



Description des workflows et des opérations

Partie 2 : gestion du cycle de vie des archives

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	03/08/2022
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	Mise à jour pour tenir compte des fonctionnalités mises en œuvre pendant la <i>Release 10</i> : <ul style="list-style-type: none"> chapitre 5, section III (« Workflow d'administration d'un référentiel des formats ») : précisions sur les cas WARNING (section III.A.1) ; précisions sur le contenu des champs suivants : EvDetDateTime, Warnings et UpdatedPUIDs. chapitre 5, section XII (« Workflow d'administration d'un référentiel des griffons ») : corrections diverses : le champ Identifier fonctionne en mode VITAM esclave, le champ ExecutableVersion n'est pas un entier. chapitre 5, section XIII (« Workflow d'administration d'un référentiel des scénarios de préservation ») : ajout de l'étape « Processus de génération du rapport d'import du référentiel des scénarios de préservation SCENARIO_REPORT » et d'une partie sur la « Structure du rapport d'administration du référentiel des scénarios de préservation ».
2.2	15/04/2019	MAF	Mise à jour et corrections diverses
3.0	24/04/2019	MRE	Finalisation pour publication Release 10
3.1	14/08/2019	MVI	Mise à jour pour tenir compte des fonctionnalités mises en œuvre pendant la <i>Release 11</i> : <ul style="list-style-type: none"> chapitre 2, section V (« Rapport de l'audit de cohérence ») : mise à jour du rapport ; chapitre 11, section II (« Rapport de

			préservation ») : création de la section.
3.2	02/09/2019	JPP	Mise à jour pour tenir des fonctionnalités mises en œuvre pendant la Release 11 : <ul style="list-style-type: none"> • chapitre 8, section III et IV.
4.0	09/09/2019	MAF	Finalisation pour publication Release 11
4.1	24/09/2019	GFO	Modification multi-stratégies
4.2	23/10/2019	GFO	Mise à jour des Audits
4.3	19/11/2019	MAF	Mise à jour pour tenir des fonctionnalités mises en œuvre pendant la Release 12
5.0	29/11/2019	AGR	Finalisation pour publication Release 12
6.0	20/03/2020	AGR	Séparation du document en 2 parties et finalisation pour publication Release 13
6.1	15/04/2020	MVI	Mise à jour pour tenir compte des fonctionnalités mises en œuvre pendant la <i>Release 14</i> : <ul style="list-style-type: none"> • chapitre 11 (« Préservation ») : ajout des étapes liées à l'extraction de métadonnées descriptives et de contrôle de sécurité sur les unités archivistiques et les objets techniques.
7.0	06/07/2020	AGR	Finalisation pour publication Release 14
7.1	26/10/2020	MVI	Relecture
8.0	26/10/2020	AGR	Finalisation pour publication Release 15
8.1	11/02/2021	MVI	Mise à jour pour tenir compte des fonctionnalités mises en œuvre pendant la <i>Release 16</i> : <ul style="list-style-type: none"> • chapitre 13 (« Mise à jour unitaire de masse ») : ajout des étapes liées à l'opération de mise à jour en masse de métadonnées descriptives, ces mises à jour différant d'une unité archivistique à l'autre. • Chapitre 14 (« Restauration des métadonnées essentielles ») : ajout des étapes liées à l'opération de restauration de métadonnées essentielles.
9.0	15/03/2021	AGR	Finalisation pour publication Release 16
9.1	09/08/2021	MVI	Mise à jour pour tenir des fonctionnalités mises en œuvre pendant la Version 5.RC : <ul style="list-style-type: none"> • partitionnement de la documentation avec une nouvelle répartition : <ul style="list-style-type: none"> ◦ partie 1 : référentiels ◦ partie 2 : gestion du cycle de vie des archives ◦ partie 3 : préservation et intégrité <p>Le présent document couvre les opérations liées à la gestion du cycle de vie des archives.</p>
10.0	27/09/2021	AGR	Finalisation pour publication Version 5.RC
10.1	19/01/2022	MVI	Mise à jour pour tenir des fonctionnalités mises en œuvre pendant la Version 5 : <ul style="list-style-type: none"> • chapitre 8 (« Export d'un DIP ») : ajout d'une vérification de la disponibilité des ressources ; • chapitre 11 (« Transfert ») : ajout d'une vérification de la disponibilité des ressources.
11.0	04/03/2022	AGR	Finalisation pour publication Version 5
11.1	03/08/2022	MVI	Mise à jour pour tenir des fonctionnalités mises en œuvre pendant la Version 6.RC : <ul style="list-style-type: none"> • chapitre 8 (« Export d'un DIP ») : ajout d'une erreur à l'étape de création du bordereau pour cause d'incompatibilité de version du SEDA ;

			<ul style="list-style-type: none">• chapitre 11 (« Transfert ») : ajout d'une erreur à l'étape de création du bordereau pour cause d'incompatibilité de version du SEDA.
11.0	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.....	11
Objectif du document.....	11
4. Ingest.....	12
4.1. Processus des contrôles préalables à l'entrée (STP_SANITY_CHECK_SIP).....	12
4.1.1. Contrôle sanitaire du SIP SANITY_CHECK_SIP (IngestExternalImpl.java).....	12
4.1.2. Contrôle du format du conteneur du SIP CHECK_CONTAINER (IngestExternalImpl.java).....	12
4.1.3. Contrôle du nom du bordereau de transfert MANIFEST_FILE_NAME_CHECK (IngestExternalImpl.java).....	12
4.1.4. Vérification de l'intégrité du bordereau de transfert MANIFEST_DIGEST_CHECK (IngestExternalImpl.java).....	13
4.2. Processus de réception du SIP dans Vitam STP_UPLOAD_SIP (IngestInternalResource.java)....	13
4.3. Processus de contrôle du SIP (STP_INGEST_CONTROL_SIP).....	13
4.3.1. Vérification globale du CHECK_SEDA (CheckSedaActionHandler.java).....	13
4.3.2. Vérification de l'en-tête du bordereau de transfert CHECK_HEADER (CheckHeaderActionHandler.java).....	14
4.3.3. Vérification de la présence et contrôle des services agents CHECK_HEADER.CHECK_AGENT....	15
4.3.4. Vérification de la présence et contrôle du contrat d'entrée CHECK_HEADER.CHECK_CONTRACT_INGEST.....	15
4.3.5. Vérification de la relation entre le contrat d'entrée et le profil d'archivage CHECK_HEADER.CHECK_IC_AP_RELATION.....	16
4.3.6. Vérification de la conformité du bordereau de transfert par le profil d'archivage CHECK_HEADER.CHECK_ARCHIVEPROFILE.....	17
4.3.7. Préparation des informations de stockage PREPARE_STORAGE_INFO (PrepareStorageInfoActionHandler.java).....	17
4.3.8. Vérification des objets et groupes d'objets CHECK_DATAOBJECTPACKAGE.....	17
4.3.9. Vérification des usages des groupes d'objets CHECK_DATAOBJECTPACKAGE.CHECK_MANIFEST_DATAOBJECT_VERSION (CheckVersionActionHandler.java).....	18
4.3.10. Vérification du nombre d'objets CHECK_DATAOBJECTPACKAGE.CHECK_MANIFEST_OBJECTNUMBER (CheckObjectsNumberActionHandler.java).....	19
4.3.11. Vérification de la cohérence du bordereau de transfert CHECK_DATAOBJECTPACKAGE.CHECK_MANIFEST (ExtractSedaActionHandler.java).....	19
4.3.12. Vérification de la cohérence entre objets, groupes d'objets et unités archivistiques CHECK_DATAOBJECTPACKAGE.CHECK_CONSISTENCY (CheckObjectUnitConsistencyActionHandler.java).....	21
4.3.13. Vérification du rattachement à un groupe d'objets ou une unité archivistique entrés sans erreur CHECK_ATTACHEMENT.....	22
(CheckAttachementActionHandler.java).....	22
4.4. Processus de contrôle et traitement des objets (STP_OG_CHECK_AND_TRANSFORME).....	22
4.4.1. Vérification de l'intégrité des objets CHECK_DIGEST (CheckConformityActionPlugin.java).....	22
4.4.2. Calcul de la taille des fichiers CHECK_OBJECT_SIZE.....	23
4.4.3. Identification des formats OG_OBJECTS_FORMAT_CHECK (FormatIdentificationActionPlugin.java).....	23
4.5. Processus de contrôle et traitement des unités archivistiques (STP_UNIT_CHECK_AND_PROCESS).....	24
4.5.1. Vérification globale de l'unité archivistique CHECK_UNIT_SCHEMA (CheckArchiveUnitSchemaActionPlugin.java).....	24
4.5.2. Vérification du profil d'unité archivistique CHECK_ARCHIVE_UNIT_PROFILE	

(CheckArchiveUnitProfileActionPlugin.java).....	25
4.5.3. Vérification du niveau de classification CHECK_CLASSIFICATION_LEVEL (CheckClassificationLevelActionPlugin.java).....	25
4.5.4. Application des règles de gestion et calcul des dates d'échéances UNITS_RULES_COMPUTE (UnitsRulesComputePlugin.java).....	26
4.6. Processus de vérification préalable à la prise en charge (STP_STORAGE_AVAILABILITY_CHECK).....	27
4.6.1. Vérification de la disponibilité de toutes les offres de stockage STORAGE_AVAILABILITY_CHECK (CheckStorageAvailabilityActionHandler.java).....	27
4.6.2. Vérification de la disponibilité de l'offre de stockage STORAGE_AVAILABILITY_CHECK.STORAGE_AVAILABILITY_CHECK (CheckStorageAvailabilityActionHandler.java).....	27
4.7. Processus d'écriture et indexation des objets et groupes d'objets (STP_OBJ_STORING).....	27
4.7.1. Écriture des objets sur l'offre de stockage OBJ_STORAGE (StoreObjectActionHandler.java).....	28
4.7.2. Indexation des métadonnées des groupes d'objets et objets OG_METADATA_INDEXATION (IndexObjectGroupActionPlugin.java).....	28
4.8. Processus d'indexation des unités archivistiques (STP_UNIT_METADATA).....	28
Indexation des métadonnées des unités archivistiques UNIT_METADATA_INDEXATION (IndexUnitActionPlugin.java).....	28
4.9. Processus d'enregistrement et écriture des métadonnées des objets et groupes d'objets (STP_OG_STORING).....	29
4.9.1. Enregistrement des journaux du cycle de vie des groupes d'objets COMMIT_LIFE_CYCLE_OBJECT_GROUP (CommitLifeCycleObjectGroupActionHandler.java).....	29
4.9.2. Écriture des métadonnées du groupe d'objets et objets sur l'offre de stockage OG_METADATA_STORAGE (StoreMetaDataSetObjectGroupActionPlugin).....	29
4.10. Processus d'enregistrement et écriture des unités archivistiques (STP_UNIT_STORING).....	29
4.10.1. Enregistrement du journal du cycle de vie des unités archivistiques COMMIT_LIFE_CYCLE_UNIT (AccessInternalModuleImpl.java).....	30
4.10.2. Écriture des métadonnées de l'unité archivistique sur l'offre de stockage UNIT_METADATA_STORAGE (AccessInternalModuleImpl.java).....	30
4.11. Processus de mise à jour des groupes d'objets (STP_UPDATE_OBJECT_GROUP).....	30
4.11.1. Mise à jour des groupes d'objets OBJECT_GROUP_UPDATE (AccessInternalModuleImpl.java).....	30
4.11.2. Enregistrement du journal du cycle de vie des groupes d'objets COMMIT_LIFE_CYCLE_OBJECT_GROUP (CommitLifeCycleObjectGroupActionHandler.java).....	31
4.11.3. Écriture des métadonnées des groupes d'objets sur l'offre de stockage OG_METADATA_STORAGE (StoreMetaDataSetObjectGroupActionPlugin).....	31
4.12. Processus d'alimentation du registre des fonds (STP_ACCESSION_REGISTRATION).....	31
Alimentation du registre des fonds ACCESSION_REGISTRATION.....	31
4.13. Processus de finalisation de l'entrée (STP_INGEST_FINALISATION).....	32
4.13.1. Notification de la fin de l'opération d'entrée ATR_NOTIFICATION (TransferNotificationActionHandler.java).....	32
4.13.2. Mise en cohérence des journaux du cycle de vie ROLL_BACK (RollBackActionHandler.java).....	32
4.14. Le cas du processus d'entrée « test à blanc ».....	32
4.15. Structure du Workflow de l'entrée.....	33
5. Mise à jour unitaire (Update).....	39
5.1. Workflow de mise à jour unitaire des unités archivistiques.....	39
5.1.1. Mise à jour d'une unité archivistique (vision métier).....	39
5.1.2. Processus de mise à jour unitaire des métadonnées descriptives d'une unité archivistique STP_UPDATE_UNIT_DESC.....	39
5.1.3. Structure de workflow de mise à jour des métadonnées descriptives.....	42
5.1.4. Mise à jour unitaire des règles de gestion d'une unité archivistique STP_UPDATE_UNIT	

(AccessInternalModuleImpl.java).....	42
5.1.5. Structure de workflow de mise à jour des métadonnées de gestion des unités archivistiques.....	46
5.2. Workflow de mise à jour des règles de gestion des unités archivistiques lors de l'import d'un nouveau référentiel.....	47
5.2.1. Mise à jour des règles de gestion des unités archivistiques (UPDATE_RULES_ARCHIVE_UNITS)	47
5.2.2. Processus de préparation des listes d'unités archivistiques à mettre à jour STP_PREPARE_LISTS (updatedRules.json).....	47
5.2.3. Invalidation des unités archivistiques dont les règles de gestion sont à invalider STP_INVALIDATE (Distribution sur LIST GUID/unitsToInvalidate.jsonl).....	48
5.2.4. Mise à jour des unités archivistiques STP_UNIT_UPDATE (Distribution sur LIST GUID/UnitsWithoutLevel).....	48
5.2.5. Mise à jour des processus d'entrée en cours – STP_UPDATE_RUNNING_INGESTS (updatedRules.json).....	49
5.2.6. Structure de workflow du processus de mise à jour des règles de gestion des unités archivistiques....	50
6. Mise à jour de masse (Mass update).....	51
6.1. Workflow de mise à jour de masse des métadonnées descriptives des unités archivistiques.....	51
6.1.1. Processus de mise à jour en masse des métadonnées descriptives des unités archivistiques (MASS_UPDATE_UNIT_DESC).....	51
6.1.2. Processus de préparation de la liste des unités archivistiques à mettre à jour et des autorisations de modification STP_CHECK_AND_COMPUTE.....	51
6.1.3. Processus de traitement de mise à jour des unités archivistiques STP_UPDATE.....	52
6.1.4. Processus de génération du rapport de mise à jour des métadonnées descriptives des unités archivistiques STP_MASS_UPDATE_FINALIZE.....	53
6.1.5. Structure du workflow de mise à jour en masse des métadonnées descriptives des unités archivistiques.....	54
6.1.6. Rapport de modification en masse des métadonnées descriptives des unités archivistiques.....	55
6.2. Workflow de mise à jour de masse des métadonnées de gestion des unités archivistiques.....	58
6.2.1. Processus de mise à jour en masse des métadonnées de gestion des unités archivistiques (MASS_UPDATE_UNIT_RULE).....	58
6.2.2. Processus de préparation de la liste des unités archivistiques à mettre à jour et des autorisations de modification STP_CHECK_AND_COMPUTE.....	58
6.2.3. Invalidation des unités archivistiques dont les règles de gestion sont à invalider STP_INVALIDATE (Distribution sur LIST GUID/unitsToInvalidate.jsonl).....	61
6.2.4. Processus de génération du rapport de mise à jour des métadonnées descriptives et de gestion des unités archivistiques STP_MASS_UPDATE_FINALIZE.....	61
6.2.5. Structure de workflow de mise à jour en masse des métadonnées de gestion des unités archivistiques.....	63
6.2.6. Rapport de modification en masse des métadonnées de gestion des unités archivistiques.....	65
6.3. Workflow de calcul des règles de gestion hérités des unités archivistiques pour faciliter les requêtes.....	68
6.3.1. Processus de préparation des règles de gestion applicables d'unités archivistiques à mettre à jour STP_COMPUTE_INHERITED_RULES_PREPARATION.....	68
6.3.2. Processus de calcul des règles de gestion applicables. STP_COMPUTE_INHERITED_RULES.....	69
6.3.3. Processus de finalisation du calcul des règles de gestion applicables. STP_COMPUTE_INHERITED_RULES_FINALIZATION.....	69
6.3.4. Structure de workflow de mise à jour du calcul des règles de gestion hérités des unités archivistiques.....	70
6.4. Workflow de suppression des règles de gestion hérités et calculés des unités archivistiques.....	71
6.4.1. Processus de préparation des règles de gestion applicables d'unités archivistiques à mettre à jour STP_COMPUTE_INHERITED_RULES_PREPARATION.....	71
6.4.2. Processus de suppression des règles de gestion applicables. STP_COMPUTE_INHERITED_RULES_DELETE.....	72
6.4.3. Processus de finalisation de la suppression des règles de gestion applicables.	

STP_COMPUTE_INHERITED_RULES_FINALIZATION.....	72
6.4.4. Structure de workflow de suppression du calcul des règles de gestion hérités des unités archivistiques.	73
7. Mise à jour unitaire de masse (Bulk update).....	74
7.1. Processus de mise à jour unitaire en masse des métadonnées descriptives des unités archivistiques (BULK_ATOMIC_UPDATE_UNIT_DESC).....	74
7.2. Processus de préparation de la liste des unités archivistiques à mettre à jour et des autorisations de modification STP_CHECK_AND_COMPUTE.....	74
7.2.1. Vérification des seuils de limitation des requêtes de traitement des unités archivistiques CHECK_QUERIES_TRESHOLD.....	75
7.2.2. Préparation de la liste des unités archivistiques à mettre à jour unitairement PREPARE_BULK_ATOMIC_UPDATE_UNIT_LIST.....	75
7.3. Processus de traitement de mise à jour unitaire des unités archivistiques STP_UPDATE.....	75
7.4. Processus de génération du rapport de mise à jour unitaire des métadonnées descriptives des unités archivistiques STP_BULK_ATOMIC_UPDATE_FINALIZE.....	76
7.5. Structure du Workflow de mise à jour unitaire des métadonnées descriptive.....	77
7.6. Rapport de modification unitaire en masse des métadonnées descriptives des unités archivistiques.	78
7.6.1. Exemple de fichier JSONL : rapport de modification unitaire en masse de métadonnées descriptives	78
7.6.2. Détail du rapport.....	78
8. Export d'un DIP.....	80
8.1. Processus de création du bordereau de mise à disposition (STP_CREATE_MANIFEST).....	80
8.1.1. Vérification des seuils de limitation de traitement des unités archivistiques CHECK_DISTRIBUTION_THRESHOLD.....	80
8.1.2. Création du bordereau CREATE_MANIFEST (CreateManifest.java).....	80
8.2. Processus de déplacement des objets binaires vers l'espace de travail interne (STP_PUT_BINARY_ON_WORKSPACE).....	80
8.2.1. Vérification de la disponibilité des ressources EXPORT_CHECK_RESOURCE_AVAILABILITY (ExportCheckResourceAvailability.java).....	81
8.2.2. Déplacement des objets binaires vers le workspace PUT_BINARY_ON_WORKSPACE (PutBinaryOnWorkspace.java).....	81
8.3. Processus de création du DIP et de son déplacement vers l'offre de stockage (STP_STORE_MANIFEST).....	81
8.4. Structure du Workflow d'export de DIP.....	82
9. Élimination.....	84
9.1. Workflow d'analyse de l'élimination des unités archivistiques.....	84
9.1.1. Processus de préparation de l'analyse de l'élimination des unités archivistiques (STP_ELIMINATION_ANALYSIS_PREPARATION).....	84
9.1.2. Processus d'indexation de l'élimination des unités archivistiques (STP_ELIMINATION_ANALYSIS_UNIT_INDEXATION).....	85
9.1.3. Processus de finalisation de l'analyse de l'élimination des unités archivistiques (STP_ELIMINATION_ANALYSIS_FINALIZATION).....	85
9.1.4. Structure de workflow d'analyse de l'élimination des unités archivistiques.....	87
9.2. Workflow d'élimination définitive des unités archivistiques.....	88
9.2.1. Processus de préparation de l'élimination définitive des unités archivistiques (STP_ELIMINATION_ACTION_PREPARATION).....	88
9.2.2. Processus d'élimination définitive des unités archivistiques éliminables (STP_ELIMINATION_ACTION_DELETE_UNIT).....	89
9.2.3. Processus de préparation de l'élimination définitive des groupes d'objets techniques dont toutes les unités archivistiques parentes ont été éliminées	

(STP_ELIMINATION_ACTION_OBJECT_GROUP_PREPARATION).....	90
9.2.4. Processus d'élimination définitive des groupes d'objets techniques dont toutes les unités archivistiques parentes sont éliminées (STP_ELIMINATION_ACTION_DELETE_OBJECT_GROUP)....	90
9.2.5. Processus de détachement des groupes d'objets techniques dont certaines unités archivistiques parentes sont éliminées (STP_ELIMINATION_ACTION_DETACH_OBJECT_GROUP).....	91
9.2.6. Processus de mise à jour du registre des fonds suite à l'élimination définitive des unités archivistiques (STP_ELIMINATION_ACTION_ACCESSION_REGISTER_PREPARATION).....	92
9.2.7. Processus de mise à jour du registre des fonds suite à l'élimination définitive des unités archivistiques (STP_ELIMINATION_ACTION_ACCESSION_REGISTER_UPDATE).....	93
9.2.8. Processus de génération du rapport d'élimination définitive des unités archivistiques (STP_ELIMINATION_ACTION_REPORT_GENERATION).....	94
9.2.9. Processus de finalisation de l'élimination définitive des unités archivistiques (STP_ELIMINATION_ACTION_FINALIZATION).....	94
9.2.10. Structure du workflow d'analyse des éliminables et action d'élimination.....	96
9.2.11. Rapport d'élimination.....	99
10. Modification d'arborescence (Reclassification).....	102
10.1. Étape de préparation de la modification d'arborescence des unités archivistiques	
STP_RECLASSIFICATION_PREPARATION.....	102
10.1.1. Vérification des processus concurrents CHECK_CONCURRENT_WORKFLOW_LOCK.....	102
10.1.2. Chargement des unités archivistiques : RECLASSIFICATION_PREPARATION_LOAD_REQUEST.....	102
10.1.3. Vérification de la cohérence du graphe : RECLASSIFICATION_PREPARATION_CHECK_GRAPH.....	103
10.1.4. Préparation de la mise à jour du graphe : RECLASSIFICATION_PREPARATION_UPDATE_DISTRIBUTION.....	103
10.2. Étape de détachement des unités archivistiques (STP_UNIT_DETACHMENT).....	103
10.2.1. Détachement des unités archivistiques UNIT_DETACHMENT.....	104
10.2.2. Enregistrement des métadonnées des unités archivistiques sur l'offre de stockage UNIT_METADATA_STORAGE.....	104
10.2.3. Établissement de la liste des objets OBJECTS_LIST_EMPTY.....	104
10.3. Processus de rattachement des unités archivistiques STP_UNIT_ATTACHMENT.....	105
10.3.1. Processus de rattachement des unités archivistiques : UNIT_ATTACHMENT.....	105
10.3.2. Enregistrement des métadonnées des unités archivistiques sur l'offre de stockage UNIT_METADATA_STORAGE.....	105
10.3.3. Établissement de la liste des objets OBJECTS_LIST_EMPTY.....	105
10.4. Mise à jour des graphes des unités archivistiques STP_UNIT_GRAPH_COMPUTE.....	106
10.5. Mise à jour des graphes des groupes d'objets STP_OBJECT_GROUP_GRAPH_COMPUTE.....	106
10.6. Finalisation de la modification d'arborescence des unités archivistiques (STP_RECLASSIFICATION_FINALIZATION).....	107
10.7. Structure de workflow de modification d'arborescence.....	108
11. Transfert.....	110
11.1. Workflow de préparation du transfert des unités archivistiques et des groupes d'objets.....	110
11.1.1. Processus de création du bordereau de mise à disposition (STP_CREATE_MANIFEST).....	110
11.1.2. Processus de déplacement des objets binaires vers l'espace de travail interne (STP_PUT_BINARY_ON_WORKSPACE).....	111
11.1.3. Processus de création du SIP de transfert et de son déplacement vers l'offre de stockage (STP_STORE_MANIFEST).....	111
11.1.4. Structure du Workflow de préparation du transfert des unités archivistiques et des groupes d'objets.....	112
11.1.5. Rapport de préparation du transfert.....	113
11.2. Workflow d'acquittement du transfert des unités archivistiques et des groupes d'objets.....	114

11.2.1. Processus de préparation de l'acquittement du transfert (STP_TRANSFER_REPLY_PREPARATION).....	114
11.2.2. Processus de purge des unités archivistiques (STP_TRANSFER_REPLY_DELETE_UNIT).....	115
11.2.3. Processus de préparation de la purge des groupes d'objets techniques (STP_TRANSFER_REPLY_OBJECT_GROUP_PREPARATION).....	116
11.2.4. Processus de purge des groupes d'objets techniques dont les unités archivistiques parentes sont purgées (STP_TRANSFER_REPLY_DELETE_OBJECT_GROUP).....	116
11.2.5. Processus de détachement des groupes d'objets techniques dont certaines unités archivistiques parentes sont purgées (STP_TRANSFER_REPLY_DETACH_OBJECT_GROUP).....	117
11.2.6. Processus de préparation du Registre des Fonds suite à l'acquittement du transfert (STP_TRANSFER_REPLY_ACCESSION_REGISTER_PREPARATION).....	117
11.2.7. Processus de mise à jour du Registre des Fonds suite à l'acquittement du transfert (STP_TRANSFER_REPLY_ACCESSION_REGISTER_UPDATE).....	117
11.2.8. Processus de génération du rapport d'acquittement du transfert (STP_TRANSFER_REPLY_REPORT_GENERATION).....	118
11.2.9. Processus de finalisation de l'acquittement du transfert (STP_TRANSFER_REPLY_FINALIZATION).....	118
11.2.10. Structure du Workflow d'acquittement du transfert des unités archivistiques et des groupes d'objets	119
11.2.11. Rapport d'acquittement du transfert.....	122
12. Restauration de métadonnées essentielles (Revert Essential Metadata).....	127
12.1. Processus de préparation de la restauration des métadonnées essentielles (STP_CHECK_REVERT).....	127
12.2. Processus de traitement de la restauration des métadonnées essentielles (STP_REVERT_UPDATE).....	128
12.3. Processus de finalisation de la restauration des métadonnées essentielles (STP_REVERT_UPDATE_FINALIZE).....	128
12.4. Structure du Workflow de mise à jour unitaire des métadonnées descriptive.....	129
12.5. Rapport de restauration de métadonnées essentielles des unités archivistiques.....	130
12.5.1. Exemple de fichier JSONL : rapport de restauration de métadonnées essentielles.....	130
12.5.2. Détail du rapport.....	130

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 seconde partie du document détail les workflows suivants :

- Entrée
- Mise à jour unitaire
- Mise à jour de masse
- Mise à jour unitaire de masse (bulk update)
- Export DIP
- Élimination
- Modification d'arborescence
- Transfert
- Restauration de métadonnées essentielles

4. INGEST

Cette section décrit le processus (workflow) d'entrée, utilisé lors du transfert d'un Submission Information Package (SIP) dans la solution logicielle Vitam. Ce workflow se décompose en deux grandes catégories : le processus d'entrée externe dit « ingest externe » et le processus d'entrée interne dit « ingest interne ». Le premier prend en charge le SIP et effectue des contrôles techniques préalables, tandis que le second débute dès le premier traitement métier. Ex: Le processus d'entrée externe comprend l'étape : STP_SANITY_CHECK_SIP (Contrôle sanitaire du SIP). Les autres étapes font partie du processus d'entrée interne.

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.

4.1. Processus des contrôles préalables à l'entrée (STP_SANITY_CHECK_SIP)

4.1.1. Contrôle sanitaire du SIP SANITY_CHECK_SIP (IngestExternalImpl.java)

- **Règle** : tâche consistant à vérifier l'absence de virus dans le SIP
- **Type** : bloquant
- **Statuts** :
 - OK : aucun virus n'a été détecté dans le SIP (SANITY_CHECK_SIP.OK = Succès du processus des contrôles préalables à l'entrée)
 - KO : un ou plusieurs virus ont été détectés dans le SIP (SANITY_CHECK_SIP.KO = Échec du processus des contrôles préalables à l'entrée)
 - FATAL : une erreur technique est survenue lors de la vérification de la présence de virus dans le SIP (SANITY_CHECK_SIP.FATAL = Erreur technique lors du processus des contrôles préalables à l'entrée)

4.1.2. Contrôle du format du conteneur du SIP CHECK_CONTAINER (IngestExternalImpl.java)

- **Règle** : tâche consistant à vérifier le format du SIP via un outil d'identification de format qui se base sur le référentiel des formats qu'il intègre
Formats acceptés : .zip, .tar, .tar.gz, .tar.bz2 et tar.gz2
- **Type** : bloquant
- **Statuts** :
 - OK : le conteneur du SIP est au bon format (CHECK_CONTAINER.OK = Succès du contrôle du format du conteneur du SIP)
 - KO : le conteneur du SIP n'est pas au bon format (CHECK_CONTAINER.KO = Échec du contrôle du format du conteneur du SIP)
 - FATAL : une erreur technique est survenue lors de la vérification du format du conteneur du SIP, liée à l'outil d'identification des formats (CHECK_CONTAINER.FATAL = Erreur technique lors du contrôle du format du conteneur du SIP)

4.1.3. Contrôle du nom du bordereau de transfert MANIFEST_FILE_NAME_CHECK (IngestExternalImpl.java)

- **Règle** : tâche consistant à vérifier le nom du bordereau de transfert. Le nom du bordereau doit être conforme avec l'expression régulière suivante :

$$\wedge([a-zA-Z0-9]{1,56}[_-]{1})_{0,1}[_]{0,1}(manifest.xml)b \gg$$
 A savoir : une chaîne de caractères débutant par des caractères alphanumériques sans accent, jusqu'à 56 caractères, suivi d'un « - » ou d'un « _ » puis suivi de « manifest.xml ». Exemples valides : « MonNouveau-manifest.xml », « UnAutreBordereau_manifest.xml ». « manifest.xml » tout court est également valide. Un SIP ne

possédant pas du tout de bordereau de transfert verra également son entrée terminer en KO à cette tâche.

- **Type** : bloquant
- **Statuts** :
 - OK : le nom du bordereau de transfert est conforme (MANIFEST_FILE_NAME_CHECK.OK = Succès du contrôle du nom du bordereau de transfert : nom du fichier conforme)
 - KO : le nom du bordereau de transfert n'est pas conforme (MANIFEST_FILE_NAME_CHECK.KO = Échec du contrôle du nom du bordereau de transfert : nom du fichier non conforme)
 - FATAL : une erreur technique est survenue lors de la vérification du nom du bordereau de transfert (MANIFEST_FILE_NAME_CHECK.FATAL = Erreur technique lors du contrôle du nom du bordereau de transfert)

4.1.4. Vérification de l'intégrité du bordereau de transfert MANIFEST_DIGEST_CHECK (IngestExternalImpl.java)

- **Règle** : tâche consistant à vérifier l'empreinte du bordereau de transfert calculée par la solution logicielle Vitam et celle déclarée dans le bordereau. Si l'empreinte déclarée dans le bordereau de transfert n'a pas été calculée avec l'algorithme SHA-512, alors l'empreinte est recalculée avec cet algorithme. Elle sera alors enregistrée dans la solution logicielle Vitam.

Algorithmes autorisés en entrée : MD5, SHA-1, SHA-256, SHA-512

- **Type** : bloquant
- **Statuts** :
 - KO : L'empreinte du bordereau de transfert n'est pas conforme (MANIFEST_DIGEST_CHECK.KO=Échec du contrôle de l'empreinte du bordereau de transfert : fichier non conforme)
 - FATAL : Une erreur technique est survenue lors du contrôle de l'empreinte (MANIFEST_DIGEST_CHECK.FATAL=Erreur technique lors du contrôle de l'empreinte du bordereau de transfert)

4.2. Processus de réception du SIP dans Vitam STP_UPLOAD_SIP (IngestInternalResource.java)

- **Règle** : étape consistant à vérifier la bonne réception du SIP sur l'espace de travail interne (« workspace »)
- **Type** : bloquant
- **Statuts** :
 - OK : le SIP a été réceptionné sur l'espace de travail interne (STP_UPLOAD_SIP.OK = Succès du processus de réception du SIP)
 - KO : le SIP n'a pas été réceptionné sur l'espace de travail interne (STP_UPLOAD_SIP.KO = Échec du processus de réception du SIP)
 - FATAL : une erreur technique est survenue lors de la réception du SIP dans la solution logicielle Vitam, par exemple suite à une indisponibilité du serveur (STP_UPLOAD_SIP.FATAL = Erreur technique lors du processus de réception du SIP)

4.3. Processus de contrôle du SIP (STP_INGEST_CONTROL_SIP)

4.3.1. Vérification globale du CHECK_SEDA (CheckSedaActionHandler.java)

- **Règle** : tâche consistant à vérifier la cohérence physique du SIP reçu par rapport au modèle de SIP accepté
Type de SIP accepté : le bordereau de transfert, obligatoire dans le SIP, doit être conforme au schéma xsd par défaut fourni avec le standard SEDA v. 2.1, le SIP doit satisfaire les exigences du document

« Structuration des SIP » et doit posséder un répertoire unique nommé « Content ».

- **Type** : bloquant
- **Statuts** :
 - OK : le SIP est présent et conforme au schéma xsd par défaut fourni avec le standard SEDA v.2.1. il satisfait aux exigences de « structuration des SIP » et possède un repertoire unique « Content » (CHECK_SEDA.OK = Succès de la vérification globale du SIP)
 - KO :
 - Cas 1 : le bordereau de transfert est introuvable dans le SIP ou n'est pas au format XML (CHECK_SEDA.NO_FILE.KO = Absence du bordereau de transfert ou bordereau de transfert au mauvais format)
 - Cas 2 : le bordereau de transfert n'est pas au format XML (CHECK_SEDA.NOT_XML_FILE.KO = Échec de la vérification globale du SIP : bordereau de transfert non conforme aux caractéristiques d'un fichier xml)
 - Cas 3 : le bordereau de transfert ne respecte pas le schéma par défaut fourni avec le standard SEDA 2.1 (CHECK_SEDA.NOT_XSD_VALID.KO = Échec de la vérification globale du SIP : bordereau de transfert non conforme au schéma SEDA 2.1)
 - Cas 4 : le SIP contient plus d'un dossier « Content » (CHECK_SEDA.CONTAINER_FORMAT.DIRECTORY.KO = Échec de la vérification globale du SIP : le SIP contient plus d'un dossier ou un dossier dont le nommage est invalide)
 - Cas 5 : le SIP contient plus d'un seul fichier à la racine (CHECK_SEDA.CONTAINER_FORMAT.FILE.KO = Échec de la vérification globale du SIP : le SIP contient plus d'un fichier à sa racine)
 - Cas 6 : l'action est déjà exécutée CHECK_SEDA.ALREADY_EXECUTED = Action déjà exécutée : Pas de vérification globale du SIP
 - FATAL :
 - Cas 1 : une erreur technique est survenue lors de la vérification globale du SIP (CHECK_SEDA.FATAL = Erreur technique lors de la vérification globale du SIP)
 - Cas 2 : une erreur technique est survenue lors de la vérification globale du SIP (CHECK_SEDA.NOT_XML_FILE.FATAL=Erreur technique lors de la vérification globale du SIP)
 - Cas 3 : une erreur technique est survenue lors de la vérification globale du SIP (CHECK_SEDA.NOT_XSD_VALID.FATAL=Erreur technique lors de la vérification globale du SIP)

4.3.2.Vérification de l'en-tête du bordereau de transfert CHECK_HEADER (CheckHeaderActionHandler.java)

- **Règles** : tâche permettant de vérifier les informations générales du bordereau de transfert (nommées « header » dans le fichier « manifest.xml ») et de l'existence du service producteur (OriginatingAgencyIdentifier)
- **Type** : bloquant
- **Statuts** :
 - OK : les informations du bordereau de transfert sont conformes et le service producteur est déclaré (CHECK_HEADER.OK = Succès de la vérification générale du bordereau de transfert)
 - KO :
 - Cas 1 : les informations du bordereau de transfert ne sont pas conformes ou il n'y a pas de service producteur déclaré (CHECK_HEADER.KO = Échec de la vérification générale du bordereau de transfert)
 - Cas 2 : les données référentielles sont inactives (CHECK_HEADER.INACTIVE.KO = Échec de la vérification générale du bordereau de transfert : donnée référentielle inactive)
 - Cas 3 : les données référentielles sont inconnues (CHECK_HEADER.UNKNOWN.KO = Échec

de la vérification générale du bordereau de transfert : donnée référentielle inconnue)

- Cas 4 : il y a une différence entre le profil déclaré dans le bordereau de transfert et celui déclaré dans le contrat (CHECK_HEADER.DIFF.KO = Échec de la vérification générale du bordereau de transfert : différence entre le profil déclaré dans le bordereau de transfert et celui déclaré dans le contrat)
- Cas 5 : un des champs obligatoires n'est pas remplie (CHECK_HEADER.EMPTY_REQUIRED_FIELD.KO = Vérification générale du bordereau de transfert : champ obligatoire vide)
- Cas 6 : la vérification a déjà été effectuée (CHECK_HEADER.ALREADY_EXECUTED = Action déjà exécutée : pas de vérification générale du bordereau de transfert)
- FATAL : une erreur technique est survenue lors des contrôles sur les informations générales du bordereau de transfert (CHECK_HEADER.FATAL = Erreur technique lors de la vérification générale du bordereau de transfert)

La tâche `check_header` contient les traitements suivants :

4.3.3. Vérification de la présence et contrôle des services agents CHECK_HEADER.CHECK_AGENT

Ce traitement n'est exécuté que si la valeur IN de `checkOriginatingAgency` est « true ».

- **Règle** : traitement consistant à vérifier le service producteur ainsi que le service versant déclarés dans le SIP par rapport au référentiel des services agents présent dans la solution logicielle Vitam
- **Type** : bloquant
- **Statuts** :
 - OK : le service producteur et, le cas échéant, le service versant déclarés dans le SIP est valide (service agent existant dans le référentiel des services agents)(CHECK_HEADER.CHECK_AGENT.OK = Succès de la vérification de la présence et du contrôle des services agents)
 - KO :
 - Cas 1 : aucun service producteur n'est déclaré dans la balise dédiée dans le bordereau de transfert (CHECK_HEADER.CHECK_AGENT.EMPTY_REQUIRED_FIELD.KO = Échec de la vérification de la présence et du contrôle des services agents : champ obligatoire vide)
 - Cas 2 : le service producteur et, le cas échéant, le service versant déclarés dans le SIP n'est pas connu du référentiel des services agents (CHECK_HEADER.CHECK_AGENT.UNKNOWN.KO = Échec de la vérification de la présence et du contrôle des services agents : services agents inconnus du référentiel des services agents)
 - Cas 3 : la balise permettant de déclarer un service producteur est absente du bordereau de transfert (CHECK_HEADER.CHECK_AGENT.KO = Échec de la vérification de la présence et du contrôle des services agents)
 - FATAL : une erreur technique est survenue lors de la vérification de la présence et du contrôle des services agents (CHECK_HEADER.CHECK_AGENT.FATAL = Erreur technique lors de la vérification de la présence et du contrôle des services agents)

4.3.4. Vérification de la présence et contrôle du contrat d'entrée CHECK_HEADER.CHECK_CONTRACT_INGEST

Ce traitement n'est exécuté que si la valeur IN de `checkContract` est « true ».

- **Règle** : traitement consistant à vérifier le contrat d'entrée déclaré dans le SIP par rapport au référentiel des contrats d'entrée présent dans la solution logicielle Vitam
- **Type** : bloquant
- **Statuts** :
 - OK : le contrat déclaré dans le SIP existant dans le référentiel des contrats d'entrée de la solution logicielle Vitam et est actif (CHECK_HEADER.CHECK_CONTRACT_INGEST.OK = Succès de la

- vérification de la présence et du contrôle du contrat d'entrée)
- KO :
 - Cas 1 : le contrat d'entrée déclaré dans le SIP est inexistant
(CHECK_HEADER.CHECK_CONTRACT_INGEST.CONTRACT_UNKNOWN.KO = Échec de la vérification de la présence du contrat d'entrée : contrat d'entrée inconnu du référentiel des contrats d'entrée)
 - Cas 2 : le contrat d'entrée déclaré dans le SIP est inactif
(CHECK_HEADER.CHECK_CONTRACT_INGEST.CONTRACT_INACTIVE.KO = Échec de la vérification du caractère actif du contrat d'entrée)
 - Cas 3 : aucun contrat d'entrée n'a été trouvé dans le manifeste
(CHECK_HEADER.CHECK_CONTRACT_INGEST.CONTRACT_NOT_IN_MANIFEST.KO = Échec de la vérification de la présence du contrat d'entrée : le champ ArchivalAgreement est absent du bordereau de transfert)
 - Cas 4 : le contrat d'entrée déclaré dans le SIP n'existe pas dans le contexte applicatif
(CHECK_HEADER.CHECK_CONTRACT_INGEST.CONTRACT_NOT_IN_CONTEXT.KO = Échec du contrôle de la présence du contrat d'entrée dans le contexte applicatif)
 - Cas 5 : le contexte applicatif est inexistant
(CHECK_HEADER.CHECK_CONTRACT_INGEST.CONTEXT_UNKNOWN.KO = Échec du contrôle de la présence du contexte applicatif : contexte inconnu du référentiel des contextes)
 - Cas 6 : le contexte applicatif est inactif
(CHECK_HEADER.CHECK_CONTRACT_INGEST.CONTEXT_INACTIVE.KO = Échec du contrôle du caractère actif du contexte applicatif)
 - Cas 7 : erreur lors de la récupération du contexte applicatif
(CHECK_HEADER.CHECK_CONTRACT_INGEST.CONTEXT_CHECK_ERROR.KO = Échec de la vérification de la présence et du contrôle du contexte applicatif)
 - Cas 8 : le contrat de gestion déclaré est inexistant
CHECK_HEADER.CHECK_CONTRACT_INGEST.MANAGEMENT_CONTRACT_UNKNOWN.KO = Échec de la vérification de la présence du contrat de gestion déclaré dans le contrat d'entrée: contrat de gestion connu dans le référentiel des contrats de gestion
 - Cas 9 : le contrat de gestion déclaré est inactif
CHECK_HEADER.CHECK_CONTRACT_INGEST.MANAGEMENT_CONTRACT_INACTIVE.KO = Échec de la vérification de la présence du contrat de gestion déclaré dans le contrat d'entrée: contrat de gestion au statut inactif dans le référentiel des contrats de gestion
 - Cas 10 : le contrat de gestion déclaré est invalide
CHECK_HEADER.CHECK_CONTRACT_INGEST.MANAGEMENT_CONTRACT_INVALID.KO = Échec de la vérification de la présence du contrat de gestion déclaré dans le contrat d'entrée: échec de validation des stratégies de stockage déclarées dans le contrat de gestion
 - FATAL : une erreur technique est survenue lors de la vérification de la présence et du contrôle du contrat d'entrée ou du contexte applicatif
(CHECK_HEADER.CHECK_CONTRACT_INGEST.FATAL = Erreur technique lors de la vérification de la présence et du contrôle du contrat d'entrée ou du contexte applicatif)

4.3.5. Vérification de la relation entre le contrat d'entrée et le profil d'archivage CHECK_HEADER.CHECK_IC_AP_RELATION

Ce traitement n'est exécuté que si la valeur IN de *checkProfile* est « true ».

