



Modèle de données

Date	Version
26/10/2020	8.0 (Release 15)

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	26/10/2020
Validation	AGR	Équipe Vitam	26/10/2020

Suivi des modifications

Version	Date	Auteur	Modifications
1.0	09/01/2019		Génération à partir de l'ancien document en RST
1.1	28/01/2019	MAF, NMO	Correction
1.2	29/01/2019	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 8 (« collection FileFormat ») : corrections des cardinalités et précisions sur les champs pouvant être vides et sur le fonctionnement du champ « CreatedDate ». Chapitre 10 (« collection Griffin ») : correction du fichier JSON mis en exemple, avec suppression du champ LastUpdate qui est peuplé par la solution logicielle Vitam ; précisions sur le champ CreatedDate (peuplé par la solution logicielle Vitam, s'il n'est pas renseigné dans le fichier JSON) ; corrections à propos du champ Identifier qui ne fonctionne qu'en mode VITAM esclave ; le champ ExecutableVersion n'est pas un entier. Chapitre 13 (« collection PreservationScenario ») : correction du fichier JSON mis en exemple, avec suppression du champ LastUpdate qui est peuplé par la solution logicielle Vitam ; précisions sur le champ CreatedDate (peuplé par la solution logicielle Vitam, s'il n'est pas renseigné dans le fichier JSON) ; corrections à propos du champ Identifier qui ne fonctionne qu'en mode VITAM esclave ; corrections de cardinalités : GriffinByFormat est obligatoire, DefaultGriffin facultatif ; ajout de cardinalités et d'un type pour les champs : FormatList, GriffinIdentifier, Timeout, MaxSize, ActionDetail, Type, Values, Args, Extension ; ajout du champ Debug.
2.2	15/04/2019	MAF	Relecture
2.3	24/04/2019	JPP	Relecture
3.0	25/04/2019	MRE	Finalisation pour publication Release 10
3.1	14/06/2019	NMO	Mise à jour pour tenir compte des fonctionnalités mises en

			œuvre pendant la <i>Release 11</i> : <ul style="list-style-type: none"> Chapitre 3 (Base Logbook, collection LogbookOperation), modification de la section relative à l'obIdIn dans le détail des champs du JSON stocké dans la collection. Chapitre 5 (Base MasterData, collection IngestContrat), modification de la section relative aux champs LinkDeclaredInManifest, LinkParentId et CheckParentId du contrat d'entrée.
3.2	29/07/2019	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 4 (Base Metadata, collection Unit) : ajout de cardinalités et de précisions. Chapitre 4 (Base Metadata, collection ObjectGroup) : ajout de cardinalités et de précisions.
4.0	09/09/2019	MAF	Finalisation pour publication Release 11
4.1	10/11/2019	MVI	Mise à jour pour tenir compte des fonctionnalités mises en œuvre pendant la <i>Release 12</i> : <ul style="list-style-type: none"> Chapitre 4 (Base MetaData, collection Unit) : ajout du champ _opts. Chapitre 5 (Base MasterData, collection ManagementContract) : création de la section relative aux contrats de gestion. Chapitre 5 (Base MasterData, collection IngestContract) : ajout des champs ManagementContractId et ComputedInheritedRules, RuleCategoryToFilter.
5.0	29/11/2019	AGR	Finalisation pour publication Release 12
5.1	03/02/2020	MVI	Mise à jour pour tenir compte des fonctionnalités mises en œuvre pendant la <i>Release 13</i> : <ul style="list-style-type: none"> Chapitre 4 (Base MetaData, collection Unit) : suppression des champs _nbc et _offerId ; Chapitre 4 (Base MetaData, collection ObjectGroup) : suppression des champs _nbc et _offerId ; Chapitre 4 (Base MetaData, collection Offset) : ajout du champ strategyId ; Chapitre 5 (Base MasterData, collection Offset) : ajout de la section.
6.0	20/03/2020	AGR	Finalisation pour publication Release 13
6.1	28/05/2020	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 5 (Base MasterData, collection Offset) : ajout du champ ComputedInheritedRulesAtIngest dans les contrats d'entrée, de la valeur « EXTRACT_AU » dans les scénarios de préservation.
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

