KO ou WARNING) ;
- « outDetail » : le code de résultat propre à l'opération ;
- « outMsg » : le message explicite de résultat ;
- « rightsStatementIdentifiant » : le contrat d'accès utilisé pour lancer l'opération.

La deuxième partie « reportSummary » est constituée du résumé du rapport :

- « evStartDateTime » : la date de l'enregistrement du premier statut dans le rapport ;
- « evEndDateTime » : la date de l'enregistrement du dernier statut dans le rapport, postérieure à la date d'enregistrement du premier statut dans le rapport ;
- « reportType » : le type de rapport, ici PROBATIVE\_VALUE
- « vitamResults » : la liste des valeurs de statut d'exécution sur chaque objet, ici OK, KO et WARNING, ainsi que le total

La quatrième et dernière partie « reportEntries » rend compte des vérifications faites sur chaque objet binaire concerné, avec, pour chacun, un objet json contenant :

- « unitIds » : le tableau des unités archivistiques sélectionnées par la requête et contenant cet objet binaire (dans le cas général un seul) ;
- « objectGroupId » : l'identifiant unique du groupe d'objets techniques contenant cet objet binaire (GUID du GOT)
- « objectId » : l'identifiant unique de l'objet binaire concerné (GUID d'objet)
- « usageVersion » : le couple « usage\_version » de l'objet concerné (ex. BinaryMaster<sub>1</sub>) ;
- « operations » : un tableau contenant les opérations à vérifier sur l'objet binaire concerné, à savoir :
  - l'opération (entrée/INGEST ou préservation/PRESERVATION) qui a conduit à la prise en charge de cet objet binaire dans le système,
  - l'opération de sécurisation de cette opération de création ;
  - l'opération de sécurisation du journal de cycle de vie au moment de la création.
- Pour chacune de ces opérations sont fournies le type de l'opération, la date de fin de l'opération, ainsi que des éléments de contexte spécifiques aux opérations d'entrée/INGEST, à savoir :
  - le contrat d'entrée « rightsStatementIdentifiant »,
  - le contexte applicatif d'entrée « agIdApp » et
  - le jeton fourni par l'applicatif d'entrée « evIdAppSession »
- « checks » : le tableau de toutes les vérifications faites sur l'objet binaire concerné. Ces vérifications sont toutes décrites selon le même modèle qui permet de comparer les deux valeurs provenant de sources différentes :
  - « name » : le code de la vérification ;



- « details » : le libellé explicite de la vérification ;
- « type » : le type de la vérification à savoir « TIMESTAMP\_CHECKING », « MERKLE\_INTEGRITY », « CHAIN » ou « LOCAL\_INTEGRITY »
- « source » : l'origine de l'information à vérifier à savoir « DATABASE » (base de données Mongo), « TRACEABILITY\_FILE » (fichier de sécurisation) ou « OFFER » (offres de stockage)
- « destination » : l'origine de l'information permettant d'assurer la vérification à savoir « DATABASE », « TRACEABILITY\_FILE » ou « OFFER »
- « sourceComparable » : la valeur à comparer venant de la source ;
- « destinationComparable » : la valeur à comparer venant de la destination ;
- « action » : l'action faite lors de la vérification, en général « COMPARAISON » mais peut être aussi « VALIDATION » si, outre la comparaison il est possible de vérifier le contenu de la valeur, ce qui est le cas pour un tampon d'horodatage ;
- « item » : la partie d'information de l'objet qui est vérifiée ;
- « status » : le résultat de la vérification
- « SecuredOperationId » : identifiant de l'opération de sécurisation
- « SecureOperationIdForOpId » : sécurisation de l'opération de sécurisation
- « Checks » : tableau retraçant les contrôles effectués sur les logbooks en question, conformité à la sécurisation et arbre de Merkle
  - « Name » : nom du logbook concerné « checkLogbookSecureInfoForOpi »
  - « Status » : statut du contrôle
  - « Name » : « CheckObjectHash », vérification du hash de l'objet
  - « Status » : statut du contrôle
  - « Name » : « checkLfcStorageEvent », vérification de la sécurisation des événements à la date de la dernière sauvegarde
  - « Status » : statut du contrôle
- « Operations Reports » : tableaux rassemblant les différents journaux des opérations logbooks, liste des opérations et vérification des logbooks en question
  - « EvTypeProc » : type d'opération, toujours « TRACEABILITY »
  - « Id » : identifiant de l'opération
  - « OperationCheckStatus » : le résultat du check de l'opération
  - « Details » : message de vérification de l'opération en question par rapport à la sécurisation
  - Dans le cas d'une opération d'entrée
    - « EvIdAppSession » : mention du contrat d'entrée utilisé lors de l'opération d'entrée
    - « agIdApp » : mention du contexte applicatif utilisé pour réaliser l'opération d'entrée

Et finalement le statut final de la vérification de l'objet.

## 15. PRÉSERVATION

Cette section décrit le processus (workflow) de préservation mis en œuvre dans la solution logicielle Vitam. Ce workflow permet d'effectuer quatre types actions sur un lot d'objets binaires pris en charge dans la solution logicielle Vitam : analyse (ex. Validation de format), extraction de métadonnées tant techniques que descriptives, génération d'objets binaires et ré-identification du format des fichiers.

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.

### Workflow de Préservation

#### 15.1. Processus de préparation du traitement de la préservation (STP\_PREPARATION)

##### 15.1.1. Vérification des seuils de limitation des traitements

###### CHECK\_DISTRIBUTION\_THRESHOLD (CheckDistributionThreshold.java)

- **Règle** : tâche consistant à vérifier les seuils de traitement des unités archivistiques
- **Type** : bloquant
- **Statuts** :
  - OK : la vérification des seuils de limitation de traitement des unités archivistiques a été effectuée (CHECK\_DISTRIBUTION\_THRESHOLD.OK = Succès de la vérification des seuils de limitation de traitement des unités archivistiques)
  - KO : une incohérence a été détectée entre le seuil et le nombre d'unités archivistiques à traiter (CHECK\_DISTRIBUTION\_THRESHOLD.KO = Échec de la vérification des seuils de limitation de traitement des unités archivistiques)
  - FATAL : une erreur technique est survenue lors de la vérification du nombre d'unités archivistiques (CHECK\_DISTRIBUTION\_THRESHOLD.FATAL = Erreur technique lors de la vérification des seuils de limitation de traitement des unités archivistiques)

##### 15.1.2. Préparation du traitement de préservation PRESERVATION\_PREPARATION (PreservationPreparationPlugin.java)

- **Règle** : tâche consistant à récupérer la liste des groupes d'objets techniques à traiter, ainsi qu'à récupérer le scénario de préservation et à créer le fichier de distribution
- **Type** : bloquant
- **Statuts** :
  - OK : succès de la récupération des groupes d'objets techniques et du scénario de préservation ainsi que de la création du fichier de distribution (PRESERVATION\_PREPARATION.OK = Succès de la préparation du traitement de préservation)
  - KO : échec de la récupération des groupes d'objets techniques et du scénario de préservation ainsi que de la création du fichier de distribution impossible (PRESERVATION\_PREPARATION.KO = Échec de la préparation du traitement de préservation)
  - FATAL : une erreur technique est survenue lors de récupération des groupes d'objets techniques et du scénario de préservation ainsi que de la création du fichier de distribution. (PRESERVATION\_PREPARATION.FATAL = Erreur technique lors de la préparation du traitement de préservation)

