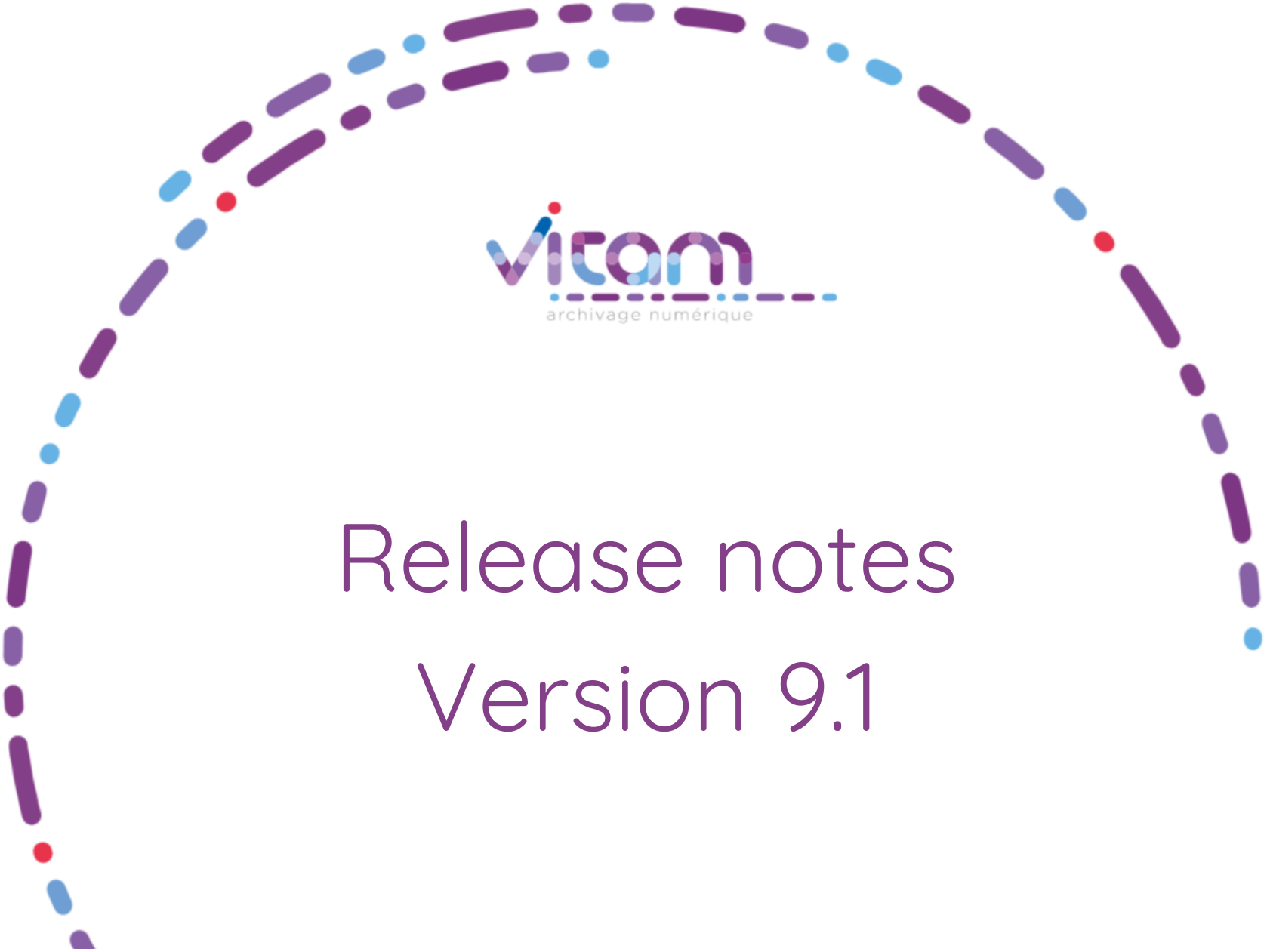

Release notes
Version 9.1

Printemps 2026

Nouveautés back-office – Vitam v9.1

Évolutions fonctionnelles

Collecte :