- **Règle** : traitement consistant à vérifier que le profil d'archivage déclaré dans le contrat d'entrée du SIP est le même que celui déclaré dans le bordereau de transfert.
- **Type** : bloquant
- **Statuts** :
 - OK : le profil d'archivage déclaré dans le contrat d'entrée et celui déclaré dans le bordereau de transfert sont les mêmes (CHECK_HEADER.CHECK_IC_AP_RELATION.OK = Succès de la

- vérification de la relation entre le contrat d'entrée et le profil)
- KO :
 - Cas 1 : le profil d'archivage déclaré dans le SIP est inexistant (CHECK_HEADER.CHECK_IC_AP_RELATION.UNKNOWN.KO = Échec du contrôle de la présence du profil d'archivage dans le référentiel des profils d'archivage)
 - Cas 2 : le profil d'archivage déclaré dans le SIP est inactif (CHECK_HEADER.CHECK_IC_AP_RELATION.INACTIVE.KO = Échec du contrôle du caractère actif du profil d'archivage)
 - Cas 3 : le profil d'archivage déclaré dans le contrat d'entrée et celui déclaré dans le bordereau de transfert ne sont pas les mêmes (CHECK_HEADER.CHECK_IC_AP_RELATION.DIFF.KO = Échec du contrôle de cohérence entre le profil d'archivage déclaré dans le bordereau de transfert et celui déclaré dans le contrat d'entrée)
- FATAL : une erreur technique est survenue lors de la vérification de la relation entre le contrat d'entrée et le profil d'archivage (CHECK_HEADER.CHECK_IC_AP_RELATION.FATAL = Erreur technique lors de la vérification de la relation entre le contrat d'entrée et le profil d'archivage)

4.3.6. Vérification de la conformité du bordereau de transfert par le profil d'archivage CHECK_HEADER.CHECK_ARCHIVEPROFILE

- **Règle** : traitement consistant à vérifier que le bordereau de transfert du SIP est conforme aux exigences du profil d'archivage. Si aucun profil d'archivage ne s'applique au SIP, ce traitement est ignoré
- **Type** : bloquant
- **Statuts** :
 - OK : le bordereau de transfert est conforme aux exigences du profil d'archivage (CHECK_HEADER.CHECK_ARCHIVEPROFILE.OK = Succès de la vérification de la conformité au profil d'archivage)
 - KO : le bordereau de transfert n'est pas conforme aux exigences du profil d'archivage (CHECK_HEADER.CHECK_ARCHIVEPROFILE.KO = Échec de la vérification de la conformité au profil d'archivage)
 - FATAL : une erreur technique est survenue lors de la vérification du bordereau de transfert par rapport au profil d'archivage (CHECK_HEADER.CHECK_ARCHIVEPROFILE.FATAL = Erreur technique lors de la vérification de la conformité au profil d'archivage)

4.3.7. Préparation des informations de stockage PREPARE_STORAGE_INFO (PrepareStorageInfoActionHandler.java)

- **Règle** : tâche consistant à récupérer les informations liées aux offres de stockage à partir des stratégies
- **Type** : bloquant
- **Statuts** :
 - OK : la récupération des informations de stockage a bien été effectuée (PREPARE_STORAGE_INFO.OK = Succès de la préparation des informations de stockage)
 - KO : la récupération des informations de stockage n'a pas pu être effectuée (PREPARE_STORAGE_INFO.KO = Échec de la préparation des informations de stockage)
 - FATAL : une erreur technique est survenue lors de la récupération des informations de stockage (PREPARE_STORAGE_INFO.FATAL = Erreur technique lors de la récupération des informations de stockage)

4.3.8. Vérification des objets et groupes d'objets CHECK_DATAOBJECTPACKAGE

- **Règle** : tâche consistant à vérifier les objets et groupes d'objets
- **Type** : bloquant
- **Statuts** :
 - OK : la vérification des objets et groupes d'objets à été effectué avec succès

- (CHECK_DATAOBJECTPACKAGE.OK=Succès de la vérification des objets et groupes d'objets)
- KO : la vérification des objets et groupes d'objets n'a pu être effectué (CHECK_DATAOBJECTPACKAGE.KO=Échec de la vérification des objets et groupes d'objets)
- FATAL : une erreur technique est survenue lors de la vérification des objets et groupes d'objets (CHECK_DATAOBJECTPACKAGE.FATAL=Erreur technique lors de la vérification des objets et groupes d'objets)

La tâche Check_DataObjectPackage contient les traitements suivants :

4.3.9. Vérification des usages des groupes d'objets

CHECK_DATAOBJECTPACKAGE.CHECK_MANIFEST_DATAOBJECT_VERSION (CheckVersionActionHandler.java)

- **Règle** : traitement consistant à vérifier que tous les objets décrits dans le bordereau de transfert du SIP déclarent un usage conforme à la liste des usages acceptés dans la solution logicielle Vitam ainsi qu'un numéro de version respectant la norme de ce champ
- **Types d'usages acceptés** :
 - original papier (PhysicalMaster),
 - original numérique (BinaryMaster),
 - diffusion (Dissemination),
 - vignette (Thumbnail),
 - contenu brut (TextContent).

Les numéros de versions sont optionnels, il s'agit d'un entier positif ou nul (0, 1, 2...).

La grammaire est : « usage_version ». Exemples : « BinaryMaster_2 », « TextContent_10 » ou sans numéro de versions « PhysicalMaster ».

- **Type** : bloquant
- **Statuts** :
 - OK : les objets contenus dans le SIP déclarent tous dans le bordereau de transfert un usage cohérent avec ceux acceptés et optionnellement un numéro de version respectant la norme de ce champ usage, par exemple « BinaryMaster_2 » (CHECK_DATAOBJECTPACKAGE.CHECK_MANIFEST_DATAOBJECT_VERSION.OK = Succès de la vérification des usages des objets)
 - KO :
 - Cas 1 : un ou plusieurs BinaryMaster sont déclarés dans un ou plusieurs objets physiques (CHECK_DATAOBJECTPACKAGE.CHECK_MANIFEST_DATAOBJECT_VERSION.PDO_DATAOBJECTVERSION_BINARYMASTER.KO = L'objet physique déclare un usage « BinaryMaster ». Cet usage n'est pas autorisé pour les objets physiques)
 - Cas 2 : un ou plusieurs PhysicalMaster sont déclarés dans un ou plusieurs objets binaires (CHECK_DATAOBJECTPACKAGE.BDO_DATAOBJECTVERSION_PHYSICALMASTER.KO = Au moins un objet binaire déclare un usage « PhysicalMaster ». Cet usage n'est pas autorisé pour les objets binaires)
 - Cas 3 : un ou plusieurs objets contenus dans le SIP déclarent dans le bordereau de transfert un usage ou un numéro de version incohérent avec ceux acceptés (CHECK_DATAOBJECTPACKAGE.CHECK_MANIFEST_DATAOBJECT_VERSION.INVALID_ID_DATAOBJECTVERSION.KO = Cet objet déclare un usage incorrect. L'usage doit s'écrire sous la forme [usage] ou [usage]_[version]. « Usage » doit être parmi l'énumération DataObjectVersion définie pour Vitam, « version » doit être un entier positif)
 - Cas 4 : une ou plusieurs URI sont vides (CHECK_DATAOBJECTPACKAGE.CHECK_MANIFEST_DATAOBJECT_VERSION.EMPTY_REQUIRED_FIELD.KO = Il existe au moins un champ non renseigné dont la valeur est obligatoire)
 - FATAL : une erreur technique est survenue lors du contrôle des usages déclarés dans le bordereau de

transfert pour les objets contenus dans le SIP
(CHECK_MANIFEST_DATAOBJECT_VERSION.FATAL = Erreur technique lors de la vérification des usages des objets)

4.3.10. Vérification du nombre d'objets

CHECK_DATAOBJECTPACKAGE.CHECK_MANIFEST_OBJECTNUMBER (CheckObjectsNumberActionHandler.java)

- **Règle** : traitement consistant à vérifier que le nombre d'objets binaires reçus dans la solution logicielle Vitam et stocké dans l'espace de travail interne (« workspace ») est strictement égal au nombre d'objets binaires déclaré dans le manifeste du SIP
- **Type** : bloquant
- **Statuts** :
 - OK : le nombre d'objets reçus dans la solution logicielle Vitam est strictement égal au nombre d'objets déclarés dans le bordereau de transfert du SIP
(CHECK_DATAOBJECTPACKAGE.CHECK_MANIFEST_OBJECTNUMBER.OK = Succès de la vérification du nombre d'objets)
 - KO :
 - Cas 1 : le nombre d'objets reçus dans la solution logicielle Vitam est supérieur au nombre d'objets déclaré dans le bordereau de transfert du SIP
(CHECK_DATAOBJECTPACKAGE.CHECK_MANIFEST_OBJECTNUMBER.MANIFEST_INFERIOR_BDO.KO = Le bordereau de transfert déclare moins d'objets binaires qu'il n'en existe dans le répertoire Content du SIP)
 - Cas 2 : le nombre d'objets reçus dans la solution logicielle Vitam est inférieur au nombre d'objets déclaré dans le bordereau de transfert du SIP
(CHECK_DATAOBJECTPACKAGE.CHECK_MANIFEST_OBJECTNUMBER.MANIFEST_SUPERIOR_BDO.KO = Le bordereau de transfert déclare plus d'objets binaires qu'il n'en existe dans le répertoire Content du SIP)
 - Cas 3 : une ou plusieurs balises URI déclarent un chemin invalide
(CHECK_DATAOBJECTPACKAGE.CHECK_MANIFEST_OBJECTNUMBER.INVALID_URI.KO = Au moins un objet déclare une URI à laquelle ne correspond pas de fichier ou déclare une URI déjà utilisée par un autre objet)
 - FATAL : une erreur technique est survenue lors de la vérification du nombre d'objets
(CHECK_DATAOBJECTPACKAGE.CHECK_MANIFEST_OBJECTNUMBER.FATAL = Erreur technique lors de la vérification du nombre d'objets)

4.3.11. Vérification de la cohérence du bordereau de transfert

CHECK_DATAOBJECTPACKAGE.CHECK_MANIFEST (ExtractSedaActionHandler.java)

- **Règle** : traitement consistant à :
 - créer les journaux du cycle de vie des unités archivistiques et des groupes d'objets,
 - extraire les unités archivistiques, objets binaires et objets physiques du bordereau de transfert,
 - vérifier la présence de récursivité dans les arborescences des unités archivistiques et à créer l'arbre d'ordre d'indexation,
 - extraire les métadonnées contenues dans la balise ManagementMetadata du bordereau de transfert pour le calcul des règles de gestion,
 - vérifier la validité des rattachements des unités du SIP aux unités présentes dans la solution logicielle Vitam si demandés,
 - détecter des problèmes d'encodage dans le bordereau de transfert et vérifier que les objets ne font pas référence directement à des unités si ces objets possèdent des groupes d'objets,
 - vérifier la présence obligatoire d'un objet de type Master pour une entrée, et vérifier les usages d'objets autorisés pour les rattachements.

- **Type** : bloquant
- **Statuts** :
 - OK : les journaux du cycle de vie des unités archivistiques et des groupes d'objets ont été créés avec succès, aucune récursivité n'a été détectée dans l'arborescence des unités archivistiques, la structure de rattachement déclarée existe, le type de structure de rattachement est autorisé, (par exemple, un SIP peut être rattaché à un plan de classement, mais pas l'inverse) aucun problème d'encodage n'a été détecté et les objets avec groupe d'objets ne référencent pas directement les unités. L'extraction des unités archivistiques, objets binaires et physiques, la création de l'arbre d'indexation et l'extraction des métadonnées des règles de gestion ont été effectuées avec succès, les vérifications au niveau des types d'usages autorisés ont bien été effectués.
(CHECK_DATAOBJECTPACKAGE.CHECK_MANIFEST.OK = Succès du contrôle de cohérence du bordereau de transfert).
 - KO :
 - Cas 1 : au moins une demande de rattachement à des unités archivistiques existantes dans le système a échoué, car le nœud de rattachement déclaré dans le contrat d'entrées a pour valeur « null »
(CHECK_DATAOBJECTPACKAGE.CHECK_MANIFEST.NULL_LINK_PARENT_ID_ATTACHMENT.KO = Le rattachement n'a pas été effectué : le contrat d'entrée ne déclare pas de nœud de rattachement)
 - Cas 2 : au moins une demande de rattachement à des unités archivistiques existantes dans le système a échoué, car le contrat d'entrée requiert un rattachement à au moins une unité archivistique
(CHECK_DATAOBJECTPACKAGE.CHECK_MANIFEST.ATTACHMENT_REQUIRED.KO = Le contrat d'entrée requiert un rattachement à au moins une unité archivistique)
 - Cas 3 : au moins une demande de rattachement à des unités archivistiques existantes dans le système a échoué, car le contrat d'entrée n'autorise pas les rattachements
(CHECK_DATAOBJECTPACKAGE.CHECK_MANIFEST.UNAUTHORIZED_ATTACHMENT_BY_CONTRACT.KO = Le rattachement n'a pas été effectué : le contrat d'entrée n'autorise pas les rattachements)
 - Cas 4 : au moins une demande de rattachement à des unités archivistiques existantes dans le système a échoué en raison d'un nombre d'unités archivistiques répondant à la requête effectuée supérieur à 1
(CHECK_DATAOBJECTPACKAGE.CHECK_MANIFEST.TOO_MANY_FOUND_ATTACHMENT.KO = Le rattachement n'a pas été effectué : l'élément de rattachement n'est pas unique dans le système)
 - Cas 5 : au moins un objet binaire dans le bordereau de transfert déclare plusieurs version d'un même usage
(CHECK_DATAOBJECTPACKAGE.CHECK_MANIFEST.TOO_MANY_VERSION_BY_USAGE.KO = Le transfert de plusieurs versions d'un même usage dans un même versement est interdit)
 - Cas 6 : au moins une demande de rattachement à des unités archivistiques existantes dans le système a échoué en raison d'un nombre d'unités archivistiques répondant à la requête, égal à 0
(CHECK_DATAOBJECTPACKAGE.CHECK_MANIFEST.NOT_FOUND_ATTACHMENT.KO = Le rattachement n'a pas été effectué : l'élément de rattachement n'existe pas dans le système)
 - Cas 7 : au moins une demande de rattachement à des unités archivistiques existantes dans le système a échoué, car le rattachement demandé n'est pas autorisé
(CHECK_DATAOBJECTPACKAGE.CHECK_MANIFEST.UNAUTHORIZED_ATTACHMENT.KO = Le rattachement n'a pas été effectué : le rattachement n'est pas situé dans le périmètre autorisé)
 - Cas 8 : au moins une demande de rattachement à des unités archivistiques existantes dans le système a échoué, car le GUID déclaré n'est pas valide
(CHECK_DATAOBJECTPACKAGE.CHECK_MANIFEST.INVALID_GUID_ATTACHMENT.KO = Le rattachement n'a pas été effectué : l'élément de rattachement est incorrect)

- Cas 9 : au moins une demande de rattachement à des unités archivistiques existantes dans le système a échoué, car elle provoquerait une récursivité de l'arborescence (CHECK_DATAOBJECTPACKAGE.CHECK_MANIFEST.MODIFY_PARENT_EXISTING_UNIT_UNAUTHORIZED.KO = Le rattachement n'a pas été effectué : impossibilité de rattacher une unité archivistique existante à une unité archivistique parente)
- Cas 10 : une ou plusieurs balises de rattachement vers un groupe d'objets techniques existant déclarent autre chose que le GUID d'un groupe d'objets techniques existant (CHECK_DATAOBJECTPACKAGE.CHECK_MANIFEST.EXISTING_OG_NOT_DECLARED.KO = Une unité archivistique déclare un objet à la place du groupe d'objets correspondant)
- Cas 11 : une récursivité a été détectée dans l'arborescence des unités archivistiques (CHECK_DATAOBJECTPACKAGE.CHECK_MANIFEST.CHECK_MANIFEST_LOOP.KO = Le bordereau de transfert présente une récursivité dans l'arborescence de ses unités archivistiques)
- Cas 12 : il y a un problème d'encodage ou des objets référencent directement des unités archivistiques (CHECK_DATAOBJECTPACKAGE.CHECK_MANIFEST.KO = Échec du contrôle de cohérence du bordereau de transfert)
- Cas 13 : présence attendue d'un objet de type Master: Binary ou physical (CHECK_DATAOBJECTPACKAGE.CHECK_MANIFEST.MASTER_MANDATORY_REQUIRED.KO = Absence d'un BinaryMaster ou PhysicalMaster dans le groupe d'objet)
- Cas 14 : le contrat d'entrée n'autorise pas un ou plusieurs usages d'objets (CHECK_DATAOBJECTPACKAGE.CHECK_MANIFEST.ATTACHMENT_OBJECTGROUP.KO = Le contrat d'entrée n'autorise pas le rattachement d'un objet à un groupe d'objets existant)
- Cas 15 : il y a une donnée malformatée (CHECK_DATAOBJECTPACKAGE.CHECK_MANIFEST_MALFORMED_DATA.KO = Le bordereau de transfert possède une donnée malformée)
- FATAL : une erreur technique est survenue lors de la vérification de la cohérence du bordereau, par exemple les journaux du cycle de vie n'ont pu être créés (CHECK_MANIFEST.FATAL = Erreur technique lors du contrôle de cohérence du bordereau de transfert)

4.3.12. Vérification de la cohérence entre objets, groupes d'objets et unités archivistiques CHECK_DATAOBJECTPACKAGE.CHECK_CONSISTENCY (CheckObjectUnitConsistencyActionHandler.java)

- **Règle** : traitement consistant à vérifier que chaque objet ou groupe d'objets est référencé par une unité archivistique, à rattacher à un groupe d'objets les objets sans groupe d'objets mais référencés par une unité archivistique, à créer la table de concordance (MAP) entre les identifiants des objets et des unités archivistiques du SIP et à générer leurs identifiants pérennes dans la solution logicielle Vitam (GUID)
- **Type** : bloquant
- **Statuts** :
 - OK : aucun objet ou groupe d'objets n'est orphelin (c'est-à-dire non référencé par une unité archivistique) et tous les objets sont rattachés à un groupe d'objets. La table de concordance est créée et les identifiants des objets et unités archivistiques ont été générés (CHECK_DATAOBJECTPACKAGE.CHECK_CONSISTENCY.OK = Succès de la vérification de la cohérence entre objets, groupes d'objets et unités archivistiques)
 - KO : au moins un objet ou groupe d'objets est orphelin (c'est-à-dire non référencé par une unité archivistique) (CHECK_DATAOBJECTPACKAGE.CHECK_CONSISTENCY.KO = Échec de la vérification de la cohérence entre objets, groupes d'objets et unités archivistiques)
 - FATAL : une erreur technique est survenue lors de la vérification de la cohérence entre objets, groupes d'objets et unités archivistiques (CHECK_DATAOBJECTPACKAGE.CHECK_CONSISTENCY.FATAL = Erreur technique lors de la vérification de la cohérence entre objets, groupes d'objets et unités archivistiques)

4.3.13. Vérification du rattachement à un groupe d'objets ou une unité archivistique entrés sans erreur CHECK_ATTACHEMENT

(CheckAttachementActionHandler.java)

- **Règle** : tâche consistant à vérifier le rattachement à des groupes d'objets techniques et des unités archivistiques entrés sans erreur dans le système.
- **Type** : bloquant
- **Statuts** :
 - OK : la vérification des objets et groupes d'objets à été effectué avec succès (CHECK_ATTACHEMENT.OK=Succès de la vérification du rattachement entre objets, groupes d'objets et unités archivistiques existantes et les nouveaux)
 - KO : la vérification des objets et groupes d'objets n'a pu être effectuée, car le groupe d'objets techniques ou l'unité archivistique devant faire l'objet d'un rattachement dans le système sont entrés en erreur (CHECK_ATTACHEMENT.KO=Échec de la vérification du rattachement entre objets, groupes d'objets et unités archivistiques existantes et les nouveaux)
 - FATAL : une erreur technique est survenue lors de la vérification des objets et groupes d'objets (CHECK_ATTACHEMENT.FATAL=Erreur technique lors de la vérification du rattachement entre objets, groupes d'objets et unités archivistiques existantes et les nouveaux)

4.4. Processus de contrôle et traitement des objets (STP_OG_CHECK_AND_TRANSFORME)

4.4.1. Vérification de l'intégrité des objets CHECK_DIGEST (CheckConformityActionPlugin.java)

- **Règle** : tâche consistant à vérifier la cohérence entre l'empreinte de l'objet binaire calculée par la solution logicielle Vitam et celle déclarée dans le bordereau de transfert. Si l'empreinte déclarée dans le bordereau de transfert n'a pas été calculée avec l'algorithme SHA-512, alors l'empreinte est recalculée avec cet algorithme. Elle sera alors enregistrée dans la solution logicielle Vitam.

Algorithmes autorisés en entrée : MD5, SHA-1, SHA-256, SHA-512

- **Type** : bloquant
- **Statuts** :
 - OK : tous les objets binaires reçus sont identiques aux objets binaires attendus. Tous les objets binaires disposent désormais d'une empreinte calculée avec l'algorithme SHA-512 (CHECK_DIGEST.OK = Succès de la vérification de l'empreinte des objets)
 - KO :
 - Cas 1 : au moins un objet reçu n'a pas d'empreinte dans le bordereau (CHECK_DIGEST.EMPTY.KO = Échec lors de la vérification de l'empreinte des objets : Il existe au moins un objet dont l'empreinte est absente dans le bordereau de transfert)
 - Cas 2 : au moins une empreinte d'un objet reçu n'est pas conforme à son empreinte dans le bordereau (CHECK_DIGEST.INVALID.KO = Échec lors de la vérification de l'empreinte des objets : Il existe au moins un objet dont l'empreinte est invalide dans le bordereau de transfert)
 - Cas 3 : le SIP soumis à la solution logicielle Vitam contient à la fois le cas 1 et le cas 2 (CHECK_DIGEST.KO = Échec de la vérification de l'empreinte des objets)
 - FATAL : une erreur technique est survenue lors de la vérification de l'intégrité des objets binaires, par exemple lorsque l'algorithme est inconnu (CHECK_DIGEST.FATAL = Erreur technique lors de la vérification de l'empreinte des objets)

4.4.2. Calcul de la taille des fichiers CHECK_OBJECT_SIZE

(CheckObjectSizeActionPlugin.java)

- **Règle** : tâche vérifier la taille de chaque objet binaire présent dans le SIP, à vérifier que la taille des objets correspond à la taille des fichiers renseignée pour chacun d'eux dans le manifeste. Le poids des fichiers est calculé en octets et comparé à la taille renseignée dans le manifeste. En cas d'incohérence entre la déclaration dans le manifeste et la taille du fichier, le SIP sera accepté, générant un avertissement. La solution logicielle Vitam se servira alors des informations qu'elle a identifiées et non de celles fournies dans le SIP
- **Type** : bloquant
- **Statuts** :
 - OK (CHECK_OBJECT_SIZE.OK = Succès de la vérification de la taille des objets) :
 - la taille des fichiers correspond à celle qui est renseignée dans le manifeste et aucune incohérence n'a été trouvée ;
 - la taille des fichiers n'est pas renseignée dans le manifeste et la solution logicielle Vitam enregistre la taille des fichiers qu'elle a calculée.
 - WARNING (CHECK_OBJECT_SIZE.WARNING = Avertissement de la vérification de la taille des objets) : au moins un objet reçu a une taille renseignée dans le manifeste qui n'est pas identique à celle des fichiers numériques.
 - FATAL : une erreur technique est survenue lors de la vérification de la taille des objets (CHECK_OBJECT_SIZE.FATAL = Erreur technique lors de la vérification de la taille des objets)

4.4.3. Identification des formats OG_OBJECTS_FORMAT_CHECK (FormatIdentificationActionPlugin.java)

- **Règle** : tâche consistant à identifier le format de chaque objet binaire présent dans le SIP, à vérifier que le format identifié des objets correspond à la liste des formats acceptés dans le contrat d'entrée et à vérifier que le format identifié des objets est référencé dans le référentiel des formats de la solution logicielle Vitam. Cette action met en œuvre un outil d'identification prenant l'objet en entrée et fournissant des informations de format en sortie. Ces informations sont comparées avec les formats enregistrés dans le référentiel des formats interne à la solution logicielle Vitam et avec celles déclarées dans le bordereau de transfert. En cas d'incohérence entre la déclaration dans le SIP et le format identifié, le SIP sera accepté, générant un avertissement. La solution logicielle Vitam se servira alors des informations qu'elle a identifiées et non de celles fournies dans le SIP
- **Type** : bloquant
- **Statuts** :
 - OK : l'identification s'est bien passée, les formats ont tous été identifiés, sont référencés dans le référentiel interne et sont soit dans la liste des formats autorisés du contrat d'entrée, soit ce contrat autorise tous les formats. De plus les informations de formats trouvées par la solution logicielle Vitam sont cohérentes avec celles déclarées dans le manifeste (OG_OBJECTS_FORMAT_CHECK.OK = Succès de la vérification des formats)
 - KO :
 - Cas 1 : au moins un objet reçu a un format qui n'a pas été trouvé et le contrat d'entrée utilisé interdit le versement d'objets aux formats non identifiés (OG_OBJECTS_FORMAT_CHECK.KO = Échec de l'identification des formats)
 - Cas 2 : au moins un objet reçu a un format qui n'est pas référencé dans le référentiel interne (OG_OBJECTS_FORMAT_CHECK.UNCHARTED.KO = Échec de l'identification des formats, le format de ou des objet(s) est identifié mais est inconnu du référentiel des formats)
 - Cas 3 : au moins objet reçu possède un format qui n'est pas indiqué dans la liste des formats

autorisés du contrat d'entrée du SIP

(OG_OBJECTS_FORMAT_CHECK.REJECTED_FORMAT.KO = Échec de l'identification des formats : le contrat d'entrée interdit le versement d'objet au format inconnu et le SIP versé contient au moins un objet au format inconnu, ou bien le SIP contient un format interdit par le contrat d'entrée)

- WARNING :
 - Cas 1 : l'identification s'est bien passée, les formats identifiés sont référencés dans le référentiel interne mais les informations ne sont pas cohérentes avec celles déclarées dans le manifeste (OG_OBJECTS_FORMAT_CHECK.WARNING = Avertissement lors de l'identification des formats)
 - Cas 2 : au moins un objet reçu a un format qui n'a pas été trouvé mais le contrat d'entrée utilisé autorise le versement d'objets aux formats non identifiés. Dans ce cas Vitam remplace le champ « FormatId » du manifest.xml par le mot « unknown » (OG_OBJECTS_FORMAT_CHECK.WARNING = Avertissement lors de l'identification des formats)
- FATAL : une erreur technique est survenue lors de l'identification des formats (OG_OBJECTS_FORMAT_CHECK.FATAL = Erreur technique lors de l'identification des formats)

4.5.Processus de contrôle et traitement des unités archivistiques (STP_UNIT_CHECK_AND_PROCESS)

4.5.1.Vérification globale de l'unité archivistique CHECK_UNIT_SCHEMA (CheckArchiveUnitSchemaActionPlugin.java)

- **Règle** : tâche consistant à :
 - contrôler que la valeur des champs déclarés dans le bordereau de transfert est d'un type conforme à celui déclaré dans l'ontologie ;
 - contrôler la validité des champs de l'unité archivistique par rapport au schéma prédéfini dans la solution logicielle Vitam. Par exemple, les champs obligatoires, comme les titres des unités archivistiques, ne doivent pas être vides. Lorsque le manifeste déclare une personne (Person) et non une société (Entity), alors au moins un champ entre « Firstname » et « Birthname » est obligatoire,
 - vérifier que la date de fin est bien supérieure ou égale à la date de début de l'unité archivistique.
- **Type** : bloquant
- **Statuts** :
 - OK : tous les champs de l'unité archivistique sont conformes à ce qui est attendu (CHECK_UNIT_SCHEMA.OK = Succès de la vérification globale de l'unité archivistique)
 - KO :
 - Cas 1 : au moins un champ d'une unité archivistique déclare un champ dont la valeur n'est pas conforme au type défini dans l'ontologie (CHECK_UNIT_SCHEMA.KO = Échec de la vérification globale de l'unité archivistique)
 - Cas 2 : au moins un champ d'une unité archivistique dont le schéma n'est pas conforme par rapport au schéma par défaut des unités archivistiques défini pour la solution logicielle Vitam. (CHECK_UNIT_SCHEMA.INVALID_UNIT.KO = Échec lors de la vérification globale de l'unité archivistique : champs non conformes)
 - Cas 3 : au moins un champ obligatoire d'une unité archivistique est vide (CHECK_UNIT_SCHEMA.EMPTY_REQUIRED_FIELD.KO = Échec lors de la vérification globale de l'unité archivistique : champs obligatoires vides)
 - Cas 4 : au moins un champ date d'une unité archivistique est supérieur à 9000 ou la date de fin des dates extrêmes est strictement inférieure à la date de début

(CHECK_UNIT_SCHEMA.RULE_DATE_THRESHOLD.KO = Échec du calcul des dates d'échéance, la date ne peut être gérée)

- Cas 5 : au moins un champ date d'une unité archivistique déclare une valeur non conforme au type attendu (CHECK_UNIT_SCHEMA.RULE_DATE_FORMAT.KO = Échec du calcul des dates d'échéance, la date ne peut être gérée)
- Cas 6 : au moins une valeur de l'unité archivistique n'est pas conforme à son schéma en raison d'un problème de cohérence entre champs. Par exemple, la valeur contenue dans le champ « StartDate » est postérieure à la date définie dans la « EndDate » (CHECK_UNIT_SCHEMA.CONSISTENCY.KO = Au moins une unité archivistique n'est pas conforme à son schéma en raison d'un problème de cohérence entre champs)
- FATAL : une erreur technique est survenue lors de la vérification de l'unité archivistique (CHECK_UNIT_SCHEMA.FATAL = Erreur technique lors de la vérification globale de l'unité archivistique)

4.5.2. Vérification du profil d'unité archivistique CHECK_ARCHIVE_UNIT_PROFILE (CheckArchiveUnitProfileActionPlugin.java)

- **Règle** : tâche consistant à vérifier la conformité des unités archivistiques au schéma défini dans les profils d'unités archivistiques qu'elles déclarent dans la balise « ArchiveUnitProfile ». Les profils d'unités archivistiques référencés doivent être en état « Actif » et ne pas avoir un schéma de contrôle vide
- **Type** : non bloquant
- **Statuts** :
 - OK : les unités archivistiques déclarant un profil d'unité archivistique de référence sont bien conformes au schéma décrit dans le profil d'unité archivistique, et le profil et le schéma existent bien dans le système en état actif (CHECK_ARCHIVE_UNIT_PROFILE.OK = Succès de la vérification de la conformité au profil d'unité archivistique)
 - KO :
 - Cas 1 : au moins une unité archivistique n'est pas conforme au schéma décrit dans le profil d'unité archivistique associé (CHECK_ARCHIVE_UNIT_PROFILE.KO = Échec de la vérification de la conformité au profil d'unité archivistique)
 - Cas 2 : au moins une unité archivistique qui déclare un lien avec un profil d'unité archivistique inexistant dans le référentiel (CHECK_ARCHIVE_UNIT_PROFILE.PROFILE_NOT_FOUND.KO = Échec de la vérification de la conformité au profil d'unité archivistique : profil d'unité archivistique non trouvé)
 - Cas 3 : au moins une unité archivistique qui n'est pas conforme au schéma décrit dans le profil d'unité archivistique associé (CHECK_ARCHIVE_UNIT_PROFILE.INVALID_UNIT.KO = Échec de la vérification de la conformité au profil d'unité archivistique : champs non conformes)
 - Cas 4 : le profil d'unité archivistique cité dans le référentiel est mal formaté (CHECK_ARCHIVE_UNIT_PROFILE.INVALID_AU_PROFILE.KO = Échec de la vérification de la conformité aux documents type : profil d'unité archivistique non conforme)
 - Cas 5 : le profil d'unité archivistique est dans l'état « inactif » (CHECK_ARCHIVE_UNIT_PROFILE.INACTIVE_STATUS.KO = Échec de la vérification de la conformité aux documents type : profil d'unité archivistique au statut « inactif »)
 - Cas 6 : le profil d'unité archivistique possède un schéma de contrôle qui est vide (CHECK_ARCHIVE_UNIT_PROFILE.EMPTY_CONTROL_SCHEMA.KO = Échec de la vérification de la conformité aux documents type : schéma de contrôle du profil d'unité archivistique vide)

4.5.3. Vérification du niveau de classification CHECK_CLASSIFICATION_LEVEL (CheckClassificationLevelActionPlugin.java)

- **Règle** : tâche consistant à vérifier les niveaux de classification associés, s'il en existe, aux unités archivistiques. Ces niveaux doivent exister dans la liste des niveaux de classification autorisés par la

plateforme (paramètre configuré dans la configuration des workers). Pour les unités archivistiques sans niveau de classification, la vérification contrôle que la plateforme autorise le versement d'unités archivistiques ne déclarant pas de niveau de classification.

- **Type** : bloquant
- **Statuts** :
 - OK : les unités archivistiques versées ont un niveau de classification autorisé par la plateforme. S'il existe dans le SIP des unités archivistiques sans niveau de classification, il faut que la plateforme autorise le versement d'unités archivistiques sans niveau de classification. (CHECK_CLASSIFICATION_LEVEL.OK = Succès de la vérification du niveau de classification)
 - KO : au moins une unité archivistique du SIP possède un niveau de classification qui n'est pas un niveau de classification autorisé par la plateforme, ou une unité archivistique n'a pas de niveau de classification alors que la plateforme requiert que toutes les unités archivistiques possèdent un niveau de classification. (CHECK_CLASSIFICATION_LEVEL.KO = Échec de la vérification du niveau de classification, non autorisé par la plateforme : le bordereau de transfert déclare un niveau de classification non autorisé par la plateforme)
 - FATAL : une erreur technique est survenue lors de la vérification des niveaux de classification (CHECK_CLASSIFICATION_LEVEL.FATAL = Erreur technique lors de la vérification du niveau de classification)

4.5.4. Application des règles de gestion et calcul des dates d'échéances UNITS_RULES_COMPUTE (UnitsRulesComputePlugin.java)

- **Règle** : tâche consistant à calculer les dates d'échéances des unités archivistiques du SIP. Pour les unités racines, c'est-à-dire les unités déclarées dans le SIP et n'ayant aucun parent dans l'arborescence, la solution logicielle Vitam utilise les règles de gestion incluses dans le bloc Management de chacune de ces unités ainsi que celles présentes dans le bloc ManagementMetadata. La solution logicielle Vitam effectue également ce calcul pour les autres unités archivistiques du SIP possédant des règles de gestion déclarées dans leurs balises Management, sans prendre en compte le ManagementMetadata. Le référentiel utilisé pour ces calculs est le référentiel des règles de gestion de la solution logicielle Vitam.
- **Type** : bloquant
- **Statuts** :
 - OK : les règles de gestion sont référencées dans le référentiel interne et ont été appliquées avec succès (UNITS_RULES_COMPUTE.OK = Succès de l'application des règles de gestion et du calcul des dates d'échéance)
 - KO :
 - Cas 1 : au moins une unité archivistique déclare un champ dont la valeur n'est pas conforme à celle attendue (UNITS_RULES_COMPUTE.KO=Au moins une unité archivistique déclare un champ dont la valeur n'est pas conforme à celle attendue)
 - Cas 2 : au moins une unité archivistique déclare une règle non référencée dans le référentiel interne (UNITS_RULES_COMPUTE.UNKNOWN.KO = Échec lors de l'application des règles de gestion et du calcul des dates d'échéance : règle de gestion inconnue)
 - Cas 3 : au moins une unité archivistique déclare une règle non cohérente avec sa catégorie (UNITS_RULES_COMPUTE.CONSISTENCY.KO = Échec lors de l'application des règles de gestion et du calcul des dates d'échéance : Au moins une unité archivistique déclare une règle non cohérente avec sa catégorie)
 - Cas 4 : au moins une unité archivistique déclare dans le champ RefNonRuleId une règle non cohérente avec sa catégorie (UNITS_RULES_COMPUTE.REF_INCONSISTENCY.KO = Échec lors de l'application des règles de gestion et du calcul des dates d'échéance : exclusion d'héritage incohérente)
 - FATAL : une erreur technique est survenue lors du calcul des dates d'échéances (UNITS_RULES_COMPUTE.FATAL = Erreur technique lors de l'application des règles de gestion et du calcul des dates d'échéance)

4.6. Processus de vérification préalable à la prise en charge (STP_STORAGE_AVAILABILITY_CHECK)

4.6.1. Vérification de la disponibilité de toutes les offres de stockage STORAGE_AVAILABILITY_CHECK (CheckStorageAvailabilityActionHandler.java)

- **Règle** : tâche consistant à vérifier la disponibilité des offres de stockage et de l'espace disponible pour y stocker le contenu du SIP compte tenu de la taille des objets à stocker
- **Type** : bloquant
- **Statuts** :
 - OK : les offres de stockage sont accessibles et disposent d'assez d'espace pour stocker le contenu du SIP (STORAGE_AVAILABILITY_CHECK.OK = Succès de la vérification de la disponibilité de toutes les offres de stockage)
 - KO :
 - Cas 1 : les offres de stockage ne sont pas disponibles (STORAGE_AVAILABILITY_CHECK.STORAGE_OFFER_KO_UNAVAILABLE.KO = Échec de la vérification de la disponibilité d'au moins une offre de stockage)
 - Cas 2 : les offres ne disposent pas d'assez d'espace pour stocker le contenu du SIP (STORAGE_AVAILABILITY_CHECK.STORAGE_OFFER_SPACE_KO.KO = Échec de la vérification de l'espace disponible)
 - FATAL : une erreur technique est survenue lors de la vérification de la disponibilité de l'offre de stockage (STORAGE_AVAILABILITY_CHECK.FATAL = Erreur technique lors de la vérification de la disponibilité d'au moins une offre de stockage)

La tâche Check_Availability_Check contient le traitement suivant :

4.6.2. Vérification de la disponibilité de l'offre de stockage STORAGE_AVAILABILITY_CHECK.STORAGE_AVAILABILITY_CHECK (CheckStorageAvailabilityActionHandler.java)

- **Règle** : traitement consistant à vérifier la disponibilité de l'offre de stockage et de l'espace disponible pour y stocker le contenu du SIP compte tenu de la taille des objets à stocker
- **Type** : bloquant
- **Statuts** :
 - OK : l'offre de stockage est accessible et dispose d'assez d'espace pour stocker le contenu du SIP (STORAGE_AVAILABILITY_CHECK.STORAGE_AVAILABILITY_CHECK.OK = Succès de la vérification de la disponibilité de l'offre de stockage)
 - KO :
 - Cas 1 : l'offre de stockage n'est pas disponible (STORAGE_AVAILABILITY_CHECK.STORAGE_AVAILABILITY_CHECK.STORAGE_OFFER_KO_UNAVAILABLE.KO = L'offre de stockage n'est pas disponible)
 - Cas 2 : l'offre de stockage ne dispose pas d'assez d'espace pour stocker le contenu du SIP (STORAGE_AVAILABILITY_CHECK.STORAGE_AVAILABILITY_CHECK.STORAGE_OFFER_SPACE_KO.KO = Disponibilité de l'offre de stockage insuffisante)
 - FATAL : une erreur technique est survenue lors de la vérification de la disponibilité de l'offre de stockage (STORAGE_AVAILABILITY_CHECK.STORAGE_AVAILABILITY_CHECK.FATAL = Erreur technique lors de la vérification de la disponibilité de l'offre de stockage)

4.7. Processus d'écriture et indexation des objets et groupes d'objets

(STP_OBJ_STORING)

4.7.1.Écriture des objets sur l'offre de stockage OBJ_STORAGE (StoreObjectActionHandler.java)

- **Règle** : tâche consistant à écrire les objets contenus dans le SIP sur les offres de stockage en fonction de la stratégie de stockage applicable
- **Type** : Bloquant
- **Statuts** :
 - OK : tous les objets binaires contenus dans le SIP ont été écrits sur les offres de stockage (OBJ_STORAGE.OK = Succès de l'écriture des objets et des groupes d'objets sur les offres de stockage)
 - KO : au moins un des objets binaires contenus dans le SIP n'a pas pu être écrit sur les offres de stockage (OBJ_STORAGE.KO = Échec de l'écriture des objets et des groupes d'objets sur les offres de stockage)
 - WARNING : le SIP ne contient pas d'objet (OBJECTS_LIST_EMPTY.WARNING = Avertissement lors de l'établissement de la liste des objets : il n'y a pas d'objet pour cette étape)
 - FATAL : une erreur technique est survenue lors de l'écriture des objets binaires sur les offres de stockage (OBJ_STORAGE.FATAL = Erreur technique lors de l'écriture des objets et des groupes d'objets sur les offres de stockage)

4.7.2.Indexation des métadonnées des groupes d'objets et objets OG_METADATA_INDEXATION (IndexObjectGroupActionPlugin.java)

- **Règle** : tâche consistant à indexer les métadonnées des groupes d'objets et objets dans les bases internes de la solution logicielle Vitam, comme la taille des objets, les métadonnées liées aux formats (Type MIME, PUID, etc.), l'empreinte des objets, etc.
- **Type** : bloquant
- **Statuts** :
 - OK : les métadonnées des groupes d'objets et objets ont été indexées avec succès (OG_METADATA_INDEXATION.OK = Succès de l'indexation des métadonnées des objets et des groupes d'objets)
 - KO : au moins une des métadonnées des groupes d'objets et objets n'a pas été indexée (OG_METADATA_INDEXATION.KO = Échec de l'indexation des métadonnées des objets et des groupes d'objets)
 - FATAL : une erreur technique est survenue lors de l'indexation des métadonnées des groupes d'objets (OG_METADATA_INDEXATION.FATAL = Erreur technique lors de l'indexation des métadonnées des objets et des groupes d'objets)

4.8.Processus d'indexation des unités archivistiques (STP_UNIT_METADATA)

Indexation des métadonnées des unités archivistiques UNIT_METADATA_INDEXATION (IndexUnitActionPlugin.java)

- **Règle** : tâche consistant à indexer les métadonnées des unités archivistiques dans les bases internes de la solution logicielle Vitam, c'est-à-dire le titre des unités, leurs descriptions, leurs dates extrêmes, etc.
- **Type** : bloquant
- **Statuts** :
 - OK : les métadonnées des unités archivistiques ont été indexées avec succès (UNIT_METADATA_INDEXATION.OK = Succès de l'indexation des métadonnées de l'unité archivistique)

- KO : au moins une des métadonnées des unités archivistiques n'a pas été indexée (UNIT_METADATA_INDEXATION.KO = Échec de l'indexation des métadonnées de l'unité archivistique)
- FATAL : une erreur technique est survenue lors de l'indexation des métadonnées des unités archivistiques (UNIT_METADATA_INDEXATION.FATAL = Erreur technique lors de l'indexation des métadonnées de l'unité archivistique)

4.9. Processus d'enregistrement et écriture des métadonnées des objets et groupes d'objets (STP_OG_STORING)

4.9.1. Enregistrement des journaux du cycle de vie des groupes d'objets COMMIT_LIFE_CYCLE_OBJECT_GROUP (CommitLifeCycleObjectGroupActionHandler.java)

- **Règle** : tâche consistant à sécuriser en base les journaux du cycle de vie des groupes d'objets. Avant cette étape, les journaux du cycle de vie des groupes d'objets sont dans une collection temporaire afin de garder une cohérence entre les métadonnées indexées et les journaux lors d'une entrée en succès ou en échec, il n'y a pas d'évènement créé dans le journal du cycle de vie.
- **Type** : bloquant
- **Statuts** :
 - OK : la sécurisation des journaux du cycle de vie s'est correctement déroulée (COMMIT_LIFE_CYCLE_OBJECT_GROUP.OK = Succès de l'enregistrement des journaux du cycle de vie des groupes d'objets)
 - FATAL : une erreur technique est survenue lors de la sécurisation du journal du cycle de vie (COMMIT_LIFE_CYCLE_OBJECT_GROUP.FATAL = Erreur technique lors de l'enregistrement des journaux du cycle de vie des groupes d'objets)

4.9.2. Écriture des métadonnées du groupe d'objets et objets sur l'offre de stockage OG_METADATA_STORAGE (StoreMetaDataObjectGroupActionPlugin)

- **Règle** : tâche consistant à sauvegarder les métadonnées liées aux groupes d'objets ainsi que leurs journaux de cycle de vie sur les offres de stockage en fonction de la stratégie de stockage
- **Type** : bloquant
- **Statuts** :
 - OK : les métadonnées des groupes d'objets et objets ont été sauvegardées avec succès (OG_METADATA_STORAGE.OK = Succès de l'écriture des métadonnées des objets et groupes d'objets sur l'offre de stockage)
 - KO : les métadonnées des groupes d'objets et objets n'ont pas été sauvegardées (OG_METADATA_STORAGE.KO = Échec de l'écriture des métadonnées des objets et groupes d'objets sur l'offre de stockage)
 - FATAL : une erreur technique est survenue lors de l'écriture des métadonnées du groupe d'objets sur les offres de stockage (OG_METADATA_STORAGE.FATAL = Erreur technique lors de l'écriture des métadonnées du groupe d'objets sur les offres de stockage)

4.10. Processus d'enregistrement et écriture des unités archivistiques (STP_UNIT_STORING)

4.10.1. Enregistrement du journal du cycle de vie des unités archivistiques **COMMIT_LIFE_CYCLE_UNIT (AccessInternalModuleImpl.java)**