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

Modèle de données.....	1
1. Introduction.....	8
1.1. Avertissement.....	8
1.2. Objectif du document.....	8
1.3. Création des index.....	8
1.4. Généralités.....	8
1.4.1. Collections et bases.....	8
1.4.2. Cardinalité.....	8
1.4.3. Nommage des champs.....	9
1.4.4. Identifiants.....	9
1.4.5. Dates.....	9
1.4.6. Limite de caractères acceptés dans les champs.....	9
1.4.7. Type d'indexation dans Elasticsearch.....	9
2. Base Identity.....	10
2.1. Collection Certificate.....	10
2.1.1. Utilisation de la collection Certificate.....	10
2.1.2. Exemple de JSON stocké en base comprenant l'exhaustivité des champs.....	10
2.1.3. Détail des champs du JSON stocké dans la collection.....	10
2.2. Collection PersonalCertificate.....	11
2.2.1. Utilisation de la collection PersonalCertificate.....	11
2.2.2. Exemple de JSON stocké en base comprenant l'exhaustivité des champs.....	11
2.2.3. Détail des champs du JSON stocké dans la collection.....	11
3. Base Logbook.....	13
3.1. Collection LogbookOperation.....	13
3.1.1. Utilisation de la collection LogbookOperation.....	13
3.1.2. Exemple de JSON stocké en base comprenant l'exhaustivité des champs de la collection LogbookOperation.....	13
3.1.3. Détail des champs du JSON stocké dans la collection.....	15
3.1.4. Champs présents dans les events.....	21
3.1.5. Détail des champs du JSON stocké en base spécifiques à une opération de sécurisation des journaux d'opération et de cycle de vie.....	21
3.2. Collection LogbookLifeCycleUnit.....	24
3.2.1. Utilisation de la collection LogbookLifeCycleUnit.....	24
3.2.2. Exemple de JSON stocké en base comprenant l'exhaustivité des champs de la collection LogbookLifeCycleUnit.....	25
3.2.3. Détail des champs du JSON stocké en base.....	25
3.2.4. Champs présents dans les events.....	28
3.2.5. Détail des champs du JSON stocké en base spécifiques à une mise à jour.....	29
3.3. Collection LogbookLifeCycleObjectGroup.....	29
3.3.1. Utilisation de la collection LogbookLifeCycleObjectGroup.....	29
3.3.2. Extrait d'un JSON stocké en base comprenant l'exhaustivité des champs.....	29
3.3.3. Détail des champs du JSON stocké en base.....	30
3.3.4. Champs présents dans les events.....	34
3.4. Collection Offset.....	34
3.4.1. Utilisation de la collection.....	34
3.4.2. Exemple de JSON stocké en base comprenant l'exhaustivité des champs.....	34
3.4.3. Détail des champs.....	34
4. Base MetaData.....	36

4.1. Collection Unit.....	36
4.1.1. Utilisation de la collection Unit.....	36
4.1.2. Exemple de XML en entrée.....	36
4.1.3. Exemple de JSON stocké dans la collection Unit.....	36
4.1.4. Détail du JSON.....	38
4.2. Collection ObjectGroup.....	56
4.2.1. Utilisation de la collection ObjectGroup.....	56
4.2.2. Exemple de XML.....	56
4.2.3. Exemple de JSON stocké en base.....	56
4.2.4. Détail des champs du JSON.....	58
4.3. Collection Offset.....	65
4.3.1. Utilisation de la collection.....	65
4.3.2. Exemple de JSON stocké en base comprenant l'exhaustivité des champs.....	65
4.3.3. Détail des champs.....	65
5. Base MasterData.....	66
5.1. Collection AccessContract.....	66
5.1.1. Utilisation de la collection AccessContract.....	66
5.1.2. Exemple d'un fichier d'import de contrat d'accès.....	66
5.1.3. Exemple de JSON stocké en base comprenant l'exhaustivité des champs de la collection AccesContract.....	67
5.1.4. Détail des champs.....	68
5.2. Collection AccessionRegisterDetail.....	70
5.2.1. Utilisation de la collection AccessionRegisterDetail.....	70
5.2.2. Exemple de la description dans le XML d'entrée.....	70
5.2.3. Exemple de JSON stocké en base comprenant l'exhaustivité des champs.....	71
5.2.4. Détail des champs.....	72
5.3. Collection AccessionRegisterSummary.....	76
5.3.1. Utilisation de la collection.....	76
5.3.2. Exemple de JSON stocké en base comprenant l'exhaustivité des champs.....	76
5.3.3. Détail des champs.....	76
5.4. Collection AccessionRegisterSymbolic.....	78
5.4.1. Utilisation de la collection.....	78
5.4.2. Détail des champs.....	78
5.5. Collection ArchiveUnitProfile.....	79
5.5.1. Utilisation de la collection.....	79
5.5.2. Exemple d'un fichier d'import de profils d'unité archivistique.....	80
5.5.3. Exemple de JSON stocké en base comprenant l'exhaustivité des champs de la collection ArchiveUnitProfile.....	80
5.5.4. Détail des champs de la collection ArchiveUnitProfile.....	80
5.6. Collection Agencies.....	82
5.6.1. Utilisation de la collection Agencies.....	82
5.6.2. Exemple de JSON stocké en base comprenant l'exhaustivité des champs de la collection Agencies.....	82
5.6.3. Détail des champs.....	83
5.7. Collection Context.....	83
5.7.1. Utilisation de la collection.....	83
5.7.2. Exemple d'un fichier d'import de contexte applicatif.....	84
5.7.3. Exemple de JSON stocké en base comprenant l'exhaustivité des champs de la collection Context.....	84
5.7.4. Détail des champs.....	85
5.8. Collection FileFormat.....	87
5.8.1. Utilisation de la collection FileFormat.....	87
5.8.2. Exemple de la description d'un format dans le XML d'entrée.....	87