- Import dans le module de collecte :
  - Il est maintenant possible d'importer successivement plusieurs dataobjectpackage de SIPs dans une même transaction. Chaque import est tracé dans le journal des opérations par une opération d'import dédiée.
  - La gestion d'erreurs lors de l'import est améliorée.
    - A l'import d'un dataobjectpackage de SIPs en cas d'erreur lors de l'import, l'opération d'import est en erreur et la présence d'un "contenu en erreur" est indiquée au niveau de la transaction. Le type de traitement concerné et un résultat "KO" sont enregistrés sur la transaction. Les archives et binaires d'un même import sont identifiables par un identifiant de traitement d'import (batchId) dont la valeur est le GUID de l'opération d'import. Il est possible de rechercher les archives en erreur par cet identifiant et de consulter les erreurs sur les unités ou objets concernés afin de pouvoir les traiter manuellement.
    - A l'import d'une arborescence en erreur avec ou sans fichier de métadonnées (CSV ou JSONL), en cas d'erreur lors de l'import, le lot en erreur est identifié par son identifiant d'import et il est immédiatement purgé. Le type de traitement concerné et un résultat "PURGED" sont enregistrés sur la transaction. Il n'y a pas de journalisation donc pas d'opérations au sens du logbookoperation associé. Un détail des erreurs est retourné par API lorsque c'est possible (jusqu'à 10 détails différents cumulés).
- Mise à jour des métadonnées au moyen d'un fichier .csv identifiant les unités archivistiques à modifier par leur identifiant technique en alternative de la colonne File (on peut identifier les unités archivistiques soit par la colonne File, soit par la colonne _id).

Simplification des entrées :
Gestion des erreurs lors des versements

Les travaux visant à rendre affichables et recherchables les erreurs liées à l'entrée afin de simplifier la reprise de versements en erreur ont abouti à des résultats sur le service d'import par dataobjectpackage de SIP dans collecte. Parallèlement d'autres améliorations ont été réalisées sur les services d'imports arborescence.

Les prochaines versions viseront à homogénéiser dans les différents services d'import les traitements réalisés et la gestion des erreurs associée.

Entrée :

- Identification de format à l'entrée : mise à jour de Siegfried de la version 1.9.6 à la version 1.11.4.

Gestion des archives :

- Modification du service producteur sur un ensemble d'archives (unités archivistiques et/ou groupes d'objets techniques) d'une même entrée par une opération de réattribution de service producteur, tracée dans le journal des opérations.
  - Si des archives appartiennent à des entrées différentes ou que la sélection des archives d'une entrée est incomplète la demande de modification est rejetée, aucune modification n'est réalisée et l'opération est en erreur.
  - Dans le cas d'une modification du service producteur sur des archives en succès :
    - L'historique des services producteurs est enregistré dans les unités archivistiques et/ou groupes d'objets techniques concernés. Il est donc possible de rechercher directement sur la base de cet historique. Cet historique, quand il existe, est également enregistré dans les rapports d'élimination, d'acquittement du transfert et de suppression d'usages/versions.
    - La modification est tracée dans le journal du cycle de vie de chaque unité d'archives et/ou groupe d'objet technique concerné.
    - Les registres de fonds du service producteur d'origine et du service producteur cible sont mis à jour en conséquence. Le détail de registre correspondant à une entrée pour laquelle le service producteur des archives de cette entrée a été changé est réattribué au fonds du service producteur cible et les synthèses statistiques (Summary) des registres du service producteur d'origine et du service producteur cible sont mises à jour (soustraction des sorties pour l'un, addition des entrées pour l'autre)
    - Les droits d'accès associés au contrat d'accès s'appliquent en prenant en compte le nouveau service producteur

Audit / Préservation :

- Possibilité de définir un post traitement des métadonnées extraites dans le cadre d'un scénario de préservation. Ainsi, il est maintenant possible de modifier avant de les enregistrer les métadonnées "brutes" (nom des métadonnées et/ou contenu des valeurs, ajout ou suppression de métadonnées, etc.) extraites des documents/binaires dans le cadre de l'application d'un scénario de préservation d'extraction de métadonnées (en utilisant par exemple les griffons Tesseract ou ImageMagick, mais non limité à ces griffons). La modification se fait par la configuration d'un traitement JSLT au niveau du scénario de préservation.

    - Exemple : cette évolution ouvre la possibilité de paramétrer et de maîtriser la modélisation des métadonnées extraites liées à l'audiovisuel et ceci quel que soit le format d'export brut proposé par les différents outils d'extraction de métadonnées utilisés.
- Réidentification de format : mise à jour de Siegfried de la version 1.9.6 à la version 1.11.4.