## 15.2. Processus de lancement du griffon de la préservation (STP\_PRESERVATION\_ACTION)

### 15.2.1. Vérification de la disponibilité des ressources

#### PRESERVATION\_CHECK\_RESOURCE\_AVAILABILITY (PreservationCheckResourceAvailability.java)

- **Règle** : tâche consistant à vérifier la disponibilité synchrone en lecture des objets binaires. En cas d'indisponibilité d'un ou plusieurs objets binaires l'exécution de l'étape courante est renvoyée au moteur de stockage pour ré-exécution ultérieure quand les données seront disponibles.
- **Type** : bloquant
- **Statuts** :
  - OK : les objets binaires sont disponibles (PRESERVATION\_CHECK\_RESOURCE\_AVAILABILITY.OK = Succès de la vérification de la disponibilité des objets binaires sur l'offre de stockage)
  - FATAL : une erreur technique est survenue lors de la vérification de la disponibilité des objets binaires (PRESERVATION\_CHECK\_RESOURCE\_AVAILABILITY.FATAL = Erreur technique lors de la vérification de la disponibilité des objets binaires sur l'offre de stockage)

### 15.2.2. Lancement du griffon PRESERVATION\_ACTION (PreservationActionPlugin.java)

- **Règle** : tâche consistant à lancer le griffon
- **Type** : bloquant
- **Statuts** :
  - OK : succès de lancement du griffon (STP\_PRESERVATION\_ACTION.OK = Succès du lancement du griffon)
  - KO : échec du lancement du griffon (STP\_PRESERVATION\_ACTION.KO = Échec du lancement du griffon)
  - FATAL : une erreur technique est survenue lors du lancement du griffon (STP\_PRESERVATION\_ACTION.FATAL = Erreur technique lors du lancement du griffon)

### 15.2.3. Identification du format des objets binaires

#### PRESERVATION\_SIEGFRIED\_IDENTIFICATION (PreservationSiegfriedPlugin.java)

- **Règle** : tâche et traitement consistant à identifier le format des objets binaires créés lors d'une action de génération de binaire et à vérifier que le format identifié est bien référencé dans le référentiel interne
- **Type** : bloquant
- **Statuts** :
  - OK : l'identification des objets binaires générés a bien été effectuée (PRESERVATION\_SIEGFRIED\_IDENTIFICATION.OK = Succès de l'identification du format des objets binaires)
  - KO : échec de l'identification des objets binaires générés (PRESERVATION\_SIEGFRIED\_IDENTIFICATION.KO = Échec de l'identification du format des objets binaires)
  - FATAL : une erreur technique est survenue lors de l'identification des objets binaires générés (PRESERVATION\_SIEGFRIED\_IDENTIFICATION.FATAL = Erreur technique lors de l'identification des objets binaires)

### 15.2.4. Calcul de l'empreinte des nouveaux objets binaires

#### PRESERVATION\_BINARY\_HASH (PreservationGenerateBinaryHash.java)

- **Règle** : traitement consistant à calculer l'empreinte pour les objets binaires

- **Type** : bloquant
- **Statuts** :
  - OK : succès du calcul d'une empreinte pour les objets binaires générés (PRESERVATION\_BINARY\_HASH.OK = Succès du calcul d'empreinte pour les objets binaires générés)
  - KO : échec du calcul d'empreinte pour les objets binaires générés (PRESERVATION\_BINARY\_HASH.KO = Échec du calcul d'empreinte pour les objets binaires générés)
  - FATAL : une erreur technique est survenue lors du calcul d'une empreinte pour les objets binaires générés (PRESERVATION\_BINARY\_HASH.FATAL = Erreur technique lors du calcul d'empreinte pour les objets binaires générés)

### 15.2.5.Écriture des objets binaires générés sur les offres de stockage PRESERVATION\_STORAGE\_BINARY (PreservationStorageBinaryPlugin.java)

- **Règle** : tâche consistant à écrire les objets binaires générés sur les offres de stockage
- **Type** : bloquant
- **Statuts** :
  - OK : succès de l'écriture des objets binaires générés sur les offres de stockage (PRESERVATION\_STORAGE\_BINARY.OK = Succès de l'écriture des objets binaires sur les offres de stockage)
  - KO : échec de l'écriture des objets binaires générés sur les offres de stockage (PRESERVATION\_STORAGE\_BINARY.KO = Échec de l'écriture des objets binaires sur les offres de stockage)
  - FATAL : une erreur technique est survenue lors de l'écriture des objets binaires sur les offres de stockage (PRESERVATION\_STORAGE\_BINARY.FATAL = Erreur technique lors de l'écriture des objets et des groupes d'objets techniques sur les offres de stockage)

### 15.2.6.Vérification de la sécurité des unités archivistiques PRESERVATION\_OBJECTGROUP\_METADATA\_SECURITY\_CHECKS

#### (PreservationObjectGroupMetadataSecurityChecks.java)

- **Règle** : tâche consistant à vérifier la sécurité des groupes d'objets techniques dans le cadre d'une opération d'extraction de métadonnées afin d'alimenter les métadonnées techniques des groupes d'objets techniques
- **Type** : bloquant
- **Statuts** :
  - OK : la vérification de la sécurité des groupes d'objets techniques a bien été effectuée (PRESERVATION\_OBJECTGROUP\_METADATA\_SECURITY\_CHECKS.OK = Succès de la vérification de la sécurité des groupes d'objets techniques)
  - KO : la vérification de la sécurité des groupes d'objets techniques n'a pas pu s'effectuer (PRESERVATION\_OBJECTGROUP\_METADATA\_SECURITY\_CHECKS.KO = Échec de la vérification de la sécurité des groupes d'objets techniques)
  - FATAL : une erreur technique est survenue lors de la vérification de la sécurité des groupes d'objets techniques (PRESERVATION\_OBJECTGROUP\_METADATA\_SECURITY\_CHECKS.FATAL = Erreur technique lors de la vérification de la sécurité des groupes d'objets techniques)

### 15.2.7.Indexation des métadonnées de préservation PRÉSERVATION\_INDEXATION\_METADATA (PreservationUpdateObjectGroupPlugin.java)

- **Règle** : tâche consistant à indexer les métadonnées des unités archivistiques et des groupes d'objets

techniques modifiées ou générées lors des actions de préservation. A la fin de cette étape, les journaux du cycle de vie des unités archivistiques et des groupes d'objets techniques seront sauvegardés dans la base de données.

- **Type** : bloquant
- **Statuts** :
  - OK : succès de l'indexation des métadonnées générées lors des actions de préservation (PRESERVATION\_INDEXATION\_METADATA.OK = Succès de l'indexation des métadonnées générées lors des actions de préservation)
  - KO : échec de l'indexation des métadonnées générées lors des actions de préservation (PRESERVATION\_INDEXATION\_METADATA.KO = Échec de l'indexation des métadonnées générées lors des actions de préservation)
  - FATAL : une erreur technique est survenue lors de l'indexation des métadonnées générées lors des actions de préservation (PRESERVATION\_INDEXATION\_METADATA.FATAL = Erreur technique lors de l'indexation des métadonnées générées lors des actions de préservation)

#### 15.2.8. Sauvegarde des métadonnées et des journaux de cycle de vie sur les offres de stockage PRESERVATION\_STORAGE\_METADATA\_LFC (PreservationStorageMetadataAndLfc.java)

- **Règle** : tâche consistant à sauvegarder les métadonnées et les journaux du cycle de vie sur les offres de stockage
- **Type** : bloquant
- **Statuts** :
  - OK : Succès de la sauvegarde des métadonnées et des journaux du cycle de vie sur les offres de stockage (PRESERVATION\_STORAGE\_METADATA\_LFC.OK = Succès de la sauvegarde des métadonnées et des journaux du cycle de vie sur les offres de stockage)
  - KO : Échec de la sauvegarde des métadonnées et des journaux du cycle de vie sur les offres de stockage (PRESERVATION\_STORAGE\_METADATA\_LFC.KO = Échec de la sauvegarde des métadonnées et des journaux du cycle de vie sur les offres de stockage)
  - FATAL : une erreur technique est survenue lors de la sauvegarde des métadonnées et des journaux du cycle de vie sur les offres de stockage (PRESERVATION\_STORAGE\_METADATA\_LFC.FATAL = Erreur technique lors de la sauvegarde des métadonnées et des journaux du cycle de vie sur les offres de stockage)