- **Règle** : tâche consistant à sécuriser en base les journaux du cycle de vie des unités archivistiques. Avant cette étape, les journaux du cycle de vie des unités archivistiques sont dans une collection temporaire afin de garder une cohérence entre les métadonnées indexées et les journaux lors d'une entrée en succès ou en échec.
- **Type** : bloquant
- **Statuts** :
 - OK : la sécurisation des journaux du cycle de vie s'est correctement déroulée (COMMIT_LIFE_CYCLE_UNIT.OK = Succès de l'enregistrement des journaux du cycle de vie des unités archivistiques)
 - FATAL : une erreur technique est survenue lors de la sécurisation des journaux du cycle de vie (COMMIT_LIFE_CYCLE_UNIT.FATAL = Erreur technique lors de l'enregistrement des journaux du cycle de vie des unités archivistiques)

4.10.2. Écriture des métadonnées de l'unité archivistique sur l'offre de stockage **UNIT_METADATA_STORAGE (AccessInternalModuleImpl.java)**

- **Règle** : tâche consistant à sauvegarder les métadonnées et des journaux de cycle de vie des unités archivistiques sur les offres de stockage en fonction de la stratégie de stockage. Pas d'évènements stockés dans le journal de cycle de vie
- **Type** : bloquant
- **Statuts** :
 - OK : l'écriture des métadonnées de l'unité archivistique sur les offres de stockage s'est correctement déroulée (UNIT_METADATA_STORAGE.OK = Succès de l'écriture des métadonnées de l'unité archivistique sur les offres de stockage)
 - KO : l'écriture des métadonnées de l'unité archivistique sur les offres de stockage n'a pas été effectuée (UNIT_METADATA_STORAGE.KO = Échec de l'écriture des métadonnées de l'unité archivistique sur les offres de stockage)
 - FATAL : une erreur technique est survenue lors de la sécurisation du journal du cycle de vie (UNIT_METADATA_STORAGE.FATAL = Erreur technique lors de l'écriture des métadonnées de l'unité archivistique sur les offres de stockage)

4.11. Processus de mise à jour des groupes d'objets **(STP_UPDATE_OBJECT_GROUP)**

4.11.1. Mise à jour des groupes d'objets **OBJECT_GROUP_UPDATE** **(AccessInternalModuleImpl.java)**

- **Règle** : Tâche consistant à mettre à jour les groupes d'objet techniques
- **Type** : bloquant
 - OK=Succès de la mise à jour des groupes d'objets existants
 - KO=Échec lors de la mise à jour des groupes d'objets existants
 - FATAL=Erreur fatale lors de la mise à jour des groupes d'objets existants
 - WARNING=Avertissement lors de la mise à jour des groupes d'objets existants

4.11.2. Enregistrement du journal du cycle de vie des groupes d'objets **COMMIT_LIFE_CYCLE_OBJECT_GROUP** (CommitLifeCycleObjectGroupActionHandler.java)

- **Règle** : tâche consistant à sécuriser en base les journaux du cycle de vie des groupes d'objets. Avant cette étape, les journaux du cycle de vie des groupes d'objets sont dans une collection temporaire afin de garder une cohérence entre les métadonnées indexées et les journaux lors d'une entrée en succès ou en échec, et il n'y a pas d'évènement créée dans le journal du cycle de vie.
- **Type** : bloquant
- **Statuts** :
 - OK : la sécurisation des journaux du cycle de vie s'est correctement déroulée (COMMIT_LIFE_CYCLE_OBJECT_GROUP.OK = Succès de l'enregistrement des journaux du cycle de vie des groupes d'objets)
 - FATAL : une erreur technique est survenue lors de la sécurisation du journal du cycle de vie (COMMIT_LIFE_CYCLE_OBJECT_GROUP.FATAL = Erreur technique lors de l'enregistrement des journaux du cycle de vie des groupes d'objets)

4.11.3. Écriture des métadonnées des groupes d'objets sur l'offre de stockage **OG_METADATA_STORAGE** (StoreMetaDataObjectGroupActionPlugin)

- **Règle** : tâche consistant à sauvegarder les métadonnées liées aux groupes d'objets ainsi que leurs journaux de cycle de vie sur les offres de stockage en fonction de la stratégie de stockage
- **Type** : bloquant
- **Statuts** :
 - OK : les métadonnées des groupes d'objets et objets ont été sauvegardées avec succès (OG_METADATA_STORAGE.OK = Succès de l'écriture des métadonnées des objets et groupes d'objets sur l'offre de stockage)
 - KO : les métadonnées des groupes d'objets et objets n'ont pas été sauvegardées (OG_METADATA_STORAGE.KO = Échec de l'écriture des métadonnées des objets et groupes d'objets sur l'offre de stockage)
 - FATAL : une erreur technique est survenue lors de l'écriture des métadonnées du groupe d'objets sur les offres de stockage (OG_METADATA_STORAGE.FATAL = Erreur technique lors de l'écriture des métadonnées du groupe d'objets sur les offres de stockage)

4.12. Processus d'alimentation du registre des fonds **(STP_ACCESSION_REGISTRATION)**

Alimentation du registre des fonds ACCESSION_REGISTRATION

- **Règle** : tâche consistant à enregistrer dans le registre des fonds des informations concernant la nouvelle entrée (nombre d'objets, volumétrie...). Ces informations viennent s'ajouter aux informations existantes pour un même service producteur. Si aucune information n'existait préalablement, alors un nouveau document est créé dans la base de données concernant ce producteur. Une fois cette action d'ajout ou de mise à jour effectuée, la solution logicielle Vitam calcule et enregistre une information agrégée de l'état des stocks du service producteur concerné (dans la collection AccessionRegisterDetail).
- **Type** : bloquant
- **Statuts** :
 - OK : le registre des fonds est correctement alimenté (ACCESSION_REGISTRATION.OK = Succès de l'alimentation du registre des fonds)
 - KO : le registre des fonds n'a pas pu être alimenté (ACCESSION_REGISTRATION.KO = Échec de l'alimentation du registre des fonds)

- FATAL : une erreur technique est survenue lors de l'alimentation du registre des fonds (ACCESSION_REGISTRATION.FATAL = Erreur technique lors de l'alimentation du registre des fonds)

4.13. Processus de finalisation de l'entrée (STP_INGEST_FINALISATION)

4.13.1. Notification de la fin de l'opération d'entrée ATR_NOTIFICATION (TransferNotificationActionHandler.java)

- **Règle** : tâche consistant à générer la notification de réponse (ArchiveTransferReply ou ATR) une fois toutes les étapes passées, en succès, avertissement ou échec, puis écriture de cette notification dans l'offre de stockage et envoi au service versant
- **Type** : non bloquant
- **Statuts** :
 - OK : le message de réponse a été correctement généré, écrit sur l'offre de stockage et envoyé au service versant (ATR_NOTIFICATION.OK = Succès de la notification de la fin de l'opération d'entrée à l'opérateur de versement)
 - KO : le message de réponse n'a pas été correctement généré, écrit sur l'offre de stockage ou envoyé au service versant (ATR_NOTIFICATION.KO = Échec de la notification de la fin de l'opération d'entrée à l'opérateur de versement)
 - FATAL : une erreur technique est survenue lors de la notification de la fin de l'opération (ATR_NOTIFICATION.FATAL = Erreur technique lors de la notification de la fin de l'opération d'entrée à l'opérateur de versement)

4.13.2. Mise en cohérence des journaux du cycle de vie ROLL_BACK (RollBackActionHandler.java)

- **Règle** : Purge des collections temporaires des journaux du cycle de vie
- **Type** : bloquant
- **Statuts** :
 - OK : la purge s'est correctement déroulée (ROLL_BACK.OK = Succès de la mise en cohérence des journaux du cycle de vie)
 - FATAL : une erreur technique est survenue lors de la purge (ROLL_BACK.FATAL = Erreur technique lors de la mise en cohérence des journaux du cycle de vie)

4.14. Le cas du processus d'entrée « test à blanc »

Il est possible de procéder à un versement dit « à blanc », pour tester la conformité du SIP par rapport à la forme attendue par la solution logicielle Vitam sans pour autant le prendre en charge. Dans ce cas, le processus d'entrée à blanc diffère du processus d'entrée « classique » en ignorant un certain nombre d'étapes.

Les étapes non exécutées dans le processus d'entrée à blanc sont les suivantes :

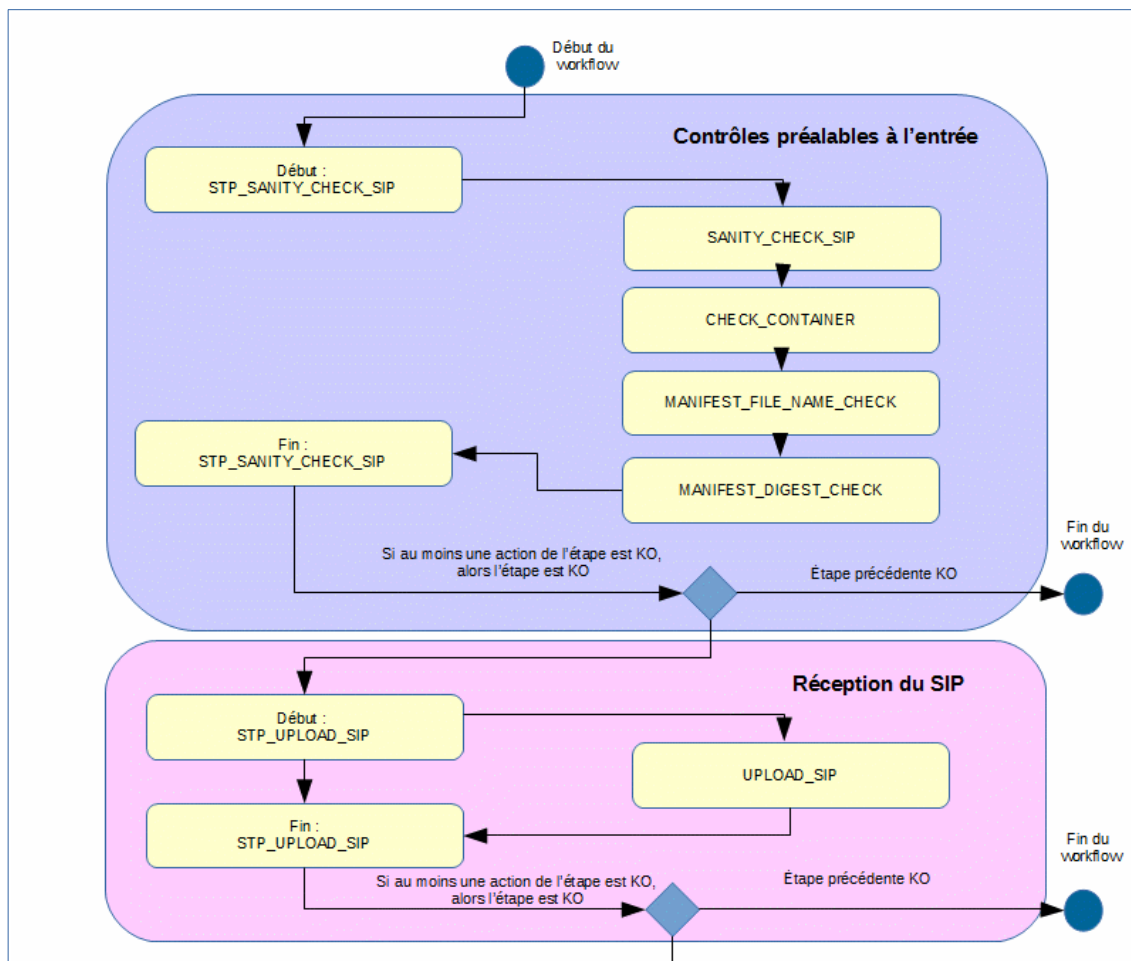
- Écriture et indexation des objets et groupes d'objets (STP_OBJ_STORING)
- Indexation des unités archivistiques (STP_UNIT_METADATA)
- Enregistrement et écriture des métadonnées des objets et groupes d'objets (STP_OG_STORING)
- Enregistrement et écriture des unités archivistiques (STP_UNIT_STORING)
- Rangement des métadonnées des objets (STP_UPDATE_OBJECT_GROUP)
- Alimentation du registre des fonds (STP_ACCESSION_REGISTRATION)

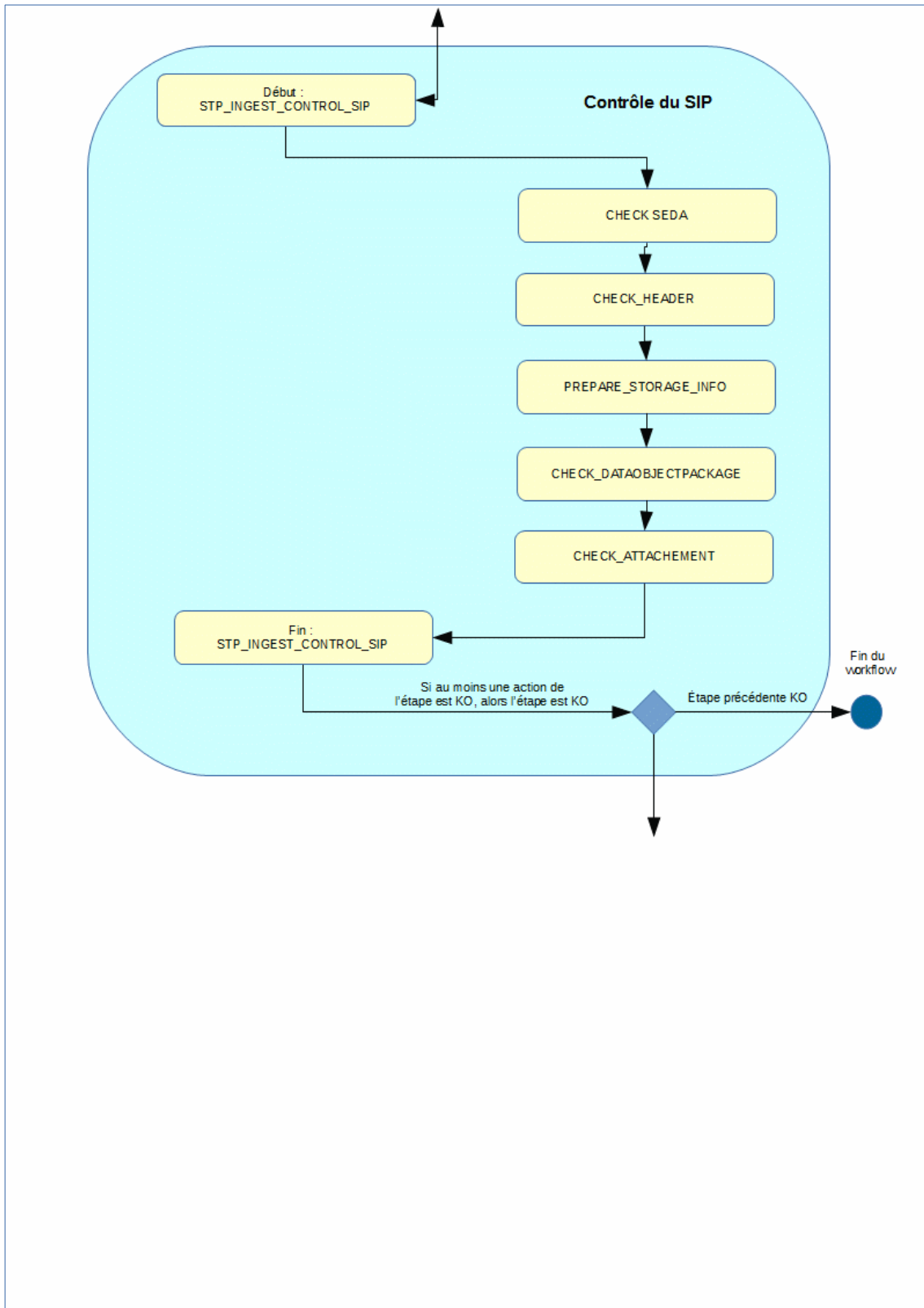
Les tâches et traitements relatifs à toutes ces étapes sont donc également ignorés.

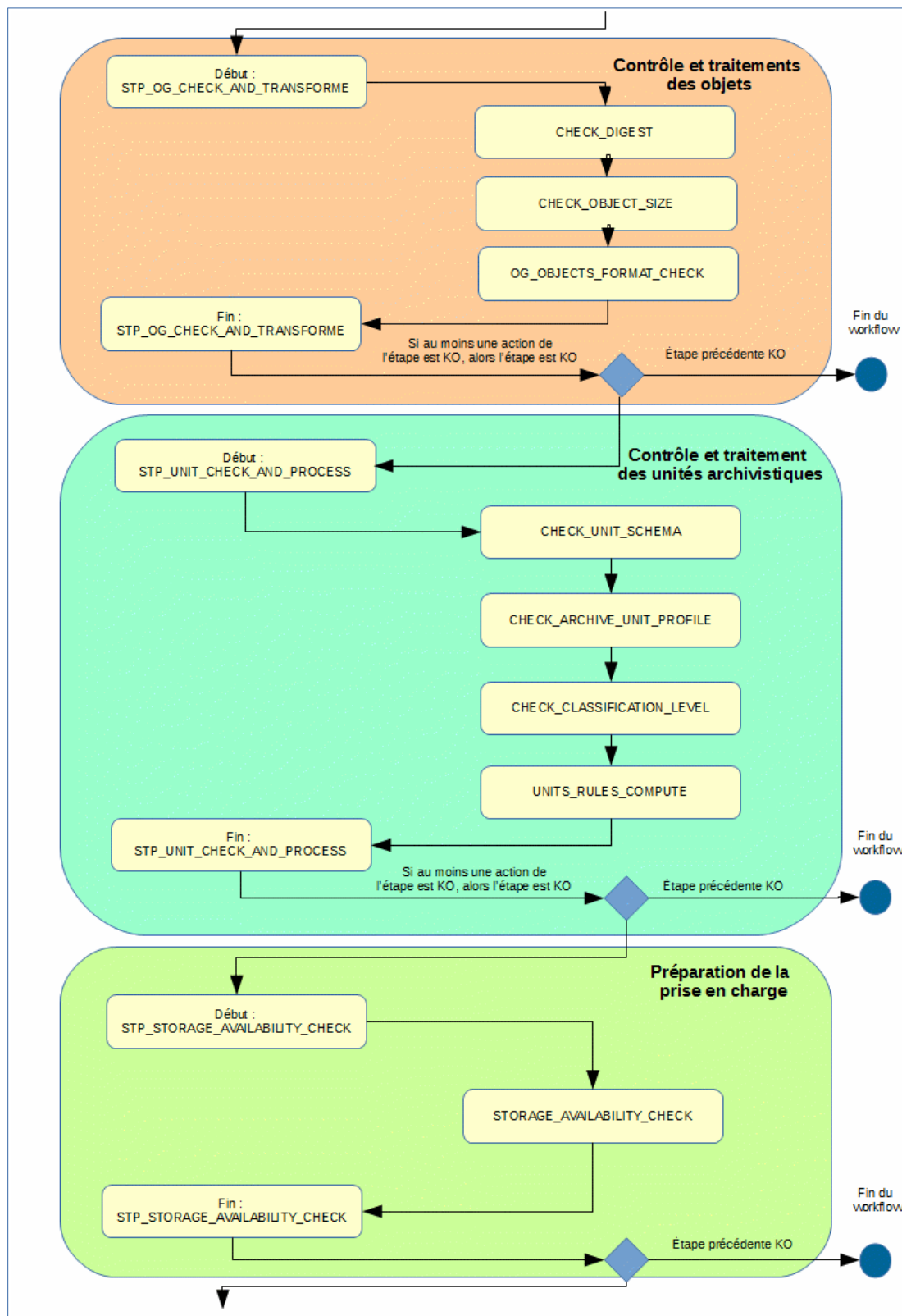
4.15. Structure du Workflow de l'entrée

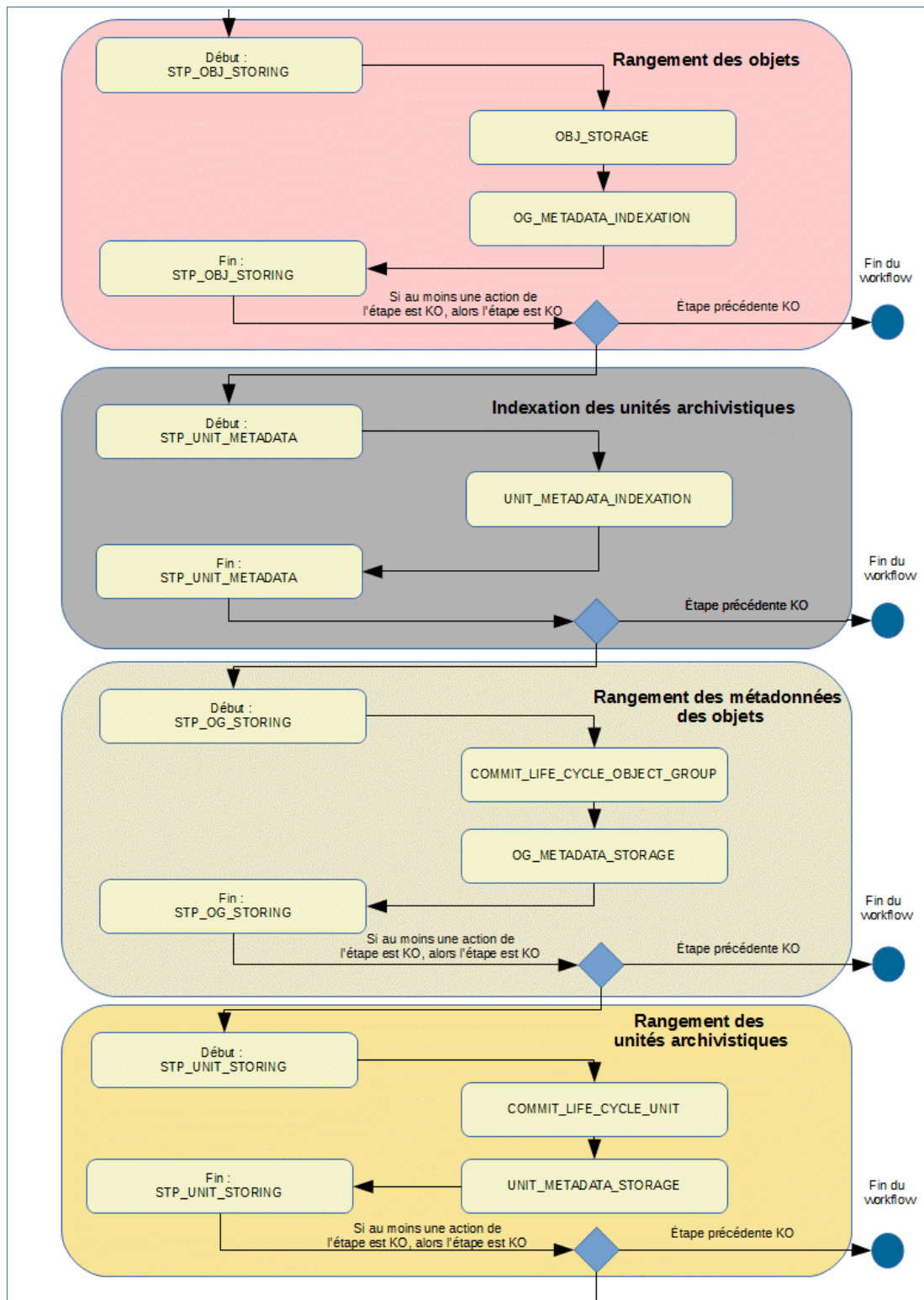
Le workflow mis en place dans la solution logicielle Vitam est défini dans l'unique fichier « *DefaultIngestWorkflow.json* ». Ce fichier est disponible dans */sources/processing/processing-management/src/main/resources/workflows*. Il décrit le processus d'entrée (hors Ingest externe) pour entrer un SIP, indexer les métadonnées et stocker les objets contenus dans le SIP.

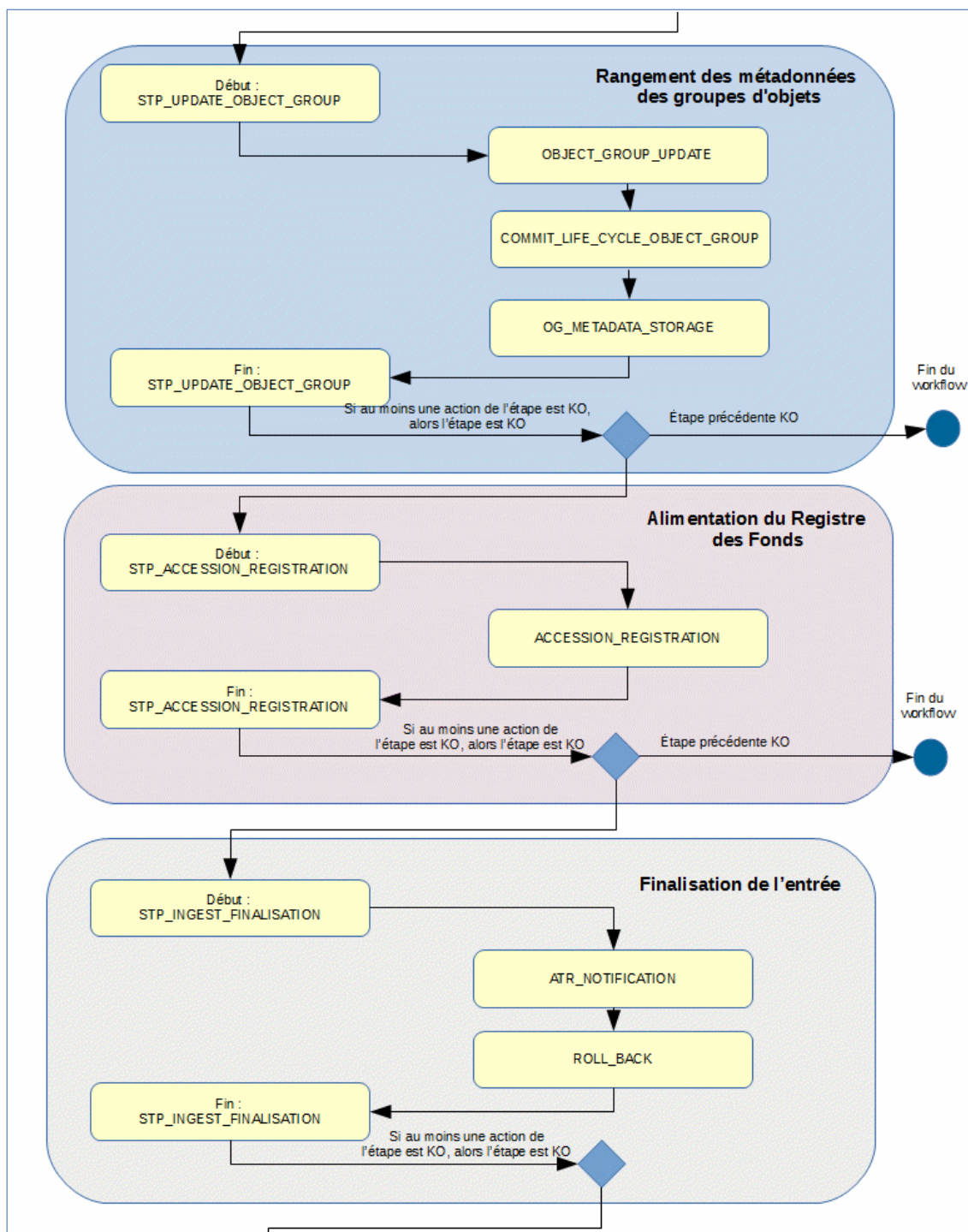
D'une façon synthétique, le workflow est décrit de cette façon :

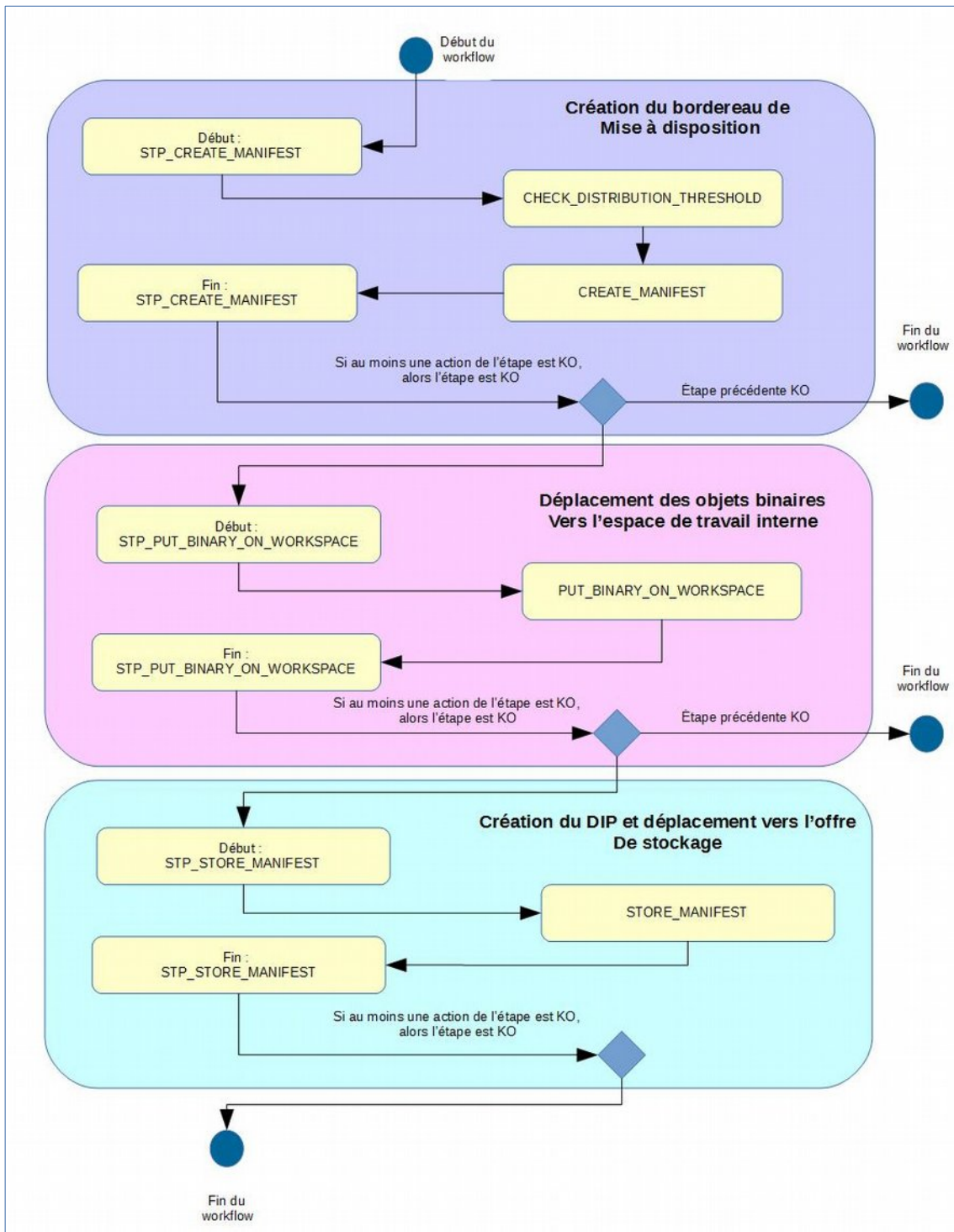












5. MISE À JOUR UNITAIRE (UPDATE)

5.1. Workflow de mise à jour unitaire des unités archivistiques

Cette section décrit le processus permettant la mise à jour unitaire 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.

5.1.1. Mise à jour d'une unité archivistique (vision métier)

Le processus de mise à jour unitaire des unités archivistiques est lancé lors d'une mise à jour de n'importe quelle métadonnée d'une unité archivistique. On distingue cependant deux cas de modifications, liés à des droits gérés via les contrats d'accès : soit les utilisateurs disposent d'un droit de modification sur métadonnées descriptives seulement, soit ils disposent des droits pour modifier les métadonnées descriptives et les métadonnées de gestion (profil d'unité archivistiques et règles de gestion).

Un certain nombre d'étapes, tâches et traitements sont journalisés dans le journal des opérations et décrivent ce processus de mise à jour (clé et description de la clé associée dans le journal des opérations).

5.1.2. Processus de mise à jour unitaire des métadonnées descriptives d'une unité archivistique STP_UPDATE_UNIT_DESC

- **Règle** : étape consistant à mettre à jour de manière unitaire les métadonnées descriptives d'une unité archivistique
- **Type** : bloquant
- **Statuts** :
 - OK : la mise à jour de l'unité archivistique a bien été effectuée (STP_UPDATE_UNIT_DESC.OK = Succès du processus de mise à jour des métadonnées descriptives de l'unité archivistique)
 - KO : la mise à jour de l'unité archivistique n'a pas été effectuée en raison d'une erreur (STP_UPDATE_UNIT_DESC.KO = Échec du processus de mise à jour des métadonnées descriptives de l'unité archivistique)
 - FATAL : une erreur technique est survenue lors de la mise à jour de l'unité archivistique (STP_UPDATE_UNIT_DESC.FATAL = Erreur technique lors du processus de mise à jour des métadonnées descriptives de l'unité archivistique)

5.1.2.1. Vérification des droits de mise à jour des métadonnées descriptives des unités archivistiques UNIT_METADATA_UPDATE_CHECK_PERMISSION

- **Règle** : tâche consistant à contrôler les droits d'écriture donnés par le contrat d'accès
- **Type** : bloquant
- **Statuts** :
 - OK : succès de la vérification des droits de mise à jour des métadonnées des unités archivistiques (UNIT_METADATA_UPDATE_CHECK_PERMISSION.OK = Succès de la vérification des droits de mise à jour des métadonnées des unités archivistiques)
 - KO : échec de la vérification des droits de mise à jour des métadonnées des unités archivistiques (UNIT_METADATA_UPDATE_CHECK_PERMISSION.KO = Échec de la vérification des droits de mise à jour des métadonnées des unités archivistiques)
 - FATAL : erreur technique lors de la vérification des droits de mise à jour des métadonnées des unités archivistiques (UNIT_METADATA_UPDATE_CHECK_PERMISSION.FATAL = Erreur technique lors de la vérification des droits de mise à jour des métadonnées des unités archivistiques)

- WARNING : avertissement lors de la vérification des droits de mise à jour des métadonnées des unités archivistiques (UNIT_METADATA_UPDATE_CHECK_PERMISSION.WARNING = Avertissement lors de la vérification des droits de mise à jour des métadonnées des unités archivistiques)

5.1.2.2. Vérification de l'association à un profil d'unité archivistique UNIT_METADATA_UPDATE_CHECK_DT

- **Règle** : tâche consistant à vérifier la conformité des unités archivistiques mise à jour vis-à-vis des profils d'unités archivistiques qu'elles déclarent. Les profils associés doivent être actifs et leur schéma de contrôle ne doit pas être vide.
- **Type** : bloquant
- **Statuts** :
 - OK : la mise à jour de l'unité archivistique est conforme au profil d'unité archivistique (document type) (UNIT_METADATA_UPDATE_CHECK_DT.OK = Succès de la vérification de l'association à un profil unité archivistique)
 - KO : la mise à jour de l'unité archivistique n'a pas été effectuée en raison de la non-conformité vis-à-vis du profil d'unité archivistique (document type) (UNIT_METADATA_UPDATE_CHECK_DT.KO = Échec de la vérification de l'association à un profil d'unité archivistique)
 - FATAL : une erreur technique est survenue lors de la vérification de la conformité à un profil d'unité archivistique (UNIT_METADATA_UPDATE_CHECK_DT.FATAL = Erreur technique lors de la vérification de l'association à un profil d'unité archivistique)

5.1.2.3. Indexation des métadonnées UNIT_METADATA_UPDATE (ArchiveUnitUpdateUtils.java)

- **Règle** : tâche consistant à indexer dans les bases internes de la solution logicielle Vitam les métadonnées de l'unité archivistique modifiée, ainsi qu'à mettre à jour son journal du cycle de vie de l'unité archivistique. Si la modification touche une métadonnée à historiser, alors un historique est créé.
- **Type** : bloquant
- **Statuts** :
 - OK : succès de la mise à jour des métadonnées de l'unité archivistique (UNIT_METADATA_UPDATE.OK = Succès de la mise à jour des métadonnées de l'unité archivistique)
 - KO : échec de la mise à jour des métadonnées de l'unité archivistique (UNIT_METADATA_UPDATE.KO = Échec de la mise à jour des métadonnées de l'unité archivistique)
 - FATAL : une erreur technique est survenue lors de la mise à jour des métadonnées de l'unité archivistique (UNIT_METADATA_UPDATE.FATAL = Erreur technique lors de la mise à jour des métadonnées de l'unité archivistique)
 - WARNING : avertissement lors de la mise à jour des métadonnées de l'unité archivistique (UNIT_METADATA_UPDATE.WARNING = Avertissement lors de la mise à jour des métadonnées de l'unité archivistique)

À propos de l'historique des données : Il existe un fichier permettant de configurer les métadonnées à historiser dans : vitam/sources/metadata/metadata-core/src/main/resources/history-triggers.json

Ce fichier contient deux variables par objet :

- FieldPathTriggeredForHistory : champ dont la modification déclenche une historisation
- ObjectPathForHistory : champ à historiser

Quand ce champ correspondant à FieldPathTriggeredForHistory est modifié, alors le champ contenu dans « ObjectPathForHistory » est enregistré dans un bloc nommé « history » dans le modèle de données.

Par défaut dans la solution logicielle Vitam, la configuration de history-triggers.json est :


```
[
{
  "FieldPathTriggeredForHistory": "_mgt.ClassificationRule.ClassificationLevel",
  "ObjectPathForHistory": "_mgt.ClassificationRule"
}
]
```

Ainsi lorsqu'un niveau de classification est modifié, alors l'intégralité de la catégorie de règle de classification est enregistré dans le bloc `_history` de l'unité archivistique.

5.1.2.4. Enregistrement du journal du cycle de vie des unités archivistiques COMMIT_LIFE_CYCLE_UNIT

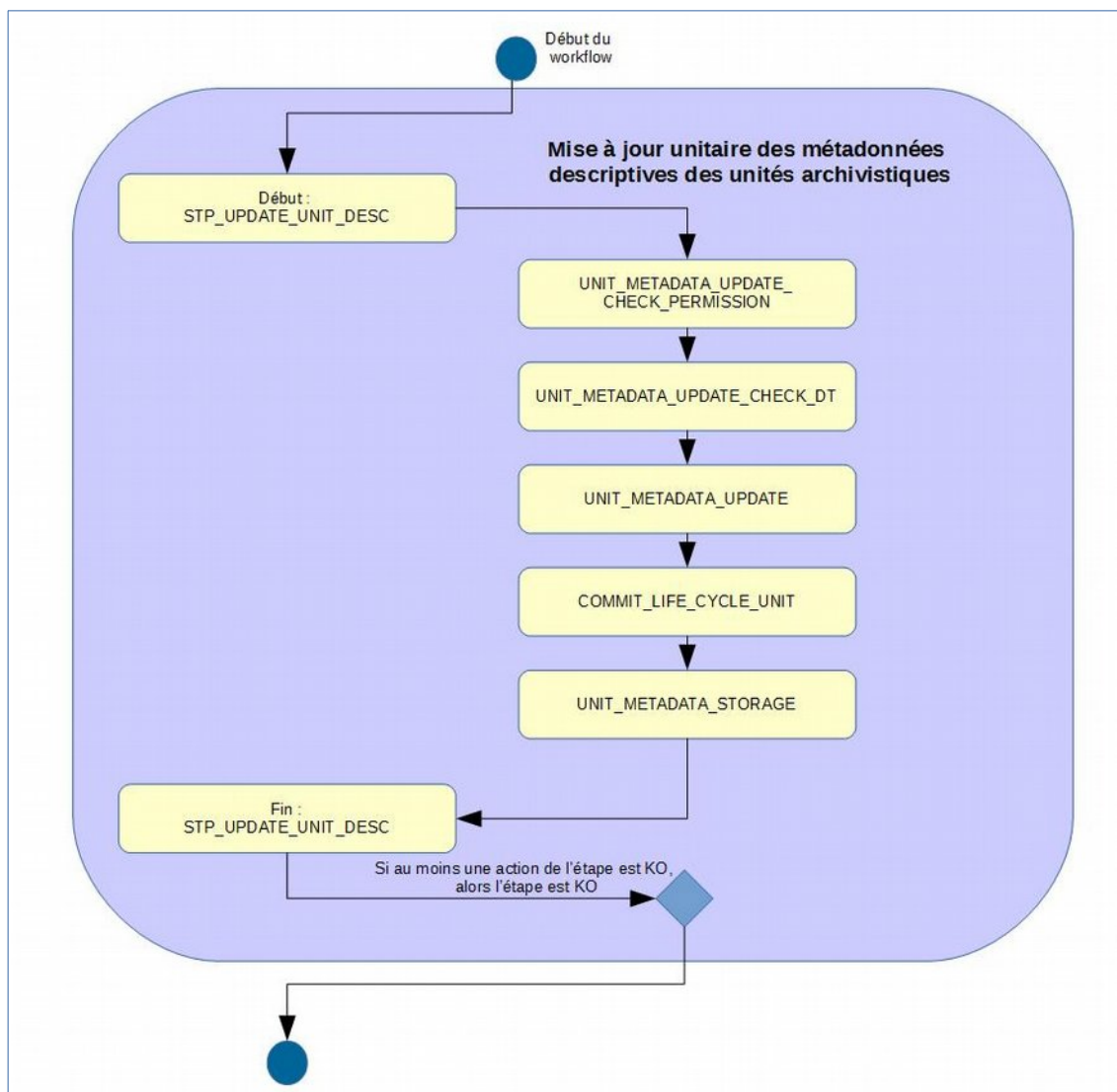
- **Règle** : tâche consistant à sécuriser en base le journal du cycle de vie de l'unité archivistique (avant cette étape, les journaux du cycle de vie des unités archivistiques sont dans une collection temporaire afin de garder une cohérence entre les métadonnées indexées et les journaux lors d'une entrée en succès ou en échec)
- **Type** : bloquant
- **Statuts** :
 - OK : le journal du cycle de vie de l'unité archivistique a été enregistré avec succès (COMMIT_LIFE_CYCLE_UNIT.OK = Succès de l'enregistrement du journal du cycle de vie de l'unité archivistique)
 - FATAL : une erreur technique est survenue lors de l'enregistrement du journal du cycle de vie de l'unité archivistique (COMMIT_LIFE_CYCLE_UNIT.FATAL = Erreur technique lors de l'enregistrement du journal du cycle de vie de l'unité archivistique)

5.1.2.5. Écriture des métadonnées de l'unité archivistique sur l'offre de stockage UNIT_METADATA_STORAGE (AccessInternalModuleImpl.java)

- **Règle** : tâche consistant à sauvegarder les métadonnées de l'unité archivistique sur les offres de stockage en fonction de la stratégie de stockage.
- **Type** : bloquant
- **Statuts** :
 - OK : les métadonnées de l'unité archivistique ont été sauvegardées sur les offres de stockage (UNIT_METADATA_STORAGE.OK = Succès de l'enregistrement des métadonnées de l'unité archivistique)
 - WARNING : avertissement lors de la sauvegarde des métadonnées de l'unité archivistique (UNIT_METADATA_STORAGE.ALREADY_EXECUTED = Action déjà exécutée : Pas d'enregistrement des métadonnées de l'unité archivistique)
 - FATAL : une erreur technique est survenue lors de la sauvegarde des métadonnées de l'unité archivistique (UNIT_METADATA_STORAGE.FATAL = Erreur technique lors de l'enregistrement des métadonnées de l'unité archivistique)

5.1.3. Structure de workflow de mise à jour des métadonnées descriptives

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



5.1.4. Mise à jour unitaire des règles de gestion d'une unité archivistique STP_UPDATE_UNIT (AccessInternalModuleImpl.java)

- **Règle** : étape consistant à mettre à jour de manière unitaire les métadonnées de gestion d'une unité archivistique
- **Type** : bloquant
- **Statuts** :
 - OK : la mise à jour de l'unité archivistique a bien été effectuée. (STP_UPDATE_UNIT.OK = Succès du processus de mise à jour des métadonnées de l'unité archivistique)
 - KO : la mise à jour de l'unité archivistique n'a pas été effectuée en raison d'une erreur (STP_UPDATE_UNIT.KO = Échec du processus de mise à jour des métadonnées de l'unité archivistique)
 - FATAL : une erreur technique est survenue lors de la mise à jour de l'unité archivistique (STP_UPDATE_UNIT.FATAL = Erreur technique lors du processus de mise à jour des métadonnées de l'unité archivistique)

5.1.4.1. Vérification des droits de mise à jour des métadonnées descriptives et de gestion des unités archivistiques UNIT_METADATA_UPDATE_CHECK_PERMISSION

- **Règle** : tâche consistant à contrôler les droits d'écriture donnés par le contrat d'accès et à vérifier que l'utilisateur a bien des droits d'écriture des métadonnées descriptives et des métadonnées de gestion
- **Type** : bloquant
- **Statuts** :
 - OK : succès de la vérification des droits de mise à jour des métadonnées de l'unité archivistique (UNIT_METADATA_UPDATE_CHECK_PERMISSION.OK = Succès de la vérification des droits de mise à jour des métadonnées de l'unité archivistique)
 - KO : échec de la vérification des droits de mise à jour des métadonnées de l'unité archivistique (UNIT_METADATA_UPDATE_CHECK_PERMISSION.KO = Échec de la vérification des droits de mise à jour des métadonnées de l'unité archivistique)
 - FATAL : erreur technique lors de la vérification des droits de mise à jour des métadonnées de l'unité archivistique (UNIT_METADATA_UPDATE_CHECK_PERMISSION.FATAL = Erreur technique lors de la vérification des droits de mise à jour des métadonnées de l'unité archivistique)
 - WARNING : avertissement lors de la vérification des droits de mise à jour des métadonnées de l'unité archivistique (UNIT_METADATA_UPDATE_CHECK_PERMISSION.WARNING = Avertissement lors de la vérification des droits de mise à jour des métadonnées de l'unité archivistique)

5.1.4.2. Vérification des règles de gestion UNIT_METADATA_UPDATE_CHECK_RULES (AccessInternalModuleImpl.java)

- **Règle** : tâche consistant à vérifier les modifications apportées aux règles de gestion de l'unité archivistique
- **Type** : bloquant
- **Statuts** :
 - OK : succès de la vérification des métadonnées de gestion de l'unité archivistique ont été vérifiées (UNIT_METADATA_UPDATE_CHECK_RULES.OK = Succès de la vérification des métadonnées de gestion de l'unité archivistique)
 - KO : oéchec de la vérification des métadonnées de gestion de l'unité archivistique (UNIT_METADATA_UPDATE_CHECK_RULES.KO = Échec de la vérification des métadonnées de gestion de l'unité archivistique)
 - FATAL : une erreur technique est survenue lors de la vérification des métadonnées de gestion (UNIT_METADATA_UPDATE_CHECK_RULES.FATAL = Erreur technique lors de la vérification des métadonnées de gestion de l'unité archivistique)

5.1.4.3. Vérification de l'association à un profil d'unité archivistique UNIT_METADATA_UPDATE_CHECK_DT

- **Règle** : cette tâche permet de vérifier la conformité des unités archivistiques mise à jour vis-à-vis de leurs profils d'unités archivistiques. Les identifiants de profils archivistiques renseignés dans les unités archivistiques doivent exister dans le référentiel des profils. Les profils associés doivent être actif et leur schéma de contrôle ne doit pas être vide.
- **Type** : bloquant
- **Statuts** :
 - OK : succès de la vérification de la conformité au profil unités archivistiques (document type) ((UNIT_METADATA_UPDATE_CHECK_DT.OK = Succès de la vérification de l'association à un profil unité archivistique)
 - KO : la mise à jour de l'unité archivistique n'a pas été effectuée en raison de la non-conformité vis-à-vis du profil d'unité archivistique (document type) (UNIT_METADATA_UPDATE_CHECK_DT.KO = Échec de la vérification de l'association à un profil unité archivistique)

- FATAL : une erreur technique est survenue lors de la vérification de la conformité aux profils d'unités archivistiques (UNIT_METADATA_UPDATE_CHECK_DT.FATAL = Erreur technique lors de la vérification de l'association à un profil unité archivistique)

5.1.4.4. Indexation des métadonnées UNIT_METADATA_UPDATE (ArchiveUnitUpdateUtils.java)

- **Règle** : tâche consistant à indexer dans les bases internes de la solution logicielle Vitam les métadonnées de l'unité archivistique modifiée, ainsi qu'à mettre à jour son journal du cycle de vie. Si la modification touche une métadonnée à historiser, alors un historique est créé.
- **Type** : bloquant
- **Statuts** :
 - OK : succès de la mise à jour des métadonnées de l'unité archivistique (UNIT_METADATA_UPDATE.OK = Succès de la mise à jour des métadonnées de l'unité archivistique)
 - KO : échec de la mise à jour des métadonnées de l'unité archivistique (UNIT_METADATA_UPDATE.KO = Échec de la mise à jour des métadonnées de l'unité archivistique)
 - FATAL : une erreur technique est survenue lors de la mise à jour des métadonnées de l'unité archivistique (UNIT_METADATA_UPDATE.FATAL = Erreur technique lors de la mise à jour des métadonnées de l'unité archivistique)
 - WARNING : avertissement lors de la mise à jour des métadonnées de l'unité archivistique (UNIT_METADATA_UPDATE.WARNING = Avertissement lors de la mise à jour des métadonnées de l'unité archivistique)

À propos de l'historique des données : Il existe un fichier permettant de configurer les métadonnées à historiser dans : vitam/sources/metadata/metadata-core/src/main/resources/history-triggers.json

Ce fichier contient deux variables par objet :

- FieldPathTriggeredForHistory : champ dont la modification déclenche une historisation
- ObjectPathForHistory : champ à historiser

Quand ce champ correspondant à FieldPathTriggeredForHistory est modifié, alors le champ contenu dans « ObjectPathForHistory » est enregistré dans un bloc nommé « history » dans le modèle de données.