5.8.3. Exemple de JSON stocké en base comprenant l'exhaustivité des champs de la collection FileFormat.....	87
5.8.4. Détail des champs du JSON stocké en base.....	88
5.9. Collection FileRules.....	90
5.9.1. Utilisation de la collection FileRules.....	90
5.9.2. Exemple de JSON stocké en base comprenant l'exhaustivité des champs de la collection FileRules.....	91
5.9.3. Détail des champs.....	91
5.10. Collection Griffin.....	93
5.10.1. Utilisation de la collection Griffin.....	93
5.10.2. Exemple d'un fichier d'import de griffon.....	93
5.10.3. Exemple de JSON stocké en base comprenant l'exhaustivité des champs de la collection Agencies.....	93
5.10.4. Détail des champs.....	94
5.11. Collection IngestContract.....	95
5.11.1. Utilisation de la collection.....	95
5.11.2. Exemple d'un fichier d'import de contrat.....	95
5.11.3. Exemple de JSON stocké en base comprenant l'exhaustivité des champs de la collection IngestContract.....	95
5.11.4. Détail des champs de la collection IngestContract.....	96
5.12. Collection ManagementContract.....	99
5.12.1. Utilisation de la collection ManagementContract.....	99
5.12.2. Exemple d'un fichier d'import de contrat de gestion.....	99
5.12.3. Exemple de JSON stocké en base comprenant l'exhaustivité des champs de la collection ManagementContract.....	100
5.12.4. Détail des champs.....	100
5.13. Collection Ontology.....	102
5.13.1. Utilisation de la collection.....	102
5.13.2. Exemple d'un fichier d'import d'ontology.....	102
5.13.3. Détail des champs de la collection Ontology.....	102
5.14. Collection PreservationScenario.....	104
5.14.1. Utilisation de la collection PreservationScenario.....	104
5.14.2. Exemple d'un fichier d'import de scénario de préservation.....	104
5.14.3. Exemple de JSON stocké en base comprenant l'exhaustivité des champs de la collection PreservationScenario.....	105
5.14.4. Détail des champs.....	106
5.15. Collection Profile.....	108
5.15.1. Utilisation de la collection Profile.....	108
5.15.2. Exemple d'un fichier d'import de notices de Profils d'archivage.....	108
5.15.3. Exemple de JSON stocké en base comprenant l'exhaustivité des champs de la collection Profile.....	109
5.15.4. Détail des champs.....	109
5.16. Collection SecurityProfile.....	111
5.16.1. Utilisation de la collection.....	111
5.16.2. Exemple de JSON stocké en base comprenant l'exhaustivité des champs.....	111
5.16.3. Détail des champs.....	111
5.17. Collection VitamSequence.....	112
5.17.1. Utilisation de la collection.....	113
5.17.2. Exemple de JSON stocké en base comprenant l'exhaustivité des champs.....	113
5.17.3. Détail des champs.....	113
5.18. Collection Offset.....	115
5.18.1. Utilisation de la collection.....	115
5.18.2. Exemple de JSON stocké en base comprenant l'exhaustivité des champs.....	115