#### 15.2.9. Vérification de la sécurité des unités archivistiques UNIT\_METADATA\_SECURITY\_CHECKS

##### (PreservationUnitMetadataSecurityChecks.java)

- **Règle** : tâche consistant à vérifier la sécurité des unités archivistiques dans le cadre d'une opération d'extraction de métadonnées afin d'alimenter les métadonnées descriptives des unités archivistiques
- **Type** : bloquant
- **Statuts** :
  - OK : la vérification de la sécurité des unités archivistiques a bien été effectuée (UNIT\_METADATA\_SECURITY\_CHECKS.OK = Succès de la vérification de la sécurité des unités archivistiques)
  - KO : la vérification de la sécurité des unités archivistiques n'a pas pu s'effectuer (UNIT\_METADATA\_SECURITY\_CHECKS.KO = Échec de la vérification de la sécurité des unités archivistiques)
  - FATAL : une erreur technique est survenue lors de la vérification de la sécurité des unités archivistiques (UNIT\_METADATA\_SECURITY\_CHECKS.FATAL = Erreur technique lors de la

vérification de la sécurité des unités archivistiques)

### 15.2.10.Extraction des métadonnées descriptives au moyen du griffon PRESERVATION\_EXTRACTION\_AU

#### (PreservationExtractionAUPlugin.java)

- **Règle** : tâche consistant à extraire les métadonnées descriptives au moyen du griffon dans le cadre d'une opération d'extraction de métadonnées afin d'alimenter les métadonnées descriptives des unités archivistiques
- **Type** : bloquant
- **Statuts** :
  - OK : l'extraction des métadonnées descriptives a bien été effectuée (PRESERVATION\_EXTRACTION\_AU.OK = Succès de l'extraction des métadonnées descriptives)
  - KO : l'extraction des métadonnées descriptives n'a pas pu s'effectuer (PRESERVATION\_EXTRACTION\_AU.KO = Échec de l'extraction des métadonnées descriptives)
  - FATAL : une erreur technique est survenue lors de l'extraction des métadonnées descriptives (PRESERVATION\_EXTRACTION\_AU.FATAL = Erreur technique lors de l'extraction des métadonnées descriptives)

### 15.3.Processus de préparation de l'insertion de métadonnées descriptives (STP\_PRESERVATION\_PREPARATION\_INSERTION\_AU\_METADATA) (PreservationPreparationInsertionAuMetadata.java)

- **Règle** : processus consistant à préparer l'insertion des métadonnées descriptives dans les unités archivistiques
- **Type** : bloquant
- **Statuts** :
  - OK : la préparation de l'insertion de métadonnées descriptives dans les unités archivistiques a bien été effectuée (STEP\_PRESERVATION\_PREPARATION\_INSERTION\_AU\_METADATA.OK = Succès du processus de préparation de l'insertion des métadonnées descriptives dans les unités archivistiques)
  - KO : la préparation de l'insertion de métadonnées descriptives dans les unités archivistiques n'a pas pu s'effectuer (STEP\_PRESERVATION\_PREPARATION\_INSERTION\_AU\_METADATA.KO = Échec du processus de préparation de l'insertion des métadonnées descriptives dans les unités archivistiques)
  - FATAL : une erreur technique est survenue lors de la préparation de l'insertion de métadonnées descriptives dans les unités archivistiques (STEP\_PRESERVATION\_PREPARATION\_INSERTION\_AU\_METADATA.FATAL = Erreur technique lors du processus de préparation de l'insertion des métadonnées descriptives dans les unités archivistiques)

### 15.4.Processus d'insertion de métadonnées descriptives dans les unités archivistiques (STP\_PRESERVATION\_INSERTION\_AU\_METADATA)

#### Insertion de métadonnées descriptives PRESERVATION\_INSERTION\_AU\_METADATA (PreservationInsertionAuMetadata.java)

- **Règle** : tâche consistant à indexer les métadonnées descriptives dans les unités archivistiques

- **Type** : bloquant
- **Statuts** :
  - OK : l'indexation des métadonnées descriptives dans les unités archivistiques a bien été effectuée (PRESERVATION\_INSERTION\_AU\_METADATA.OK = Succès de l'indexation des métadonnées des unités archivistiques)
  - KO : l'indexation de métadonnées descriptives dans les unités archivistiques n'a pas pu s'effectuer (PRESERVATION\_INSERTION\_AU\_METADATA.KO = Échec de l'insertion des métadonnées des unités archivistiques)
  - FATAL : une erreur technique est survenue lors de l'indexation de métadonnées descriptives dans les unités archivistiques (PRESERVATION\_INSERTION\_AU\_METADATA.FATAL = Erreur technique lors de l'indexation des métadonnées des unités archivistiques)

## Enregistrement des métadonnées des unités archivistiques dans l'offre de stockage

### UNIT\_METADATA\_STORAGE (StoreMetaDataUnitActionPlugin.java)

- **Règle** : tâche consistant à enregistrer les métadonnées descriptives des unités archivistiques dans les offres de stockage
- **Type** : bloquant
- **Statuts** :
  - OK : l'enregistrement des métadonnées des unités archivistiques dans les offres de stockage a bien été effectuée (UNIT\_METADATA\_STORAGE.OK = Succès de l'insertion des métadonnées dans les unités archivistiques)
  - KO : l'enregistrement des métadonnées des unités archivistiques dans les offres de stockage n'a pas pu s'effectuer (UNIT\_METADATA\_STORAGE.KO = Échec de l'insertion des métadonnées dans les unités archivistiques)
  - FATAL : une erreur technique est survenue lors de l'enregistrement des métadonnées des unités archivistiques dans les offres de stockage (UNIT\_METADATA\_STORAGE.FATAL = Erreur technique lors de l'enregistrement des métadonnées des unités archivistiques)

## 15.5.Processus d'alimentation du registre des fonds (STP\_ACCESSION\_REGISTRATION)

### Mise à jour du registre des fonds PRESERVATION\_ACCESSION\_REGISTRATION (PreservationAccessionRegistrationHandler.java)

- **Règle** : tâche consistant à mettre à jour le registre des fonds suite à la réalisation d'actions de préservation
- **Type** : bloquant
- **Statuts** :
  - OK : la mise à jour du registre des fonds a bien été effectuée (PRESERVATION\_ACCESSION\_REGISTRATION.OK = Succès de la mise à jour du registre des fonds)
  - KO : la mise à jour du registre des fonds n'a pas pu s'effectuer (PRESERVATION\_ACCESSION\_REGISTRATION.KO = Échec de la mise à jour du registre des fonds)
  - FATAL : une erreur technique est survenue lors de la mise à jour du registre des fonds

(PRESERVATION\_ACCESSION\_REGISTRATION.FATAL = Erreur technique lors de la mise à jour du registre des fonds)