Par défaut dans la solution logicielle Vitam, la configuration de history-triggers.json est :

```
[
  {
    "FieldPathTriggeredForHistory": "_mgt.ClassificationRule.ClassificationLevel",
    "ObjectPathForHistory": "_mgt.ClassificationRule"
  }
]
```

Ainsi, lorsqu'un niveau de classification est modifié, alors l'intégralité de la catégorie de règle de classification est enregistré dans le bloc _history de l'unité archivistique.

5.1.4.5. Enregistrement du journal du cycle de vie des unités archivistiques COMMIT_LIFE_CYCLE_UNIT

- **Règle** : tâche consistant à sécuriser en base le journal du cycle de vie de l'unité archivistique (avant cette étape, les journaux du cycle de vie des unités archivistiques sont dans une collection temporaire afin de garder une cohérence entre les métadonnées indexées et les journaux lors d'une entrée en succès ou en échec).
- **Type** : bloquant
- **Statuts** :
 - OK : le journal du cycle de vie de l'unité archivistique a été enregistré avec succès

(COMMIT_LIFE_CYCLE_UNIT.OK = Succès de l'enregistrement du journal du cycle de vie de l'unité archivistique)

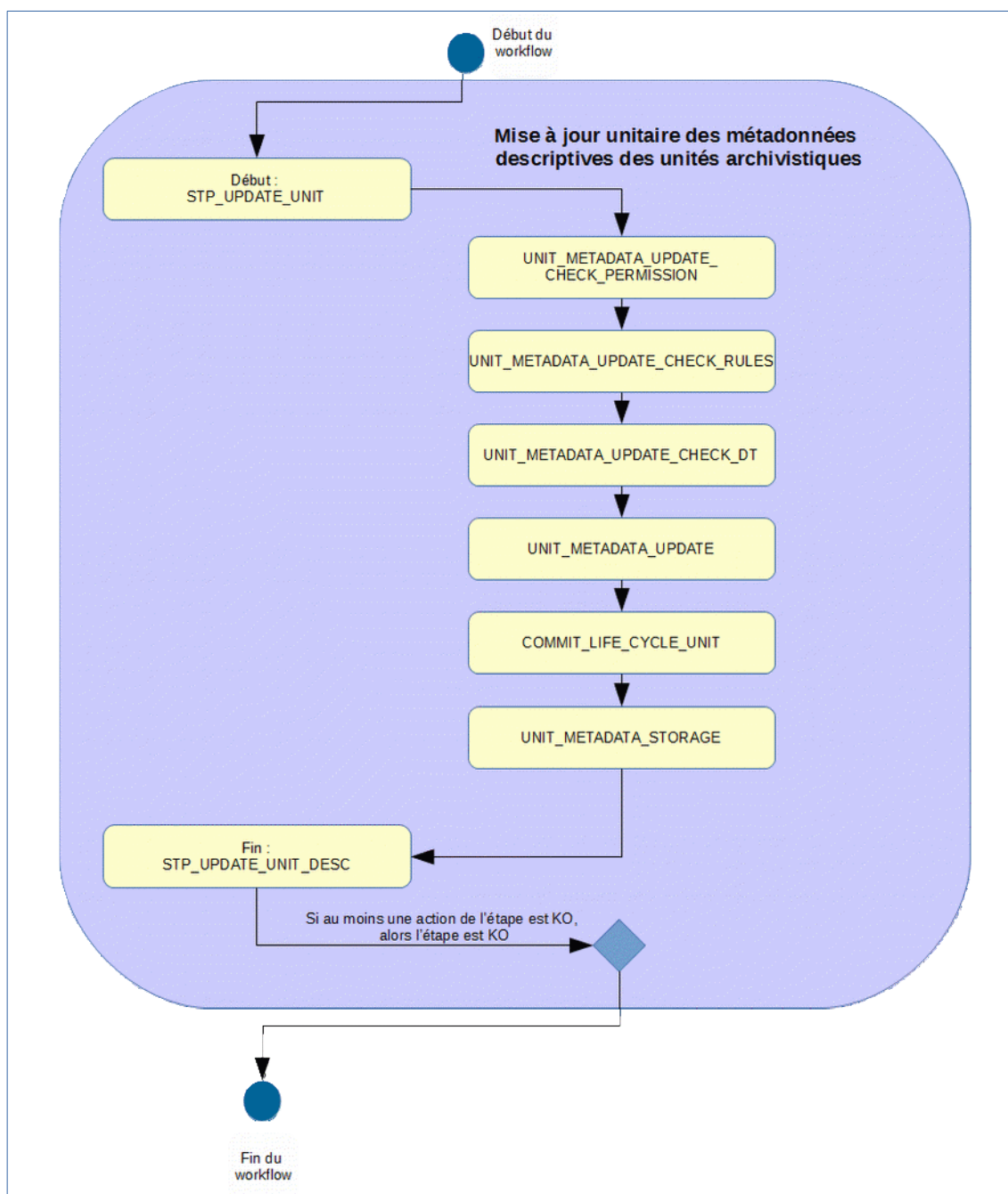
- FATAL : une erreur technique est survenue lors de l'enregistrement du journal du cycle de vie de l'unité archivistique (COMMIT_LIFE_CYCLE_UNIT.FATAL = Erreur technique lors de l'enregistrement du journal du cycle de vie de l'unité archivistique)

5.1.4.6.Écriture des métadonnées de l'unité archivistique sur l'offre de stockage UNIT_METADATA_STORAGE (AccessInternalModuleImpl.java)

- **Règle** : tâche consistant à sauvegarder les métadonnées de l'unité archivistique sur les offres de stockage en fonction de la stratégie de stockage
- **Type** : bloquant
- **Statuts** :
 - OK : les métadonnées de l'unité archivistique ont été sauvegardées sur les offres de stockage (UNIT_METADATA_STORAGE.OK = Succès de l'enregistrement des métadonnées de l'unité archivistique)
 - WARNING : avertissement lors de la sauvegarde des métadonnées de l'unité archivistique (UNIT_METADATA_STORAGE.ALREADY_EXECUTED = Action déjà exécutée : Pas d'enregistrement des métadonnées de l'unité archivistique)
 - FATAL : une erreur technique est survenue lors de la sauvegarde des métadonnées de l'unité archivistique (UNIT_METADATA_STORAGE.FATAL = Erreur technique lors de l'enregistrement des métadonnées de l'unité archivistique)

5.1.5. Structure de workflow de mise à jour des métadonnées de gestion des unités archivistiques

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



5.2. Workflow de mise à jour des règles de gestion des unités archivistiques lors de l'import d'un nouveau référentiel

Cette section décrit le processus (workflow) permettant la mise à jour des règles de gestion des unités archivistiques suite à l'import d'un nouveau référentiel des règles de gestion.

Le workflow mis en place dans la solution logicielle Vitam est défini dans le fichier « DefaultRulesUpdateWorkflow.json ». Ce fichier est disponible dans : 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.

5.2.1. Mise à jour des règles de gestion des unités archivistiques (UPDATE_RULES_ARCHIVE_UNITS)

Le processus de mise à jour des règles de gestion des unités archivistiques est lancé à la suite d'une mise à jour des règles de gestion lorsque la solution logicielle Vitam détecte qu'une règle de gestion a été modifiée et est utilisée par une ou plusieurs unités archivistiques. Toutes les étapes et actions sont journalisées dans le journal des opérations.

5.2.2. Processus de préparation des listes d'unités archivistiques à mettre à jour STP_PREPARE_LISTS (updatedRules.json)

5.2.2.1. Établissement de la liste des entrées en cours d'exécution – LIST_RUNNING_INGESTS – fichier de sortie : GUID/PROCESSING/runningIngests.json

- **Règle** : étape consistant à vérifier si des entrées sont en cours d'exécution. Un fichier runningIngests.json est rempli avec les identifiants des entrées en cours. Le fichier est vide si aucune entrée n'est en cours.
- **Type** : bloquant
- **Statuts** :
 - **OK** : le fichier listant les entrées (qu'il soit vide ou non) a bien été créé (LIST_RUNNING_INGESTS.OK = Succès du processus de préparation des listes des unités archivistiques à mettre à jour).
 - **KO** : la liste des entrées en cours n'a pas pu être récupérée, ou alors la liste des entrées n'a pas pu être enregistrée sur l'espace de travail interne (LIST_RUNNING_INGESTS.KO = Échec du processus de préparation des listes des unités archivistiques à mettre à jour)
 - **FATAL** : une erreur technique est survenue lors de la préparation de la liste des entrées (LIST_RUNNING_INGESTS.FATAL = Erreur technique lors du processus de préparation des listes des unités archivistiques à mettre à jour)

5.2.2.2. Établissement de la liste des unités archivistiques à mettre à jour – LIST_ARCHIVE_UNITS (fichier de sortie : GUID/PROCESSING/auToBeUpdated.json)

- **Règle** : tâche consistant à établir la liste des unités archivistiques à mettre à jour. Pour chaque unité archivistique concernée, un fichier est créé et déposé sur l'espace de travail interne (le workspace) pour pouvoir être traité plus tard dans le workflow.
- **Type** : bloquant
- **Statuts** :

- OK : la liste des unités archivistiques et les fichiers associés ont bien pu être créés. Les fichiers associés ont bien été créés (LIST_ARCHIVE_UNITS.OK = Succès lors du processus de l'établissement de la liste des unités archivistiques à mettre à jour)
- FATAL : une erreur technique est survenue lors d'établissement de la liste des unités archivistiques à mettre à jour (LIST_ARCHIVE_UNITS.FATAL = Erreur technique lors du processus d'établissement de la liste des unités archivistiques à mettre à jour)

5.2.2.1.Établissement de la liste des unités archivistiques à mettre à jour – COMPUTE_INHERITED_RULES_PROGENY_IDENTIFIER (fichier de sortie : GUID/PROCESSING/unitsToInvalidate.json)

- **Règle** : tâche consistant à établir la liste des unités archivistiques dont les règles de gestion applicables sont à invalider. Pour chaque unité archivistique concernée, un fichier est créé et déposé sur l'espace de travail interne (le workspace) pour pouvoir être traité plus tard dans le workflow.
- **Type** : bloquant
- **Statuts** :
 - OK : la liste des unités archivistiques et les fichiers associés ont bien pu être créés. Les fichiers associés ont bien été créés (COMPUTE_INHERITED_RULES_PROGENY_IDENTIFIER .OK = Succès lors du processus d'identification des unités archivistiques dont les règles de gestion applicables sont à invalider)
 - FATAL : une erreur technique est survenue lors d'établissement de la liste des unités archivistiques à mettre à jour (COMPUTE_INHERITED_RULES_PROGENY_IDENTIFIER .FATAL = Erreur technique lors du processus d'identification des unités archivistiques dont les règles de gestion applicables sont à invalider)

5.2.3.Invalidation des unités archivistiques dont les règles de gestion sont à invalider STP_INVALIDATE (Distribution sur LIST GUID/unitsToInvalidate.json)

5.2.3.1.Invalidation des règles de gestion applicables dans les unités archivistiques à invalider COMPUTE_INHERITED_RULES_INVALIDATOR

- **Règle** : traitement consistant, pour chaque unité archivistique à invalider, à supprimer les règles de gestion applicable.
- **Type** : bloquant
- **Statuts** :
 - OK : les règles de gestion applicables ont bien été invalidés dans les unités archivistiques (COMPUTE_INHERITED_RULES_INVALIDATOR.OK = Succès de la mise à jour des règles de gestion applicables des unités archivistiques).
 - FATAL : une erreur technique est survenue lors de la mise à jour de l'unité archivistique (COMPUTE_INHERITED_RULES_INVALIDATOR .FATAL = Erreur technique lors de la mise à jour des règles de gestion des unités archivistiques)

5.2.4.Mise à jour des unités archivistiques STP_UNIT_UPDATE (Distribution sur LIST GUID/UnitsWithoutLevel)

5.2.4.1.Mise à jour des règles de gestion d'une unité archivistique - UPDATE_UNIT_RULES

- **Règle** : traitement consistant, pour chaque unité archivistique à mettre à jour, à vérifier les règles de gestion impactées et à recalculer les échéances des règles

- **Type** : bloquant
- **Statuts** :
 - OK : l'unité archivistique a bien été mise à jour (UPDATE_UNIT_RULES.OK = Succès de la mise à jour des règles de gestion des unités archivistiques).
 - KO : l'unité archivistique n'a pas été trouvée, ou n'a pas pu être mise à jour (UPDATE_UNIT_RULES.KO = Échec de la mise à jour des règles de gestion des unités archivistiques)
 - FATAL : une erreur technique est survenue lors de la mise à jour de l'unité archivistique (UPDATE_UNIT_RULES.FATAL = Erreur technique lors de la mise à jour des règles de gestion des unités archivistiques)

5.2.4.2.Écriture des métadonnées de l'unité archivistique sur l'offre de stockage UNIT_METADATA_STORAGE (AccessInternalModuleImpl.java)

- **Règle** : tâche consistant à sauvegarder les métadonnées de l'unité archivistique sur les offres de stockage en fonction de la stratégie de stockage
- **Type** : bloquant
- **Statuts** :
 - OK : les métadonnées de l'unité archivistique ont été sauvegardées sur les offres de stockage (UNIT_METADATA_STORAGE.OK = Succès de l'enregistrement des métadonnées de l'unité archivistique)
 - WARNING : avertissement lors de la sauvegarde des métadonnées de l'unité archivistique (UNIT_METADATA_STORAGE.ALREADY_EXECUTED = Action déjà exécutée : Pas d'enregistrement des métadonnées de l'unité archivistique)
 - FATAL : une erreur technique est survenue lors de la sauvegarde des métadonnées de l'unité archivistique (UNIT_METADATA_STORAGE.FATAL = Erreur technique lors de l'enregistrement des métadonnées de l'unité archivistique)

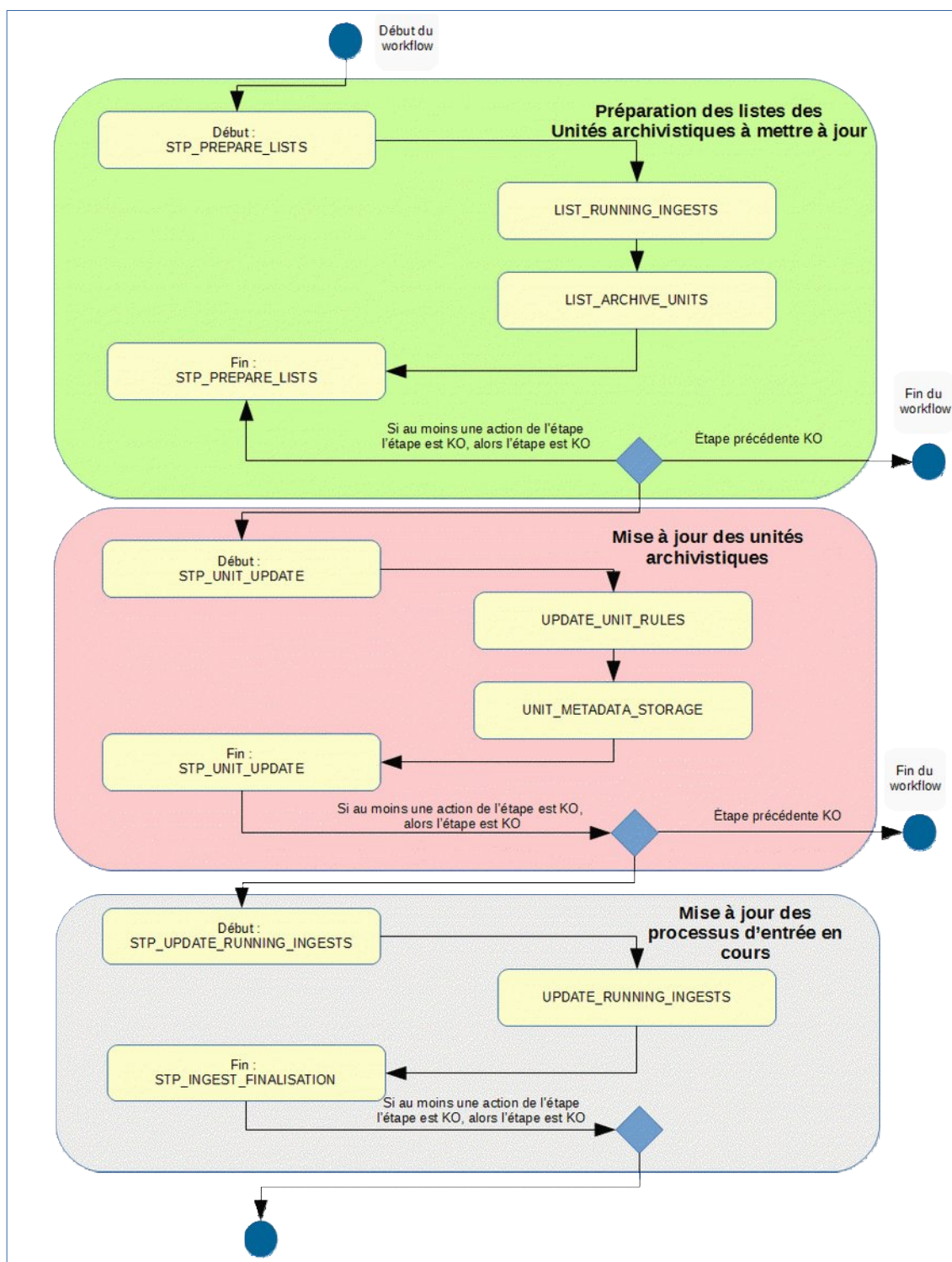
5.2.5.Mise à jour des processus d'entrée en cours – STP_UPDATE_RUNNING_INGESTS (updatedRules.json).

5.2.5.1.Mise à jour des entrées en cours – UPDATE_RUNNING_INGESTS (runningIngests.json)

- **Règle** : traitement consistant, pour une liste d'entrées en cours, à vérifier que chaque entrée est finalisée, à vérifier les règles de gestion impactées, et à recalculer les échéances des règles.
- **Type** : bloquant
- **Statuts** :
 - OK : les entrées en cours ont été finalisées, et les unités archivistiques ont bien été mises à jour (STP_UPDATE_RUNNING_INGESTS.OK = Succès de la mise à jour des entrées en cours).
 - KO : un problème a été rencontré avec le fichier des règles de gestion mises à jour (STP_UPDATE_RUNNING_INGESTS.KO = Échec de la mise à jour des entrées en cours)
 - FATAL : une erreur technique est survenue lors de la mise à jour des processus d'entrées (STP_UPDATE_RUNNING_INGESTS.FATAL = Erreur technique lors de la mise à jour des entrées en cours)

5.2.6. Structure de workflow du processus de mise à jour des règles de gestion des unités archivistiques

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



6. MISE À JOUR DE MASSE (MASS UPDATE)

6.1. Workflow de mise à jour de masse des métadonnées descriptives des unités archivistiques

Cette section décrit le processus permettant d'effectuer des actions sur un grand nombre d'unités archivistiques stockées dans la solution logicielle Vitam. Cette fonctionnalité nécessite d'avoir les droits requis pour pouvoir intervenir sur les métadonnées. Les autorisations de modifications en masse portent soit sur les métadonnées descriptives soit sur les métadonnées de gestion.

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

6.1.1. Processus de mise à jour en masse des métadonnées descriptives des unités archivistiques (MASS_UPDATE_UNIT_DESC)

Le processus de mise à jour en masse des métadonnées descriptives des unités archivistiques permet d'effectuer des modifications sur un ensemble conséquent d'archives en une seule opération.

- **Règle** : opération consistant à mettre à jour en masse les métadonnées descriptives des unités archivistiques
- **Type** : bloquant
- **Statuts** :
 - OK : la mise à jour en masse des métadonnées descriptives des unités archivistiques a bien été effectuée (MASS_UPDATE_UNIT_DESC.OK = Succès de la mise à jour en masse des métadonnées descriptives des unités archivistiques)
 - KO : la mise à jour en masse des métadonnées descriptives des unités archivistiques n'a pas été effectuée en raison d'une erreur (MASS_UPDATE_UNIT_DESC.KO = Échec de la mise à jour en masse des métadonnées descriptives des unités archivistiques)
 - FATAL : une erreur technique est survenue lors de la mise à jour en masse des métadonnées descriptives des unités archivistiques (MASS_UPDATE_UNIT_DESC.FATAL = Erreur technique lors de la mise à jour en masse des métadonnées descriptives des unités archivistiques)

6.1.2. Processus de préparation de la liste des unités archivistiques à mettre à jour et des autorisations de modification STP_CHECK_AND_COMPUTE

- **Règle** : étape consistant à préparer la liste des unités archivistiques à mettre à jour et à vérifier les autorisations de modification
- **Type** : bloquant
- **Statuts** :
 - OK : le processus de préparation de la liste des unités archivistiques à mettre à jour et de vérification des droits de modification a bien été effectué (STP_CHECK_AND_COMPUTE.OK = Succès du processus de préparation de la liste des unités archivistiques à mettre à jour et de vérification des autorisations de modification)
 - KO : le processus de préparation de la liste des unités archivistiques à mettre à jour et de vérification des droits de modification n'a pas été effectué en raison d'une erreur (STP_CHECK_AND_COMPUTE.KO = Échec du processus de préparation de la liste des unités archivistiques à mettre à jour et de vérification des autorisations de modification)
 - FATAL : une erreur technique est survenue lors du processus de préparation de la liste des unités archivistiques à mettre à jour et de vérification des autorisations de modification (STP_CHECK_AND_COMPUTE.FATAL = Erreur technique lors du processus de préparation de la

liste des unités archivistiques à mettre à jour et de vérification des autorisations de modification)

6.1.2.1. Vérification des droits de mise à jour des métadonnées descriptives des unités archivistiques MASS_UPDATE_CHECK

- **Règle** : tâche consistant à contrôler les droits d'écriture donnés par le contrat d'accès et à vérifier que l'utilisateur a bien des droits d'écriture des métadonnées descriptives
- **Type** : bloquant
- **Statuts** :
 - OK : la vérification des droits de mise à jour des métadonnées des unités archivistiques a bien été effectuée (MASS_UPDATE_CHECK.OK = Succès de la vérification des droits de mise à jour des métadonnées descriptives des unités archivistiques)
 - KO : l'utilisateur n'a pas les droits de mise à jour des métadonnées descriptives des unités archivistiques (MASS_UPDATE_CHECK.KO = Échec de la vérification des droits de mise à jour des métadonnées descriptives des unités archivistiques)
 - FATAL : une erreur technique est survenue lors de la vérification des droits de mise à jour des métadonnées descriptives des unités archivistiques (MASS_UPDATE_CHECK.FATAL = Erreur technique lors de la vérification des droits de mise à jour des métadonnées descriptives des unités archivistiques)

6.1.2.2. 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)

6.1.2.3. Préparation de la liste des unités archivistiques à mettre à jour PREPARE_UPDATE_UNIT_LIST

- **Règle** : tâche consistant à préparer la liste des unités archivistiques à mettre à jour
- **Type** : bloquant
- **Statuts** :
 - OK : la préparation de la liste des unités archivistiques à mettre à jour a bien été effectuée (PREPARE_UPDATE_UNIT_LIST.OK = Succès de la préparation de la liste des unités archivistiques à mettre à jour)
 - KO : la préparation de la liste des unités archivistiques à mettre à jour n'a pas été effectuée en raison d'une erreur (PREPARE_UPDATE_UNIT_LIST.KO = Échec de la préparation de la liste des unités archivistiques à mettre à jour)
 - FATAL : une erreur technique est survenue lors de la préparation de la liste des unités archivistiques à mettre à jour (PREPARE_UPDATE_UNIT_LIST.FATAL = Erreur technique lors de la préparation de la liste des unités archivistiques à mettre à jour)

6.1.3. Processus de traitement de mise à jour des unités archivistiques STP_UPDATE

Préparation de la liste des unités archivistiques à mettre à jour MASS_UPDATE_UNIT

- **Règle** : tâche et traitement consistant à mettre à jour les métadonnées descriptives des unités archivistiques
- **Type** : bloquant
- **Statuts** :
 - OK : la mise à jour des métadonnées descriptives des unités archivistiques a bien été effectué (MASS_UPDATE_UNITS.OK = Succès de la mise à jour des métadonnées descriptives des unités archivistiques)
 - KO : la mise à jour des métadonnées descriptives des unités archivistiques n'a pas été effectuée en raison d'une erreur (MASS_UPDATE_UNITS.KO = Échec de la mise à jour des métadonnées descriptives des unités archivistiques)
 - FATAL : une erreur technique est survenue lors de la mise à jour des métadonnées descriptives des unités archivistiques (MASS_UPDATE_UNITS.FATAL = Erreur technique lors de la mise à jour des métadonnées descriptives des unités archivistiques)

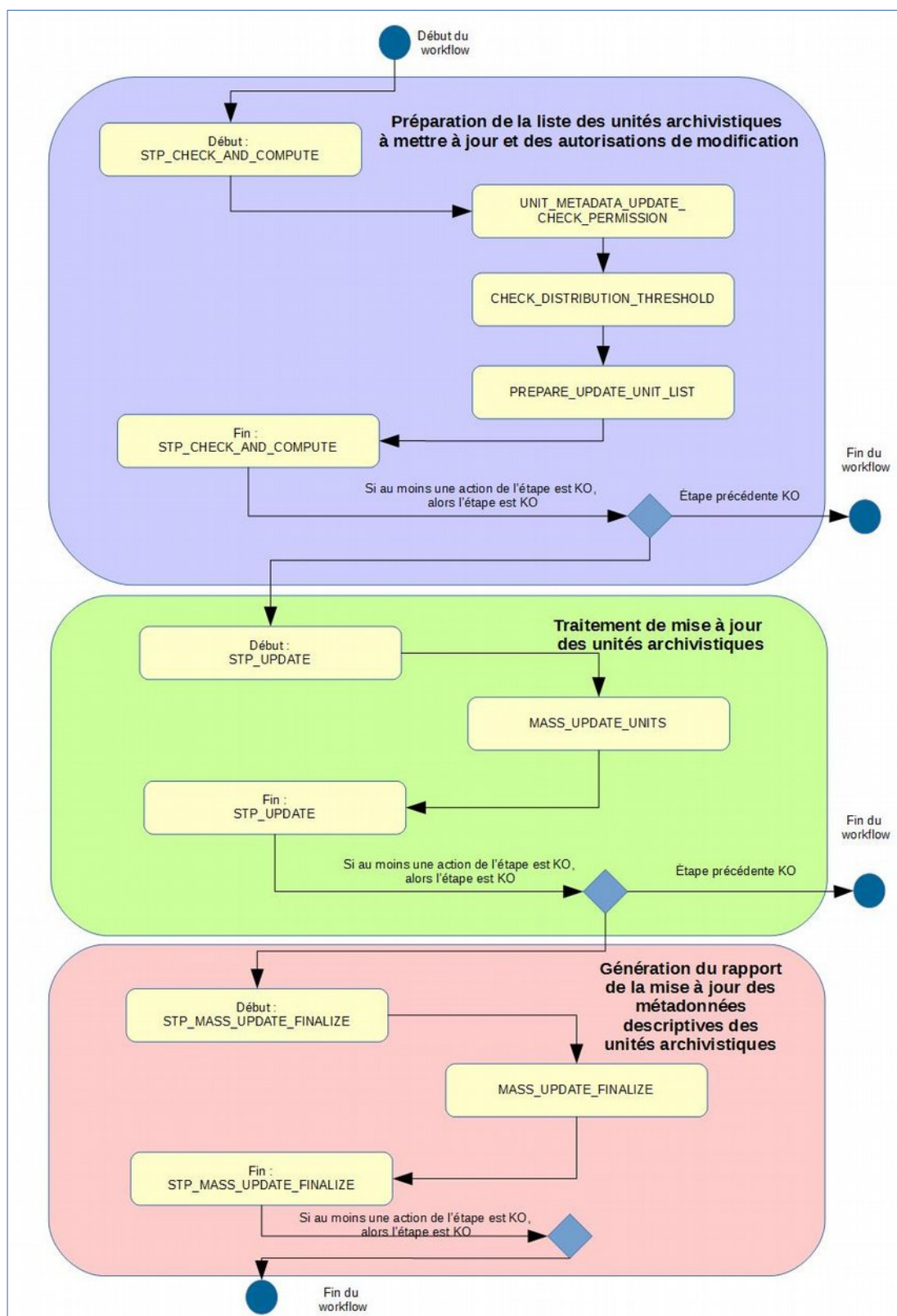
6.1.4.Processus de génération du rapport de mise à jour des métadonnées descriptives des unités archivistiques STP_MASS_UPDATE_FINALIZE

Génération du rapport de mise à jour des métadonnées descriptives des unités archivistiques MASS_UPDATE_FINALIZE

- **Règle** : tâche consistant à générer le rapport de mise à jour des métadonnées descriptives des unités archivistiques
- **Type** : bloquant
- **Statuts** :
 - OK : le processus de génération du rapport de mise à jour des métadonnées descriptives des unités archivistiques a bien été effectué (MASS_UPDATE_FINALIZE.OK = Succès du processus de génération du rapport de mise à jour des métadonnées descriptives des unités archivistiques)
 - KO : le processus de génération du rapport de mise à jour des métadonnées descriptives des unités archivistiques à mettre à jour n'a pas été effectuée en raison d'une erreur (MASS_UPDATE_FINALIZE.KO = Échec du processus de génération du rapport de mise à jour des métadonnées descriptives des des unités archivistiques)
 - FATAL : une erreur technique est survenue lors du processus de génération du rapport de mise à jour des métadonnées descriptives des des unités archivistiques à mettre à jour (MASS_UPDATE_FINALIZE.FATAL = Erreur technique lors du processus de génération du rapport de mise à jour des métadonnées descriptives des unités archivistiques)

6.1.5. Structure du workflow de mise à jour en masse des métadonnées descriptives des unités archivistiques

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



6.1.6. Rapport de modification en masse des métadonnées descriptives des unités archivistiques.

Le rapport de modification en masse des métadonnées descriptives des unités archivistiques est un fichier JSONL généré par la solution logicielle Vitam lorsqu'une opération de modification en masse des métadonnées descriptives se termine. Cette section décrit la manière dont ce rapport est structuré.

6.1.6.1. Exemple de JSONL : rapport de modification en masse des métadonnées descriptives des unités archivistiques

```

{
  → "tenant": "8",
  → "evId": "aaaaaaaaabchia7juaabddesalomutkfwiaaaaq",
  → "evType": "MASS_UPDATE_UNITS",
  → "outcome": "KO",
  → "outDetail": "MASS_UPDATE_UNITS.KO",
  → "outMsg": "\u0009chec de la mise \u000E0 jour des unit\u000E9a archivistiques Detail-
KO:1",
  → "rightsStatementIdentifier": {
    → "AccessContract": "AC-000001"
  }
}

"evDetData": {
  → "Event": "Invalid unit format : Document schema validation failed :
\n{"validateJson":{"level":"error","schema":
{"loadingURI":"#","pointer":"/definitions/agent-
type+signature+validation"},"instance":
{"pointer":"/Writer/0"},"domain":"validation","keyword":"oneOf","message":
"instance failed to match exactly one schema (matched 2 out of
2)","matched":2,"nSchemas":2,"reports":{"definitions/agent-
type+signature+validation/oneOf/0":[],"definitions/agent-
type+signature+validation/oneOf/1":[]},"level":"error","schema":
{"loadingURI":"#","pointer":"/definitions/agent-
type+signature+validation"},"instance":
{"pointer":"/Writer/0"},"domain":"validation","keyword":"type","message":
"instance type (string) does not match any allowed primitive type (allowed:
[\"object\"])}","found":"string","expected":["object"]}}"}
}

}

{
  → "evStartDate": "2019-11-13T14:24:21.577",
  → "evEndDate": "2019-11-13T14:24:30.743",
  → "reportType": "UPDATE_UNIT",
  → "vitamResults": {
    → "OK": 0,
    → "KO": 1,
    → "WARNING": 0,
    → "total": 1
  }
  → "extendedInfo": null
}

{
  → "$roots": [],
  → "$query": {
    → "for": {

```

OperationSummary

ReportSummary

Context

```

→ → → → → "$sin": -{[
→ → → → → → "#id": -["aaaaaaaabahfpcbnablg6alol3q3l6iaaaa"]}]
→ → → → → }
→ → → → → 1. -{[
→ → → → → → "$sin": -{[
→ → → → → → → "#allunitups": -{[]
→ → → → → → → }
→ → → → → → }
→ → → → → 1.}]
→ → → → → "$depth": -1000}]
→ → → }
→ 1.}]
→ "$filter": -{[]}]
→ "$action": -{[[]
→ → → → "$act": -{[]
→ → → → → "Writer": -"Graf"}
→ → → → }
→ → → }
→ 1}]

```

```

→ "_id": -"aaaaaaaaahocvtxabtmualomutmanmyaaaaq",}]
→ "$resultKey": -"CHECK_UNIT_SCHEMA",}]
→ "$processId": -"aaaaaaaaabchia7jwabddaalomutkwiaaaaaq",}]
→ "$tenant": -8,}]
→ "$status": -"KO",}]
→ "$message": -"Invalid unit format : Document schema validation failed :
n{"validateJson":{"level":"error","schema":
{"loadingURI":"#","pointer":"/definitions/agent-
type+signature+validation"},"instance":
{"pointer":"/Writer/0"},"domain":"validation","keyword":"oneOf","message":
"instance failed to match exactly one schema (matched 2 out of
)}","matched":2,"nSchemas":2,"reports":{"definitions/agent-
type+signature+validation/oneOf/0":[],"definitions/agent-
type+signature+validation/oneOf/1":[]},"level":"error","schema":
{"loadingURI":"#","pointer":"/definitions/agent-
type+signature+validation"},"instance":
{"pointer":"/Writer/0"},"domain":"validation","keyword":"type","message":
"instance type (string) does not match any allowed primitive type (allowed:
["object"])}","found":"string","expected":["object"]}]},}]
→ "$outcome": -"MASS_UPDATE_UNITS.KO",}]
→ "$detailType": -"UPDATE_UNIT",}]
→ "$id": -"aaaaaaaabahfpcbnablg6alol3q3l6iaaaa"}

```

ReportDetail

6.1.6.2.Détails du rapport

La partie « OperationSummary » c’est-à-dire le bloc racine du rapport et correspondant au résumé de l’opération est composé des champs suivants :

- « Tenant » : tenant sur lequel l’opération de modification en masse a été lancée
- « evId »:identifiant de l’événement
- « outcome » : statut de l’événement

- « outMsg » : détail du résultat des données de l'événement
- « rightsStatementIdentifiant » : identifiant des données référentielles en vertu desquelles l'opération peut s'exécuter.
- « evDetData » : détails des données de l'événement

La partie « ReportSummary » c'est-à-dire le bloc situé sous la racine du rapport et correspondant au résumé du rapport est composé 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 de l'opération (dernier evDateTime de l'event master de l'opération dans le journal des opérations)
- « reportType » : correspond au modèle du rapport.
- « vitamResults » : correspond au nombre de OK, KO et 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 ce champ est vide.

La partie « Context » correspond à la requête DSL utilisée pour créer la distribution sur chaque unité archivistique.

La partie « ReportDetail » contient les détails de l'opération de modification en masse effectuée sur chacune des unités archivistiques, uniquement quand ces dernières sont aux statuts « KO » ou « Warning ».

- « _id » : identifiant de la ligne d'erreur dans la base MongoDB
- « resultKey » : information sur le problème lié au KO
- « processId » : identifiant de l'opération
- « _tenant » : tenant sur lequel l'opération de modification en masse a été lancée
- « status » : statut l'opération
- « message » message exacte de l'erreur
- « outcome » : statut de l'événement
- « detailType » : type de rapport dans la base de données MongoDB
- « id » : identifiant de l'unité archivistique

6.2.Workflow de mise à jour de masse des métadonnées de gestion des unités archivistiques

Cette section décrit le processus permettant d'effectuer des actions sur un grand nombre d'unités archivistiques stockées dans la solution logicielle Vitam. Cette fonctionnalité nécessite d'avoir les droits requis pour pouvoir intervenir sur les métadonnées. Les autorisations de modifications en masse portent soit sur les métadonnées descriptives soit sur les métadonnées de gestion.

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

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.

6.2.1.Processus de mise à jour en masse des métadonnées de gestion des unités archivistiques (MASS_UPDATE_UNIT_RULE)

Le processus de mise à jour en masse des métadonnées descriptives des unités archivistiques permet d'effectuer des modifications sur un ensemble conséquent d'archives en une seule opération.