Stockage, gestion de la donnée :

- Audit de cohérence interne d'offre de stockage : voir chapitre "Evolution technique", partie "Exploitation"

Administration fonctionnelle :

- Scénario de préservation :
    - Ajout d'un paramètre permettant de définir des règles de transformation lors d'une extraction de métadonnées techniques avec le griffon ImageMagick ou descriptives avec le griffon Tesseract.
- Référentiel des formats :
    - Mise à jour du référentiel Pronom de la version V109 à la version V122.

Évolutions techniques

Exploitation

- Gestion des logs, ElasticSearch-Log, optimisation de robustesse et performance :
  - Définition de mappings dédiés permettant d'éviter la double indexation de tous les champs des logs récoltés.
  - Ajout de la possibilité de configurer le nombre de shards/replicas par indices (défaut : 1 shard / 1 replica).
  - Désactivation par défaut des processors filebeat (rendu configurable par l'exploitant) permettant d'éviter d'enregistrer des données non utilisées dans les dashboards livrés.
    - Gains constatés : environ -50% d'espace disque utilisé en moins par document stocké.
    - Permet d'alléger la pression (cpu/ram) sur le cluster ElasticSearch-log et/ou de stocker sur une plus longue période les logs.
- Outillage pour l'investigation des erreurs :
  - Fourniture de 2 nouveaux playbooks d'exploitation (mongo_find & elastic_find) pour effectuer des recherches simples et récupérer les résultats à destination du support dans les bases Vitam (Mongo-Data & Elastic-Data).
- Audit de cohérence interne d'une offre de stockage :
  - Playbook permettant d'exécuter le diagnostic complet du statut de stockage d'une offre. Il permet de comparer le contenu stocké d'une offre à son contenu théorique - d'après son journal interne en base. (ansible-vitam-exploitation/offer_diag.yml).
- Correction du nommage des variables Ansible selon la convention établie :
  - Permet d'assurer la cohérence entre le nommage des services et les variables de configuration associées. Simplifie les automatisations dans les scripts et le support du code.
  - Renommage des variables suivantes :
    - Vitam :
      - accessexternal -> access_external
      - accessinternal -> access_internal
      - batchreport -> batch_report
      - ingestexternal -> ingest_external
      - ingestinternal -> ingest_internal

        - elastickibanainterceptor -> elastic_kibana_interceptor
        - storageengine -> storage
        - storageofferdefault -> offer
    - Renommage des groupes suivants dans les fichiers d'inventaires :
        - [hosts_storage_engine] -> [hosts_storage]
        - [hosts_storage_offer_default] -> [hosts_offer]
- Possibilité de surcharger certains paramètres de vitam_defaults par composant. Permet d'appliquer simplement une configuration spécifique sur certains composants seulement.
    - Ex. vitam.worker.elasticSearchScrollTimeoutInMilliseconds peut maintenant être spécifié uniquement pour le worker au lieu de l'appliquer sur tous les composants vitam.
- Possibilité de définir une liste de tenant sous forme de range au lieu d'une liste unitaire. Permet d'améliorer la maintenabilité et lisibilité de la configuration.
    - Ex. *vitam_tenants_ids: [0-10]* au lieu de *vitam_tenant_ids: [0, 1, 2, 3, 4, 5, 6, 7, 8, 9, 10]*

Réindexation différentielle

- Pour les collections Unit et ObjectGroup, il est maintenant possible de faire une réindexation différentielle à partir d'une date donnée plutôt qu'une réindexation complète.
    - Dans un contexte de très forte volumétrie sur ces collections (centaines de millions, milliards), quand les durées de réindexation peuvent être très longue, cela permet par exemple lors de montée de version nécessitant une réindexation de conserver le service vitam **partiellement** ouvert le temps d'une première réindexation complète avant d'interrompre le service et basculer vers ce nouvel index pour procéder à une réindexation différentielle des éléments correspondant à un dernier delta non réindexé (i.e delta correspondant aux éléments modifiés, créés ou supprimé pendant la période de réindexation). Le service fourni est limité, les suppressions et les réorganisations sont proscrites en l'état. Le service peut rester ouvert en lecture et en écriture sans suppression d'AU et de GOT car le dernier delta ne peut détecter les suppressions. Ainsi les services d'élimination et de suppressions ne doivent pas être utilisés pendant la réindexation en parallèle mais les consultations, les versements et les modifications de métadonnées descriptives et de gestions peuvent toujours être utilisés.