## 15.6. Processus de création du rapport de préservation (STP\_PRESERVATION\_FINALIZATION)

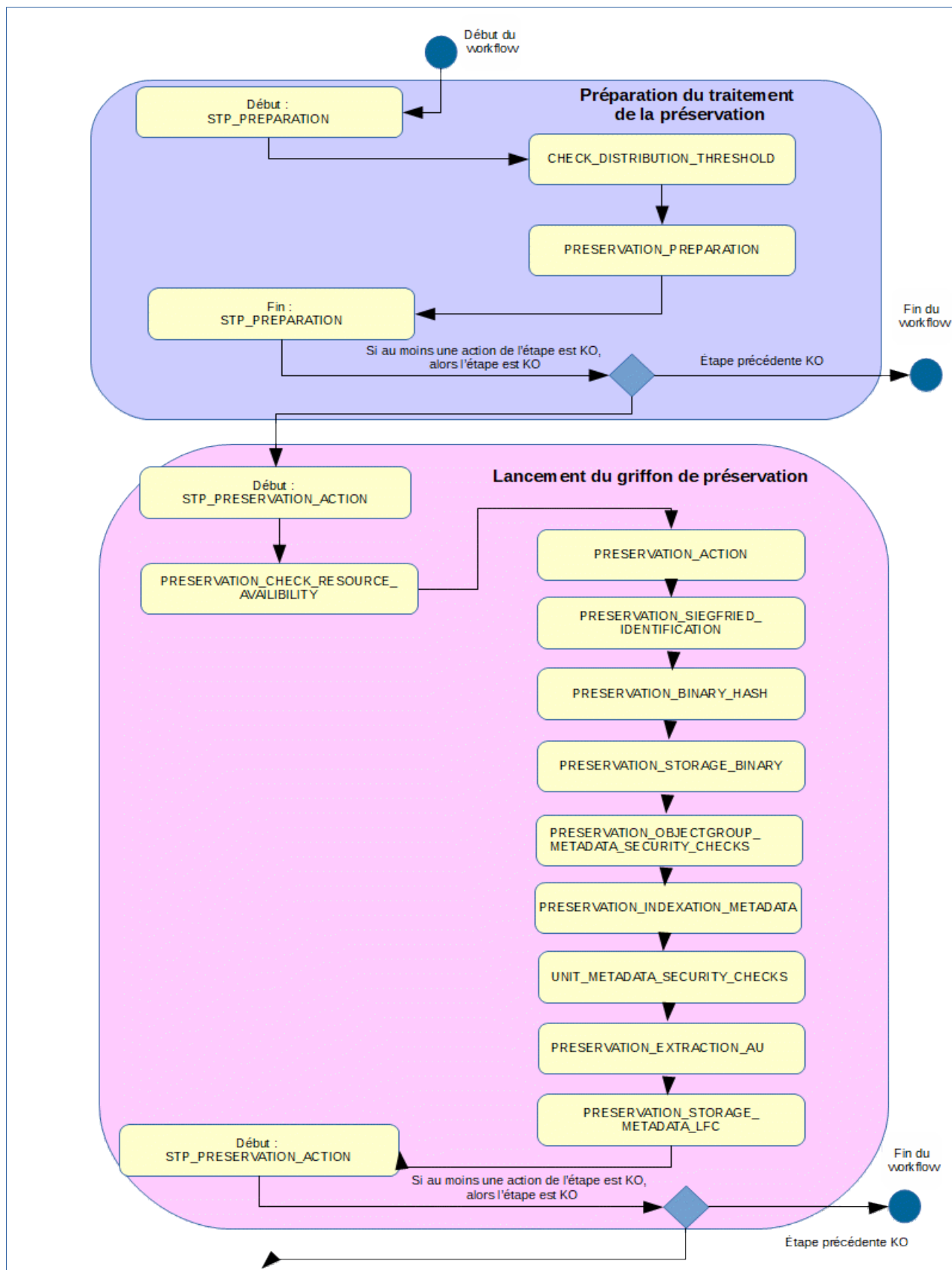
### Création du rapport de préservation PRESERVATION\_FINALIZATION (PreservationFinalizationPlugin.java)

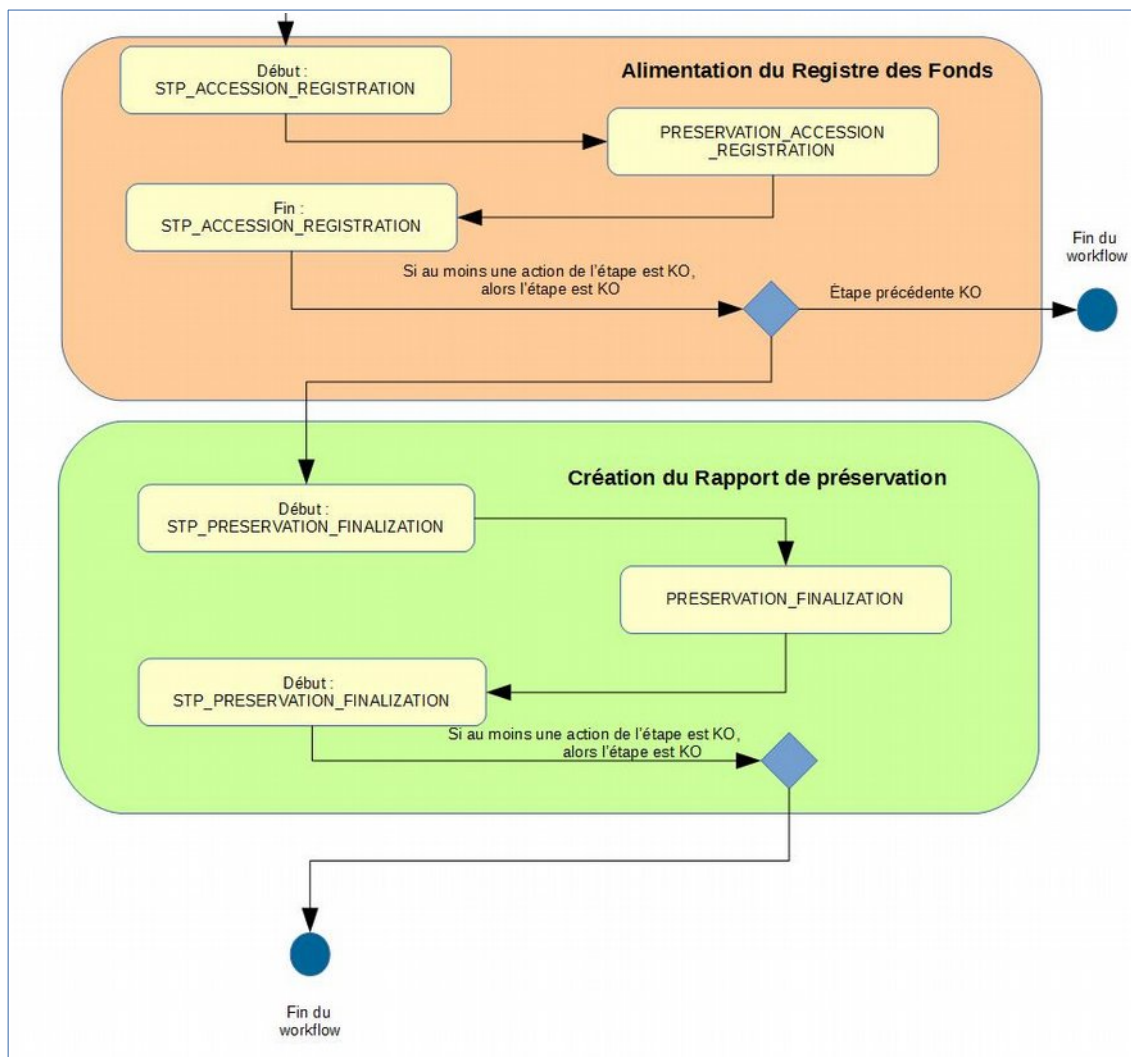
- **Règle** : tâche consistant à finaliser la préservation
- **Type** : bloquant
- **Statuts** :
  - OK : la finalisation de la préservation a bien été effectuée (PRESERVATION\_FINALIZATION.OK = Succès de la finalisation du traitement de préservation)
  - KO : la finalisation de la préservation n'a pas pu s'effectuer (PRESERVATION\_FINALIZATION.KO = Échec de la finalisation du traitement de préservation)
  - FATAL : une erreur technique est survenue lors de la finalisation de la préservation (PRESERVATION\_FINALIZATION.FATAL = Erreur technique lors de la finalisation du traitement de préservation)