- **Règle** : opération consistant à mettre à jour en masse les métadonnées descriptives et de gestion des unités archivistiques
- **Type** : bloquant
- **Statuts** :
 - OK : la mise à jour en masse des métadonnées descriptives et de gestion des unités archivistiques a bien été effectuée (MASS_UPDATE_UNIT_RULE.OK = Succès de la mise à jour en masse des métadonnées descriptives et de gestion des unités archivistiques)
 - KO : la mise à jour en masse des métadonnées descriptives et de gestion des unités archivistiques n'a pas été effectuée en raison d'une erreur (MASS_UPDATE_UNIT_RULE.KO = Échec de la mise à jour en masse des métadonnées descriptives et de gestion des unités archivistiques)
 - FATAL : une erreur technique est survenue lors de la mise à jour en masse des métadonnées descriptives et de gestion des unités archivistiques (MASS_UPDATE_UNIT_RULE.FATAL = Erreur technique lors de la mise à jour en masse des métadonnées descriptives et de gestion des unités archivistiques)

6.2.2.Processus de préparation de la liste des unités archivistiques à mettre à jour et des autorisations de modification STP_CHECK_AND_COMPUTE

- **Règle** : étape consistant à préparer le processus de préparation de la liste des unités archivistiques à mettre à jour et à vérifier les autorisations de modification
- **Type** : bloquant
- **Statuts** :
 - OK : le processus de préparation de la liste des unités archivistiques à mettre à jour et de vérification des droits de modification a bien été effectué (STP_CHECK_AND_COMPUTE.OK = Succès du processus de préparation de la liste des unités archivistiques à mettre à jour et de vérification des autorisations de modification)
 - KO : le processus de préparation de la liste des unités archivistiques à mettre à jour et de vérification des droits de modification n'a pas été effectué en raison d'une erreur (STP_CHECK_AND_COMPUTE.KO = Échec du processus de préparation de la liste des unités archivistiques à mettre à jour et de vérification des autorisations de modification)
 - FATAL : une erreur technique est survenue lors du processus de préparation de la liste des unités

archivistiques à mettre à jour et de vérification des droits de modification (STP_CHECK_AND_COMPUTE.FATAL = Erreur technique lors du processus de préparation de la liste des unités archivistiques à mettre à jour et de vérification des autorisations de modification)

6.2.2.1. Vérification des droits de mise à jour des métadonnées descriptives des unités archivistiques MASS_UPDATE_RULES_CHECK

- **Règle** : tâche consistant à contrôler les droits d'écriture donnés par le contrat d'accès et à vérifier que l'utilisateur a bien des droits d'écriture des métadonnées descriptives et de gestion
- **Type** : bloquant
- **Statuts** :
 - OK : la vérification des droits de mise à jour des métadonnées des unités archivistiques a bien été effectuée (MASS_UPDATE_RULES_CHECK.OK = Succès de la vérification des droits de mise à jour des métadonnées des unités archivistiques)
 - KO : l'utilisateur n'a pas les droits de mise à jour des métadonnées descriptives et de gestion des unités archivistiques (MASS_UPDATE_RULES_CHECK.KO = Échec de la vérification des droits de mise à jour des métadonnées descriptives et de gestion des unités archivistiques)
 - FATAL : une erreur technique est survenue lors de la vérification des droits de mise à jour des métadonnées descriptives et de gestion des unités archivistiques (MASS_UPDATE_RULES_CHECK.FATAL = Erreur technique lors de la vérification des droits de mise à jour des métadonnées descriptives et de gestion des unités archivistiques)

6.2.2.2. Vérification de cohérence des actions lors de la mise à jour des métadonnées descriptives et de gestion des unités archivistiques UNIT_METADATA_CHECK_CONSISTENCY

- **Règle** : tâche consistant à :
 - vérifier l'existence des métadonnées de gestion demandées ainsi que leur cohérence avec leur catégorie de règle
 - vérifier les niveaux de classification associés aux unités archivistiques. Ces niveaux doivent exister dans la liste des niveaux de classifications autorisés par la plateforme (paramètre configuré dans la configuration des workers). Pour les unités archivistiques sans niveau de classification, la vérification contrôle que la plateforme autorise le versement d'unités archivistiques ne déclarant pas de niveau de classification.
 - vérifier que le profil d'unité archivistique existe dans le référentiel interne, est référencé au statut « Actif » et n'a pas un schéma de contrôle vide
- **Type** : bloquant
- **Statuts** :
 - OK : la vérification de cohérence des actions lors de la mise à jour des métadonnées descriptives et de gestion des unités archivistiques a bien été effectuée (UPDATE_UNIT_RULES_CONSISTENCY.OK = Succès de la vérification de cohérence des actions lors de la mise à jour des métadonnées descriptives et de gestion des unités archivistiques)
 - KO : la vérification de cohérence des actions lors de la mise à jour des métadonnées descriptives et de gestion des unités archivistiques n'a pas été effectuée en raison d'une erreur (UPDATE_UNIT_RULES_CONSISTENCY.KO = Échec de la vérification de cohérence des actions lors de la mise à jour des métadonnées descriptives et de gestion des unités archivistiques)
 - FATAL : une erreur technique est survenue lors de la vérification de cohérence des actions lors de la mise à jour des métadonnées descriptives et de gestion des unités archivistiques (UPDATE_UNIT_RULES_CONSISTENCY.FATAL = Erreur technique lors de la vérification de cohérence des actions lors de la mise à jour des métadonnées descriptives et de gestion des unités archivistiques)

6.2.2.3. Vérification des seuils de limitation de traitement des unités archivistiques CHECK_DISTRIBUTION_THRESHOLD

- **Règle** : Vérification des 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 effectué (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)

6.2.2.4. Préparation de la liste des unités archivistiques à mettre à jour PREPARE_UPDATE_UNIT_LIST

- **Règle** : tâche consistant à préparer la liste des unités archivistiques à mettre à jour
- **Type** : bloquant
- **Statuts** :
 - OK : la préparation de la liste des unités archivistiques à mettre à jour a bien été effectuée (PREPARE_UPDATE_UNIT_LIST.OK = Succès de la préparation de la liste des unités archivistiques à mettre à jour)
 - KO : la préparation de la liste des unités archivistiques à mettre à jour n'a pas été effectuée en raison d'une erreur (PREPARE_UPDATE_UNIT_LIST.KO = Échec de la préparation de la liste des unités archivistiques à mettre à jour)
 - FATAL : une erreur technique est survenue lors de la préparation de la liste des unités archivistiques à mettre à jour (PREPARE_UPDATE_UNIT_LIST.FATAL = Erreur technique lors de la préparation de la liste des unités archivistiques à mettre à jour)

6.2.2.5. Établissement de la liste des unités archivistiques à mettre à jour – COMPUTE_INHERITED_RULES_PROGENY_IDENTIFIER (fichier de sortie : GUID/PROCESSING/unitsToInvalidate.json)

- **Règle** : tâche consistant à établir la liste des unités archivistiques dont les règles de gestion applicables sont à invalider. Pour chaque unité archivistique concernée, un fichier est créé et déposé sur l'espace de travail interne (le workspace) pour pouvoir être traité plus tard dans le workflow.
- **Type** : bloquant
- **Statuts** :
 - OK : la liste des unités archivistiques et les fichiers associés ont bien pu être créés. Les fichiers associés ont bien été créés (COMPUTE_INHERITED_RULES_PROGENY_IDENTIFIER .OK = Succès lors du processus d'identification des unités archivistiques dont les règles de gestion applicables sont à invalider)
 - FATAL : une erreur technique est survenue lors d'établissement de la liste des unités archivistiques à mettre à jour (COMPUTE_INHERITED_RULES_PROGENY_IDENTIFIER .FATAL = Erreur technique lors du processus d'identification des unités archivistiques dont les règles de gestion applicables sont à invalider)

6.2.3. Invalidation des unités archivistiques dont les règles de gestion sont à invalider STP_INVALIDATE (Distribution sur LIST GUID/unitsToInvalidate.jsonl)

6.2.3.1. Invalidation des règles de gestion applicables dans les unités archivistiques à invalider COMPUTE_INHERITED_RULES_INVALIDATOR

- **Règle** : traitement consistant, pour chaque unité archivistique à invalider, à supprimer les règles de gestion applicable.
- **Type** : bloquant
- **Statuts** :
 - **OK** : les règles de gestion applicables ont bien été invalidés dans les unités archivistiques (COMPUTE_INHERITED_RULES_INVALIDATOR.OK = Succès de la mise à jour des règles de gestion applicables des unités archivistiques).
 - **FATAL** : une erreur technique est survenue lors de la mise à jour de l'unité archivistique (COMPUTE_INHERITED_RULES_INVALIDATOR.FATAL = Erreur technique lors de la mise à jour des règles de gestion des unités archivistiques)Processus de traitement de mise à jour des unités archivistiques STP_UPDATE

6.2.3.2. Préparation de la liste des unités archivistiques à mettre à jour MASS_UPDATE_UNIT_RULES

- **Règle** : tâche et traitement consistant à mettre à jour les métadonnées descriptives et de gestion des unités archivistiques
- **Type** : bloquant
- **Statuts** :
 - **OK** : la mise à jour des métadonnées descriptives et de gestion des unités archivistiques a bien été effectué (MASS_UPDATE_UNITS_RULES.OK = Succès de la mise à jour des métadonnées descriptives et de gestion des unités archivistiques)
 - **KO** : la mise à jour des métadonnées descriptives et de gestion des unités archivistiques n'a pas été effectuée en raison d'une erreur (MASS_UPDATE_UNITS_RULES.KO = Échec de la mise à jour des métadonnées descriptives et de gestion des unités archivistiques)
 - **WARNING** : avertissement lors de la mise à jour des métadonnées descriptives et de gestion des unités archivistiques (MASS_UPDATE_UNITS_RULES.WARNING = Avertissement lors de la mise à jour des métadonnées descriptives et de gestion des unités archivistiques)
 - **FATAL** : une erreur technique est survenue lors de la mise à jour des métadonnées descriptives et de gestion des unités archivistiques (MASS_UPDATE_UNITS_RULES.FATAL = Erreur technique lors de la mise à jour des métadonnées descriptives et de gestion des unités archivistiques)

6.2.4. Processus de génération du rapport de mise à jour des métadonnées descriptives et de gestion des unités archivistiques STP_MASS_UPDATE_FINALIZE

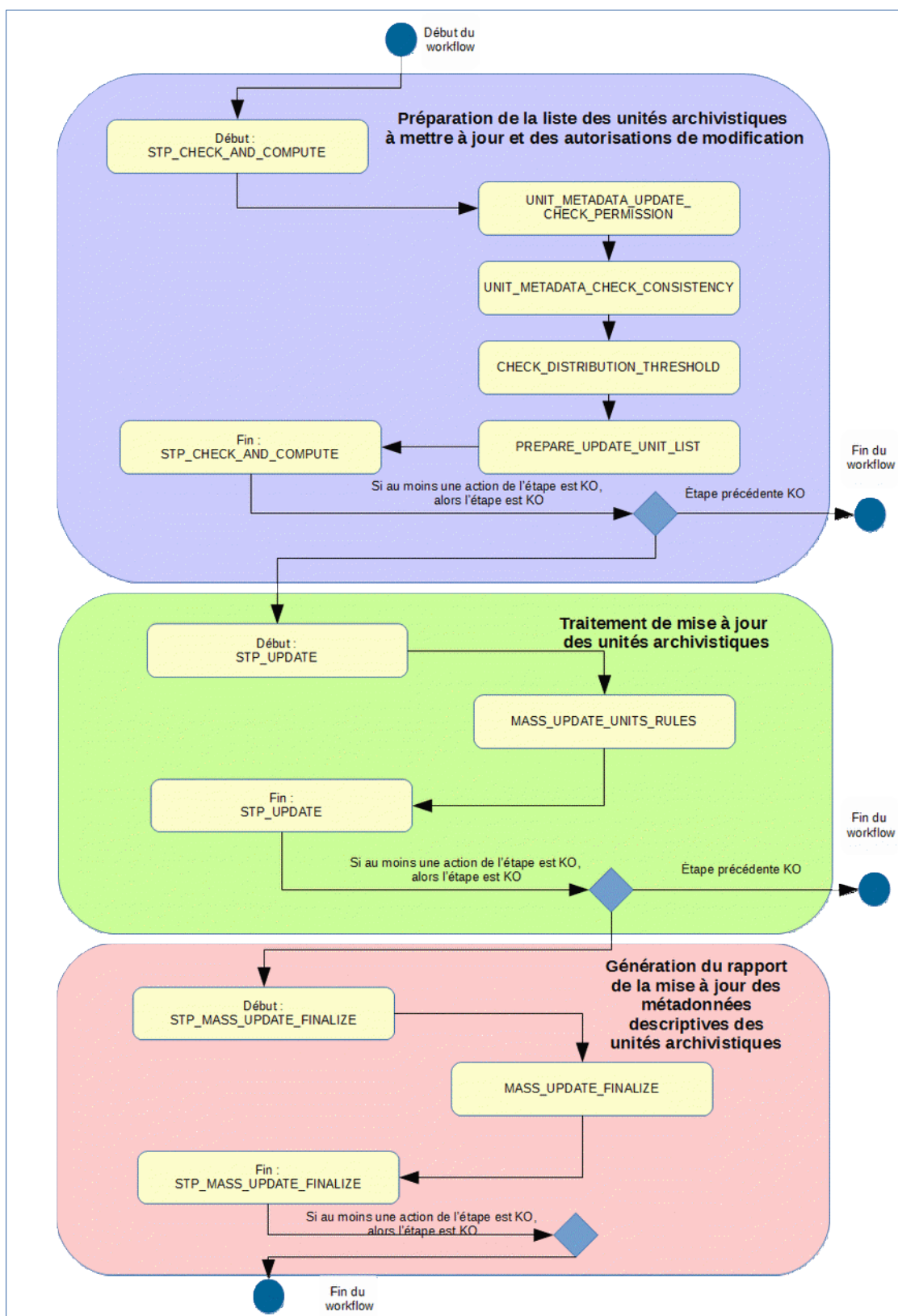
6.2.4.1. Génération du rapport de mise à jour des métadonnées descriptives et de gestion des unités archivistiques MASS_UPDATE_FINALIZE

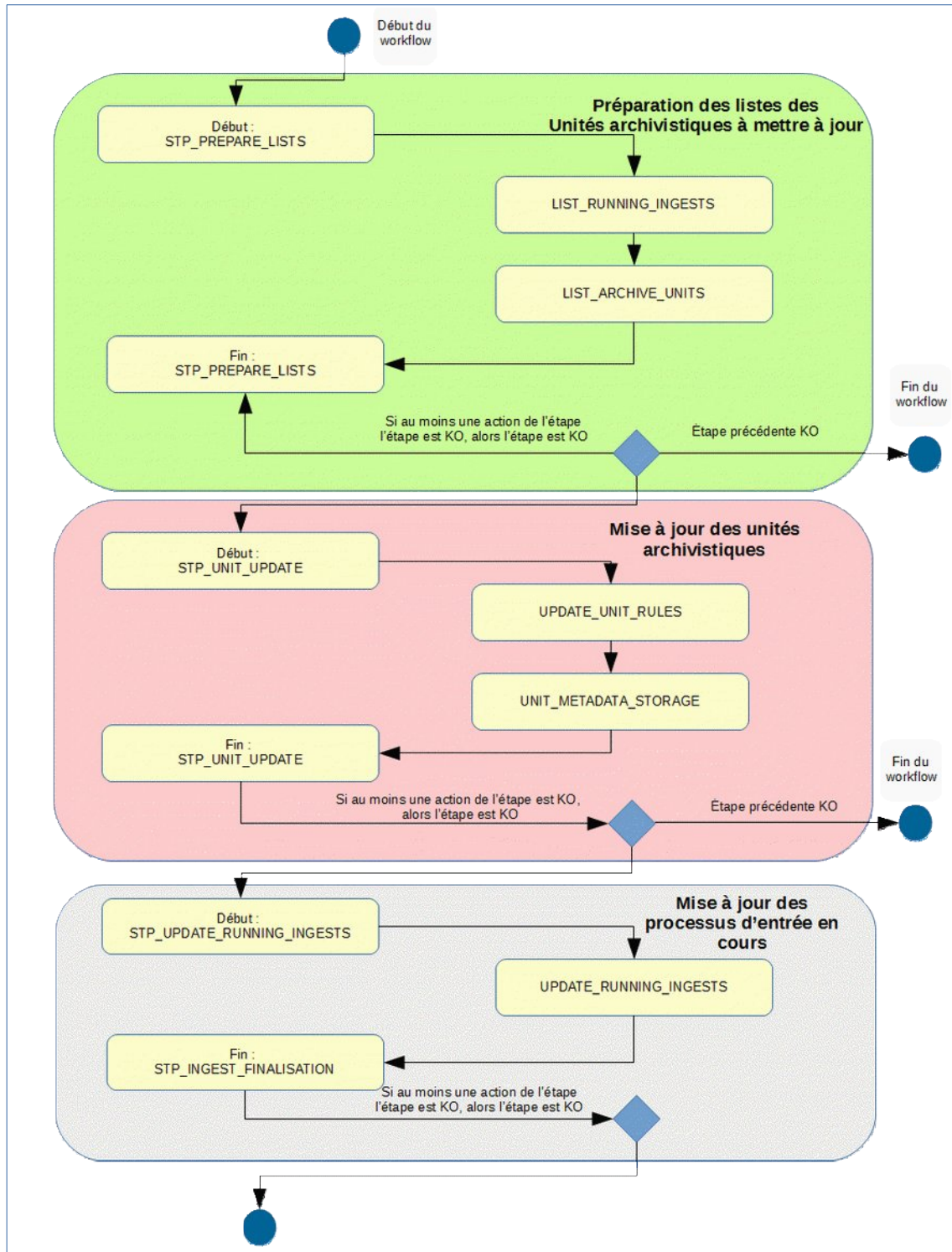
- **Règle** : tâche consistant à générer le rapport de mise à jour des métadonnées descriptives et de gestion des unités archivistiques
- **Type** : bloquant
- **Statuts** :
 - **OK** : le processus de génération du rapport de mise à jour des métadonnées descriptives et de gestion des unités archivistiques à mettre à jour a bien été effectué (MASS_UPDATE_FINALIZE.OK = Succès du processus de génération du rapport de mise à jour des métadonnées descriptives et de

- gestion des unités archivistiques)
- KO : le processus de génération du rapport de mise à jour des métadonnées descriptives et de gestion des unités archivistiques à mettre à jour n'a pas été effectuée (MASS_UPDATE_FINALIZE.KO = Échec du processus de génération du rapport de mise à jour des métadonnées descriptives et de gestion des unités archivistiques)
 - FATAL : une erreur technique est survenue lors du processus de génération du rapport de mise à jour des métadonnées descriptives et de gestion des unités archivistiques à mettre à jour (MASS_UPDATE_FINALIZE.FATAL = Erreur technique lors du processus de génération du rapport de mise à jour des métadonnées descriptives et de gestion des unités archivistiques)

6.2.5. Structure de workflow de mise à jour en masse des métadonnées de gestion des unités archivistiques

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





6.2.6.Rapport de modification en masse des métadonnées de gestion des unités archivistiques.

Le rapport de modification en masse des métadonnées de gestion des unités archivistiques est un fichier JSONL généré par la solution logicielle Vitam lorsqu'une opération de modification en masse des métadonnées de gestion se termine. Cette section décrit la manière dont ce rapport est structuré.

6.2.6.1.Exemple de JSONL : rapport de modification en masse des métadonnées descriptives des unités archivistiques



```

→ → → → → $in : {}
→ → → → → → "#allunitaps": {}
→ → → → → {}
→ → → → {}
→ → → {}
→ → → "$depth": 1000
→ → {}
→ {}
→ "$filter": {},
→ "$action": {}
{}

```

```

{}
→ {}
→ "$id": "aaaaaaaaahds2chsbtncalonfotj3iaaaaq",
→ "$resultKey": "CHECK_UNIT_SCHEMA",
→ "$processId": "aaaaaaaaabchfcj4xasr5calonfbs2ngaaaaq",
→ "$tenant": "B",
→ "$status": "KO",
→ "$message": "Archive unit profile validation failed: Document schema validation
failed: \n{"validateJson":{"level":"error","schema":
{"loadingURI":"#","pointer":"","instance":
{"pointer":"","domain":"validation","keyword":"additionalProperties","mess
age":"object instance has properties which are not allowed by the schema:
[[{"management","CreatedDate","RegisteredDate"}]"],"unwanted":
[{"management","CreatedDate","RegisteredDate"}]":{"level":"error","schema":
{"loadingURI":"#","pointer":"","instance":
{"pointer":"","domain":"validation","keyword":"required","message":"obje
ct has missing required properties:
[["Address","Management","OriginatingSystemId","SentDate"],"
Writer"]],"required":
[["Address","ArchiveUnitProfile","Description","DescriptionLevel","Management
","OriginatingSystemId","ReceivedDate","SentDate","Title","Writer"],"missing\
":["Address","Management","OriginatingSystemId","SentDate","Writer"]}]}}",
→ "$outcome": "MASS UPDATE UNITS RULES KO",
→ "$detailType": "UPDATE_UNIT",
→ "$id": "aaagaaabahnbfxqasmqalone7741iaaaaq"
{}

```

6.2.6.2.Détails du rapport

La partie « OperationSummary » c’est-à-dire le bloc racine du rapport et correspondant au résumé de l’opération est composé des champs suivants :

- « Tenant » : tenant sur lequel l’opération de modification en masse a été lancée
- « evId »:identifiant de l’événement
- « outcome » : statut de l’événement
- « outMsg » : détail du résultat des données de l’événement
- « rightsStatementIdentifiant » : identifiant des données référentielles en vertu desquelles l’opération peut s’exécuter.
- « evDetData » : détails des données de l’événement

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

- « evStartDateTime »: date du début de l'opération (evDateTime de l'événement master de l'opération dans le journal des opérations)
- « evEndDateTime » : date de fin de l'opération (dernier evDateTime de l'événement master de l'opération dans le journal des opérations)
- « reportType » : correspond au modèle du rapport.
- « vitamResults » : correspond au nombre de OK, KO et 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 ce champ est vide.

La partie « Context » correspond à la requête DSL utilisée pour créer la distribution sur chaque unité archivistique.

La partie « ReportDetail » contient les détails de l'opération de modification en masse effectuée sur chacune des unités archivistiques, uniquement quand ces dernières sont aux statuts « KO » ou « Warning ».

- « _id » : identifiant de la ligne d'erreur dans la base MongoDB
- « resultKey » : information sur le problème lié au KO
- « processId » : identifiant de l'opération
- « _tenant » : tenant sur lequel l'opération de modification en masse a été lancée
- « status » : statut l'opération
- « message » message exacte de l'erreur
- « outcome » : statut de l'événement
- « detailType » : type de rapport dans la base de données MongoDB
- « id » : identifiant de l'unité archivistique

6.3. Workflow de calcul des règles de gestion hérités des unités archivistiques pour faciliter les requêtes

Cette section décrit le processus (workflow) permettant le calcul des règles de gestion des unités archivistiques afin de faciliter et d'accélérer les requêtes des unités archivistiques en fonction des règles de gestions effectifs.

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

Toutes les étapes, tâches et traitements ne sont pas 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.

6.3.1. Processus de préparation des règles de gestion applicables d'unités archivistiques à mettre à jour STP_COMPUTE_INHERITED_RULES_PREPARATION

6.3.1.1. 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)

6.3.1.2. Préparation du calcul des règles de gestion applicables des unités archivistiques COMPUTE_INHERITED_RULES_PREPARATION

- **Règle** : étape consistant à préparer le calcul des règles de gestion applicables des unités archivistiques.
- **Type** : bloquant
- **Statuts** :
 - OK : la préparation du calcul des règles de gestion applicables des unités archivistiques a bien été effectuée (COMPUTE_INHERITED_RULES_PREPARATION.OK = Succès du processus de préparation du calcul des règles de gestion applicables).
 - FATAL : la préparation du calcul des règles de gestion applicables des unités archivistiques n'a pas pu être effectuée. (COMPUTE_INHERITED_RULES_PREPARATION.FATAL = Erreur technique lors du processus de préparation du calcul des règles de gestion applicables)

6.3.2. Processus de calcul des règles de gestion applicables. STP_COMPUTE_INHERITED_RULES

6.3.2.1. Calcul des règles de gestion applicables des unités archivistiques – COMPUTE_INHERITED_RULES (pas journalisé dans la base de données MongoDB)

- **Règle** : étape consistant à calculer et persister les règles de gestion applicables des unités archivistiques.
- **Type** : bloquant
- **Statuts** :
 - OK : les règles de gestion applicables des unités archivistiques ont bien été calculées et persistées. (COMPUTE_INHERITED_RULES.OK = Succès du calcul des règles de gestion applicables).
 - KO : les règles de gestion applicables des unités archivistiques n'ont pas pu être calculées ou persistées. (COMPUTE_INHERITED_RULES.KO = Échec du calcul des règles de gestion applicables)

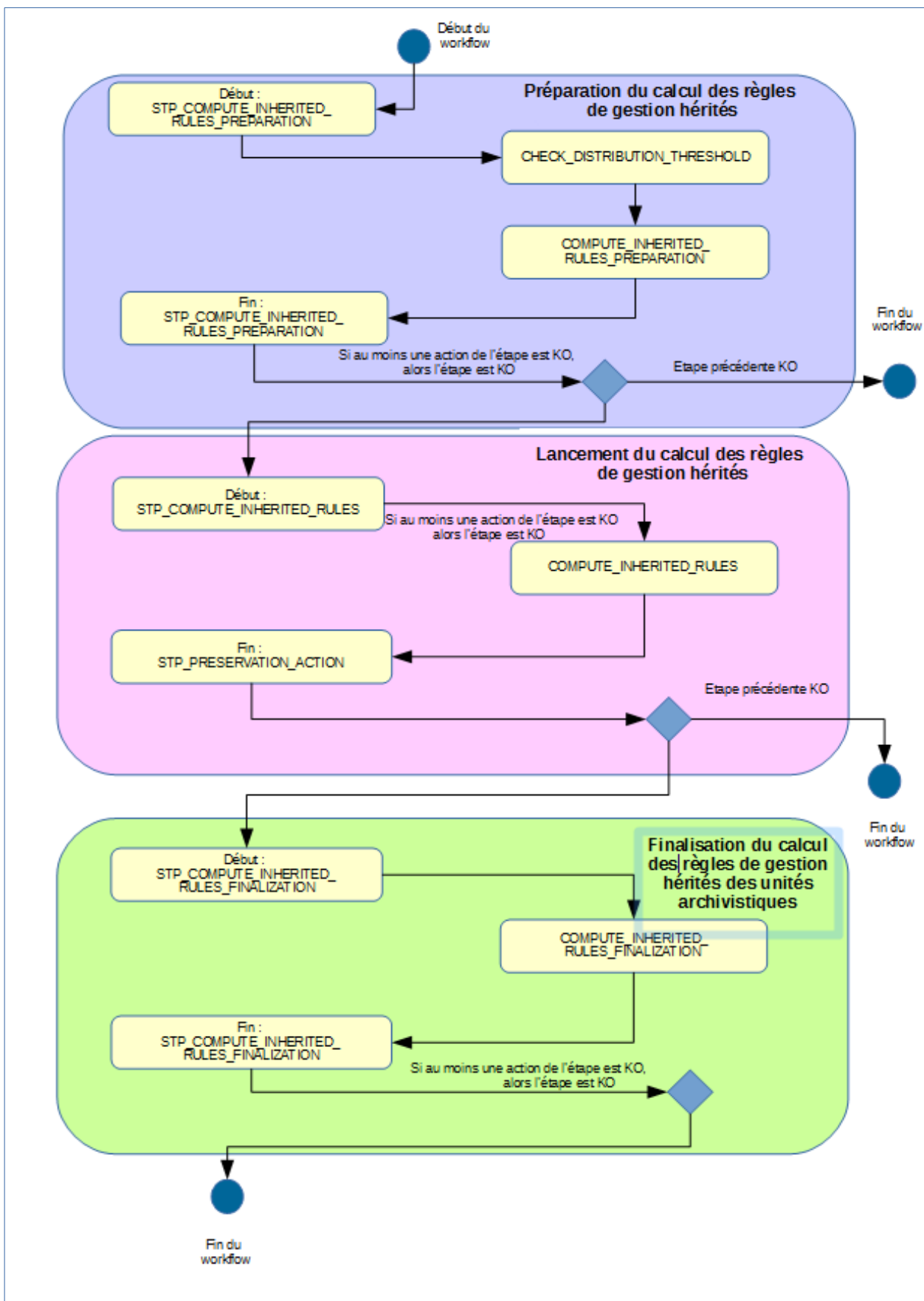
6.3.3. Processus de finalisation du calcul des règles de gestion applicables. STP_COMPUTE_INHERITED_RULES_FINALIZATION

6.3.3.1. Finalisation du calcul des règles de gestion applicables – COMPUTE_INHERITED_RULES_FINALIZATION

- **Règle** : étape consistant à finaliser le calcul des règles de gestion applicables des unités archivistiques.
- **Type** : bloquant
- **Statuts** :
 - OK : la finalisation du calcul des règles de gestion applicables a bien été effectuée (COMPUTE_INHERITED_RULES_FINALIZATION .OK = Succès du processus de finalisation du calcul des règles de gestion applicables).
 - FATAL : une erreur technique est survenue lors de la finalisation du calcul des règles de gestion applicables (COMPUTE_INHERITED_RULES_FINALIZATION .FATAL = Erreur technique lors du processus de finalisation du calcul des règles de gestion applicables).

6.3.4. Structure de workflow de mise à jour du calcul des règles de gestion hérités des unités archivistiques.

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



6.4. Workflow de suppression des règles de gestion hérités et calculés des unités archivistiques

Cette section décrit le processus (workflow) permettant la suppression des règles de gestion des unités archivistiques applicables dans les unités archivistiques.

Le workflow mis en place dans la solution logicielle Vitam est défini dans le fichier « ComputedInheritedRulesDeleteWorkflow.json ». Ce fichier est disponible dans : 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.

6.4.1. Processus de préparation des règles de gestion applicables d'unités archivistiques à mettre à jour STP_COMPUTE_INHERITED_RULES_PREPARATION

6.4.1.1. 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)

6.4.1.2. Préparation du calcul des règles de gestion applicables des unités archivistiques COMPUTE_INHERITED_RULES_PREPARATION

- **Règle** : étape consistant à préparer le calcul des règles de gestion applicables des unités archivistiques.
- **Type** : bloquant
- **Statuts** :
 - OK : la préparation du calcul des règles de gestion applicables des unités archivistiques a bien été effectuée (COMPUTE_INHERITED_RULES_PREPARATION.OK = Succès du processus de préparation du calcul des règles de gestion applicables).
 - FATAL : la préparation du calcul des règles de gestion applicables des unités archivistiques n'a pas pu être effectuée. (COMPUTE_INHERITED_RULES_PREPARATION.FATAL = Erreur technique lors du processus de préparation du calcul des règles de gestion applicables)
 -

6.4.2. Processus de suppression des règles de gestion applicables. STP_COMPUTE_INHERITED_RULES_DELETE

6.4.2.1. suppression des règles de gestion applicables des unités archivistiques – COMPUTE_INHERITED_RULES_DELETE

- **Règle** : étape consistant supprimer les règles de gestion applicables des unités archivistiques.
- **Type** : bloquant
- **Statuts** :
 - OK : les règles de gestion applicables des unités archivistiques ont bien été supprimés. (COMPUTE_INHERITED_RULES_DELETE.OK = Succès du processus de suppression du calcul des règles de gestion applicables
 -).
 - KO : les règles de gestion applicables des unités archivistiques n'ont pas pu être suppression. (COMPUTE_INHERITED_RULES_DELETE.KO = Échec du calcul des règles de gestion applicables)

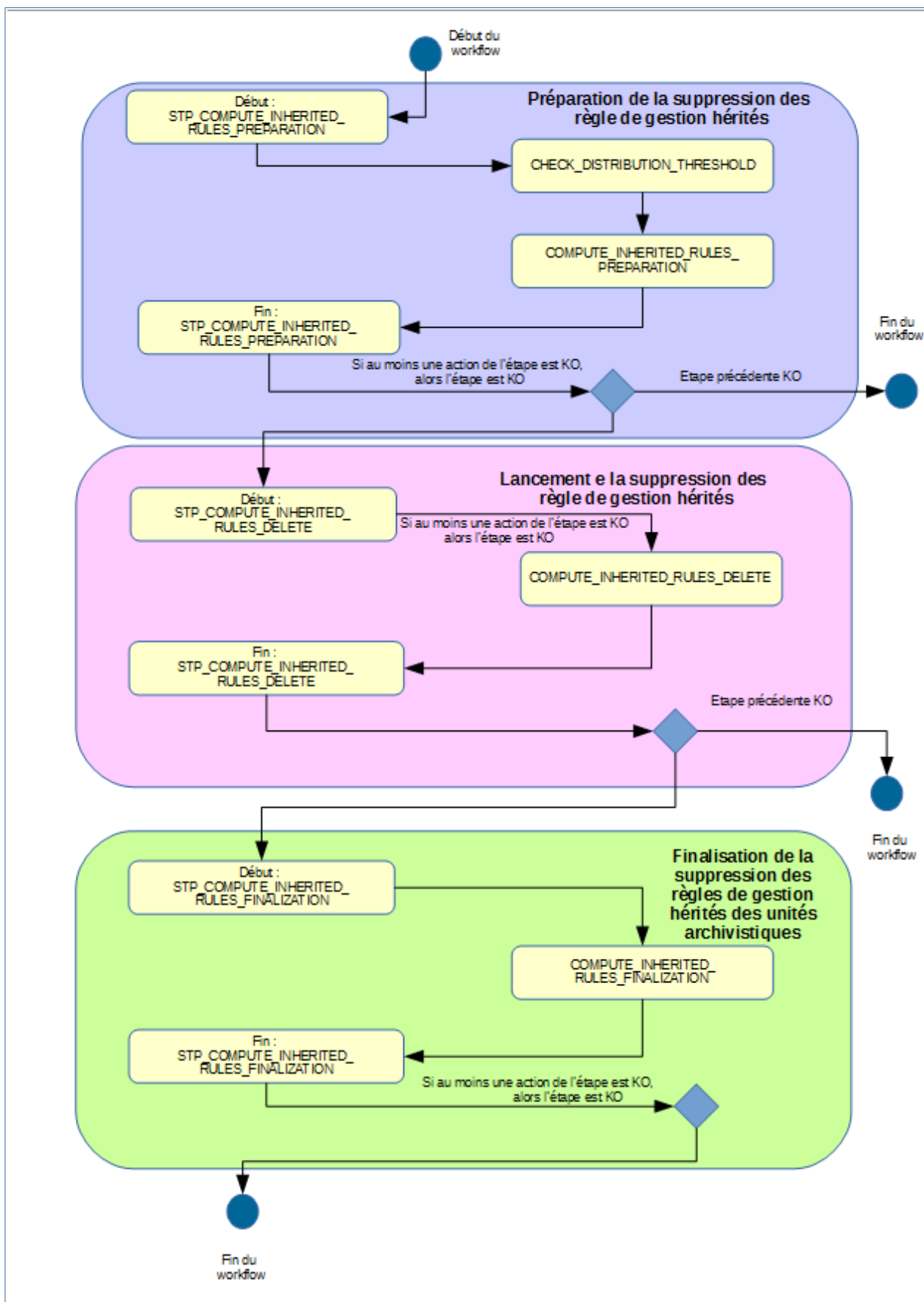
6.4.3. Processus de finalisation de la suppression des règles de gestion applicables. STP_COMPUTE_INHERITED_RULES_FINALIZATION

● Finalisation du calcul des règles de gestion applicables – COMPUTE_INHERITED_RULES_FINALIZATION

- **Règle** : étape consistant à finaliser la suppression des règles de gestion applicables des unités archivistiques.
- **Type** : bloquant
- **Statuts** :
 - OK : la finalisation du calcul des règles de gestion applicables a bien été effectuée (COMPUTE_INHERITED_RULES_FINALIZATION .OK = Succès du processus de finalisation du calcul des règles de gestion applicables
 -).
 - FATAL : une erreur technique est survenue lors de la finalisation du calcul des règles de gestion applicables (COMPUTE_INHERITED_RULES_FINALIZATION .FATAL = Erreur technique lors du processus de finalisation du calcul des règles de gestion applicables)

6.4.4. Structure de workflow de suppression du calcul des règles de gestion hérités des unités archivistiques.

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



7. MISE À JOUR UNITAIRE DE MASSE (BULK UPDATE)

Workflow de mise à jour unitaire de masse des métadonnées descriptives des unités archivistiques

Cette section décrit le processus permettant d'effectuer des actions unitaires sur un grand nombre d'unités archivistiques stockées dans la solution logicielle Vitam. Cette fonctionnalité nécessite d'avoir les droits requis pour pouvoir intervenir sur les métadonnées. Les autorisations de modifications en masse portent sur les métadonnées descriptives.

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.

7.1.Processus de mise à jour unitaire en masse des métadonnées descriptives des unités archivistiques (BULK_ATOMIC_UPDATE_UNIT_DESC)

Le processus de mise à jour unitaire en masse des métadonnées descriptives des unités archivistiques permet d'effectuer des modifications unitaires différenciées sur plusieurs unités archivistiques en une seule opération.

- **Règle** : opération consistant à mettre à jour unitairement en masse les métadonnées descriptives des unités archivistiques
- **Type** : bloquant
- **Statuts** :
 - OK: le mise à jour unitaire en masse des métadonnées descriptives des unités archivistiques ont bien été effectuées (BULK_ATOMIC_UPDATE_UNIT_DESC.OK = Succès du processus de mise à jour unitaire en masse des métadonnées descriptives des unités archivistiques)
 - KO: la mise à jour unitaire en masse des métadonnées descriptives des unités archivistiques n'a pas été effectuée en raison d'une erreur (BULK_ATOMIC_UPDATE_UNIT_DESC.KO = Échec du processus de mise à jour unitaire en masse des métadonnées descriptives des unités archivistiques)
 - FATAL: une erreur technique est survenue lors de la mise à jour atomique en masse des métadonnées descriptives des unités archivistiques (BULK_ATOMIC_UPDATE_UNIT_DESC.FATAL = Erreur technique lors du début du processus de mise à jour unitaire en masse des métadonnées descriptives des unités archivistiques)

7.2.Processus de préparation de la liste des unités archivistiques à mettre à jour et des autorisations de modification STP_CHECK_AND_COMPUTE

- **Règle** : étape consistant à préparer la liste des unités archivistiques à mettre à jour et à vérifier les autorisations de modification
- **Type** : bloquant
- **Statuts** :
 - OK : le processus de préparation de la liste des unités archivistiques à mettre à jour et de vérification des droits de modification a bien été effectué (STP_CHECK_AND_COMPUTE.OK = Succès du processus de préparation de la liste des unités archivistiques à mettre à jour et de vérification des autorisations de modification)
 - KO : le processus de préparation de la liste des unités archivistiques à mettre à jour et de vérification des droits de modification n'a pas été effectué en raison d'une erreur (STP_CHECK_AND_COMPUTE.KO = Échec du processus de préparation de la liste des unités archivistiques à mettre à jour et de vérification des autorisations de modification)
 - FATAL : une erreur technique est survenue lors du processus de préparation de la liste des unités archivistiques à mettre à jour et de vérification des autorisations de modification (STP_CHECK_AND_COMPUTE.FATAL = Erreur technique lors du processus de préparation de la

liste des unités archivistiques à mettre à jour et de vérification des autorisations de modification)

7.2.1. Vérification des seuils de limitation des requêtes de traitement des unités archivistiques CHECK_QUERIES_TRESHOLD

- **Règle** : tâche consistant à vérifier les seuils de limitation des requêtes de traitement des unités archivistiques.
- **Type** : bloquant
- **Statuts** :
 - OK: la vérification des seuils de limitation de traitement des requêtes de mise à jour des unités archivistiques a bien été effectuée (CHECK_DISTRIBUTION_THRESHOLD.OK = Succès de la vérification des seuils de limitation des requêtes de traitement des unités archivistiques)
 - KO: une incohérence a été détectée entre le seuil et le nombre de requêtes de mise à jour des unités archivistiques à traiter (CHECK_DISTRIBUTION_THRESHOLD.KO = Échec de la vérification des seuils de limitation des requêtes de traitement des unités archivistiques)
 - WARNING: une incohérence a été détectée entre le seuil fourni en entrée et le seuil par défaut (CHECK_DISTRIBUTION_THRESHOLD.WARNING = Avertissement lors de la vérification des seuils de limitation des requêtes de traitement des unités archivistiques)
 - FATAL: une erreur technique est survenue lors de la vérification des seuils de limitation de traitement des requêtes de mise à jour des unités archivistiques (CHECK_DISTRIBUTION_THRESHOLD.FATAL = Erreur technique lors de la vérification des seuils de limitation des requêtes de traitement des unités archivistiques)

7.2.2. Préparation de la liste des unités archivistiques à mettre à jour unitairement PREPARE_BULK_ATOMIC_UPDATE_UNIT_LIST

- **Règle** : tâche consistant à préparer la liste des unités archivistiques à mettre à jour. Lors de cette préparation vont être vérifiées pour chaque requête de mise à jour sa validité (pas de champ interdit “_”, valide selon le DSL) et que son nombre de résultats n’est pas EXACTEMENT égal à 1 unité archivistique.
- **Type** : bloquant
- **Statuts** :
 - OK: la préparation de la liste des unités archivistiques à mettre à jour unitairement a bien été effectuée (PREPARE_BULK_ATOMIC_UPDATE_UNIT_LIST.OK = Succès de la préparation de la liste des unités archivistiques à mettre à jour unitairement)
 - KO: la préparation de la liste des unités archivistiques à mettre à jour unitairement n’a pas été effectuée en raison d’une erreur (PREPARE_BULK_ATOMIC_UPDATE_UNIT_LIST.KO = Échec de la préparation de la liste des unités archivistiques à mettre à jour unitairement)
 - WARNING: un avertissement est survenu lors de la préparation de la liste des unités archivistiques à mettre à jour unitairement (PREPARE_BULK_ATOMIC_UPDATE_UNIT_LIST.WARNING = Avertissement lors de la préparation de la liste des unités archivistiques à mettre à jour unitairement)
 - FATAL: une erreur technique est survenue lors de la préparation de la liste des unités archivistiques à mettre à jour unitairement (PREPARE_BULK_ATOMIC_UPDATE_UNIT_LIST.FATAL = Erreur technique lors de la préparation de la liste des unités archivistiques à mettre à jour unitairement)

7.3. Processus de traitement de mise à jour unitaire des unités archivistiques STP_UPDATE

Mise à jour unitaire des métadonnées descriptives BULK_ATOMIC_UPDATE_UNITS

- **Règle** : tâche et traitement consistant à mettre à jour les métadonnées descriptives des unités archivistiques

- **Type** : non bloquant
- **Statuts** :
 - OK: la mise à jour unitaire des métadonnées descriptives des unités archivistiques a bien été effectué (BULK_ATOMIC_UPDATE_UNITS.OK = Succès de la mise à jour unitaire des unités archivistiques)
 - KO: la mise à jour unitaire des métadonnées descriptives des unités archivistiques n'a pas été effectuée en raison d'une erreur (BULK_ATOMIC_UPDATE_UNITS.KO = Échec de la mise à jour des métadonnées descriptives des unités archivistiques)
 - WARNING: un avertissement est survenu lors de la mise à jour unitaire des métadonnées descriptives des unités archivistiques (BULK_ATOMIC_UPDATE_UNITS.WARNING = Avertissement lors de la mise à jour unitaire des unités archivistiques)
 - FATAL: une erreur technique est survenue lors de la mise à jour unitaire des métadonnées descriptives des unités archivistiques (BULK_ATOMIC_UPDATE_UNITS.FATAL = Erreur technique lors de la mise à jour des métadonnées descriptives des unités archivistiques)

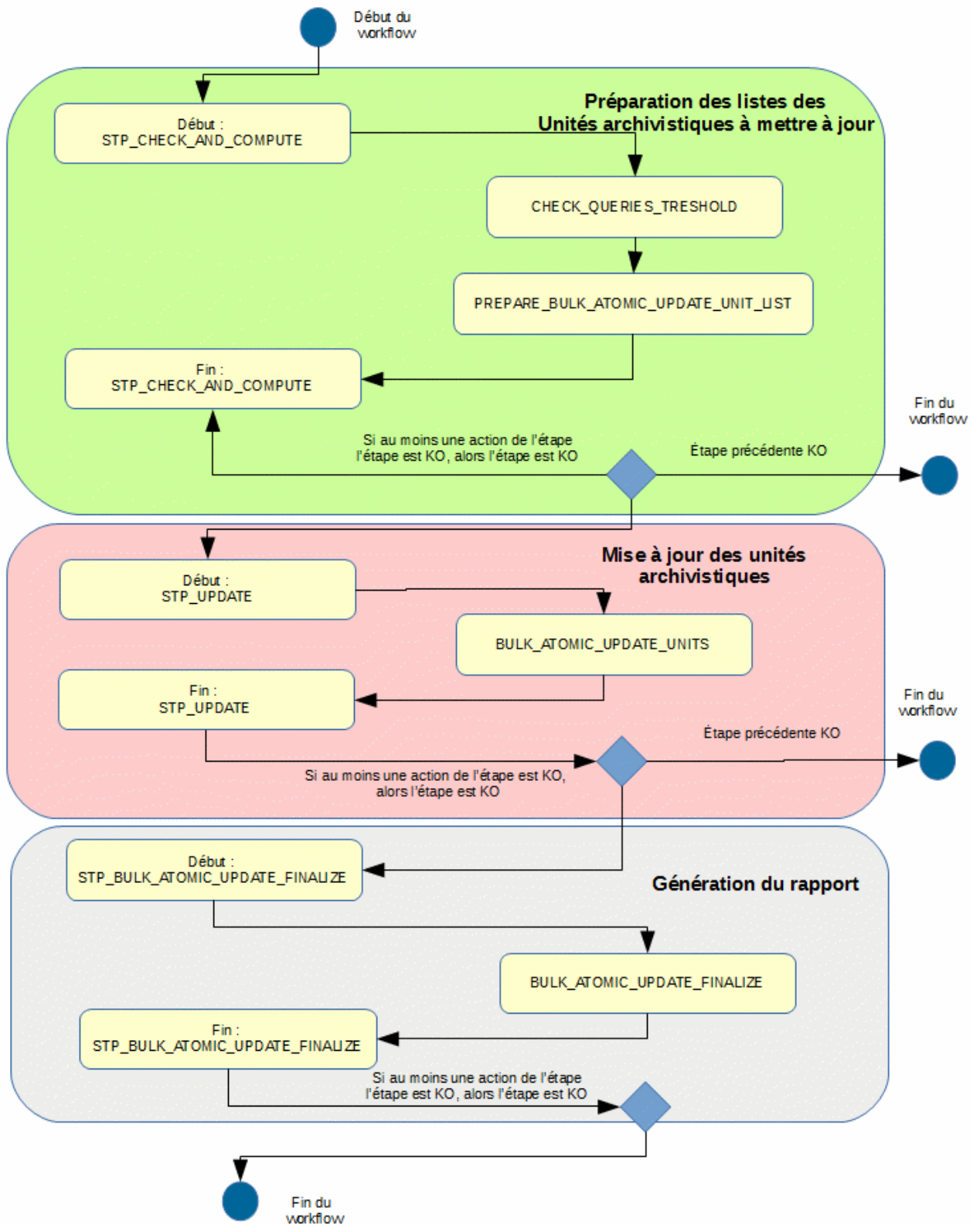
7.4.Processus de génération du rapport de mise à jour unitaire des métadonnées descriptives des unités archivistiques **STP_BULK_ATOMIC_UPDATE_FINALIZE**

Génération du rapport de mise à jour unitaire des métadonnées descriptives des unités archivistiques BULK_ATOMIC_UPDATE_FINALIZE

- **Règle** : tâche consistant à générer le rapport de mise à jour unitaire des métadonnées descriptives des unités archivistiques
- **Type** : bloquant
- **Statuts** :
 - OK: le processus de génération du rapport de mise à jour unitaire des métadonnées descriptives des unités archivistiques a bien été effectué (BULK_ATOMIC_UPDATE_FINALIZE.OK = Succès du processus de génération du rapport de mise à jour des métadonnées descriptives des unités archivistiques)
 - KO: le processus de génération du rapport de mise à jour unitaire des métadonnées descriptives des unités archivistiques à mettre à jour n'a pas été effectuée en raison d'une erreur (BULK_ATOMIC_UPDATE_FINALIZE.KO = Échec du processus de génération du rapport de mise à jour des métadonnées descriptives des des unités archivistiques)
 - FATAL: une erreur technique est survenue lors du processus de génération du rapport de mise à jour unitaire des métadonnées descriptives des des unités archivistiques à mettre à jour (BULK_ATOMIC_UPDATE_FINALIZE.FATAL = Erreur technique lors du processus de génération du rapport de mise à jour des métadonnées descriptives des unités archivistiques). Toutes les lignes de détail sont exportées

7.5. Structure du Workflow de mise à jour unitaire des métadonnées descriptive

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



7.6. Rapport de modification unitaire en masse des métadonnées descriptives des unités archivistiques.

Le rapport de modification unitaire en masse des métadonnées descriptives des unités archivistiques est un fichier JSONL généré par la solution logicielle Vitam lorsqu'une opération de modification unitaire en masse des métadonnées descriptives se termine. Cette section décrit la manière dont ce rapport est structuré.