  Veuillez-vous référer à la documentation d'exploitation pour plus d'informations.

Reconstruction

- Les lectures d'objets sur l'offre de stockage ont été parallélisées, permettant un gain de performances d'environ 80%, particulièrement dans le cas de cluster MongoDB multi-shardés.

Evolutions du scheduler

- Le scheduler est maintenant un singleton et utilise une base de données locale.
    - L'usage d'une base de données MongoDb pour le scheduler a été exclu après le constat par des utilisateurs d'une mauvaise gestion de l'unicité des traitements. Dans ce contexte, plusieurs instances d'une même tache planifiée ont pu être lancée. L'évolution corrige ce défaut mais introduit un SPOF (*Single Point of Failure*), nous recommandons donc de superviser activement ce service.

- **Reconstruction des chaînes de sécurisation (Fonctionnalité en cours d'évaluation - à ne pas utiliser en production en l'état)**
- Cette fonction permet de construire de nouvelles versions des différentes chaînes de sécurisation en changeant les paramètres de version de chaque chaîne spécifique dans la configuration de Vitam. Elle s'applique aux chaines de sécurisation des journaux des opérations et des cycles de vie des groupes d'objets et des unités archivistiques. La chaîne précédente est conservée mais inactive, pour des usages futurs.
- Dans cette version, cette fonctionnalité n'apporte pas de plus-value fonctionnelle mais elle permet de reconstruire des chaînes endommagées sous supervision du programme Vitam.

Workflow

- Protection contre les annulations intempestives : depuis la version V8.1, il est possible de marquer (configuration) les étapes des workflows pour lesquelles l'annulation peut entrainer des conséquences graves. Dans ce cas l'annulation exige le passage d'un paramètre pour "forcer" l'annulation. Le workflow d'Ingest (entrée) était le seul à avoir reçu par défaut cette protection sur toutes les étapes dites de "comite", qui suivent les étapes de contrôles métiers et qui correspondent à des étapes d'enregistrement en base de

données et sur les offres de stockage des métadonnées et des données d'archives. Dans cette version, cette protection a été étendue aux workflows et étapes par défaut suivants :

Fichier	Id	Nom	Première étape non annulable
BigIngestWorkflow	BIG_WORKFLOW	Big Ingest Workflow	STP_OBJ_STORING
BulkAtomicUpdateUnitDescWorkflow	BULK_ATOMIC_UPDATE_UNIT_DESC	Bulk Atomic Update Descriptive Workflow	STP_UPDATE
CollectDeletionActionWorkflow	COLLECT_DELETION_ACTION	Deletion workflow in Collect	STP_DELETION_ACTION_DELETE_UNIT
CollectEliminationActionWorkflow	COLLECT_ELIMINATION_ACTION	Elimination action workflow in Collect	STP_ELIMINATION_ACTION_DELETE_UNIT
CollectIngestWorkflow	COLLECT_SIP_INGEST	Collect Ingest Workflow	STP_OBJ_STORING
CollectReclassificationWorkflow	COLLECT_RECLASSIFICATION	Reclassification in Collect	STP_UNIT_DETACHMENT
ComputeInheritedRulesDeleteWorkflow	COMPUTE_INHERITED_RULES_DELETE	inherited rules action workflow delete	STP_COMPUTE_INHERITED_RULES_DELETE
ComputeInheritedRulesWorkflow	COMPUTE_INHERITED_RULES	inherited rules action workflow	STP_COMPUTE_INHERITED_RULES
DataMigrationWorkflow	DATA_MIGRATION	migration Workflow	STP_MIGRATION_UNITS
DefaultFilingSchemeWorkflow	FILING_SCHEME	Default Filing Scheme Workflow	STP_UNIT_METADATA
DefaultHoldingSchemeWorkflow	HOLDING_SCHEME	Default Holding Scheme Workflow	STP_UNIT_METADATA
DefaultRulesUpdateWorkflow	UPDATE_RULES_ARCHIVE_UNITS	Default Update Rules in Archive Units	STP_INVALIDATE
DefaultUnitLifecycleTraceability	UNIT_LFC_TRACEABILITY	Default Unit Lifecycle Traceability	STP_UNIT_LFC_TRACEABILITY_FINALIZATION
DeleteGotVersionsWorkflow	DELETE_GOT_VERSIONS	Delete got versions workflow	STP_DELETE_GOT_VERSIONS_ACTION
EliminationActionWorkflow	ELIMINATION_ACTION	Elimination action workflow	STP_ELIMINATION_ACTION_DELETE_UNIT
ExportProbativeValueWorkflowV2	EXPORT_PROBATIVE_VALUE	export probative value	STP_PROBATIVE_VALUE_GENERATE_REPORT
IngestCleanupWorkflow	INGEST_CLEANUP	Ingest cleanup workflow	STP_INGEST_CLEANUP_DELETE_UNIT
MassUpdateUnitDescWorkflow	MASS_UPDATE_UNIT_DESC	Mass Update Workflow	STP_UPDATE
MassUpdateUnitRuleWorkflow	MASS_UPDATE_UNIT_RULE	Mass Update Units Rules Workflow	STP_INVALIDATE
PreservationWorkflow	PRESERVATION	preservation workflow	STP_PRESERVATION_ACTION
ReclassificationWorkflow	RECLASSIFICATION	Reclassification	STP_UNIT_DETACHMENT
RectificationAuditWorkflow	RECTIFICATION_AUDIT	Rectification Audit Workflow	STP_CORRECTIVE_AUDIT
RevertEssentialMetadataWorkflow	REVERT_ESSENTIAL_METADATA	Revert Essential Metadata Workflow	STP_REVERT_UPDATE
TransferReplyWorkflow	TRANSFER_REPLY	Transfer reply workflow	STP_TRANSFER_REPLY_PREPARATION