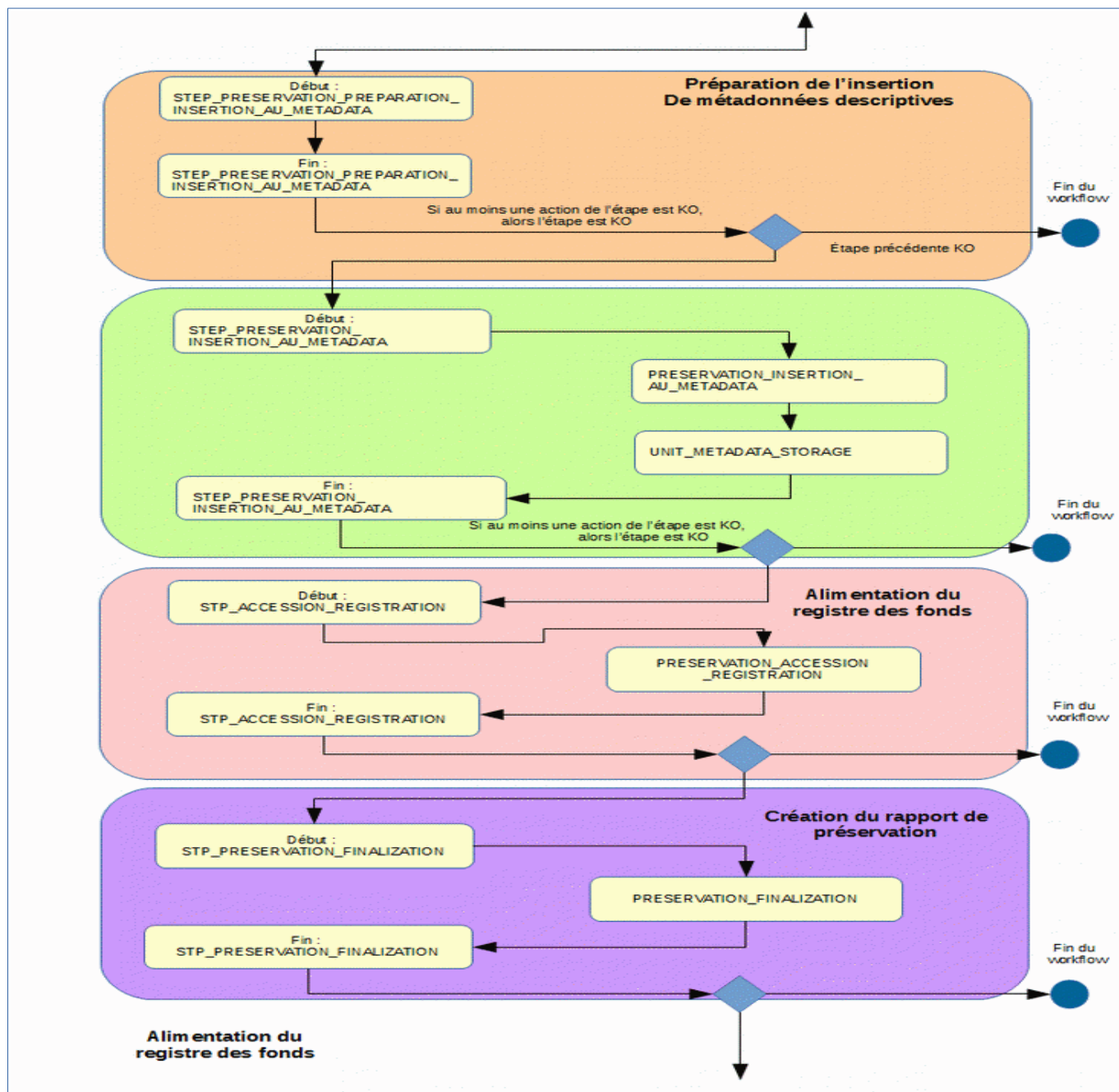
### 15.7. Structure du workflow de préservation

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









## **15.8. Rapport de préservation**

Le rapport de préservation est un fichier JSONL généré par la solution logicielle Vitam lorsqu'une opération de préservation se termine. Cette section décrit la manière dont ce rapport est structuré.

### **15.8.1. Exemple de fichier JSONL : rapport de préservation OK**





élimination...) aura un « reportType » associé ;

- « vitamResults » : indique le nombre total d'objets traités lors de l'opération de préservation et correspondant pour chacun des trois statuts, « OK », « KO », « WARNING » ;
- « extendedInfo » : partie libre où chaque type de rapport pourra ajouter des informations qui lui est propre. Ici on y trouve :
  - « nbUnits » : nombre total d'unités archivistiques ayant fait l'objet de l'opération de préservation,
  - « nbObjectGroups » : nombre total de groupes d'objets techniques ayant fait l'objet de l'opération de préservation,
  - « nbStatusKo » : nombre total d'objets techniques en erreur,
  - « nbStatusWarning » : nombre total d'objets techniques en avertissement,
  - « nbActionAnalyses » : nombre total d'objets techniques ayant fait l'objet d'une opération d'analyse,
  - « nbActionGenerates » : nombre total d'objets techniques ayant fait l'objet d'une opération de génération d'objet binaire,
  - « nbActionIdentifys » : nombre total d'objets techniques ayant fait l'objet d'une opération de réidentification de format,
  - « nbActionExtracts » : nombre total d'objets techniques ayant fait l'objet d'une opération d'extraction de métadonnées techniques,
  - « analyseResults » : résultat d'une opération d'analyse, pouvant contenir les valeurs suivantes :
    - « VALID\_ALL » : nombre d'objets techniques dont le format est valide et non corrompu,
    - « NOT\_VALID » : nombre d'objets techniques dont le format est identifié, mais n'est pas valide ;
    - « WRONG\_FORMAT » : nombre d'objets techniques dont le format est faux et corrompu,

La partie « Context » correspond au contexte dans lequel l'opération de préservation a été effectuée et destiné à créer la distribution sur chaque unité archivistique. On y retrouve les éléments suivants :

- « dslQuery » : requête DSL utilisée pour l'opération de préservation ;
- « preservationScenarioModel » : contenu du scénario de préservation utilisé pour l'opération de préservation ;
- « griffinModel » : contenu du (des) griffon(s) utilisé(s) pour l'opération de préservation.

La partie « ReportDetail » contient les détails de l'opération de préservation effectuée sur chacun des objets techniques, que ces derniers soient aux statuts « OK », « KO » ou « WARNING ».

- « \_id » : identifiant de l'objet technique ayant fait l'objet de l'opération de préservation ;
- « preservationReportId » : identifiant du rapport de préservation ;
- « processId » : identifiant de l'opération de préservation, correspondant à « EvIdProc » ;
- « creationDateTime » : date de l'opération ;
- « unitId » : identifiant de l'unité archivistique dont dépend l'objet technique ayant fait l'objet de l'opération de préservation ;
- « objectGroupId » : identifiant du groupe d'objets techniques auquel appartient l'objet technique ayant fait l'objet de l'opération de préservation ;
- « status » : résultat de l'opération de préservation pour l'objet technique concerné, pouvant



correspondre aux valeurs suivantes : « OK », « KO », « WARNING » ;

- « inputObjectId » : identifiant de l'objet technique faisant l'objet de l'opération de préservation ;
- « outputObjectId » : dans le cas d'une génération d'objets binaires, identifiant du nouvel objet technique généré ;
- « griffinId » : identifiant du scénario de préservation utilisé pour cet objet technique en particulier ;
- « preservationScenarioId » : identifiant du scénario de préservation utilisé pour l'opération.