7.6.1. Exemple de fichier JSONL : rapport de modification unitaire en masse de métadonnées descriptives

```

{"tenant":9,
 "evId":"aeaaaaabghcnjz2aarx6alxq6aizyaaaaaq",
 "evType":"BULK_ATOMIC_UPDATE_UNITS",
 "outcome":"WARNING",
 "outDetail":"BULK_ATOMIC_UPDATE_UNITS.WARNING",
 "outMsg":"Avertissement lors de la mise \u00E9 jour unitaire des unit\u00E9s archivistiques Detail= OK:1 WARNING:1",
 "rightsStatementIdentifier":{"AccessContract":"AC-000001"},
 "evDetData":{"Event":"Bulk atomic update OK"}
}

{"evStartDateTime":"2021-02-09T15:54:44.879",
 "evEndDateTime":"2021-02-09T15:54:46.595",
 "reportType":"BULK_UPDATE_UNIT",
 "vitamResults":{"OK":1,"KO":0,"WARNING":3,"total":4},
 "extendedInfo":null
}

{}

{"_id":"aeaaaaaaah15ctvabavcalxq6ajepaaaaaq",
 "resultKey":"UNIT_METADATA_NO_NEW_DATA",
 "processId":"aeaaaaabghcnjz2aarx6alxq6aizyaaaaaq",
 "_tenant":9,
 "status":"WARNING",
 "query":{"$query":{"$seq":{"ArchivalAgencyArchiveUnitIdentifier":{"Value3"}}, "$action":{"$set":{"Description doit h\u00E9riter de ManagementMetadata de la r\u00E9gion ACC-00002 avec pour StartDate 01/01/2000 et a comme r\u00E9gion 01/01/2000","Title":"Fracture"}}}},
 "unitId":"aeaaaaabehcnjz2aac6calxq47fkmyaaadq",
 "message":"Unit not updated.",
 "outcome":"BULK_ATOMIC_UPDATE_UNITS.WARNING",
 "detailType":"BULK_UPDATE_UNIT",
 "id":"3"
}
    
```

7.6.2. Détail du rapport

La partie **OperationSummary**, c'est-à-dire le bloc racine du rapport et correspondant au résumé de l'opération est composé des champs suivants :

- Tenant : tenant sur lequel l'opération de modification unitaire en masse a été lancée
- evId : identifiant de l'événement
- outcome : statut de l'événement
- outMsg : détail du résultat des données de l'événement
- rightsStatementIdentifier : identifiants des données référentielles en vertu desquelles l'opération peut s'exécuter.
- EvDetData : détails des données de l'événement

La partie **ReportSummary**, c'est-à-dire le bloc situé sous la racine du rapport et correspondant au résumé du

rapport est composé des champs suivants :

- evStartDateTime : date du début de l'opération (evDateTime de l'événement master de l'opération dans le journal des opérations)
- evEndDateTime : date de fin de l'opération (dernier evDateTime de l'événement master de l'opération dans le journal des opérations)
- reportType : correspond au modèle du rapport.
- vitamResults: correspond au nombre de OK, KO et 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 ce champ est vide.

La partie **Context** est vide.

La partie **ReportDetail** contient les détails de l'opération de modification en masse effectuée sur chacune des unités archivistiques pour chaque requête fournie en entrée, et ce que le statut soit OK, KO ou WARNING. Les résultats sont ordonnés selon la criticité du statut (KO, puis WARNING, puis OK)

- _id : identifiant de la ligne d'erreur dans la base MongoDB
- resultKey : information sur le statut de résultat
- processId : identifiant de l'opération
- _tenant : tenant sur lequel l'opération de modification en masse a été lancée
- status : statut l'opération
- message : message exact de l'erreur ou du résultat d'exécution
- outcome : statut de l'événement
- detailType : type de rapport dans la base de données MongoDB
- query : requête liée au détail de résultat au format "String"
- id : index de la requête dans la list des requêtes
- unitId : unité archivistique impactée à la requête (présent uniquement si outcome commencer par BULK_ATOMIC_UPDATE_UNITS)

8. EXPORT D'UN DIP

Workflow d'export d'un DIP

Cette section décrit le processus (workflow) d'export, utilisé lors de l'export d'un Dissemination Information Package (DIP) dans la solution logicielle Vitam.

Le workflow d'export de DIP actuel mis en place dans la solution logicielle Vitam est défini dans l'unique fichier *"ExportUnitWorkflow.json"*. Ce fichier est disponible dans `/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.

8.1. Processus de création du bordereau de mise à disposition (STP_CREATE_MANIFEST)

8.1.1. 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)

8.1.2. Création du bordereau CREATE_MANIFEST (CreateManifest.java)

- **Règle** : tâche consistant à créer le bordereau contenant les unités archivistiques soumises au service d'export de DIP, ainsi que les groupes d'objets techniques et objets qui leur sont associés
- **Type** : bloquant
- **Statuts** :
 - OK : le bordereau contenant les descriptions des unités archivistiques, groupes d'objets techniques et objets-données a été créé avec succès (CREATE_MANIFEST.OK = Succès de la création du bordereau de mise à disposition)
 - KO : la création du bordereau contenant les descriptions des unités archivistiques, groupes d'objets techniques et objets-données a échoué, car des informations étaient manquantes, erronées ou inconnues, la version du SEDA demandée est incompatible (CREATE_MANIFEST.KO = Échec de la création du bordereau de mise à disposition)
 - FATAL : une erreur technique est survenue lors de la création du bordereau (CREATE_MANIFEST.FATAL = Erreur technique lors de la création du bordereau de mise à disposition)

8.2. Processus de déplacement des objets binaires vers l'espace de travail interne (STP_PUT_BINARY_ON_WORKSPACE)

8.2.1. Vérification de la disponibilité des ressources **EXPORT_CHECK_RESOURCE_AVAILABILITY** (ExportCheckResourceAvailability.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 (EXPORT_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 (EXPORT_CHECK_RESOURCE_AVAILABILITY.FATAL = Erreur technique lors de la vérification de la disponibilité des objets binaires sur l'offre de stockage)

8.2.2. Déplacement des objets binaires vers le workspace **PUT_BINARY_ON_WORKSPACE** (PutBinaryOnWorkspace.java)

- **Règle** : tâche consistant à déplacer les objets binaires mentionnés dans le bordereau vers l'espace de travail interne (« workspace »)
- **Type** : bloquant
- **Statuts** :
 - OK : les objets binaires ont été déplacés vers le workspace avec succès (PUT_BINARY_ON_WORKSPACE.OK = Succès du déplacement des objets binaires de l'offre de stockage vers l'espace de travail interne)
 - KO : le déplacement des objets binaires vers le workspace a échoué car un ou plusieurs de ces objets étaient introuvables (PUT_BINARY_ON_WORKSPACE.KO = Échec du déplacement des objets binaires de l'offre de stockage vers l'espace de travail interne)
 - FATAL : une erreur technique est survenue lors du déplacement des objets binaires de l'offre de stockage vers le workspace (PUT_BINARY_ON_WORKSPACE.FATAL = Erreur technique lors du déplacement des objets binaires de l'offre de stockage vers l'espace de travail interne)

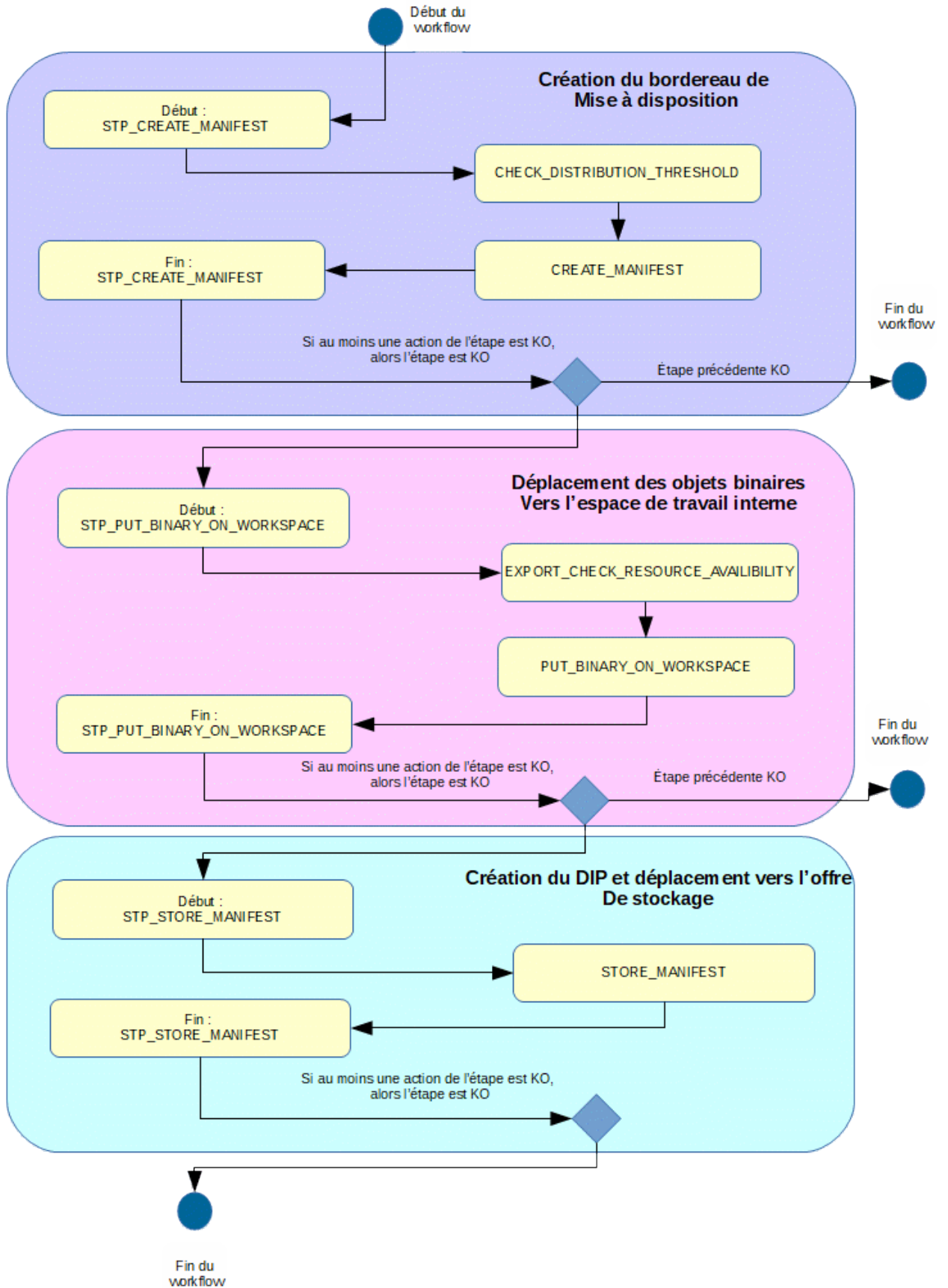
8.3. Processus de création du DIP et de son déplacement vers l'offre de stockage (STP_STORE_MANIFEST)

Stockage du DIP STORE_MANIFEST (StoreDIP.java)

- **Règle** : tâche consistant à créer le DIP et à le déplacer vers l'offre de stockage
- **Type** : bloquant
- **Statuts** :
 - OK : le DIP a été créé et enregistré sur les offres de stockage avec succès (STORE_MANIFEST.OK = Succès du processus de la création du DIP et de son déplacement vers l'offre de stockage)
 - FATAL : une erreur technique est survenue lors de la création et de l'enregistrement du DIP sur les offres de stockage déplacement des objets binaires de stockage vers l'espace de travail interne (STORE_MANIFEST.FATAL = Erreur technique lors de la création du DIP et de son déplacement vers l'offre de stockage)

8.4. Structure du Workflow d'export de DIP

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



9. ÉLIMINATION

L'élimination est un traitement de masse permettant d'évaluer dans un lot conséquent d'unités archivistiques, celles qui sont éliminables (la durée d'utilité administrative est échuée et le sort final déclaré est « Détruire », arbre de positionnement, workflow d'analyse) et de procéder à leur élimination du système (workflow d'action).

Ces deux étapes ne sont pas liées, l'action d'élimination peut être exécutée sans avoir lancé préalablement de phase d'analyse. La phase d'analyse peut servir à déterminer une liste d'unités archivistiques potentiellement éliminables.

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.

9.1. Workflow d'analyse de l'élimination des unités archivistiques

Lors de la phase d'analyse, le système procède à une analyse du caractère éliminable ou non des unités archivistiques, en vérifiant les règles de gestion de celles-ci, qu'elles soient déclarées ou héritées. Le résultat de l'analyse consiste à calculer un statut global de l'unité archivistique au regard de son caractère éliminable, et à l'indexer, si son statut est DESTROY ou CONFLICT

9.1.1. Processus de préparation de l'analyse de l'élimination des unités archivistiques (STP_ELIMINATION_ANALYSIS_PREPARATION)

- **Règle** : tâche consistant à préparer l'analyse du caractère éliminable ou non des unités archivistiques
- **Type** : bloquant
- **Statuts** :
 - OK : la préparation de l'analyse du caractère éliminable ou non des unités archivistiques a bien été effectuée (STP_ELIMINATION_ANALYSIS_PREPARATION.OK = Succès de la préparation de l'analyse du caractère éliminable ou non des unités archivistiques)
 - KO : la préparation de l'analyse du caractère éliminable ou non des unités archivistiques n'a pas été effectuée (STP_ELIMINATION_ANALYSIS_PREPARATION.KO = Échec de la préparation de l'analyse du caractère éliminable ou non des unités archivistiques)
 - FATAL : une erreur technique est survenue lors de la préparation de l'analyse du caractère éliminable ou non des unités archivistiques (STP_ELIMINATION_ANALYSIS_PREPARATION.FATAL = Erreur technique lors de la préparation de l'analyse du caractère éliminable ou non des unités archivistiques)

9.1.1.1. Vérification des seuils de l'analyse de l'élimination des unités archivistiques ELIMINATION_ANALYSIS_CHECK_DISTRIBUTION_THRESHOLD

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

9.1.1.2. Préparation de l'analyse de l'élimination des unités archivistiques **ELIMINATION_ANALYSIS_PREPARATION**

- **Règle** : tâche consistant à établir la liste des unités archivistiques concernées par l'analyse en prenant en compte les règles de gestion héritées et à créer le fichier de distribution
- **Type** : bloquant
- **Statuts** :
 - OK : l'établissement de la liste des unités archivistiques concernées par l'analyse a bien été effectuée (ELIMINATION_ANALYSIS_PREPARATION.OK = Succès de la préparation de l'analyse de l'élimination des unités archivistiques)
 - KO : l'établissement de la liste des unités archivistiques concernées par l'analyse n'a pas été effectuée (ELIMINATION_ANALYSIS_PREPARATION.KO = Échec de la préparation de l'analyse de l'élimination des unités archivistiques)
 - FATAL : une erreur technique est survenue lors de l'établissement de la liste des unités archivistiques concernées par l'analyse (ELIMINATION_ANALYSIS_PREPARATION.FATAL = Erreur technique lors de la préparation de l'analyse de l'élimination des unités archivistiques)

9.1.2. Processus d'indexation de l'élimination des unités archivistiques **(STP_ELIMINATION_ANALYSIS_UNIT_INDEXATION)**

9.1.2.1. Indexation de l'analyse d'élimination des unités archivistiques **ELIMINATION_ANALYSIS_UNIT_INDEXATION**

- **Règle** : tâche et traitement consistant à indexer les unités archivistiques éliminables au regard de la requête lancée (échéance de la règle de durée d'utilité administrative antérieure à la date saisie dans la requête)
- **Type** : bloquant
- **Statuts** :
 - OK : l'indexation des unités archivistiques a bien été effectuée (ELIMINATION_ANALYSIS_UNIT_INDEXATION.OK = Succès de l'indexation de l'analyse du caractère éliminable ou non des unités archivistiques)
 - KO : l'indexation des unités archivistiques n'a pas été effectuée (ELIMINATION_ANALYSIS_UNIT_INDEXATION.KO = Échec lors de l'indexation de l'analyse du caractère éliminable ou non des unités archivistiques)
 - FATAL : une erreur technique est survenue lors de l'indexation des unités archivistiques (ELIMINATION_ANALYSIS_UNIT_INDEXATION.FATAL = Erreur technique lors de l'indexation du caractère éliminable ou non d'élimination des unités archivistiques)

9.1.2.2. Établissement de la liste des objets **OBJECTS_LIST_EMPTY**

- **Règle** : tâche consistant à établir la liste des objets
- **Statuts** :
 - OK : la liste des objets a été établie avec succès (OBJECTS_LIST_EMPTY.OK = Succès lors de l'établissement de la liste des objets : il n'y a pas d'objet pour cette étape)
 - FATAL : une erreur technique est survenue lors de l'établissement de la liste des objets (OBJECTS_LIST_EMPTY.FATAL = Erreur technique lors de l'établissement de la liste des objets)

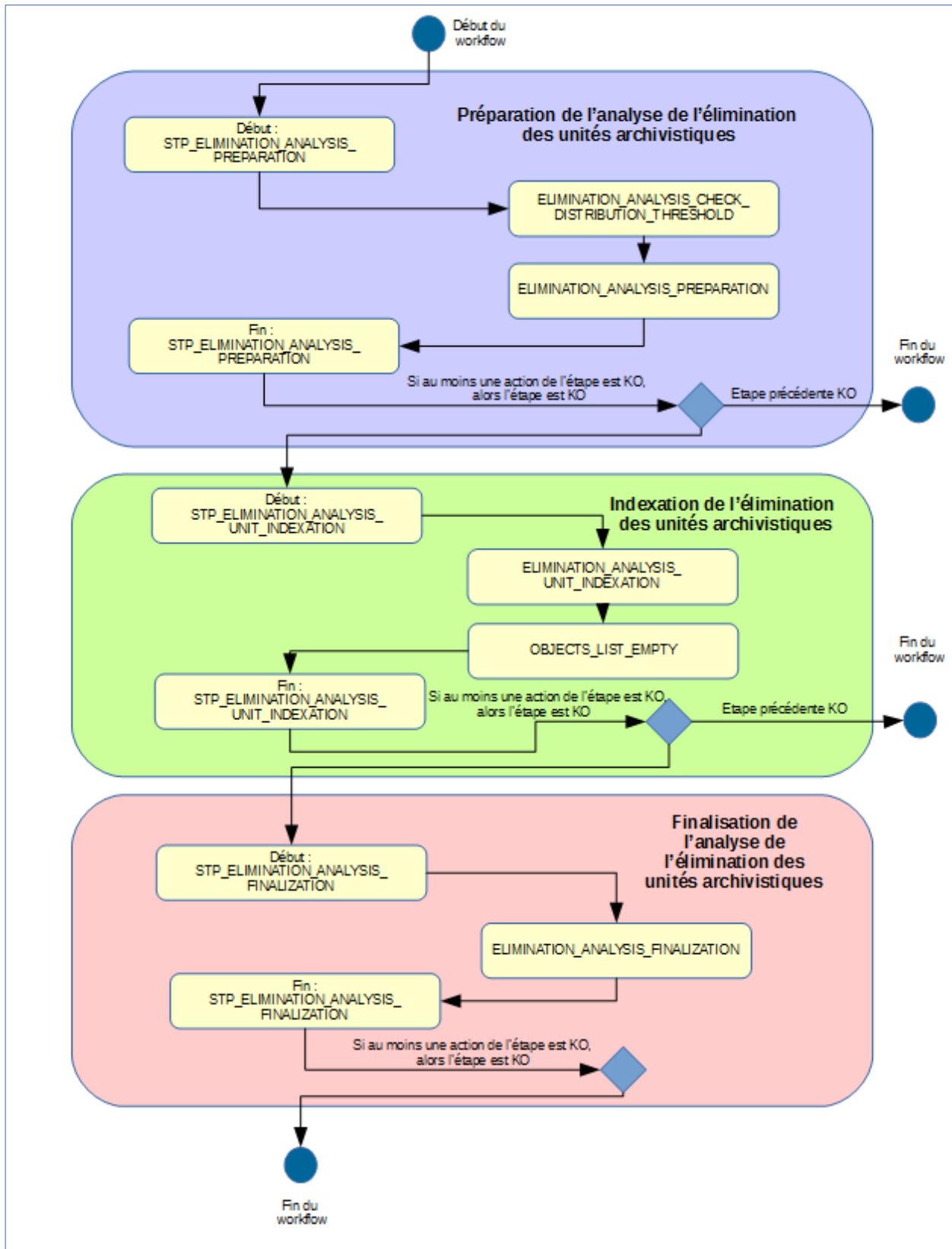
9.1.3. Processus de finalisation de l'analyse de l'élimination des unités archivistiques **(STP_ELIMINATION_ANALYSIS_FINALIZATION)**

9.1.3.1.Finalisation de l'analyse de l'élimination des unités archivistiques ELIMINATION_ANALYSIS_FINALIZATION

- **Règle** : tâche consistant à finaliser l'analyse du caractère éliminable ou non des unités archivistiques
- **Type** : bloquant
- **Statuts** :
 - OK : la finalisation de l'analyse du caractère éliminable ou non des unités archivistiques a bien été effectuée (ELIMINATION_ANALYSIS_FINALIZATION.OK = Succès de la finalisation de l'analyse du caractère éliminable ou non des unités archivistiques)
 - KO : la finalisation de l'analyse du caractère éliminable ou non des unités archivistiques n'a pas été effectuée (ELIMINATION_ANALYSIS_FINALIZATION.KO = Échec lors de la finalisation de l'analyse du caractère éliminable ou non des unités archivistiques)
 - FATAL : une erreur technique est survenue lors de l'analyse du caractère éliminable ou non des unités archivistiques (ELIMINATION_ANALYSIS_FINALIZATION.FATAL = Erreur technique lors de la finalisation de l'analyse du caractère éliminable ou non des unités archivistiques)

9.1.4. Structure de workflow d'analyse de l'élimination des unités archivistiques

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



9.2. Workflow d'élimination définitive des unités archivistiques

Lors de la phase d'analyse, le système procède d'une part à la vérification du caractère éliminable des unités archivistiques – toutes doivent avoir un statut « DESTROY » et ne pas d'avoir d'unités archivistiques non éliminables dans leur descendance –, et d'autre part à mettre en œuvre à proprement parler l'élimination.

9.2.1. Processus de préparation de l'élimination définitive des unités archivistiques (STP_ELIMINATION_ACTION_PREPARATION)

9.2.1.1. Vérification des processus concurrents (CHECK_CONCURRENT_WORKFLOW_LOCK)

- **Règle** : tâche consistant à détecter s'il n'y a pas d'autre processus d'élimination en cours. Si tel est le cas, le processus n'est pas lancé.
- **Type** : bloquant
- **Statuts** :
 - OK : aucun processus concurrent d'élimination n'a été détecté (CHECK_CONCURRENT_WORKFLOW_LOCK.OK = Succès de la détection de processus concurrents d'élimination)
 - KO : un processus concurrent d'élimination a été détecté (CHECK_CONCURRENT_WORKFLOW_LOCK.KO = Échec de la détection de processus concurrents d'élimination)
 - WARNING : avertissement lors de la détection de processus concurrents d'élimination (CHECK_CONCURRENT_WORKFLOW_LOCK.WARNING = Avertissement lors de la détection de processus concurrents d'élimination)
 - FATAL : une erreur technique est survenue lors de la détection de processus concurrents d'élimination (CHECK_CONCURRENT_WORKFLOW_LOCK.FATAL = Erreur technique lors de la détection de processus concurrents d'élimination)

9.2.1.2. Vérification des seuils de l'élimination définitive des unités archivistiques (ELIMINATION_ACTION_CHECK_DISTRIBUTION_THRESHOLD)

- **Règle** : tâche consistant à vérifier les seuils de traitement des unités archivistiques par rapport à la liste des unités archivistiques à traiter
- **Type** : bloquant
- **Statuts** :
 - OK : la vérification des seuils de l'élimination définitive des unités archivistiques a bien été effectuée (ELIMINATION_ACTION_CHECK_DISTRIBUTION_THRESHOLD.OK = Succès de la vérification des seuils de l'élimination définitive des unités archivistiques)
 - KO : une incohérence a été détectée entre le seuil et le nombre d'unités archivistiques à traiter (ELIMINATION_ACTION_CHECK_DISTRIBUTION_THRESHOLD.KO = Échec de la vérification des seuils de l'élimination définitive des unités archivistiques)
 - WARNING : avertissement lors de la vérification des seuils de l'élimination définitive des unités archivistiques (ELIMINATION_ACTION_CHECK_DISTRIBUTION_THRESHOLD.WARNING = Avertissement lors de la vérification des seuils de l'élimination définitive des unités archivistiques)
 - FATAL : une erreur technique est survenue lors de la vérification des seuils de l'élimination définitive des unités archivistiques (ELIMINATION_ACTION_CHECK_DISTRIBUTION_THRESHOLD.FATAL = Erreur technique lors de la vérification des seuils de l'élimination définitive des unités archivistiques)

9.2.1.3. Préparation de l'élimination définitive des unités archivistiques ELIMINATION_ACTION_UNIT_PREPARATION

- **Règle** : tâche consistant à établir la liste des unités archivistiques concernées par l'analyse en prenant en compte les règles de gestion héritées – toutes doivent avoir un statut « DESTROY » et ne pas d'avoir d'unités archivistiques non éliminables dans leur descendance – et à créer le fichier de distribution
- **Type** : bloquant
- **Statuts** :
 - OK : la préparation de l'élimination définitive des unités archivistiques a bien été effectuée (ELIMINATION_ACTION_UNIT_PREPARATION.OK = Succès de la préparation de l'élimination définitive des unités archivistiques)
 - KO : la préparation de l'élimination définitive des unités archivistiques n'a pas été effectuée (ELIMINATION_ACTION_UNIT_PREPARATION.KO = Échec de la préparation de l'élimination définitive des unités archivistiques)
 - WARNING : avertissement lors de la préparation de l'élimination définitive des unités archivistiques suite à la détection d'unités archivistiques à conserver (ELIMINATION_ACTION_UNIT_PREPARATION.WARNING = Avertissement lors de la préparation de l'élimination définitive des unités archivistiques suite à la détection d'unités archivistiques à conserver)
 - FATAL : une erreur technique est survenue lors de la préparation de l'élimination définitive des unités archivistiques (ELIMINATION_ACTION_UNIT_PREPARATION.FATAL = Erreur technique lors de la préparation de l'élimination définitive des unités archivistiques)

9.2.2. Processus d'élimination définitive des unités archivistiques éliminables (STP_ELIMINATION_ACTION_DELETE_UNIT)

9.2.2.1. Élimination définitive des unités archivistiques éliminables ELIMINATION_ACTION_DELETE_UNIT

- **Règle** : tâche consistant à éliminer définitivement les unités archivistiques éliminables
- **Type** : bloquant
- **Statuts** :
 - OK : l'élimination définitive des unités archivistiques éliminables a bien été effectuée (ELIMINATION_ACTION_DELETE_UNIT.OK = Succès de l'élimination définitive des unités archivistiques éliminables)
 - KO : l'élimination définitive des unités archivistiques éliminables n'a pas été effectuée (ELIMINATION_ACTION_DELETE_UNIT.KO = Échec de l'élimination définitive des unités archivistiques éliminables)
 - WARNING : avertissement lors de l'élimination définitive des unités archivistiques éliminables (STP_ELIMINATION_ACTION_DELETE_UNIT.WARNING = Avertissement lors de l'élimination définitive des unités archivistiques éliminables)
 - FATAL : une erreur technique est survenue lors de l'élimination définitive des unités archivistiques éliminables (STP_ELIMINATION_ACTION_DELETE_UNIT.FATAL = Erreur technique lors de l'élimination définitive des unités archivistiques éliminables)

9.2.2.2. Établissement de la liste des objets OBJECTS_LIST_EMPTY

- **Règle** : tâche consistant à établir la liste des objets
- **Statuts** :
 - OK : la liste des objets a été établie avec succès (OBJECTS_LIST_EMPTY.OK = Succès lors de l'établissement de la liste des objets : il n'y a pas d'objet pour cette étape)

- FATAL : une erreur technique est survenue lors de l'établissement de la liste des objets (OBJECTS_LIST_EMPTY.FATAL = Erreur technique lors de l'établissement de la liste des objets)

9.2.3. Processus de préparation de l'élimination définitive des groupes d'objets techniques dont toutes les unités archivistiques parentes ont été éliminées (STP_ELIMINATION_ACTION_OBJECT_GROUP_PREPARATION)

9.2.3.1. Préparation de l'élimination définitive des groupes d'objets techniques dont toutes les unités archivistiques parentes ont été éliminées ELIMINATION_ACTION_OBJECT_GROUP_PREPARATION

- **Règle** : tâche et traitement consistant à préparer l'élimination définitive des groupes d'objets techniques dont les unités archivistiques parentes ont été éliminées
- **Type** : bloquant
- **Statuts** :
 - OK : la préparation de l'élimination définitive des groupes d'objets techniques a bien été effectuée (ELIMINATION_ACTION_OBJECT_GROUP_PREPARATION.OK = Succès de la préparation de l'élimination définitive des groupes d'objets techniques)
 - KO : la préparation de l'élimination définitive des groupes d'objets techniques n'a pas été effectuée (ELIMINATION_ACTION_OBJECT_GROUP_PREPARATION.KO = Échec de la préparation de l'élimination définitive des groupes d'objets techniques)
 - WARNING : la préparation de l'élimination définitive des groupes d'objets techniques est en warning (ELIMINATION_ACTION_OBJECT_GROUP_PREPARATION.WARNING = Avertissement lors de la préparation de l'élimination définitive des groupes d'objets techniques)
 - FATAL : une erreur technique est survenue lors de la préparation de l'élimination définitive des groupes d'objets techniques (ELIMINATION_ACTION_OBJECT_GROUP_PREPARATION.FATAL = Erreur technique lors de la préparation de l'élimination définitive des groupes d'objets techniques)

9.2.4. Processus d'élimination définitive des groupes d'objets techniques dont toutes les unités archivistiques parentes sont éliminées (STP_ELIMINATION_ACTION_DELETE_OBJECT_GROUP)

9.2.4.1. Élimination définitive des groupes d'objets techniques dont toutes les unités archivistiques parentes ont été éliminées ELIMINATION_ACTION_DELETE_OBJECT_GROUP

- **Règle** : tâche et traitement consistant à éliminer définitivement les groupes d'objets techniques dont toutes les unités archivistiques parentes ont été éliminées
- **Type** : bloquant
- **Statuts** :
 - OK : l'élimination définitive des groupes d'objets techniques dont les unités archivistiques parentes sont éliminées a bien été effectuée (ELIMINATION_ACTION_DELETE_OBJECT_GROUP.OK = Succès de l'élimination définitive des groupes d'objets techniques dont les unités archivistiques parentes sont éliminées)
 - KO : l'élimination définitive des groupes d'objets techniques dont les unités archivistiques parentes sont éliminées n'a pas été effectuée (ELIMINATION_ACTION_DELETE_OBJECT_GROUP.KO = Échec de l'élimination définitive des groupes d'objets techniques dont les unités archivistiques parentes sont éliminées)
 - WARNING : avertissement lors de l'élimination définitive des groupes d'objets techniques dont les

unités archivistiques parentes sont éliminées

(ELIMINATION_ACTION_DELETE_OBJECT_GROUP.WARNING = Avertissement lors de l'élimination définitive des groupes d'objets techniques dont les unités archivistiques parentes sont éliminées)

- FATAL : une erreur technique est survenue lors de l'élimination définitive des groupes d'objets techniques dont les unités archivistiques parentes sont éliminées (ELIMINATION_ACTION_DELETE_OBJECT_GROUP.FATAL = Erreur technique lors de l'élimination définitive des groupes d'objets techniques dont les unités archivistiques parentes sont éliminées)

9.2.4.2.Établissement de la liste des objets OBJECTS_LIST_EMPTY

- **Règle** : tâche consistant à établir la liste des objets
- **Statuts** :
 - OK : le processus d'établissement de la liste des objets a été établie avec succès (OBJECTS_LIST_EMPTY.OK = Succès lors de l'établissement de la liste des objets : il n'y a pas d'objet pour cette étape)
 - FATAL : une erreur technique est survenue lors de l'établissement de la liste des objets (OBJECTS_LIST_EMPTY.FATAL = Erreur technique lors de l'établissement de la liste des objets)

9.2.5.Processus de détachement des groupes d'objets techniques dont certaines unités archivistiques parentes sont éliminées (STP_ELIMINATION_ACTION_DETACH_OBJECT_GROUP)

9.2.5.1.Détachement des groupes d'objets techniques dont certaines unités archivistiques parentes ont été éliminées ELIMINATION_ACTION_DETACH_OBJECT_GROUP

- **Règle** : tâche et traitement consistant à détacher les groupes d'objets techniques des unités archivistiques parentes qui ont été éliminées
- **Type** : bloquant
- **Statuts** :
 - OK : le détachement des groupes d'objets techniques dont certaines unités archivistiques parentes sont éliminées a bien été effectué (ELIMINATION_ACTION_DETACH_OBJECT_GROUP.OK = Succès du détachement des groupes d'objets techniques dont certaines unités archivistiques parentes sont éliminées)
 - KO : le détachement des groupes d'objets techniques dont certaines unités archivistiques parentes sont éliminées n'a pas été effectuée (ELIMINATION_ACTION_DETACH_OBJECT_GROUP.KO = Échec du détachement des groupes d'objets techniques dont certaines unités archivistiques parentes sont éliminées)
 - WARNING : avertissement lors du détachement des groupes d'objets techniques dont certaines unités archivistiques parentes sont éliminées (ELIMINATION_ACTION_DETACH_OBJECT_GROUP.WARNING = Avertissement lors du processus de détachement des groupes d'objets techniques dont certaines unités archivistiques parentes sont éliminées)
 - FATAL : une erreur technique est survenue lors du détachement des groupes d'objets techniques dont certaines unités archivistiques parentes sont éliminées (ELIMINATION_ACTION_DETACH_OBJECT_GROUP.FATAL = Erreur technique lors du processus de détachement des groupes d'objets techniques dont certaines unités archivistiques parentes sont éliminées)

9.2.5.2.Écriture des métadonnées des groupes d'objets techniques sur les offres de stockage (OG_METADATA_STORAGE)

- **Règle** : tâche et traitement consistant, en cas de détachement de groupes d'objets techniques, à écrire les métadonnées des groupes d'objets techniques sur les offres de stockage
- **Type** : bloquant
- **Statuts**
 - OK : succès de l'écriture des métadonnées des groupes d'objets techniques sur les offres de stockage (OG_METADATA_STORAGE.OK = Succès de de l'écriture des métadonnées des groupes d'objets techniques sur les offres de stockage)
 - KO : échec de l'écriture des métadonnées des groupes d'objets techniques sur les offres de stockage (OG_METADATA_STORAGE.KO = Échec de l'écriture des métadonnées des groupes d'objets techniques sur les offres de stockage)
 - FATAL : erreur technique lors de de l'écriture des métadonnées des groupes d'objets techniques sur les offres de stockage (OG_METADATA_STORAGE.FATAL = Erreur technique lors de l'écriture des métadonnées des groupes d'objets techniques sur les offres de stockage)

9.2.5.3.Recalcul des graphes des groupes d'objets techniques (OBJECT_GROUP_GRAPH_COMPUTE)

- **Règle** : tâche et traitement consistant, en cas de détachement de groupes d'objets techniques, à recalculer les graphes des groupes d'objets techniques.
- **Type** : bloquant
- **Statuts**
 - OK : succès de la mise à jour des graphes des groupes d'objets techniques (OBJECT_GROUP_GRAPH_COMPUTE.OK = Succès de l'étape de mise à jour des graphes des groupes d'objets techniques)
 - KO : échec de la mise à jour des graphes du groupe d'objets techniques (OBJECT_GROUP_GRAPH_COMPUTE.KO = Échec de la mise à jour des graphes des groupes d'objets techniques)
 - FATAL : erreur technique lors de la mise à jour des graphes des groupes d'objets techniques (OBJECT_GROUP_GRAPH_COMPUTE.FATAL = Erreur technique lors de l'étape de mise à jour des graphes des groupes d'objets techniques)

9.2.5.4.Établissement de la liste des objets OBJECTS_LIST_EMPTY

- **Règle** : tâche consistant à établir la liste des objets
- **Statuts** :
 - OK : le processus d'établissement de la liste des objets a été établie avec succès (OBJECTS_LIST_EMPTY.OK = Succès lors de l'établissement de la liste des objets : il n'y a pas d'objet pour cette étape)
 - FATAL : une erreur technique est survenue lors de l'établissement de la liste des objets (OBJECTS_LIST_EMPTY.FATAL = Erreur technique lors de l'établissement de la liste des objets)

9.2.6.Processus de mise à jour du registre des fonds suite à l'élimination définitive des unités archivistiques (STP_ELIMINATION_ACTION_ACCESSION_REGISTER_PREPARATION)

9.2.6.1. Préparation de la mise à jour du registre des fonds suite à l'élimination définitive des unités archivistiques

ELIMINATION_ACTION_ACCESSION_REGISTER_PREPARATION

- **Règle** : tâche consistant à préparer la mise à jour du registre des fonds suite à l'élimination définitive d'unités archivistiques
- **Type** : bloquant
- **Statuts** :
 - OK : la préparation du registre des fonds suite à l'élimination définitive d'unités archivistiques a bien été effectuée (ELIMINATION_ACTION_ACCESSION_REGISTER_PREPARATION.OK = Succès de la préparation du registre des fonds suite à l'élimination définitive des unités archivistiques)
 - KO : la préparation du registre des fonds suite à l'élimination définitive d'unités archivistiques n'a pas été effectuée (ELIMINATION_ACTION_ACCESSION_REGISTER_PREPARATION.KO = Échec de la préparation du registre des fonds suite à l'élimination définitive des unités archivistiques)
 - WARNING : avertissement lors de la préparation du registre des fonds suite à l'élimination définitive d'unités archivistiques (ELIMINATION_ACTION_ACCESSION_REGISTER_PREPARATION.WARNING = Avertissement lors de la préparation du registre des fonds suite à l'élimination définitive des unités archivistiques)
 - FATAL : une erreur technique est survenue lors de la préparation du registre des fonds suite à l'élimination définitive d'unités archivistiques (ELIMINATION_ACTION_ACCESSION_REGISTER_PREPARATION.FATAL = Erreur technique lors de la préparation du registre des fonds suite à l'élimination définitive des unités archivistiques)

9.2.7. Processus de mise à jour du registre des fonds suite à l'élimination définitive des unités archivistiques

(STP_ELIMINATION_ACTION_ACCESSION_REGISTER_UPDATE)

9.2.7.1. Mise à jour du registre des fonds suite à l'élimination définitive des unités archivistiques

ELIMINATION_ACTION_ACCESSION_REGISTER_UPDATE

- **Règle** : tâche consistant à mettre à jour le registre des fonds suite à l'élimination définitive d'unités archivistiques
- **Type** : bloquant
- **Statuts** :
 - OK : la mise à jour du registre des fonds suite à l'élimination définitive d'unités archivistiques a bien été effectuée (ELIMINATION_ACTION_ACCESSION_REGISTER_UPDATE.OK = Succès de la mise à jour du registre des fonds suite à l'élimination définitive d'unités archivistiques)
 - KO : la mise à jour du registre des fonds suite à l'élimination définitive d'unités archivistiques n'a pas été effectuée (ELIMINATION_ACTION_ACCESSION_REGISTER_UPDATE.KO = Échec de la mise à jour du registre des fonds suite à l'élimination définitive d'unités archivistiques)
 - WARNING : avertissement lors de la mise à jour du registre des fonds suite à l'élimination définitive d'unités archivistiques (ELIMINATION_ACTION_ACCESSION_REGISTER_UPDATE.WARNING = Avertissement lors de la mise à jour du registre des fonds suite à l'élimination définitive d'unités archivistiques)
 - FATAL : une erreur technique est survenue lors de la mise à jour du registre des fonds suite à l'élimination définitive d'unités archivistiques (ELIMINATION_ACTION_ACCESSION_REGISTER_UPDATE.FATAL = Erreur technique lors de la mise à jour du registre des fonds suite à l'élimination définitive d'unités archivistiques)

9.2.7.2.Établissement de la liste des objets OBJECTS_LIST_EMPTY

- **Règle** : tâche consistant à établir la liste des objets
- **Statuts** :
 - OK : le processus d'établissement de la liste des objets a été établie avec succès (OBJECTS_LIST_EMPTY.OK = Succès lors de l'établissement de la liste des objets : il n'y a pas d'objet pour cette étape)
 - FATAL : une erreur technique est survenue lors de l'établissement de la liste des objets (OBJECTS_LIST_EMPTY.FATAL = Erreur technique lors de l'établissement de la liste des objets)

9.2.8.Processus de génération du rapport d'élimination définitive des unités archivistiques (STP_ELIMINATION_ACTION_REPORT_GENERATION)

9.2.8.1.Génération du rapport d'élimination définitive des unités archivistiques ELIMINATION_ACTION_REPORT_GENERATION

- **Règle** : tâche consistant à générer le rapport d'élimination définitive des unités archivistiques
- **Type** : bloquant
- **Statuts** :
 - OK : la génération du rapport d'élimination définitive des unités archivistiques a bien été effectuée (ELIMINATION_ACTION_REPORT_GENERATION.OK = Succès de la génération du rapport d'élimination définitive des unités archivistiques)
 - KO : la génération du rapport d'élimination définitive des unités archivistiques n'a pas été effectuée (ELIMINATION_ACTION_REPORT_GENERATION.KO = Échec de la génération du rapport d'élimination définitive des unités archivistiques)
 - WARNING : avertissement lors de la génération du rapport d'élimination définitive des unités archivistiques (ELIMINATION_ACTION_REPORT_GENERATION.WARNING = Avertissement lors de la génération du rapport d'élimination définitive des unités archivistiques)
 - FATAL : une erreur technique est survenue lors de la génération du rapport d'élimination définitive des unités archivistiques (ELIMINATION_ACTION_REPORT_GENERATION.FATAL = Erreur technique lors de la génération du rapport d'élimination définitive des unités archivistiques)

9.2.9.Processus de finalisation de l'élimination définitive des unités archivistiques (STP_ELIMINATION_ACTION_FINALIZATION)