Mise à jour des COTS et librairies

- Montée de version de consul vers la 1.22.6
- Mise à jour de Siegfried de la version 1.9.6 à la version 1.11.4
- Mise à jour du référentiel Pronom de la V109 à la V122
- Montée de version de la suite ElasticSearch de la version 8.18.0 à la version 9.3.2
- Montée de version de MongoDB à la version 8.0.23

- Mise à jour de Curator à la version 9.0.0 pour compatibilité avec ES9+
  - Curator n'est plus livré par Elasticsearch au format deb/rpm. En conséquence il est remplacé par un build Vitam du binaire curator et livré par le paquet vitam-elasticsearch-curator au format deb/rpm.
- Montée de version des outils de monitoring :
  - Grafana : v11.6.1 → v12.4.2
  - Prometheus : v3.3.1 → v3.10.0
  - Alertmanager : v0.28.1 → v0.31.1
  - Elasticsearch exporter : v1.9.0 → v1.10.0
  - Node exporter : v1.9.1 → v1.10.2
  - Blackbox exporter : v0.26.0 → v0.28.0
  - MongoDB exporter : v0.44.0 → v0.49.0
- Suppression des outils extras suivants :
  - metricbeat
  - packetbeat
  - gatling
  - openldap_server

Evolution des API

Evolution d'API non rétro-compatible

- Mise à jour de l'API Collect External :
  - /transactions/{transactionId}/close

Nouvelles API

- Une nouvelle API Access External :
  - originatingAgencyReassignment
    - "sourceOriginatingAgency"
    - "targetOriginatingAgency"
    - "dslRequest"
    - PropagateToObjectGroups

Tests de non-régression

Collecte

- Pouvoir importer plusieurs SIPs dans une transaction
  - Adaptation des TNR existants
  - Création de nouveaux TNR

Montée de version / migration de données

OS et distribution

- Vitam V9.1 est compatible avec les OS AlmaLinux 9 et Debian 12.
-

Montée de version

- La mise à jour d'Elasticsearch utilisé dans la version v9.1 de Vitam (ES v9) impose une réindexation complète des indices créés initialement dans une ancienne version d'ES (ES V8-) lors de la montée de version. Autrement dit peu importe la version de Vitam considérée à l'instant t par une montée de version Vitam V9.1, pour toutes les instances ayant démarrée initialement sur une version inférieure ou égale à la V7.1 (en ES7), une réindexation complète des indices ES est obligatoire lors de la montée en version Vitam V9.1.
  - Pour les collections volumineuses tel que unit et objectgroup, il est maintenant possible de faire la réindexation complète en maintenant le service ouvert sauf en élimination / suppression puis en terminant par la réindexation différentielle des éléments manquants ajoutés ou modifiés (cf. Documentation d'exploitation sur la réindexation).