## 16. SUPPRESSION DES VERSIONS DES GOT

Cette section décrit le workflow de suppression des versions d'objets techniques .

Lors des opérations de préservation, des versions supplémentaires d'usages / version peuvent être créées. Si des usages / version ainsi générés sont illisibles ou non conformes et qu'une nouvelle opération de préservation est envisageable pour créer une nouvelle version lisible et/ou conforme, il convient de pouvoir supprimer l'usage / version défectueux afin de ne pas présenter un objet en défaut.

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

### Workflow de suppression des versions de GOT

#### 16.1. Processus de préparation de la suppression des versions des GOT (STP\_DELETE\_GOT\_VERSIONS\_PREPARATION)

**Règle :** tâche consistant à préparer la suppression des versions des GOT.

**Type :** bloquant

**Statuts :**

- OK : la préparation de la suppression des versions de GOT a bien été effectuée (STP\_DELETE\_GOT\_VERSIONS\_PREPARATION.OK = Succès du processus de préparation de la suppression des versions des groupes d'objets)
- KO : la préparation de la suppression des versions de GOT n'a pas bien été effectuée (STP\_DELETE\_GOT\_VERSIONS\_PREPARATION.KO = Échec du processus de la préparation de la suppression des versions des groupes d'objets)
- FATAL : une erreur technique est survenue lors de la la préparation de la suppression des versions de GOT (STP\_DELETE\_GOT\_VERSIONS\_PREPARATION.FATAL=Erreur technique lors du processus de la préparation de la suppression des versions des groupes d'objets)
- WARNING : (STP\_DELETE\_GOT\_VERSIONS\_PREPARATION.WARNING=Avertissement lors du processus de préparation de la suppression des versions des groupes d'objets)

#### 16.1.1.Vérification des seuils de la suppression des versions des GOT (CHECK\_DISTRIBUTION\_THRESHOLD :CheckDistributionThreshold.java)

- **Règle :** tâche consistant à vérifier les seuils de la suppression des versions des GOT.
- **Type :** bloquant
- **Statuts :**
  - OK : la vérification des seuils de la suppression des versions des GOT a bien été effectuée (CHECK\_DISTRIBUTION\_THRESHOLD.OK= Succès de la vérification des seuils de limitation de traitement des unités archivistiques)
  - KO : la vérification des seuils de la suppression des versions des GOT n'a pas bien été effectuée (CHECK\_DISTRIBUTION\_THRESHOLD.KO= Échec de la vérification des seuils de limitation de traitement des unités archivistiques)
  - FATAL : une erreur technique est survenue lors de la vérification des seuils limitation de traitement des unités archivistiques (CHECK\_DISTRIBUTION\_THRESHOLD.FATAL=

Erreur technique lors de la vérification des seuils de limitation de traitement des unités archivistiques)

### 16.1.2. Préparation de la suppression des versions des GOT

**(DELETE\_GOT\_VERSIONS\_PREPARATION :DeleteGotVersionsPreparationPlugin.java)**

- **Règle** : tâche consistant à établir la préparation de la suppression des versions des objets et préparer l'output "distributionFileOG"
- **Type** : bloquant
- **Statuts** :
  - OK : la préparation de la suppression des versions des objets a bien été effectuée (DELETE\_GOT\_VERSIONS\_PREPARATION.OK = Succès de la préparation de la suppression des versions des groupes d'objets )
  - KO : la préparation de la suppression des version des objets n'a pas été effectuée (DELETE\_GOT\_VERSIONS\_PREPARATION.KO = Échec de la préparation de la suppression des versions des groupes d'objets)
  - FATAL : une erreur technique est survenue lors de la préparation de la suppression des versions des objets (DELETE\_GOT\_VERSIONS\_PREPARATION.FATAL = Erreur technique lors de la préparation de la suppression des versions des groupes d'objets)

## 16.2. Processus de suppression des versions des GOT

**(STP\_DELETE\_GOT\_VERSIONS\_ACTION ) (Distribution sur LIST GUID/distributionFileOG.jsonl**

### 16.2.1. Action de suppression (DELETE\_GOT\_VERSIONS\_ACTION : DeleteGotVersionsActionPlugin.java)

- **Règle** : tâche consistant à l'action de la suppression des versions des GOT.
- **Type** : bloquant
- **Statuts** :
  - OK : l'action de la suppression des versions de GOT a bien été effectuée (STP\_DELETE\_GOT\_VERSIONS\_ACTION.OK = Succès du processus de la suppression des versions des groupes d'objets)
  - KO : l'action de la suppression des versions de GOT n'a pas bien été effectuée (STP\_DELETE\_GOT\_VERSIONS\_ACTION.KO = Échec du processus de la suppression des versions des groupes d'objets)
  - FATAL : une erreur technique est survenue lors de l'action de la suppression des versions de GOT (STP\_DELETE\_GOT\_VERSIONS\_ACTION.FATAL=Erreur technique lors de la suppression des versions des groupes d'objets)
  - WARNING : (STP\_DELETE\_GOT\_VERSIONS\_ACTION.WARNING=Avertissement lors du processus de l'action de la suppression des versions des groupes d'objets)

### 16.2.2. Stockage des groupes d'objets dans workspace

**(STORE\_METADATA\_AND\_LFC\_WORKSPACE : DeleteGotVersionsStoreMetadataAndLfcPlugin.java)**

**Règle** : tâche consistant au stockage des groupes d'objets dans workspace.

**Type :** bloquant

**Statuts :**

- OK : l'action de stockage des groupes d'objets a bien été effectuée (STORE\_METADATA\_AND\_LFC\_WORKSPACE.OK = Succès du stockage des groupes d'objets dans workspace)
- KO : l'action de stockage des groupes d'objets n'a pas bien été effectuée (STORE\_METADATA\_AND\_LFC\_WORKSPACE.KO = Échec du stockage des groupes d'objets dans workspace)
- FATAL : une erreur technique est survenue lors de l'action de stockage des groupes d'objets (STORE\_METADATA\_AND\_LFC\_WORKSPACE.FATAL=Erreur technique lors du stockage des groupes d'objets dans workspace )
- WARNING : Avertissement lors de l'action de stockage des groupes d'objets (STORE\_METADATA\_AND\_LFC\_WORKSPACE.WARNING= Avertissement lors du stockage des groupes d'objets dans workspace)

### 16.2.3. Suppression des groupes d'objets depuis les offres (DELETE\_GOT\_VERSIONS\_STORAGE : DeleteGotVersionsStoragePlugin.java)

**Règle :** tâche consistant à la suppression des groupes d'objets depuis les offres.

**Type :** bloquant

**Statuts :**

- OK : la suppression des groupes d'objets depuis les offres a bien été effectuée (DELETE\_GOT\_VERSIONS\_STORAGE.OK = Succès du stockage des groupes d'objets dans workspace)
- KO : la suppression des groupes d'objets depuis les offres n'a pas bien été effectuée (DELETE\_GOT\_VERSIONS\_STORAGE.KO = Échec du stockage des groupes d'objets dans workspace)
- FATAL : une erreur technique est survenue lors de la suppression des groupes d'objets depuis les offres (DELETE\_GOT\_VERSIONS\_STORAGE.FATAL=Erreur technique lors du stockage des groupes d'objets dans workspace)
- WARNING : Avertissement lors de la suppression des groupes d'objets depuis les offres (DELETE\_GOT\_VERSIONS\_STORAGE.WARNING= Avertissement lors du stockage des groupes d'objets dans workspace)