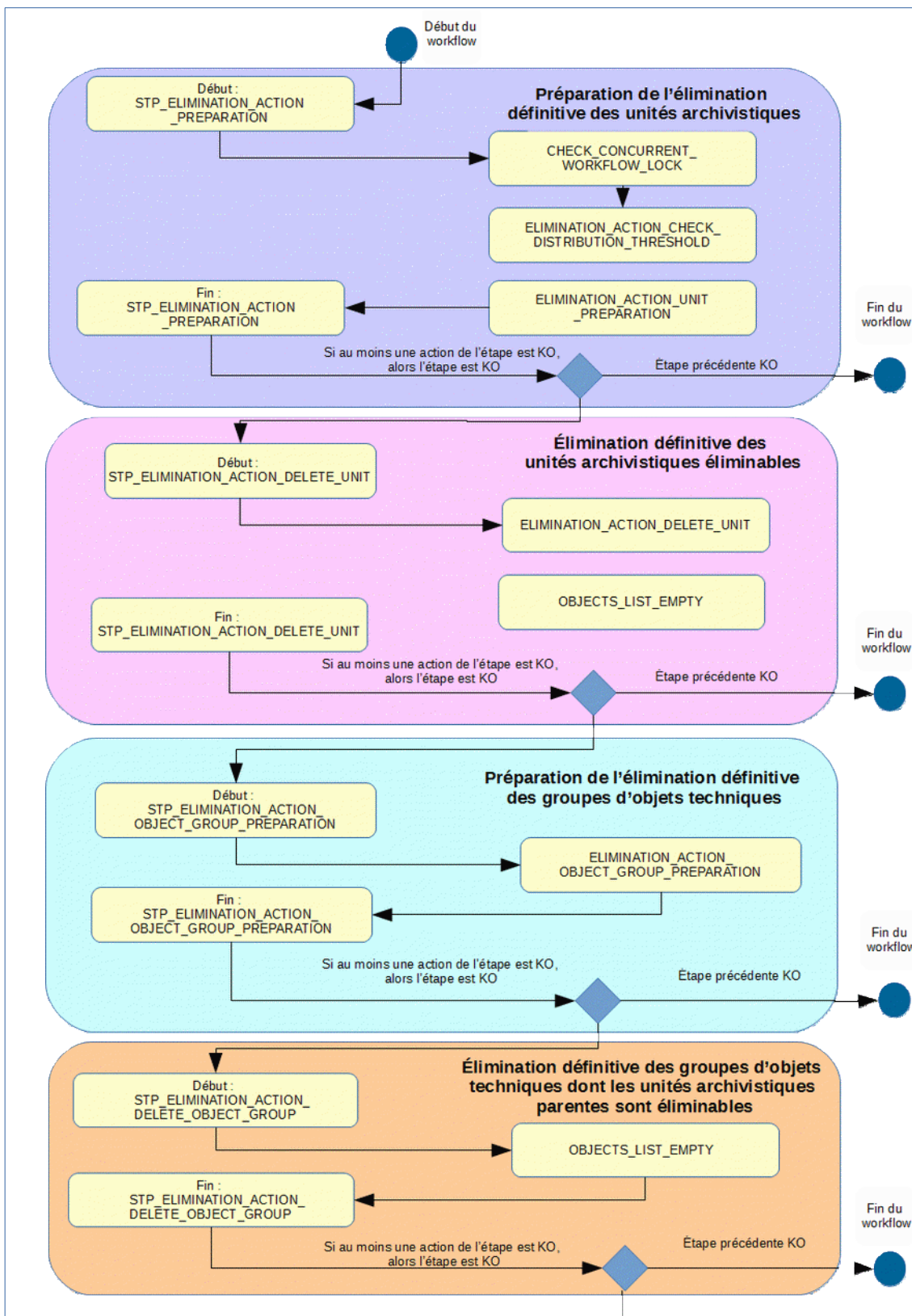
9.2.9.1.Finalisation de l'élimination définitive des unités archivistiques ELIMINATION_ACTION_FINALIZATION

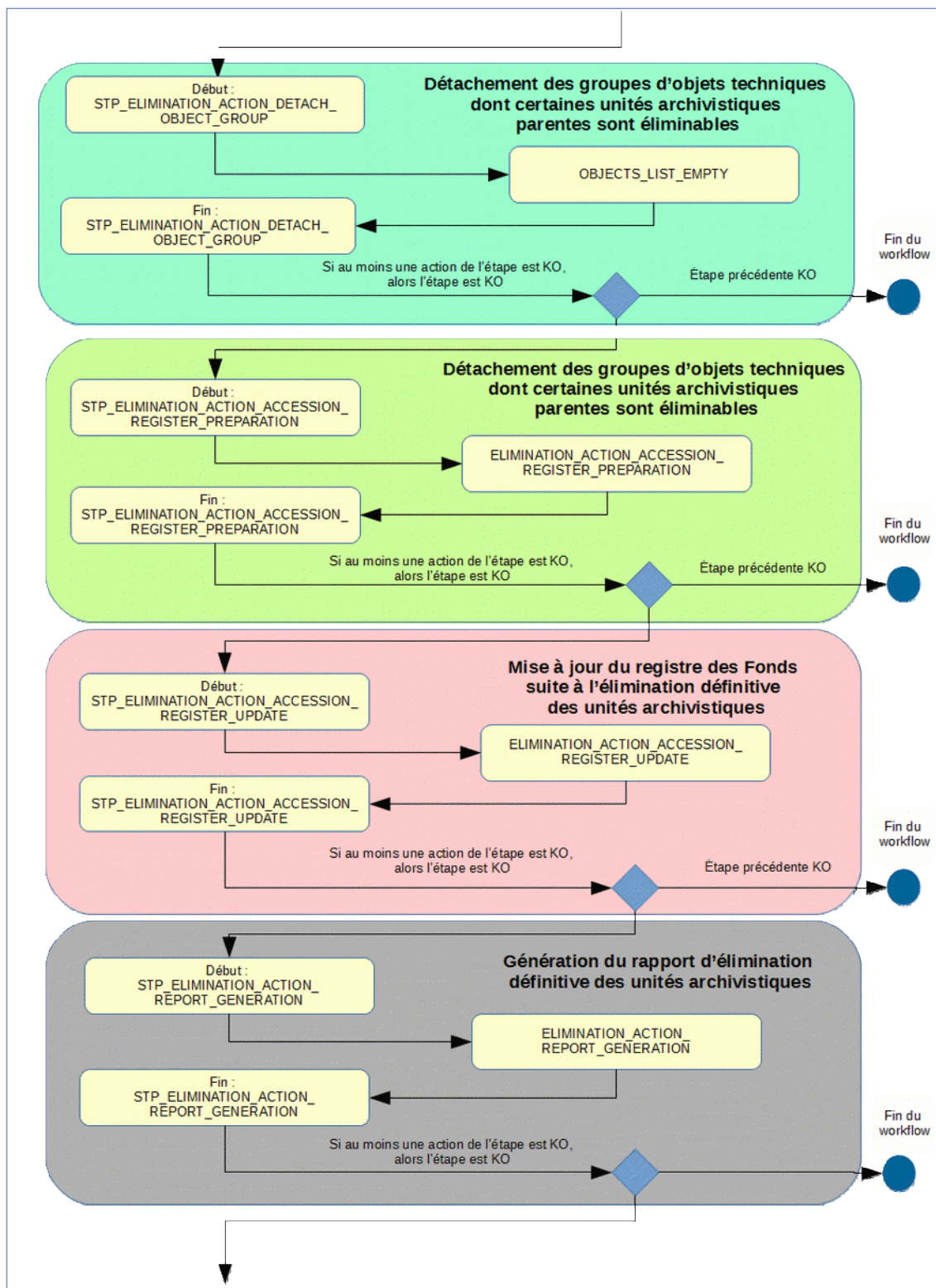
- **Règle** : tâche consistant à finaliser l'élimination définitive des unités archivistiques
- **Type** : bloquant
- **Statuts** :
 - OK : La finalisation de l'élimination définitive des unités archivistiques a bien été effectuée (ELIMINATION_ACTION_FINALIZATION.OK = Succès de la finalisation de l'élimination définitive des unités archivistiques)
 - KO : La finalisation de l'élimination définitive des unités archivistiques n'a pas été effectuée (ELIMINATION_ACTION_FINALIZATION.KO = Échec de la finalisation de l'élimination définitive des unités archivistiques)
 - WARNING : avertissement lors de la finalisation de l'élimination définitive des unités archivistiques (ELIMINATION_ACTION_FINALIZATION.WARNING = Avertissement lors de la finalisation de l'élimination définitive des unités archivistiques)

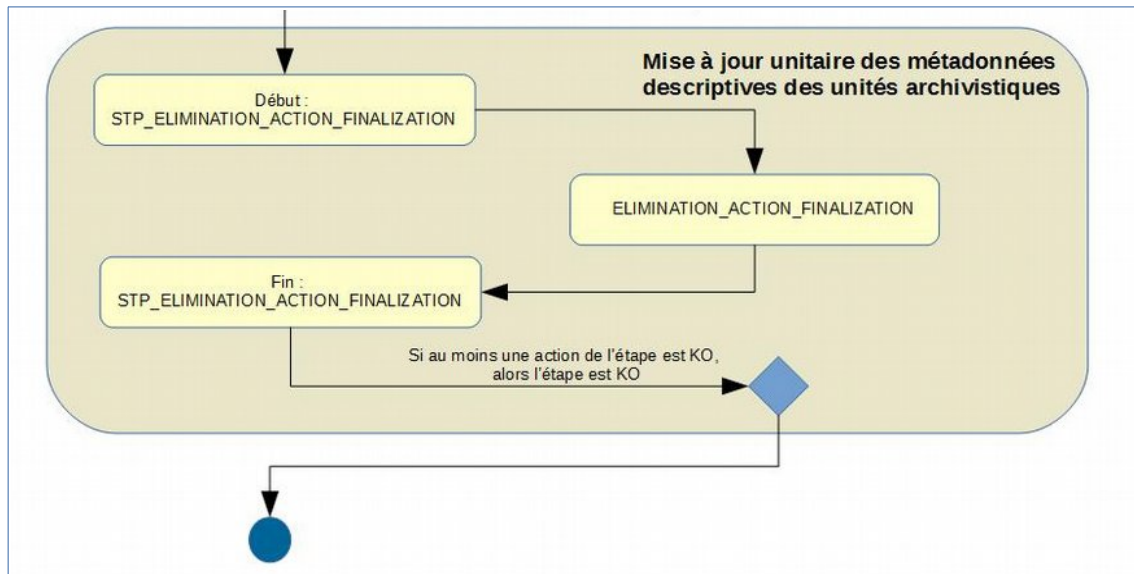
- FATAL : une erreur technique est survenue lors de la finalisation de l'élimination définitive des unités archivistiques (ELIMINATION_ACTION_FINALIZATION.FATAL = Erreur technique lors de la finalisation de l'élimination définitive des unités archivistiques)

9.2.10. Structure du workflow d'analyse des éliminables et action d'élimination

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







9.2.11. Rapport d'élimination

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

9.2.11.1. Exemple de JSON : rapport d'élimination

```

11
+ "tenant": "S", X
+ "evId": "aee aaaa abc hfcj 4xaa 3a6 alo new dbe 4aaa ag", X
+ "evType": "", X
+ "outcome": "", X
+ "outDetail": "", X
+ "outMsg": "", X
+ "rightStatementIdentifier": {}, X
+ "evDet Data": {} X
14 X
15 X
+ "evEndDate Time": "2018-11-14T10:14:16.848", X
+ "reportType": "ELIMINATION_ACTION", X
+ "writeResults": {}, X
+ "extendedInfo": {} X
18 X
19 X
+ "id": "aee aaaa abc hfcj 4xaa 3a6 alo new dbe 4aaa ag", X
+ "params": {} X
+   + "id": "aee aaaa abc hfcj 4xaa 3a6 alo new dbe 4aaa ag", X
+   + "type": "Unit", X
+   + "status": "GLOBAL_STATUS_KEEP", X
+   + "opId": "a eaaa aaaa chfcj 4xaa 3a6 alo new dbe 4aaa ag", X
+   + "originatingAgency": "RAIP" X
+ } X
22 X
+ "id": "aee aaaa abc hfcj 4xaa 3a6 alo new dbe 4aaa ag", X
+ "params": {} X
+   + "id": "aee aaaa abc hfcj 4xaa 3a6 alo new dbe 4aaa ag", X
+   + "type": "Unit", X
+   + "status": "GLOBAL_STATUS_KEEP", X
+   + "opId": "a eaaa aaaa chfcj 4xaa 3a6 alo new dbe 4aaa ag", X
+   + "originatingAgency": "RAIP" X
+ } X
25 X
+ "id": "aee aaaa abc hfcj 4xaa 3a6 alo new dbe 4aaa ag", X
+ "params": {} X
+   + "id": "aee aaaa abc hfcj 4xaa 3a6 alo new dbe 4aaa ag", X
+   + "type": "Unit", X
+   + "status": "GLOBAL_STATUS_KEEP", X
+   + "opId": "a eaaa aaaa chfcj 4xaa 3a6 alo new dbe 4aaa ag", X
+   + "originatingAgency": "RAIP" X
+ } X
28 X

```

OperationSummary

ReportSummary

ReportDetail

```

→ "id": "aeagaaaabahnbfxgaavmgalonednqaaaaba", ¶
→ "params": {¶
→   → "id": "aeagaaaabahnbfxgaavmgalonednqaaaaba", ¶
→   → "type": "Unit", ¶
→   → "status": "DELETED", ¶
→   → "opi": "aeaaaaabchfcj4xaa3s6alonedbeiaaaaq", ¶
→   → "originatingAgency": "RATP" ¶
→ } ¶
} ¶
→ "id": "aeagaaaabahnbfxgaavmgalonednqaaaaca", ¶
→ "params": {¶
→   → "id": "aeagaaaabahnbfxgaavmgalonednqaaaaca", ¶
→   → "type": "Unit", ¶
→   → "status": "DELETED", ¶
→   → "opi": "aeaaaaabchfcj4xaa3s6alonedbeiaaaaq", ¶
→   → "originatingAgency": "RATP", ¶
→   → "objectGroupId": "aebaaaaabahnbfxgaavmgalonednayaaaaq" ¶
→ } ¶
} ¶
→ "id": "aebaaaaabahnbfxgaavmgalonednayaaaaq", ¶
→ "params": {¶
→   → "id": "aebaaaaabahnbfxgaavmgalonednayaaaaq", ¶
→   → "type": "ObjectGroup", ¶
→   → "status": "DELETED", ¶
→   → "opi": "aeaaaaabchfcj4xaa3s6alonedbeiaaaaq", ¶
→   → "originatingAgency": "RATP", ¶
→   → "objectIds": ["aeaaaaaaahnbfxgaavmgalonednagaaaaq"] ¶
→ } ¶
} ¶

```

9.2.11.2.Détails du rapport

La partie « OperationSummary » c’est-à-dire le bloc racine du rapport et correspondant au résumé de l’opération est composé des champs suivants :

- « Tenant » : tenant sur lequel l’opération de modification en masse a été lancée
- « evId »:identifiant de l’événement
- « outcome » : statut de l’événement
- « outMsg » : détail du résultat des données de l’événement
- « rightsStatementIdentifier » : identifiant des données référentielles en vertu desquelles l’opération peut s’exécuter.
- « evDetData » : détails des données de l’événement

La partie « ReportSummary » c’est-à-dire le bloc situé sous la racine du rapport et correspondant au résumé du rapport est composé 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 de l’opération (dernier evDateTime de l’event master de

l'opération dans le journal des opérations)

- « reportType » : correspond au modèle du rapport.
- « vitamResults » : correspond au nombre de OK, KO et 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 ce champ est vide.

La partie « ReportDetail » contient les détails de l'opération d'élimination effectuée sur chacune des unités archivistiques.

- « _id » : identifiant de l'unité archivistique ou de l'objet sur lequel l'action d'élimination s'est effectuée.
- « params » : informations sur l'unité archivistique ou le groupe d'objet
 - « id » : identifiant de l'unité archivistique ou de l'objet
 - « type » : collection sur laquelle l'opération a été réalisée (UNIT ou OBJECTGROUP)
 - « status » : statut au regard de l'action d'élimination :
 - **GLOBAL_STATUS_KEEP** : unité archivistique non éliminable au regard des règles de gestion.
 - **GLOBAL_STATUS_CONFLICT** : unité archivistique portant des règles contradictoires
 - **NOT_DESTROYABLE_HAS_CHILD_UNITS** : unité non supprimable, car elle a des enfants et sa suppression entraînerait une incohérence dans le graphe.
 - **DELETED** : l'unité a effectivement été supprimée
 - « opi » : identifiant de l'opération d'entrée
 - « originatingAgency » : service producteur
 - « objectIds » : identifiant du groupe d'objets techniques rattaché à l'unité archivistique
-

10. MODIFICATION D'ARBORESCENCE (RECLASSIFICATION)

Workflow de modification d'arborescence

Cette section décrit le processus permettant la modification d'arborescence d'archives, c'est-à-dire de modifier les rattachements d'une unité archivistique présente dans le système.

Le processus de modification d'arborescence est lié à des ajouts et des suppressions de liens de parenté, en sachant que les unités archivistiques modifiées ou rattachées doivent être accessibles par le même contrat d'accès.

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

10.1.Étape de préparation de la modification d'arborescence des unités archivistiques STP_RECLASSIFICATION_PREPARATION

- **Règle** : étape de préparation de la modification d'arborescence des unités archivistiques
- **Type** : bloquant
- **Statuts**
 - OK : l'étape de préparation de la modification d'arborescence des unités archivistiques a bien été effectué (STP_RECLASSIFICATION_PREPARATION.OK = Succès du processus de préparation de la modification d'arborescence des unités archivistiques)
 - KO : erreur lors de la préparation de la modification d'arborescence des unités archivistiques n'a pas été effectué (STP_RECLASSIFICATION_PREPARATION.KO = Échec du processus de préparation de la modification d'arborescence des unités archivistiques)
 - FATAL : erreur technique a eu lieu lors de l'étape de préparation de la modification d'arborescence des unités archivistiques (STP_RECLASSIFICATION_PREPARATION.FATAL = Erreur technique lors du processus de préparation de la modification d'arborescence des unités archivistiques)

10.1.1.Vérification des processus concurrents CHECK_CONCURRENT_WORKFLOW_LOCK

- **Règle** : tâche ayant pour but de détecter s'il n'y a pas d'autre processus de modification d'arborescence en cours. Si tel est le cas, le processus n'est pas lancé afin d'éviter les cycles rattachements/détachements concernant plusieurs unités archivistiques.
- **Type** : bloquant
- **Statuts** :
 - OK : aucun processus concurrent de modification d'arborescence n'a été détecté (CHECK_CONCURRENT_WORKFLOW_LOCK.OK = Succès de la vérification des processus concurrents)
 - KO : un processus concurrent de modification d'arborescence a été détecté (CHECK_CONCURRENT_WORKFLOW_LOCK.KO = Échec lors de la vérification des processus concurrents)
 - FATAL : une erreur technique est survenue lors de la détection de processus concurrents de modification d'arborescence (CHECK_CONCURRENT_WORKFLOW_LOCK.FATAL = Erreur technique lors de la détection de processus concurrents de modification d'arborescence)

10.1.2.Chargement des unités archivistiques : RECLASSIFICATION_PREPARATION_LOAD_REQUEST

- **Règle** : tâche consistant à charger les unités archivistiques
- **Type** : bloquant
- **Statuts** :

- OK : le chargement des unités archivistiques a bien été effectué (RECLASSIFICATION_PREPARATION_LOAD_REQUEST.OK = Succès du chargement des unités archivistiques au moment de la modification d'arborescence des unités archivistiques)
- KO : le chargement des unités archivistiques n'a pas pu s'effectuer (RECLASSIFICATION_PREPARATION_LOAD_REQUEST.KO = Échec du chargement des unités archivistiques au moment de la modification d'arborescence des unités archivistiques)
- FATAL : une erreur technique est survenue lors du chargement des unités archivistiques (RECLASSIFICATION_PREPARATION_LOAD_REQUEST.FATAL = Erreur technique lors du chargement des unités archivistiques au moment de la modification d'arborescence des unités archivistiques)

10.1.3. Vérification de la cohérence du graphe : RECLASSIFICATION_PREPARATION_CHECK_GRAPH

- **Règle** : tâche consistant à vérifier que le graphe est cohérent, c'est-à-dire à vérifier que les rattachements s'effectuent de façon cohérente entre les différents types d'unités archivistiques : arbre de positionnement, plan de classement, unités simples
- **Type** : bloquant
- **Statuts** :
 - OK : le graphe des unités archivistiques est cohérent (RECLASSIFICATION_PREPARATION_CHECK_GRAPH.OK = Succès du contrôle de cohérence du graphe au moment de la modification d'arborescence des unités archivistiques)
 - KO : le graphe de l'arborescence des unités archivistiques est incohérent (RECLASSIFICATION_PREPARATION_CHECK_GRAPH.KO = Échec du contrôle de cohérence du graphe au moment de la modification d'arborescence des unités archivistiques)
 - FATAL : une erreur technique est survenue lors du contrôle de cohérence du graphe des unités archivistiques (RECLASSIFICATION_PREPARATION_CHECK_GRAPH.FATAL = Erreur technique lors du contrôle de cohérence du graphe au moment de la modification d'arborescence des unités archivistiques)

10.1.4. Préparation de la mise à jour du graphe : RECLASSIFICATION_PREPARATION_UPDATE_DISTRIBUTION

- **Règle** : tâche consistant à préparer la mise à jour du graphe des unités archivistiques
- **Type** : bloquant
- **Statuts** :
 - OK : la mise à jour du graphe au moment de la modification d'arborescence des unités a bien été préparée (RECLASSIFICATION_PREPARATION_UPDATE_DISTRIBUTION.OK = Succès de la préparation de la mise à jour du graphe au moment de la modification d'arborescence des unités archivistiques)
 - KO : la mise à jour du graphe au moment de la modification d'arborescence des unités n'a pas pu être préparée (RECLASSIFICATION_PREPARATION_UPDATE_DISTRIBUTION.KO = Échec lors de la préparation de la mise à jour du graphe au moment de la modification d'arborescence des unités archivistiques)
 - FATAL : une erreur technique est survenue lors de la préparation de la mise à jour du graphe (RECLASSIFICATION_PREPARATION_UPDATE_DISTRIBUTION.FATAL = Erreur technique lors de la préparation de la mise à jour du graphe au moment de la modification d'arborescence des unités archivistiques)

10.2. Étape de détachement des unités archivistiques (STP_UNIT_DETACHMENT)

- **Règle** : étape consistant à supprimer des liens entre unités archivistiques

- **Type** : bloquant
- **Statuts**
 - OK : la suppression de liens entre unités archivistiques a bien été effectué (STP_UNIT_DETACHMENT.OK = Succès de l'étape de suppression de liens entre unités archivistiques)
 - KO : la suppression de liens entre unités archivistiques n'a pas pu s'effectuer (STP_UNIT_DETACHMENT.KO = Échec lors de l'étape de suppression de liens entre unités archivistiques)
 - FATAL : une erreur technique est survenue lors de la suppression de liens entre unités archivistiques (STP_UNIT_DETACHMENT.FATAL = Erreur technique lors de l'étape de suppression de liens entre unités archivistiques)

10.2.1. Détachement des unités archivistiques UNIT_DETACHMENT

- **Règle** : tâche ou traitement consistant à supprimer des liens entre unités archivistiques
- **Type** : bloquant
- **Statuts** :
 - OK : la suppression de liens entre unités archivistiques a bien été effectué (UNIT_DETACHMENT.OK = Succès de la suppression de liens entre unités archivistiques)
 - KO : la suppression de liens entre des unités archivistiques n'a pas pu s'effectuer (UNIT_DETACHMENT.KO = Échec lors de la suppression de liens entre les unités archivistiques)
 - FATAL : une erreur technique est survenue lors de la suppression de liens entre unités archivistiques (UNIT_DETACHMENT.FATAL = Erreur technique lors de la suppression de liens entre unités archivistiques)

10.2.2. Enregistrement des métadonnées des unités archivistiques sur l'offre de stockage UNIT_METADATA_STORAGE

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

10.2.3. Établissement de la liste des objets OBJECTS_LIST_EMPTY

- **Règle** : tâche consistant à établir la liste des objets, en cas d'absence de liens à supprimer
- **Statuts** :
 - OK : la liste des objets a été établie avec succès (OBJECTS_LIST_EMPTY.OK = Succès lors de l'établissement de la liste des objets : il n'y a pas d'objet pour cette étape)
 - FATAL : une erreur technique est survenue lors de l'établissement de la liste des objets (OBJECTS_LIST_EMPTY.FATAL = Erreur technique lors de l'établissement de la liste des objets)

10.3.Processus de rattachement des unités archivistiques STP_UNIT_ATTACHMENT

- **Règle** : étape consistant à établir des liens entre unités archivistiques
- **Type** : bloquant
- **Statuts**
 - OK : l'établissement de liens entre unités archivistiques a bien été effectué (STP_UNIT_ATTACHMENT.OK = Succès du processus d'établissement de liens entre unités archivistiques)
 - KO : l'établissement de liens entre unités archivistiques n'a pas pu s'effectuer (STP_UNIT_ATTACHMENT.KO = Échec du processus d'établissement de liens entre unités archivistiques)
 - FATAL : Erreur technique lors de l'établissement de liens entre unités archivistiques (STP_UNIT_ATTACHMENT.FATAL = Erreur technique lors du processus d'établissement de liens entre unités archivistiques)

10.3.1.Processus de rattachement des unités archivistiques : UNIT_ATTACHMENT

- **Règle** : tâche et traitement consistant à établir des liens entre unités archivistiques
- **Type** : bloquant
- **Statuts** :
 - OK : l'établissement de liens entre unités archivistiques a bien été effectué (UNIT_ATTACHMENT.OK = Succès de l'établissement de liens entre unités archivistiques)
 - KO : l'établissement de liens entre unités archivistiques n'a pas pu s'effectuer (UNIT_ATTACHMENT.KO = Échec lors de l'établissement de liens entre unités archivistiques)
 - FATAL : Erreur technique lors de l'établissement de liens entre unités archivistiques (UNIT_ATTACHMENT.FATAL = Erreur technique lors de l'établissement de liens entre unités archivistiques)

10.3.2.Enregistrement des métadonnées des unités archivistiques sur l'offre de stockage UNIT_METADATA_STORAGE

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

10.3.3.Établissement de la liste des objets OBJECTS_LIST_EMPTY

- **Règle** : tâche consistant à établir la liste des objets, en cas d'absence de liens à ajouter
- **Statuts** :
 - OK : la liste des objets a été établie avec succès (OBJECTS_LIST_EMPTY.OK = Succès lors de l'établissement de la liste des objets : il n'y a pas d'objet pour cette étape)
 - FATAL : une erreur technique est survenue lors de l'établissement de la liste des objets

(OBJECTS_LIST_EMPTY.FATAL = Erreur technique lors de l'établissement de la liste des objets)

10.4. Mise à jour des graphes des unités archivistiques **STP_UNIT_GRAPH_COMPUTE**

- **Règle** : étape consistant à recalculer le graphe des unités archivistiques
- **Type** : bloquant
- **Statuts**
 - OK : la mise à jour des graphes des unités archivistiques a bien été effectuée (STP_UNIT_GRAPH_COMPUTE.OK = Succès du processus de mise à jour des graphes des unités archivistiques)
 - KO : la mise à jour des graphes des unités archivistiques n'a pas pu s'effectuer (STP_UNIT_GRAPH_COMPUTE.KO = Échec lors du processus de mise à jour des graphes des unités archivistiques)
 - FATAL : une erreur technique est survenue lors de la mise à jour des graphes des unités archivistiques (STP_UNIT_GRAPH_COMPUTE.FATAL = Erreur technique lors du processus de mise à jour des graphes des unités archivistiques)

Calcul du graphe des unités archivistiques : UNIT_GRAPH_COMPUTE

- **Règle** : tâche consistant à recalculer le graphe des unités archivistiques
- **Type** : bloquant
- **Statuts** :
 - OK : la mise à jour des graphes des unités archivistiques a bien été effectuée (UNIT_GRAPH_COMPUTE.OK = Succès de la mise à jour des graphes des unités archivistiques)
 - KO : la mise à jour des graphes des unités archivistiques n'a pas pu s'effectuer (UNIT_GRAPH_COMPUTE.KO = échec de la mise à jour des graphes des unités archivistiques)
 - FATAL : une erreur technique est survenue lors de la mise à jour des graphes des unités archivistiques (UNIT_GRAPH_COMPUTE.FATAL = Erreur technique lors de la mise à jour des graphes des unités archivistiques)

10.5. Mise à jour des graphes des groupes d'objets **STP_OBJECT_GROUP_GRAPH_COMPUTE**

- **Règle** : étape consistant à recalculer le graphe des groupes d'objets.
- **Type** : bloquant
- **Statuts**
 - OK : le processus de mise à jour des graphes des groupes d'objets techniques a bien été effectué (STP_OBJECT_GROUP_GRAPH_COMPUTE.OK = Succès de l'étape de mise à jour des graphes des groupes d'objets techniques)
 - KO : le processus de mise à jour de mise à jour des graphes du groupe d'objets techniques n'a pas pu s'effectuer (STP_OBJECT_GROUP_GRAPH_COMPUTE.KO = Échec de l'étape de mise à jour des graphes du groupe d'objets techniques)
 - FATAL : erreur technique lors du processus de mise à jour des graphes du groupe d'objets techniques (STP_OBJECT_GROUP_GRAPH_COMPUTE.FATAL = Erreur technique lors de l'étape de mise à jour des graphes du groupe d'objets techniques)

Calcul des graphes des groupes d'objets (OBJECT_GROUP_GRAPH_COMPUTE)

- **Règle** : tâche ou traitement consistant à recalculer le graphe des groupes d'objets.

- **Type** : bloquant
- **Statuts**
 - OK : la mise à jour des graphes du groupe d'objets techniques a bien été effectuée (OBJECT_GROUP_GRAPH_COMPUTE.OK = Succès de l'étape de mise à jour des graphes des groupes d'objets techniques)
 - KO : la mise à jour des graphes du groupe d'objets techniques n'a pas pu s'effectuer (OBJECT_GROUP_GRAPH_COMPUTE.KO = Échec de l'étape de mise à jour des graphes des groupes d'objets techniques)
 - FATAL : erreur technique lors de la mise à jour des graphes des groupes d'objets techniques (OBJECT_GROUP_GRAPH_COMPUTE.FATAL = Erreur technique lors de l'étape de mise à jour des graphes du groupe d'objets techniques)

10.6.Finalisation de la modification d'arborescence des unités archivistiques (STP_RECLASSIFICATION_FINALIZATION)

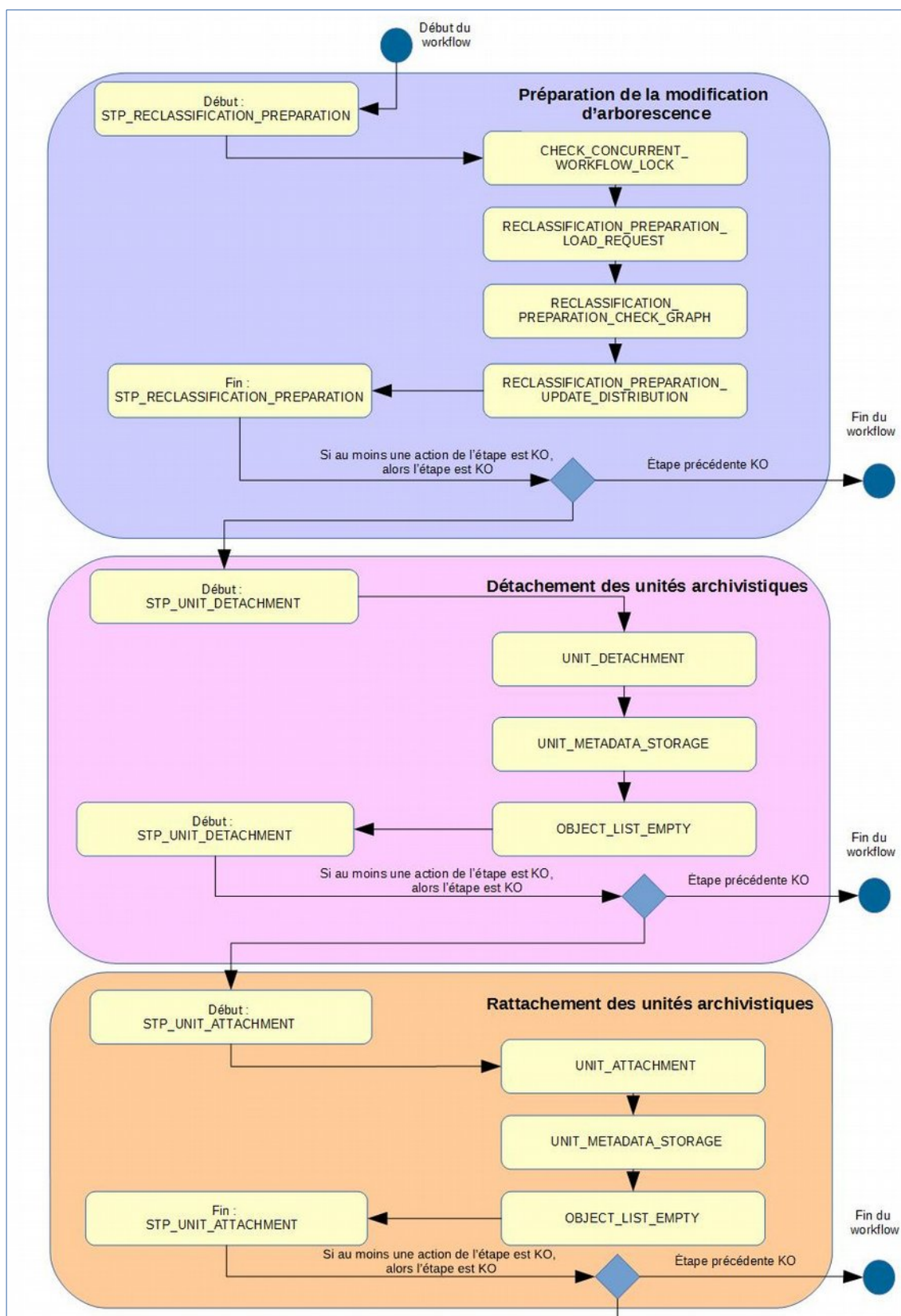
- **Règle** : étape consistant à finaliser le processus de modification d'arborescence pour des unités archivistiques présentes dans le système
- **Type** : bloquant
- **Statuts**
 - OK : le processus de finalisation de la modification d'arborescence des unités archivistiques a été effectué (STP_RECLASSIFICATION_FINALIZATION.OK = Succès du processus de finalisation de la modification d'arborescence des unités archivistiques)
 - KO : le processus de finalisation de la modification d'arborescence des unités archivistiques n'a pas pu être effectué (STP_RECLASSIFICATION_FINALIZATION.KO = Échec du processus de finalisation de la modification d'arborescence des unités archivistiques)
 - FATAL : erreur technique lors du processus de finalisation de la modification d'arborescence des unités archivistiques (STP_RECLASSIFICATION_FINALIZATION.FATAL = Erreur technique lors du processus de finalisation de la modification d'arborescence des unités archivistiques)

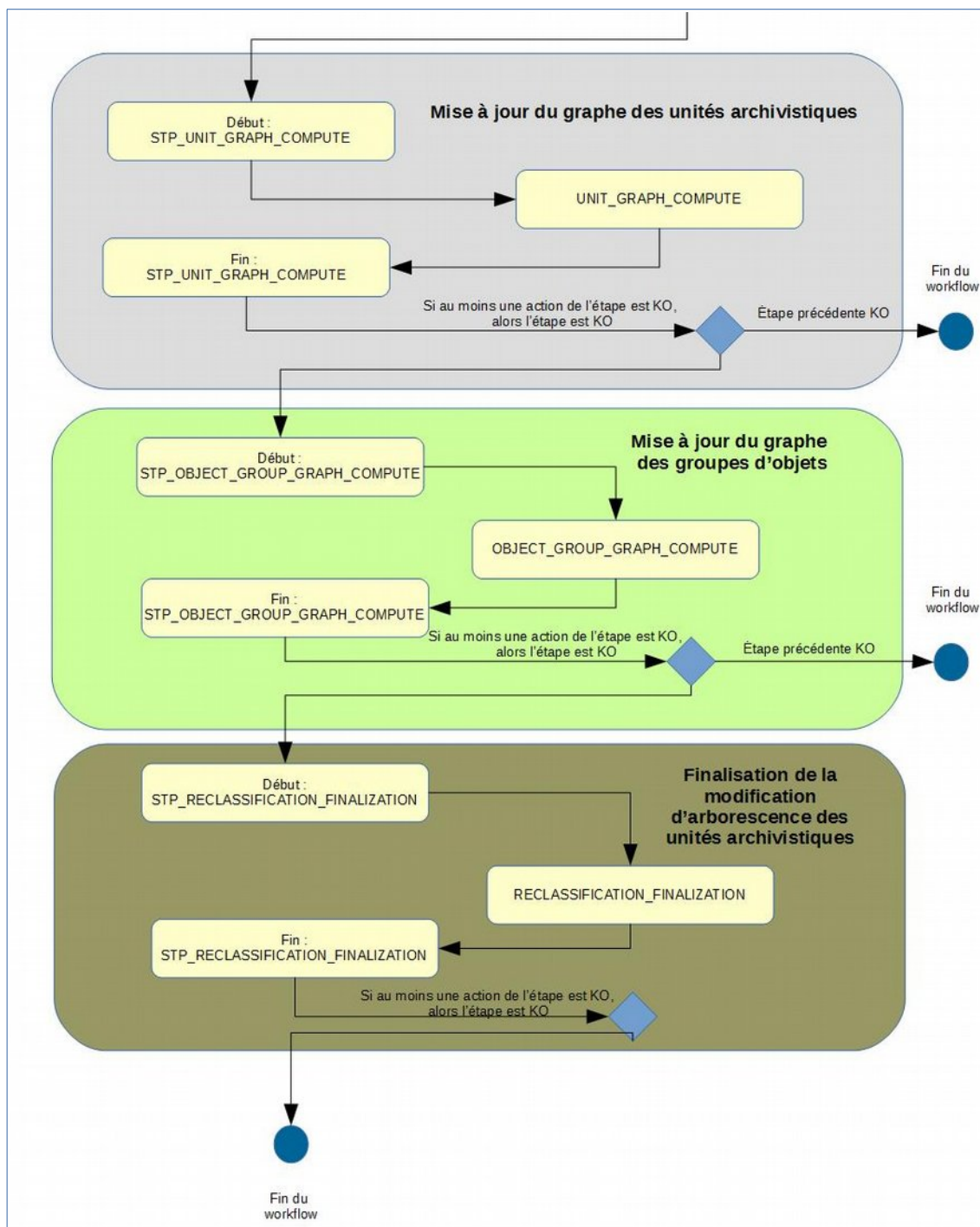
Finalisation de la modification d'arborescence des unités archivistiques (RECLASSIFICATION_FINALIZATION)

- **Règle** : tâche consistant à finaliser le processus de modification d'arborescence pour des unités archivistiques existantes dans le système.
- **Type** : bloquant
- **Statuts**
 - OK : la finalisation de la modification d'arborescence des unités archivistiques a bien été effectuée (RECLASSIFICATION_FINALIZATION.OK = Succès de la finalisation de la modification d'arborescence des unités archivistiques)
 - KO : la finalisation de la modification d'arborescence des unités archivistiques n'a pas pu être effectuée (RECLASSIFICATION_FINALIZATION.KO = Échec lors de la finalisation de la modification d'arborescence des unités archivistiques)
 - FATAL : erreur technique lors de la mise à jour des graphes du groupe d'objets (RECLASSIFICATION_FINALIZATION.FATAL = Erreur technique lors de la finalisation de la modification d'arborescence des unités archivistiques)

10.7. Structure de workflow de modification d'arborescence

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





11. TRANSFERT

11.1. Workflow de préparation du transfert des unités archivistiques et des groupes d'objets

Cette section décrit le processus (workflow) de préparation du transfert mis en œuvre dans la solution logicielle Vitam.

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.

11.1.1. Processus de création du bordereau de mise à disposition (STP_CREATE_MANIFEST)

11.1.1.1. Vérification du seuil 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)

11.1.1.2. Création du bordereau CREATE_MANIFEST (CreateManifest.java)

- **Règle** : tâche consistant à créer le bordereau contenant les unités archivistiques soumises au service de préparation du transfert, ainsi que les groupes d'objets techniques et objets qui leur sont associés
- **Type** : bloquant
- **Statuts** :
 - OK : le bordereau contenant les descriptions des unités archivistiques, groupes d'objets techniques et objets-données a été créé avec succès (CREATE_MANIFEST.OK = Succès de la création du bordereau de mise à disposition)
 - KO : la création du bordereau contenant les descriptions des unités archivistiques, groupes d'objets techniques et objets-données a échoué, car des informations étaient manquantes, erronées ou inconnues, la version du SEDA demandée est incompatible (CREATE_MANIFEST.KO = Échec de la création du bordereau de mise à disposition)
 - FATAL : une erreur technique est survenue lors de la création du bordereau (CREATE_MANIFEST.FATAL = Erreur technique lors de la création du bordereau de mise à disposition)

11.1.2. Processus de déplacement des objets binaires vers l'espace de travail interne (STP_PUT_BINARY_ON_WORKSPACE)

11.1.2.1. Vérification de la disponibilité des ressources

EXPORT_CHECK_RESOURCE_AVAILABILITY

(ExportCheckResourceAvailability.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 (EXPORT_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 (EXPORT_CHECK_RESOURCE_AVAILABILITY.FATAL = Erreur technique lors de la vérification de la disponibilité des objets binaires sur l'offre de stockage)

11.1.2.2. Déplacement des objets binaires vers le workspace

PUT_BINARY_ON_WORKSPACE (PutBinaryOnWorkspace.java)

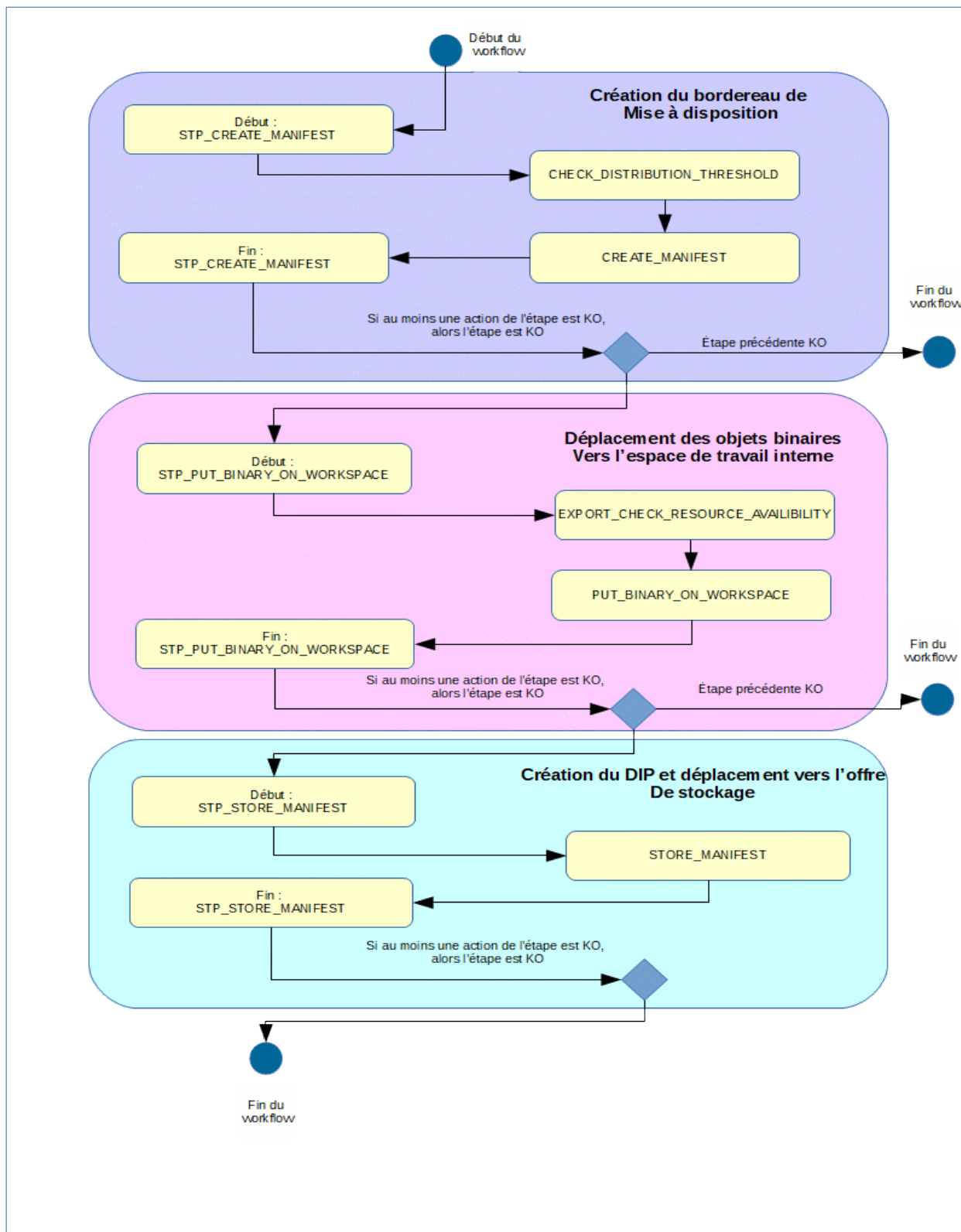
- **Règle** : tâche consistant à déplacer les objets binaires mentionnés dans le bordereau vers l'espace de travail interne (« workspace »)
- **Type** : bloquant
- **Statuts** :
 - OK : les objets binaires ont été déplacés vers le workspace avec succès (PUT_BINARY_ON_WORKSPACE.OK = Succès du déplacement des objets binaires de l'offre de stockage vers l'espace de travail interne)
 - KO : le déplacement des objets binaires vers le workspace a échoué car un ou plusieurs de ces objets étaient introuvables (PUT_BINARY_ON_WORKSPACE.KO = Échec du déplacement des objets binaires de l'offre de stockage vers l'espace de travail interne)
 - FATAL : une erreur technique est survenue lors du déplacement des objets binaires de l'offre de stockage vers le workspace (PUT_BINARY_ON_WORKSPACE.FATAL = Erreur technique lors du déplacement des objets binaires de l'offre de stockage vers l'espace de travail interne)

11.1.3. Processus de création du SIP de transfert et de son déplacement vers l'offre de stockage (STP_STORE_MANIFEST)

- **Règle** : tâche consistant à créer le SIP de transfert et à le déplacer vers l'offre de stockage
- **Type** : bloquant
- **Statuts** :
 - OK : le SIP de transfert a été créé et enregistré sur les offres de stockage avec succès (STP_STORE_MANIFEST.OK = Succès du processus de création de l'archive et de son déplacement vers l'offre de stockage)
 - FATAL : une erreur technique est survenue lors de la création et de l'enregistrement du SIP de transfert sur les offres de stockage déplacement des objets binaires de stockage vers l'espace de travail interne (STORE_MANIFEST.FATAL = Erreur technique lors du processus de création de l'archive et de son déplacement vers l'offre de stockage)

11.1.4. Structure du Workflow de préparation du transfert des unités archivistiques et des groupes d'objets

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



11.2.Workflow d'acquittement du transfert des unités archivistiques et des groupes d'objets

Cette section décrit le processus (workflow) d'acquittement du transfert mis en œuvre dans la solution logicielle Vitam.