Nouveautés front-office - VitamUI v9.1

Ajout de fonctionnalités

APP Recherche, consultation et gestion des archives :

- Modification du service producteur sur un ensemble d'archives (unités archivistiques et/ou groupes d'objets techniques) par une opération de réattribution de service producteur. La sélection s'effectue sur une à plusieurs entrées d'archives. Les archives ne peuvent pas être réattribuées si elles sont de type "arbre de positionnement", si elles n'ont pas le même service producteur, s'il y a dépassement de seuil.
- Améliorations lors d'une mise à jour de règles de gestion :
  - Des améliorations ergonomiques ont été apportées au sein de la page dédiée aux fonctionnalités de mises à jour des règles de gestion afin de rendre plus intuitif la sortie de la page et le retour à la page d'accueil de l'APP Recherche, consultation et gestion des archives.

APP Profils documentaires :

- Possibilité d'ajouter des contrôles sur les rattachements dans les profils d'archivage (bloc UpdateOperation) et de générer un profil d'archivage incluant ces nouveaux contrôles.

APP Collecte :

- Amélioration de l'affichage des erreurs à la suite de l'import d'un SIP contenant des erreurs :
  - Dans la liste des archives de la transaction, ajout d'une pastille rouge pour identifier les unités archivistiques et/ou leurs objets associés ayant des erreurs.
  - Dans le détail d'une unité archivistique, présence d'une pastille rouge en haut du panneau latéral et nouveau composant plus visible, permettant de signaler davantage d'informations sur les erreurs rencontrées.

- Mise à jour des métadonnées au moyen d'un fichier .csv identifiant les unités archivistiques à modifier par leur identifiant technique en remplacement de la colonne File. En sachant que l'on peut identifier les unités archivistiques soit par la colonne File, soit par la colonne _id.
- Pouvoir ajouter un SIP ou un ZIP dans une transaction déjà existante.
- Affichage dans les IHM d'une à plusieurs erreurs à la suite de :
  - L'import d'une arborescence zippée ou non zippée contenant un fichier metadata.csv ou metadata.jsonl
  - La mise à jour par un réimport d'un fichier .csv ou .jsonl

  Le retour erreur est limité à 10 erreurs.

Ergonomie

Harmonisation du comportement des composants

- Suite au travail sur les fondations du design système et à l'harmonisation des composants, de nouveaux composants (Checkbox et Radio button) ont été modifiés, complétés, simplifiés en termes de couleurs, taille des textes, états (default, hover, etc.)
- Gestion de tous les cas de validation et d'erreurs possibles pour le composant d'upload.

Evolutions techniques

Sécurité

- Limitation des permissions accordées à l'administrateur générique lors de la subrogation d'une organisation. Cet utilisateur générique, à destination du support pour l'initialisation d'une organisation, est maintenant limité exclusivement à la création du premier administrateur nominatif de l'organisation.

Exploitation

- Refonte complète de la PKI VitamUI :
  - Séparation des certificats clients/serveurs.

- Mise à jour des stores au format p12 au lieu de jks.
- Arborescence d'implémentation plus simple pour les intégrateurs.

- Nouveaux dashboards Grafana permettant de superviser les composants Springboot ainsi que l'API Gateway (CPU, RAM, nombre de requêtes traitées, etc.).

Mise à jour des COTS et librairies

- Montée de version CAS V6.6.12 vers V7.0.10.1

Tests de non-régression

Améliorations :

- Réécriture du moteur des tests end-to-end avec Playwright :
  - Amélioration du temps d'exécution (40 min -> 7 min).
  - Ajout de traces permettant de faciliter le débuggage a posteriori un cas d'erreur.
  - À déjà permis de détecter et corriger des bugs VitamUI (en raison notamment de la vitesse d'exécution).

v9.1

Ajout de fonctionnalités

- Possibilité d'ajouter des contrôles sur les rattachements dans les profils d'archivage (bloc UpdateOperation) et de générer un profil d'archivage incluant ces nouveaux contrôles.