### 16.3. Processus de mise à jour du registre des fonds (STP\_DELETE\_GOT\_VERSIONS\_ACCESSION\_REGISTER\_UPDATE)

**Règle :** tâche consistant à l'alimentation du Registre des Fonds.

**Type :** bloquant

**Statuts :**

- OK : l'alimentation du Registre des Fonds a bien été effectuée (STP\_DELETE\_GOT\_VERSIONS\_ACCESSION\_REGISTER\_UPDATE.OK = Succès du processus l'alimentation du Registre des Fonds)
- KO : l'alimentation du Registre des Fonds n'a pas bien été effectuée (STP\_DELETE\_GOT\_VERSIONS\_ACCESSION\_REGISTER\_UPDATE.KO = Échec du processus de l'alimentation du Registre des Fonds)

- FATAL : une erreur technique est survenue lors de l'alimentation du Registre des Fonds (STP\_DELETE\_GOT\_VERSIONS\_ACCESSION\_REGISTER\_UPDATE .FATAL=Erreur technique lors du processus de l'alimentation du Registre des Fonds)
- WARNING : Avertissement lors de l'alimentation du Registre des Fonds (STP\_DELETE\_GOT\_VERSIONS\_ACCESSION\_REGISTER\_UPDATE.WARNING=Avertissement lors du processus de l'alimentation du Registre des Fonds)

### **Alimentation du Registre des Fonds (DELETE\_GOT\_VERSIONS\_ACCESSION\_REGISTER\_UPDATE : DeleteGotVersionsAccessionRegisterUpdatePlugin.java)**

**Règle :** tâche consistant à l'alimentation du Registre des Fonds.

**Type :** bloquant

**Statuts :**

- OK : l'alimentation du Registre des Fonds a bien été effectuée (DELETE\_GOT\_VERSIONS\_ACCESSION\_REGISTER\_UPDATE.OK = Succès de l'alimentation du Registre des Fonds)
- KO : l'alimentation du Registre des Fonds n'a pas bien été effectuée (DELETE\_GOT\_VERSIONS\_ACCESSION\_REGISTER\_UPDATE.KO = Échec de l'alimentation du Registre des Fonds)
- FATAL : une erreur technique est survenue lors de l'alimentation du Registre des Fonds (DELETE\_GOT\_VERSIONS\_ACCESSION\_REGISTER\_UPDATE .FATAL=Erreur technique lors du processus de l'alimentation du Registre des Fonds)  
WARNING : Avertissement lors de l'alimentation du Registre des Fonds (DELETE\_GOT\_VERSIONS\_ACCESSION\_REGISTER\_UPDATE.WARNING=Avertissement lors de l'alimentation du Registre des Fonds)

### **16.4. Processus de finalisation de la suppression des versions des groupes d'objets (STP\_DELETE\_GOT\_VERSIONS\_FINALIZATION)**

**Règle :** tâche consistant à la finalisation de la suppression des versions des groupes d'objets

**Type :** bloquant

**Statuts :**

- OK : Le process de la finalisation de la suppression des versions des groupes d'objets a bien été effectuée (STP\_DELETE\_GOT\_VERSIONS\_FINALIZATION.OK = Succès du processus de finalisation de la suppression des versions des groupes d'objets)
- KO : Le process de la finalisation de la suppression des versions des groupes d'objets n'a pas bien été effectuée (STP\_DELETE\_GOT\_VERSIONS\_FINALIZATION .KO = Échec du processus de finalisation de la suppression des versions des groupes d'objets)
- FATAL : une erreur technique est survenue lors de process de la finalisation de la suppression des versions des groupes d'objets (STP\_DELETE\_GOT\_VERSIONS\_FINALIZATION .FATAL=Erreur technique lors du processus de la finalisation de la suppression des versions des groupes d'objets)
- WARNING : Avertissement lors de l'alimentation du Registre des Fonds (STP\_DELETE\_GOT\_VERSIONS\_FINALIZATION .WARNING= Avertissement lors du processus de finalisation de la suppression des versions des groupes d'objets)

## **Finalisation de la suppression des versions des groupes d'objets (DELETE\_GOT\_VERSIONS\_FINALIZATION : handlers.DeleteGotVersionsFinalizationPlugin.java)**