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.

11.2.1.Processus de préparation de l'acquittement du transfert (STP_TRANSFER_REPLY_PREPARATION)

11.2.1.1.Vérification de l'ATR de l'acquittement du transfert VERIFY_ARCHIVAL_TRANSFER_REPLY

- **Règle** : tâche consistant à vérifier l'ATR de l'acquittement du transfert
- **Type** : bloquant
- **Statuts** :
 - OK : la vérification de l'ATR de l'acquittement du transfert a bien été effectuée. (VERIFY_ARCHIVAL_TRANSFER_REPLY.OK = Succès de la vérification de l'ATR de l'acquittement du transfert)
 - KO : la vérification de l'ATR de l'acquittement du transfert n'a pas été effectuée. (VERIFY_ARCHIVAL_TRANSFER_REPLY.KO = Échec de la vérification de l'ATR de l'acquittement du transfert)
 - FATAL : une erreur technique est survenue lors de la vérification de l'ATR de l'acquittement du transfert (VERIFY_ARCHIVAL_TRANSFER_REPLY..FATAL = Erreur technique lors de la vérification de l'ATR de l'acquittement du transfert)

11.2.1.2.Sauvegarde de l'ATR de l'acquittement du transfert SAVE_ARCHIVAL_TRANSFER_REPLY

- **Règle** : tâche consistant à sauvegarder l'ATR de l'acquittement du transfert
- **Type** : bloquant
- **Statuts** :
 - OK : la sauvegarde de l'ATR de l'acquittement du transfert a bien été effectuée. (SAVE_ARCHIVAL_TRANSFER_REPLY.OK = Succès de la sauvegarde de l'ATR de l'acquittement du transfert)
 - KO : la sauvegarde de l'ATR de l'acquittement du transfert n'a pas été effectuée. (SAVE_ARCHIVAL_TRANSFER_REPLY.KO = Échec de la sauvegarde de l'ATR de l'acquittement du transfert)
 - FATAL : une erreur technique est survenue lors de la sauvegarde de l'ATR de l'acquittement du transfert (SAVE_ARCHIVAL_TRANSFER_REPLY.FATAL = Erreur technique lors de la sauvegarde de l'ATR de l'acquittement du transfert)

11.2.1.3.Vérification du statut de l'ATR et stockage sur l'espace de travail CHECK_ATR_AND_ADD_IT_TO_WORKSPACE

- **Règle** : tâche consistant à vérifier le statut de l'ATR et procéder au stockage sur l'espace de travail.
- **Type** : bloquant
- **Statuts** :

- OK : la vérification du statut de l'ATR et le stockage sur l'espace de travail à bien été effectué (CHECK_ATR_AND_ADD_IT_TO_WORKSPACE.OK = Succès de la vérification du statut de l'ATR et du stockage sur l'espace de travail)
- KO : la vérification du statut de l'ATR et le stockage sur l'espace de travail n'a pas été effectuée. (CHECK_ATR_AND_ADD_IT_TO_WORKSPACE.KO = Échec de la vérification du statut de l'ATR et du stockage sur l'espace de travail)
- FATAL : une erreur technique est survenue lors de la vérification du statut de l'ATR et du stockage sur l'espace de travail (CHECK_ATR_AND_ADD_IT_TO_WORKSPACE.FATAL = Erreur technique lors de la vérification du statut de l'ATR et du stockage sur l'espace de travail)

11.2.1.4. Préparation des unités archivistiques à purger TRANSFER_REPLY_UNIT_PREPARATION

- **Règle** : tâche consistant à préparer les unités archivistiques à purger
- **Type** : bloquant
- **Statuts** :
 - OK : la préparation des unités archivistiques à purger à bien été effectué (TRANSFER_REPLY_UNIT_PREPARATION.OK = Succès de la préparation des unités archivistiques à purger)
 - KO : la préparation des unités archivistiques à purger n'a pas été effectué (TRANSFER_REPLY_UNIT_PREPARATION.KO = Échec de la préparation des unités archivistiques à purger)
 - FATAL : une erreur technique est survenue lors de la préparation des unités archivistiques à purger (TRANSFER_REPLY_UNIT_PREPARATION.FATAL = Erreur technique lors de la la préparation des unités archivistiques à purger)

11.2.1.5. Suppression du SIP de transfert TRANSFER_REPLY_DELETE_SIP

- **Règle** : tâche consistant à supprimer le SIP de transfert
- **Type** : bloquant
- **Statuts** :
 - OK : la suppression du SIP de transfert à bien été effectué (TRANSFER_REPLY_DELETE_SIP.OK = Succès de la suppression du SIP de transfert)
 - KO : la suppression du SIP de transfert n'a pas été effectué (TRANSFER_REPLY_DELETE_SIP.KO = Échec de la suppression du SIP de transfert)
 - FATAL : une erreur technique est survenue lors de la suppression du SIP de transfert (TRANSFER_REPLY_DELETE_SIP.FATAL = Erreur technique lors de la suppression du SIP de transfert)

11.2.2. Processus de purge des unités archivistiques (STP_TRANSFER_REPLY_DELETE_UNIT)

Purge des unités archivistiques TRANSFER_REPLY_DELETE_UNIT

- **Règle** : tâche consistant à supprimer les unités archivistiques
- **Type** : bloquant
- **Statuts** :
 - OK : la suppression des unités archivistiques à bien été effectué (TRANSFER_REPLY_DELETE_UNIT.OK = Succès de la suppression des unités archivistiques)
 - KO : la suppression des unités archivistiques n'a pas été effectué (TRANSFER_REPLY_DELETE_SIP.KO = Échec de la suppression des unités archivistiques)

- FATAL : une erreur technique est survenue lors de la suppression du SIP de transfert (TRANSFER_REPLY_DELETE_SIP.FATAL = Erreur technique lors de la suppression des unités archivistiques)

11.2.3. Processus de préparation de la purge des groupes d'objets techniques (STP_TRANSFER_REPLY_OBJECT_GROUP_PREPARATION)

Préparation de la purge des groupes d'objets techniques TRANSFER_REPLY_OBJECT_GROUP_PREPARATION

- **Règle** : tâche consistant à préparer la purge des groupes d'objets techniques.
- **Type** : bloquant
- **Statuts** :
 - OK : la préparation de la purge des groupes d'objets techniques à bien été effectué (TRANSFER_REPLY_OBJECT_GROUP_PREPARATION.OK = Succès de la préparation de la purge des groupes d'objets techniques)
 - KO : la préparation de la purge des groupes d'objets techniques n'a pas été effectué (TRANSFER_REPLY_OBJECT_GROUP_PREPARATION.KO = Échec de la préparation de la purge des groupes d'objets techniques)
 - FATAL : une erreur technique est survenue lors de la préparation de la purge des groupes d'objets techniques (TRANSFER_REPLY_OBJECT_GROUP_PREPARATION.FATAL = Erreur technique lors de la suppression des unités archivistiques)

11.2.4. Processus de purge des groupes d'objets techniques dont les unités archivistiques parentes sont purgées (STP_TRANSFER_REPLY_DELETE_OBJECT_GROUP)

Purge des groupes d'objets techniques dont les unités archivistiques parentes sont purgées TRANSFER_REPLY_DELETE_OBJECT_GROUP

- **Règle** : tâche consistant à purger les groupes d'objets techniques dont les unités archivistiques parentes sont purgées
- **Type** : bloquant
- **Statuts** :
 - OK : la purge des groupes d'objets techniques dont les unités archivistiques parentes sont purgées à bien été effectué (TRANSFER_REPLY_DELETE_OBJECT_GROUP.OK = Succès de la purge des groupes d'objets techniques dont les unités archivistiques parentes sont purgées)
 - KO : la purge des groupes d'objets techniques dont les unités archivistiques parentes sont purgées n'a pas été effectué (TRANSFER_REPLY_DELETE_OBJECT_GROUP.KO = Échec de la purge des groupes d'objets techniques dont les unités archivistiques parentes sont purgées)
 - FATAL : une erreur technique est survenue lors de la purge des groupes d'objets techniques dont les unités archivistiques parentes sont purgées (TRANSFER_REPLY_DELETE_OBJECT_GROUP.FATAL = Erreur technique lors de la purge des groupes d'objets techniques dont les unités archivistiques parentes sont purgées)

11.2.5. Processus de détachement des groupes d'objets techniques dont certaines unités archivistiques parentes sont purgées (STP_TRANSFER_REPLY_DETACH_OBJECT_GROUP)

Établissement de la liste des éléments OBJECTS_LIST_EMPTY

- **Règle** : tâche consistant à établir la liste des objets
- **Type** : bloquant
- **Statuts** :
 - OK : l'établissement de la liste des objets a été réalisé avec succès (OBJECTS_LIST_EMPTY.OK = Succès de l'établissement de la liste des objets)
 - KO : l'établissement de la liste des objets n'a pas été réalisé (OBJECTS_LIST_EMPTY.KO = Échec de l'établissement de la liste des objets)
 - FATAL : une erreur technique est survenue lors de l'établissement de la liste des objets (OBJECTS_LIST_EMPTY.FATAL = Erreur technique lors de l'établissement de la liste des objets)
 - WARNING : il n'y a pas d'objet dans le SIP (Avertissement lors de l'établissement de la liste des objets : il n'y a pas d'objet pour cette étape)

11.2.6. Processus de préparation du Registre des Fonds suite à l'acquittement du transfert (STP_TRANSFER_REPLY_ACCESSION_REGISTER_PREPARATION)

Préparation du Registre des Fonds suite à l'acquittement du transfert TRANSFER_REPLY_ACCESSION_REGISTER_PREPARATION

- **Règle** : tâche consistant à préparer le Registre des Fonds suite à l'acquittement du transfert
- **Type** : bloquant
- **Statuts** :
 - OK : la préparation du Registre des Fonds suite à l'acquittement du transfert a été réalisé avec succès (TRANSFER_REPLY_ACCESSION_REGISTER_PREPARATION.OK = Succès de la préparation du Registre des Fonds suite à l'acquittement du transfert)
 - KO : la préparation du Registre des Fonds suite à l'acquittement du transfert n'a pas été réalisé (TRANSFER_REPLY_ACCESSION_REGISTER_PREPARATION.KO = Échec de la préparation du Registre des Fonds suite à l'acquittement du transfert)
 - FATAL : une erreur technique est survenue lors de la préparation du Registre des Fonds suite à l'acquittement de transfert (TRANSFER_REPLY_ACCESSION_REGISTER_PREPARATION.FATAL = Erreur technique lors de la préparation du Registre des Fonds suite à l'acquittement de transfert)

11.2.7. Processus de mise à jour du Registre des Fonds suite à l'acquittement du transfert (STP_TRANSFER_REPLY_ACCESSION_REGISTER_UPDATE)

Mise à jour du Registre des Fonds suite à l'acquittement du transfert TRANSFER_REPLY_ACCESSION_REGISTER_UPDATE

- **Règle** : tâche consistant à mettre à jour le Registre des Fonds suite à l'acquittement du transfert
- **Type** : bloquant
- **Statuts** :
 - OK : la mise à jour du Registre des Fonds suite à l'acquittement du transfert a été réalisé avec succès (TRANSFER_REPLY_ACCESSION_REGISTER_UPDATE.OK = Succès de la mise à jour du

Registre des Fonds suite à l’acquittement du transfert)

- KO : la mise à jour du Registre des Fonds suite à l’acquittement du transfert n’a pas été réalisé (TRANSFER_REPLY_ACCESSION_REGISTER_UPDATE.KO = Échec de la mise à jour du Registre des Fonds suite à l’acquittement du transfert)
- FATAL : une erreur technique est survenue lors de la mise à jour du Registre des Fonds suite à l’acquittement du transfert (TRANSFER_REPLY_ACCESSION_REGISTER_UPDATE.FATAL = Erreur technique lors de la mise à jour du Registre des Fonds suite à l’acquittement du transfert)

11.2.8.Processus de génération du rapport d’acquittement du transfert (STP_TRANSFER_REPLY_REPORT_GENERATION)

Génération du rapport d’acquittement du transfert TRANSFER_REPLY_REPORT_GENERATION

- **Règle** : tâche consistant à générer le rapport d’acquittement du transfert
- **Type** : bloquant
- **Statuts** :
 - OK : la génération du rapport d’acquittement du transfert a été réalisé avec succès (TRANSFER_REPLY_REPORT_GENERATION.OK = Succès de la génération du rapport d’acquittement du transfert)
 - KO : la génération du rapport d’acquittement du transfert n’a pas été réalisé (TRANSFER_REPLY_REPORT_GENERATION.KO = Échec de la génération du rapport d’acquittement du transfert)
 - FATAL : une erreur technique est survenue lors de la génération du rapport d’acquittement du transfert (TRANSFER_REPLY_REPORT_GENERATION.FATAL = Erreur technique lors de la génération du rapport d’acquittement du transfert)
 - WARNING : avertissement lors de la génération du rapport d’acquittement du transfert (TRANSFER_REPLY_REPORT_GENERATION.WARNING = Avertissement lors de la génération du rapport d’acquittement du transfert)

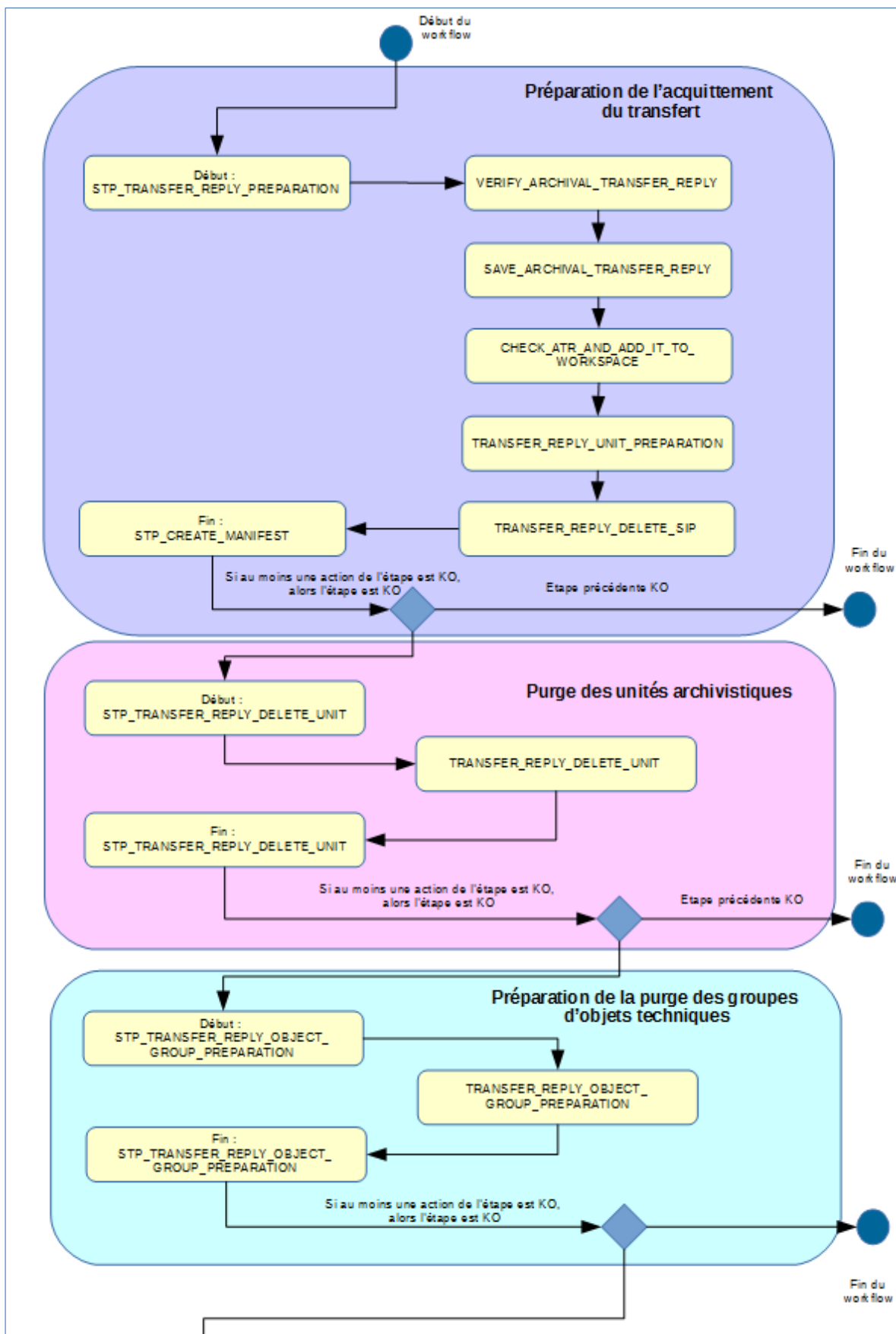
11.2.9.Processus de finalisation de l’acquittement du transfert (STP_TRANSFER_REPLY_FINALIZATION)

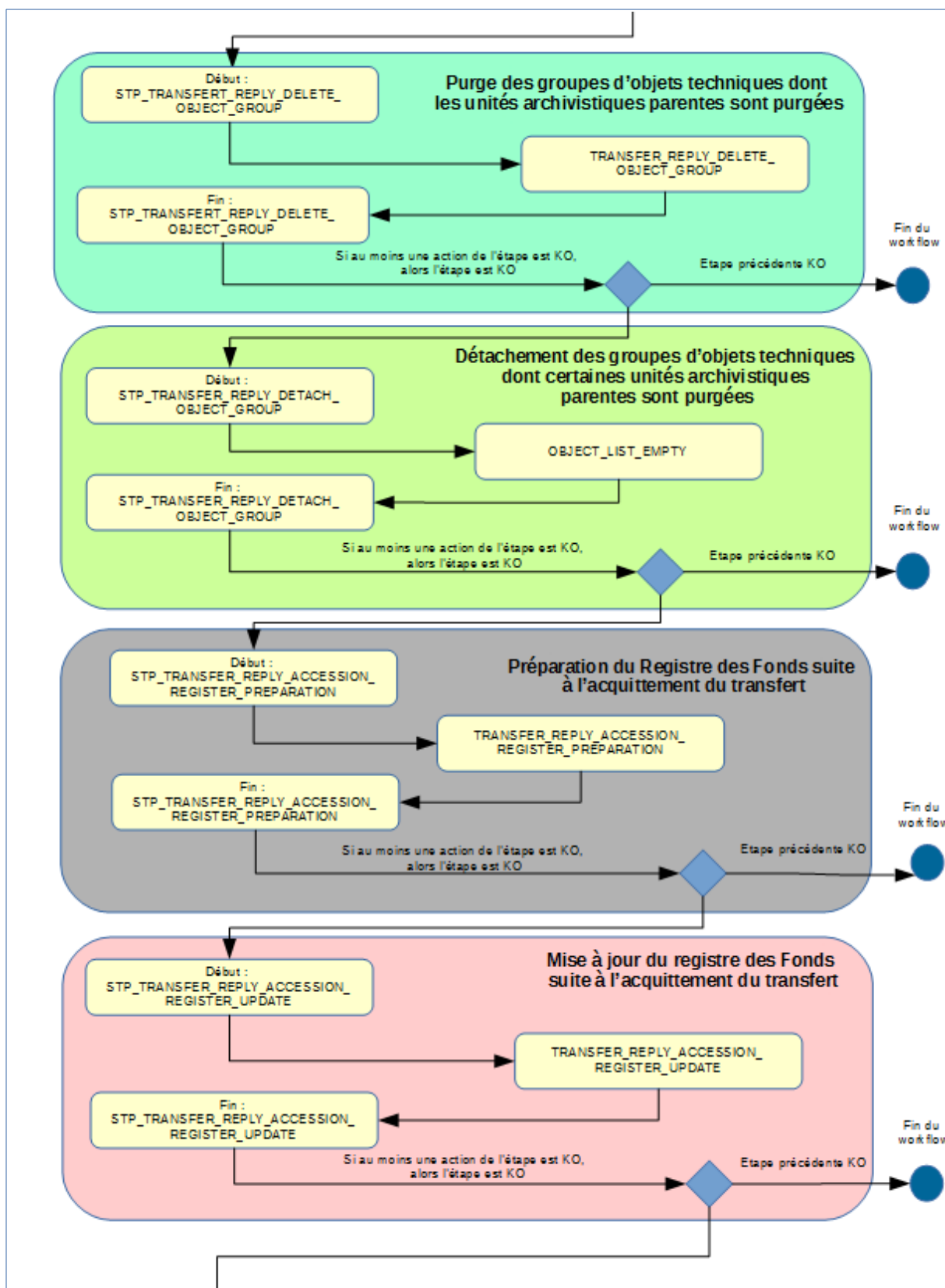
Finalisation de l’acquittement de transfert TRANSFER_REPLY_FINALIZATION

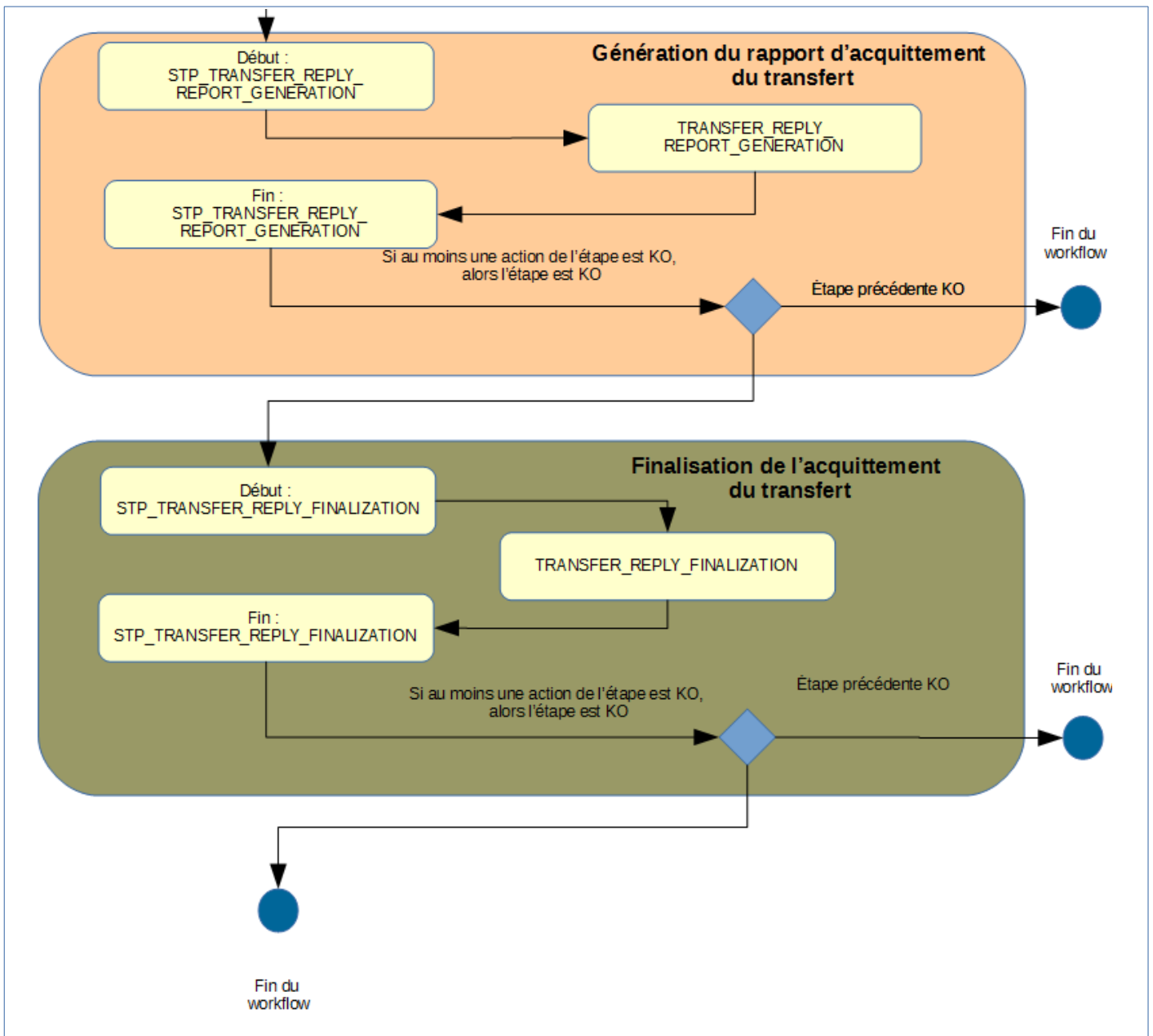
- **Règle** : tâche consistant à finaliser l’acquittement du transfert
- **Type** : bloquant
- **Statuts** :
 - OK : la finalisation de l’acquittement du transfert a été réalisé avec succès (TRANSFER_REPLY_FINALIZATION.OK = Succès de la finalisation de l’acquittement du transfert)
 - KO : la finalisation de l’acquittement du transfert n’a pas été réalisé (TRANSFER_REPLY_FINALIZATION.KO = Échec de la finalisation de l’acquittement du transfert)
 - FATAL : une erreur technique est survenue lors de la finalisation de l’acquittement du transfert (TRANSFER_REPLY_FINALIZATION.FATAL = Erreur technique lors de la finalisation de l’acquittement du transfert)

11.2.10. Structure du Workflow d'acquittement du transfert des unités archivistiques et des groupes d'objets

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







11.2.11. Rapport d'acquittement du transfert

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

11.2.11.1. Exemple de fichier JSONL : rapport d'acquittement du transfert

```

{
  "id": "6",
  "type": "TRANSFER",
  "status": "DELETED",
  "originatingAgency": "Service_parcheur",
  "reportType": "TRANSFER_REPLY",
  "endDateTime": "2019-11-14T13:56:59.925",
  "parameters": [
    {
      "id": "1",
      "type": "Unit",
      "status": "DELETED",
      "originatingAgency": "Service_parcheur"
    },
    {
      "id": "2",
      "type": "Unit",
      "status": "DELETED",
      "originatingAgency": "Vitam"
    },
    {
      "id": "3",
      "type": "Unit",
      "status": "DELETED",
      "originatingAgency": "Service_parcheur"
    }
  ]
}
    
```



```

+ + type = "LIRE";
+ + status = "DELETED";
+ + log = "##### h fjt x m3 st dlon cvd loggamaq";
+ + log_instaling_Agency = "Service_prestateur";
+ }
}
+ id = "ach_sommaire h h fjt gva mg alon cvd x2 lammaq";
+ genera = (
+ + id = "ach_sommaire h h fjt gva mg alon cvd x2 lammaq";
+ + type = "ObjectGroup";
+ + status = "DELETED";
+ + log = "##### h fjt x m3 st dlon cvd loggamaq";
+ + log_instaling_Agency = "Service_prestateur";
+ + log_cofid = "##### n h fjt gva mg alon cvd x2 lammaq";
+ }
}
+ id = "ach_sommaire h h fjt gva mg alon cvd f jammaq";
+ genera = (
+ + id = "ach_sommaire h h fjt gva mg alon cvd f jammaq";
+ + type = "ObjectGroup";
+ + status = "DELETED";
+ + log = "##### h fjt x m3 st dlon cvd f jammaq";
+ + log_instaling_Agency = "Service_prestateur";
+ + log_cofid = "##### n h fjt gva mg alon cvd f jammaq";
+ }
}
+ id = "ach_sommaire h h fjt gva mg alon cvk mcamaq";
+ genera = (
+ + id = "ach_sommaire h h fjt gva mg alon cvk mcamaq";
+ + type = "ObjectGroup";
+ + status = "DELETED";
+ + log = "##### h fjt x m3 st dlon cvk mcamaq";
+ + log_instaling_Agency = "Vitam";
+ + log_cofid = "##### n h fjt gva mg alon cvk mcamaq";
+ }
}
+ id = "ach_sommaire h h fjt gva mg alon cvk mg amaq";
+ genera = (
+ + id = "ach_sommaire h h fjt gva mg alon cvk mg amaq";
+ + type = "ObjectGroup";
+ + status = "DELETED";
+ + log = "##### h fjt x m3 st dlon cvk mcamaq";
+ + log_instaling_Agency = "Vitam";
+ + log_cofid = "##### n h fjt gva mg alon cvk mcamaq";
+ }
}
+ id = "ach_sommaire h h fjt gva mg alon cvk mg amaq";

```

```

→ "params": {
→   → "id": "aebaaaaavhnbfxgaavmgalonewkmcqaabq",
→   → "type": "ObjectGroup",
→   → "status": "DELETED",
→   → "opi": "aeaaaaaa2hfcj4xaa3s6alonewkafiaaaa",
→   → "originatingAgency": "Vitam",
→   → "objectIds": ["aeaaaaaaahnbfxgaavmgalonewkmcqaaba"]
→ }
}
→ "id": "aebaaaaavhnbfxgaavmgalonewpxzyaaba",
→ "params": {
→   → "id": "aebaaaaavhnbfxgaavmgalonewpxzyaaba",
→   → "type": "ObjectGroup",
→   → "status": "DELETED",
→   → "opi": "aeaaaaaa2hfcj4xaa3s6alonewplyaaaa",
→   → "originatingAgency": "RATP",
→   → "objectIds": ["aeaaaaaaahnbfxgaavmgalonewpxzyaaa"]
→ }
}

```

11.2.11.2. Détail du rapport

La partie « OperationSummary » c'est-à-dire le bloc racine du rapport et correspondant au résumé de l'opération est composé des champs suivants :

- « Tenant » : tenant sur lequel l'opération de modification en masse a été lancée
- « evId »:identifiant de l'événement
- « outcome » : statut de l'événement
- « outMsg » : détail du résultat des données de l'événement
- « rightsStatementIdentifiant » : identifiant des données référentielles en vertu desquelles l'opération peut s'exécuter.
- « evDetData » : détails des données de l'événement

La partie « ReportSummary » c'est-à-dire le bloc situé sous la racine du rapport et correspondant au résumé du rapport est composé 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 de l'opération (dernier evDateTime de l'event master de l'opération dans le journal des opérations)
- « reportType » : correspond au modèle du rapport.
- « vitamResults » : correspond au nombre de OK, KO et 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 ce champ est vide.

La partie « ReportDetail » contient les détails de l'opération d'acquittement du transfert effectuée sur chacune des unités archivistiques.

- « `_id` » : identifiant de l'unité archivistique ou de l'objet sur lequel l'action d'acquittement du transfert s'est effectuée.
- « `params` » : informations sur l'unité archivistique ou le groupe d'objet
 - « `id` » : identifiant de l'unité archivistique ou de l'objet
 - « `type` » : collection sur laquelle l'opération a été réalisée (UNIT ou OBJECTGROUP)
 - « `status` » : statut au regard de l'action d'acquittement du transfert :
 - **ALREADY_IN_TRANSFER** : unité archivistique ayant déjà subi une demande de transfert.
 - **NOT_DESTROYABLE_HAS_CHILD_UNITS** : unité non supprimable, car elle a des enfants et sa suppression entraînerait une incohérence dans le graphe.
 - **DELETED** : l'unité a effectivement été supprimée
 - « `opi` » : identifiant de l'opération d'entrée
 - « `originatingAgency` » : service producteur
 - « `objectIds` » : identifiant du groupe d'objets techniques rattaché à l'unité archivistique

12. RESTAURATION DE MÉTADONNÉES ESSENTIELLES (REVERT ESSENTIAL METADATA)

Cette section décrit le processus (workflow REVERT_ESSENTIAL_METADATA) utilisé lors d'une opération de restauration de métadonnées essentielles. Cette fonctionnalité nécessite d'avoir les droits requis pour pouvoir intervenir sur les métadonnées.

Cette restauration peut être demandée suite à la perte de métadonnées consécutive à une erreur métier (et non technique) car ces métadonnées ont vocation à être conservées dans le temps. Il s'agit de restaurer les métadonnées enregistrées dans la collection Unit, telle qu'existantes avant une opération particulière caractérisée par son identifiant. Les métadonnées concernées sont limitées pour l'instant au titre (Title ou Title_*), au niveau de description (DescriptionLevel) et à la description (Description ou Description_*). 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.

Workflow de restauration de métadonnées essentielles (REVERT_ESSENTIAL_METADATA)

La restauration des métadonnées essentielles peut être demandée suite à la perte de données et/ou de métadonnées consécutive à une erreur métier (et non technique).

Règle : opération consistant restaurer les métadonnées essentielles d'une ou plusieurs AU à partir d'un identifiant d'opération et éventuellement de la liste des métadonnées concernées.

Type : bloquant

Statuts :

- OK : la restauration des métadonnées essentielles a bien été effectuée (REVERT_ESSENTIAL_METADATA.OK=Succès de la restauration des métadonnées essentielles)
- KO : la restauration des métadonnées essentielles n'a pas été effectuée en raison d'une erreur (REVERT_ESSENTIAL_METADATA.KO=Échec de la restauration des métadonnées essentielles)
- FATAL : une erreur technique est survenue lors de la restauration des métadonnées essentielles (REVERT_ESSENTIAL_METADATA.FATAL=Erreur technique lors de la restauration des métadonnées essentielles)
- WARNING : la restauration des métadonnées essentielles n'a pas été effectuée en raison d'une erreur de mise à jour d'une ou plusieurs unités archivistiques (REVERT_ESSENTIAL_METADATA.WARNING=Avertissement lors de la restauration des métadonnées essentielles)

12.1. Processus de préparation de la restauration des métadonnées essentielles (STP_CHECK_REVERT)

Préparation de la restauration des métadonnées essentielles REVERT_CHECK

- **Règle :** Etape de contrôle des droits d'écriture donnés par le contrat d'accès et de vérification que l'utilisateur a bien des droits d'écriture sur les métadonnées descriptives. Préparation de la requête de restauration des métadonnées essentielles à partir de l'identifiant de l'opération et des éléments de la requête.
- **Type :** bloquant
- **Statuts :**
 - OK : La liste des requêtes de restauration a été constituée avec succès à partir des éléments renseignés dans la requête (REVERT_CHECK.OK=Succès de la préparation de la restauration des métadonnées essentielles)

- KO : Il n’y a pas d’unités archivistiques concernées par la demande de restauration ou une opération plus récente a déjà modifié les métadonnées dont on a demandé la restauration dans le cas où l’option « Force » est à « false » (REVERT_CHECK.KO=Échec de la préparation de la restauration des métadonnées essentielles)
- FATAL : Une erreur technique est survenue lors de la préparation de la requête de restauration des métadonnées essentielles (REVERT_CHECK.FATAL=Erreur technique lors de la préparation de la restauration des métadonnées essentielles)

12.2. Processus de traitement de la restauration des métadonnées essentielles (STP_REVERT_UPDATE)

Restauration des métadonnées essentielles REVERT_UPDATE_UNITS

- **Règle** : Exécution de la liste des requêtes de restauration des métadonnées et enregistrement des mises à jour des unités archivistiques en base, dans les index et dans les journaux de cycle de vie.
- **Type** : Non bloquant
- **Statuts** :
 - OK : la restauration des métadonnées essentielles des unités archivistiques a bien été effectuée (REVERT_UPDATE_UNITS.OK=Succès de la restauration des métadonnées essentielles)
 - WARNING : la restauration des métadonnées des unités archivistiques n’a pas été effectuée ou a été effectuée partiellement (REVERT_UPDATE_UNITS.WARNING=Avertissement lors de la restauration des métadonnées essentielles)
 - FATAL : une erreur technique est survenue lors de la restauration des métadonnées essentielles (REVERT_UPDATE_UNITS.FATAL=Erreur technique lors de la restauration des métadonnées essentielles)

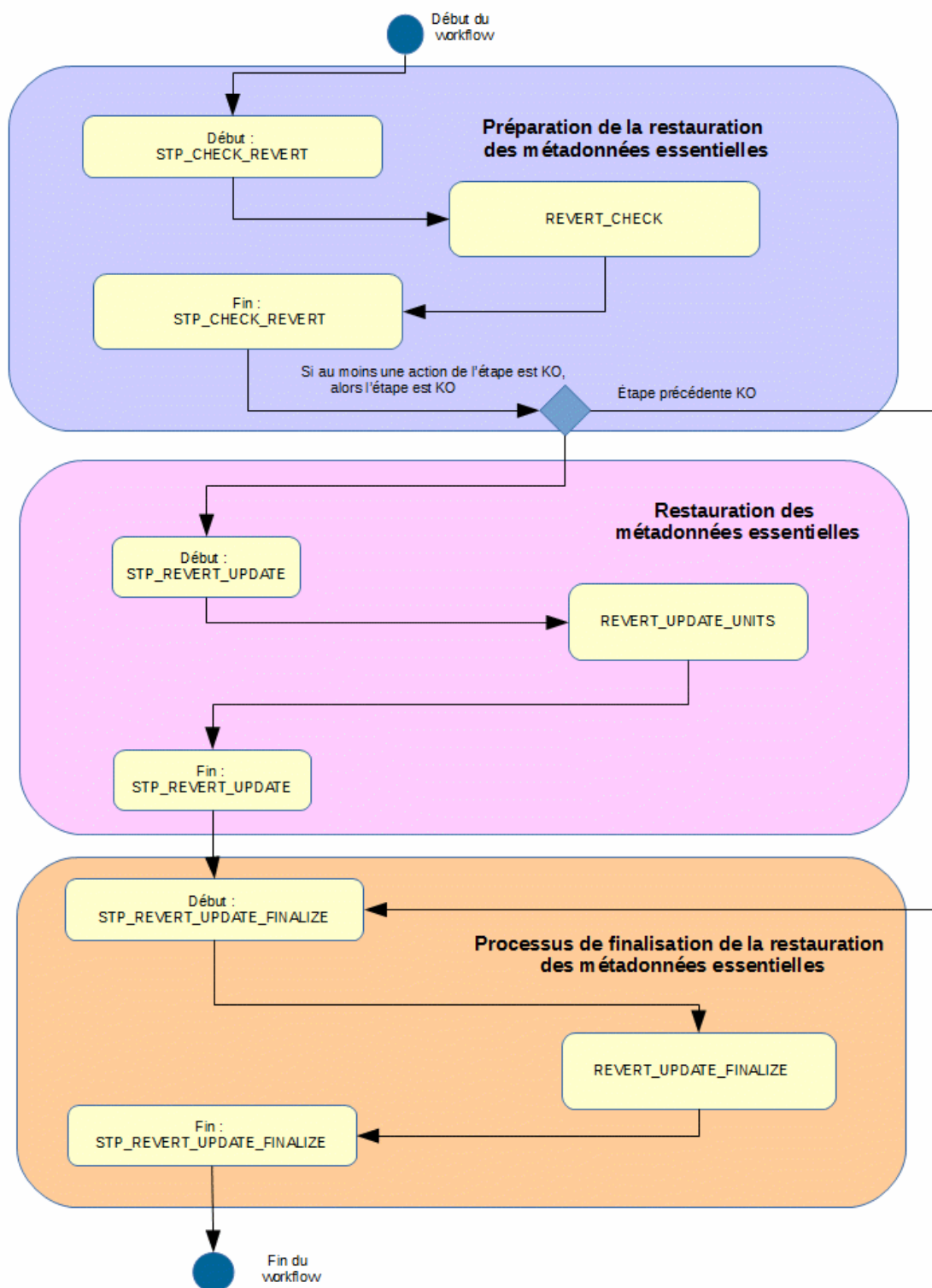
12.3. Processus de finalisation de la restauration des métadonnées essentielles (STP_REVERT_UPDATE_FINALIZE)

Finalisation de la restauration des métadonnées essentielles REVERT_UPDATE_FINALIZE

- **Règle** : Tâche consistant à générer le rapport de restauration des métadonnées essentielles des unités archivistiques concernées par la requête.
- **Type** : bloquant
- **Statuts** :
 - OK : la génération du rapport de restauration des métadonnées essentielles a bien été effectuée (REVERT_UPDATE_FINALIZE.OK=Succès de la finalisation de la restauration des métadonnées essentielle)
 - KO : la génération du rapport de restauration des métadonnées essentielles n’a pas été effectuée en raison d’une erreur (REVERT_UPDATE_FINALIZE.KO=Échec de la finalisation de la restauration des métadonnées essentielles)
 - WARNING : pas de résultats dans le rapport de restauration des métadonnées essentielles (REVERT_UPDATE_FINALIZE.WARNING=Avertissement lors de la finalisation de la restauration des métadonnées essentielles)
 - FATAL : une erreur technique est survenue lors de la génération du rapport de restauration des métadonnées essentielles (REVERT_UPDATE_FINALIZE.FATAL=Erreur technique lors de la finalisation de la restauration des métadonnées essentielles)

12.4. Structure du Workflow de mise à jour unitaire des métadonnées descriptive

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



12.5. Rapport de restauration de métadonnées essentielles des unités archivistiques.

Le rapport de restauration de métadonnées essentielles des unités archivistiques est un fichier JSONL généré par la solution logicielle Vitam lorsqu'une opération de restauration de métadonnées essentielles se termine. Cette section décrit la manière dont ce rapport est structuré.

12.5.1. Exemple de fichier JSONL : rapport de restauration de métadonnées essentielles

```

{
  "tenant":1,
  "evId":"aaaaaaaaaghd4unabuloalxzmbhjjaaaag",
  "evType":"REVERT_UPDATE_UNITS",
  "outcome":"OK",
  "outDetail":"REVERT_UPDATE_UNITS.OK",
  "outMsg":"Succ\u00E8s de la restauration des m\u00E9tadonn\u00E9es essentielles Detail- OK:1",
  "rightsStatementIdentifier":{"AccessContract":"ContratINR"},
  "evDetData":{"Event":"Mass update OK"}
}

{"evStartDateTime":"2021-02-23T10:03:59.624",
"evEndDateTime":"2021-02-23T10:04:01.481",
"reportType":"UPDATE_UNIT",
"vitamResults":{"OK":1,"KO":0,"WARNING":0,"total":1},
"extendedInfo":null}

{"(\\"#roots\\":{\\"aaaaaaahlsbxib4tkalxzgbejraaaaa\\"},\\"#query\\":{\\"#filter\\":{\\"#action\\":{\\"#set\\":{\\"Title\\":\\"BAD0431E2C5E80E5BD42D547A3ED596655

```

12.5.2. Détail du rapport

La partie **OperationSummary**, c'est-à-dire le bloc racine du rapport et correspondant au résumé de l'opération est composé des champs suivants :

- tenant : tenant sur lequel l'opération de restauration de métadonnées essentielles a été lancée
- evId : identifiant de l'événement
- evType : type d'opération
- outcome : statut de l'événement
- outDetail : détail de l'opération
- outMsg : détail du résultat des données de l'événement
- rightsStatementIdentifier : identifiants des données référentielles en vertu desquelles l'opération peut s'exécuter.
- EvDetData : détails des données de l'événement

La partie **ReportSummary**, c'est-à-dire le bloc situé sous la racine du rapport et correspondant au résumé du rapport est composé 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 de l'opération (dernier evDateTime de l'event master de l'opération dans le journal des opérations)
- reportType : correspond au modèle du rapport.
- vitamResults: correspond au nombre de OK, KO et 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 ce champ est vide.

La partie **Context** est vide.

La partie **ReportDetail** contient les détails de l'opération de modification en masse effectuée sur chacune des unités archivistiques.

- \$root : identifiant de l'unité archivistique impactée à la requête
- \$query : requête liée au détail de résultat au format "String"