**Règle :** tâche consistant à la finalisation de la suppression des versions des groupes d'objets

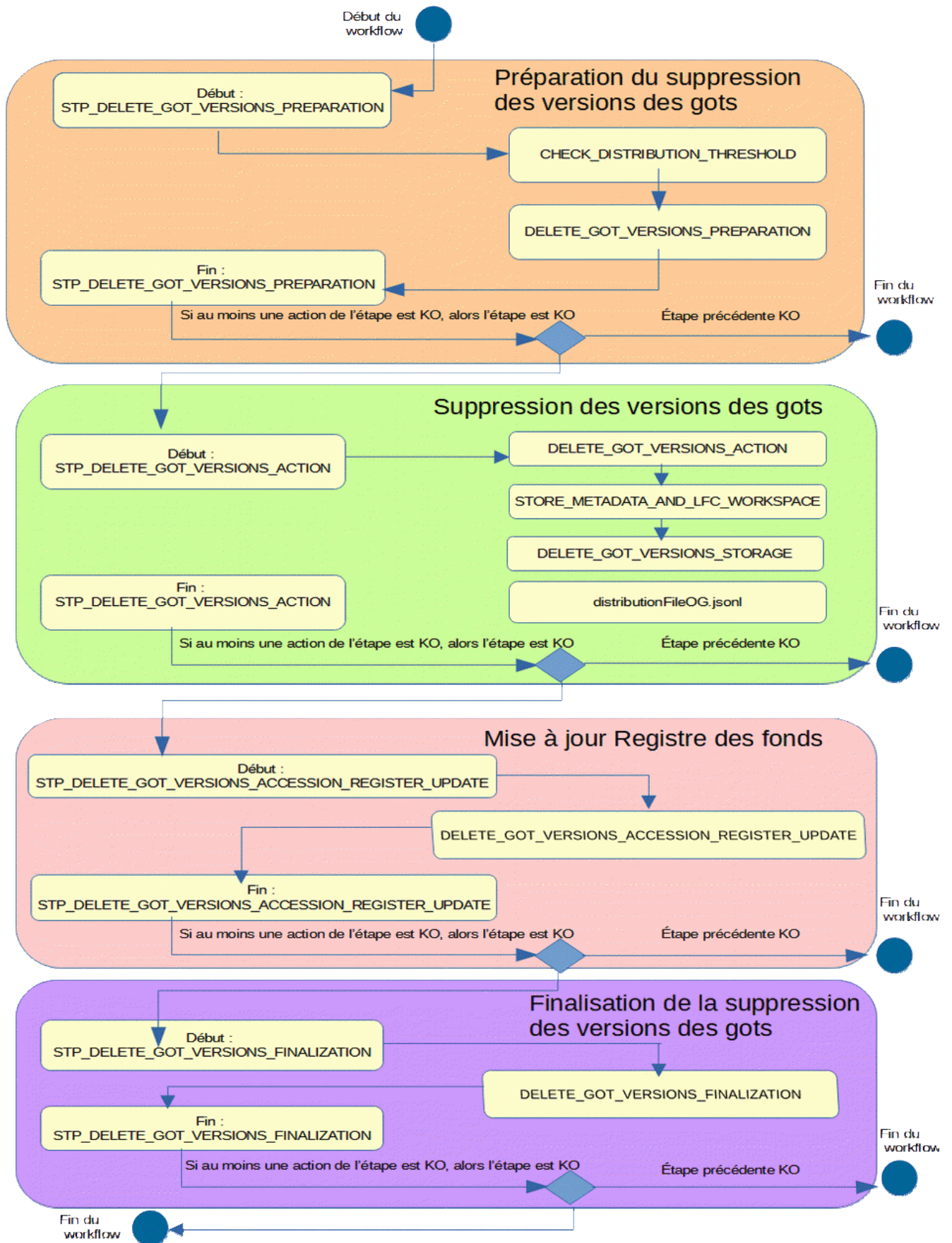
**Type :** bloquant

**Statuts :**

- OK : La finalisation de la suppression des versions des groupes d'objets a bien été effectuée (DELETE\_GOT\_VERSIONS\_FINALIZATION.OK = Succès de la finalisation de la suppression des versions des groupes d'objets)
- KO : La finalisation de la suppression des versions des groupes d'objets n'a pas bien été effectuée (DELETE\_GOT\_VERSIONS\_FINALIZATION.KO = Échec de la finalisation de la suppression des versions des groupes d'objets)
- FATAL : une erreur technique est survenue lors de finalisation de la suppression des versions des groupes d'objets (DELETE\_GOT\_VERSIONS\_FINALIZATION.FATAL=Erreur technique lors de la finalisation de la suppression des versions des groupes d'objets)
- WARNING : Avertissement lors de l'alimentation du Registre des Fonds (DELETE\_GOT\_VERSIONS\_FINALIZATION.WARNING= Avertissement lors de la finalisation de la suppression des versions des groupes d'objets)

### **16.5. Structure du Workflow de suppression des versions des GOTS**

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





## 16.6. Rapport de suppression des versions des groupes d'objets

Le rapport de suppression des versions des groupes d'objets (got) est un fichier JSON généré par la solution logicielle Vitam lorsqu'une opération suppression des versions des groupes d'objets se termine.

Cette section décrit la manière dont ce rapport est structuré.

### 16.6.1. Exemple de JSON : rapport de suppression des versions des got

<pre>{   "tenant":0,   "evId":"aeaaaaaachnqhzoaaomal2qd6nlsaaaaaq",   "evType":"DELETE_GOT_VERSIONS_ACCESSION_REGISTER_UPDATE",   "outcome":"OK",   "outDetail":"DELETE_GOT_VERSIONS_ACCESSION_REGISTER_UPDATE.OK",   "outMsg":"Succ\u00E8s de l'alimentation du Registre des Fonds Detail= OK:1",   "rightsStatementIdentifier":{"AccessContract":"ContratTNR"},   "evDetData":{} }</pre>	<b>Operation Summary</b>
<pre>{   "evStartDateTime":"2021-07-07T12:41:31.387",   "evEndDateTime":"2021-07-07T12:41:33.265",   "reportType":"DELETE_GOT_VERSIONS",   "vitamResults":{"OK":1,"KO":0,"WARNING":0,"total":1},   "extendedInfo":{} }</pre>	<b>Report Summary</b>
<pre>{    "UsageName":"BinaryMaster",   "SpecificVersions":[2],   "DslQuery":{"\$roots":[],   "\$query":[{"\$in":{"#id":["aeaqaaaaahnqhzoab6qoal2qd3fowqaaaaq"]},   "\$depth":1000}],   "\$filter":{},   "\$projection":{},   "\$facets":[] }</pre>	<b>Contexte</b>
<pre>{   "_id":"aeaaaaaaaahl3ntaanlyal2qd6npxyaaaaq",   "id":"aebaaaaaahnqhzoab6qoal2qd3fouyaaaba",   "processId":"aeaaaaaachnqhzoaaomal2qd6nlsaaaaaq",   "_tenant":0,   "creationDateTime":"2021-07-07T12:41:31.864",   "detailType":"DELETE_GOT_VERSIONS",   "objectGroupGlobal":{     {"status":"OK",     "deletedVersions":[       {"id":"aeaaaaaaaahnqhzoab6qoal2qd4rl2aaaaaq","DataObjectVersion":"BinaryMaster_2","Size":12843,"strategyId":"default","opi":"aeaaaaaachnqhzoaa73wal2qd3e6cqaaaaq","opc":"aeaaaaaachnqhzoaaomal2qd4qtxyaaaaq"}     ]   },   "objectGroupId":"aebaaaaaahnqhzoab6qoal2qd3fouyaaaba",   "unitIds":["aeaqaaaaahnqhzoab6qoal2qd3fowqaaaaq"] }</pre>	<b>Report Detail</b>

### 16.6.2. Détail du rapport

Le rapport de suppression des versions des groupes d'objets (GOT) est composé de quatre parties :

La partie « OperationSummary », c'est-à-dire le bloc à la racine du rapport et correspondant au résumé de



L'opération (sans indentation) est composée des champs suivants :

- « tenant » : tenant sur lequel l'opération a été lancée ;
- « evId » : identifiant de l'événement (l'opération) ;
- « evType » : code du type de l'opération (avant la dernière step);
- « outcome »: statut de l'événement, pouvant correspondre aux valeurs « OK », « KO » ou « WARNING » ;
- « outDetail » : code correspondant au résultat de l'événement ;
- « outMsg » : détail du résultat de l'événement ;
- « rightsStatementIdentifiant »:identifiant des données référentielles en vertu desquelles l'opération peut s'exécuter ;
- « evDetData » : détails des données l'événement.

La partie « ReportSummary », c'est-à-dire le bloc situé sous la racine du rapport et correspondant au résumé du rapport est composée des champs suivants :

- « evStartDateTime » : date du début de l'opération (evDateTime de l'event master de l'opération dans le journal des opérations) ;
- « evEndDateTime » : date de fin d'opération (dernier evDateTime dans l'opération dans le journal des opérations) ;
- « reportType »: correspond au modèle du rapport. Chaque opération de masse (audit, élimination...) aura un « reportType » associé. Ici pour le rapport de suppression des versions des got est égale au "DELETE\_GOT\_VERSIONS"
- « vitamResults » : indique le nombre total d'objets traités lors de l'opération de suppression des versions des got et correspondant pour chacun des trois statuts, « OK », « KO » et « WARNING » et le « total » ;
- « extendedInfo » : partie libre où chaque type de rapport pourra ajouter des informations qui lui est propre. Ici pas d'information complémentaire ajoutée

La partie « Context » correspond à la requête DSL d'entrée utilisé pour supprimer des versions des gots:

- « UsageName » : Nom d'usage de la version de got à supprimer. Prend comme valeur "BinaryMaster", "Dissemination", "Thumbnail", "TextContent" et "PhysicalMaster".
- « SpecificVersion » : un tableau des versions qu'on souhaite les supprimer
- « dslQuery » : requête DSL utilisée pour l'opération de suppression des version des gots ;

La partie « ReportDetail » contient les détails de l'opération de suppression des version des gots :

- « \_id » : id de l'opération ;
- « id » : id de got ;
- « processId » : id de rapport ;
- « \_tenant » : tenant sur lequel l'opération a été lancée ;
- « creationDateTime » : le temps et la date de création de rapport ;
- « detailType » : type de rapport="DELETE\_GOT\_VERSIONS"
- « objectGroupGlobal » : tableau pour chaque objet (version) à supprimer
- « status » : résultat de suppression
- « deletedVersions » : tableau des versions à supprimer avec le détail de chaque version : id (id de GOT), DataObjectVersion, Size, strategyId, opi (ingest original ou préservation qui a crée le GOT) et opc
- « objectGroupId » : Id de l'objet groupe
- « unitIds » : une liste des unités archivistiques associées à la version de GOT