5.18.3. Détail des champs.....	115
6. Base Report.....	116
6.1. Collection AuditObjectGroup.....	116
6.1.1. Utilisation de la collection.....	116
6.1.2. Détail des champs.....	116
6.2. Collection EliminationActionUnit.....	117
6.2.1. Utilisation de la collection.....	117
6.2.2. Détail des champs.....	117
6.3. Collection EliminationActionObjectGroup.....	118
6.3.1. Utilisation de la collection.....	118
6.3.2. Détail des champs.....	118
6.4. Collection PreservationReport.....	118
6.4.1. Utilisation de la collection.....	119
6.4.2. Détail des champs.....	119
7. Annexes.....	121
7.1. Valeurs possibles pour le champ evType du LogBook Operation.....	121
7.2. Valeurs possibles pour le champ evType du LogBook LifeCycle.....	121
7.3. Valeurs possibles pour le champ evTypeProc (type de processus).....	121
7.4. Catégories de règles possibles.....	122
7.5. Valeurs possibles pour le champ Status de la collection AccessionRegisterDetail.....	122
7.6. Valeurs possibles pour le champ Name de la collection VitamSequence.....	122
7.7. Type d’indexation des chaînes de caractères dans ElasticSearch par collection et par champ	124
7.8. Correspondances des champs spéciaux dans Vitam.....	139

1. Introduction

1.1. Avertissement

Ce document fait état du travail en cours. Il est susceptible de changer de manière conséquente au fur et à mesure de l'avancée des développements.

1.2. Objectif du document

Ce document a pour objectif de présenter la structure générale des collections utilisées dans la solution logicielle Vitam. Il est destiné principalement aux développeurs, afin de leur présenter l'organisation des données dans la solution logicielle Vitam, ainsi qu'à tous les autres acteurs du programme pour leur permettre de connaître ce qui existe en l'état actuel.

Il explicite chaque champ, précise la relation avec les sources (par exemple bordereau de transfert conforme au standard SEDA v.2.1, référentiel Pronom, etc.) et la structuration JSON stockée dans la base de données MongoDB. Ce document est structuré de façon à suivre l'ordre des bases et collections dans Mongo.

Pour chacun des champs, cette documentation apporte :

- une liste des valeurs licites,
- la sémantique ou syntaxe du champ,
- la codification en JSON.

Il décrit aussi parfois une utilisation particulière faite à une itération donnée. Cette indication différant de la cible finale, le numéro de l'itération de cet usage est mentionné.

1.3. Création des index

Les différents index sont créés par ansible, plate-forme logicielle libre. Les fichiers à renseigner pour rajouter un nouvel index sont stockés dans le répertoire `deployment/ansible-vitam/roles/mongo_configure/templates/init-{nom-base}-database.js.j2`

1.4. Généralités

1.4.1. Collections et bases

Les bases Mongo sont organisées en bases et collections.

Les bases contiennent différentes collections. Les collections peuvent être rapprochées du concept de tables en SQL.

1.4.2. Cardinalité

La cardinalité présentée pour chacun des champs correspond aux exigences de la base de données Mongo.

Certains champs ayant une cardinalité 1-1 sont directement renseignés par la solution logicielle Vitam et sont donc obligatoirement présents dans la base de données, mais ne le sont pas forcément dans les données envoyées.

1.4.3. Nommage des champs

Les champs des fichiers JSON présents dans les collections peuvent être nommés de deux manières :

- « champ » : un champ sans underscore est modifiable via les API,
- « _champ » : un champ commençant par un underscore n'est pas modifiable via les API. Une fois renseigné dans la solution logicielle Vitam par le bordereau de transfert ou la solution logicielle Vitam, il ne pourra plus être modifié depuis l'extérieur.

1.4.4. Identifiants

Il existe plusieurs types d'identifiants :

- GUID : identifiant unique de 36 caractères généré par la solution logicielle Vitam,
- PUID : identifiant des formats dans le référentiel Pronom,
- PID : identifiant de processus Unix.

1.4.5. Dates

Toutes les dates décrites dans ce document sont au format ISO 8601.

Exemple : "2017-11-02T13:50:28.922".

1.4.6. Limite de caractères acceptés dans les champs

Mongo est un type de base de données dite « schemaless », soit sans schéma. Ainsi, les champs contenus dans les collections décrites dans ce document sont, sauf mention contraire, sans limite de caractères.

1.4.7. Type d'indexation dans Elasticsearch

Les champs peuvent être indexés de deux façons différentes dans Elasticsearch :

- **les champs analysés** : les informations contenues dans ces champs peuvent être retrouvées par une recherche full-text. Par exemple, les champs *Description*, *Name*.
- **les champs non analysés** : les informations contenues dans ces champs peuvent être retrouvées par une recherche exacte uniquement. Par exemple, les champs *Identifier* ou *OriginatingAgency*.

2. Base Identity

La base Identity contient les collections relatives aux certificats applicatifs et personnels utilisés par la solution logicielle Vitam.

2.1. Collection Certificate

2.1.1. Utilisation de la collection Certificate

La collection Certificate permet de référencer et décrire unitairement les certificats utilisés par les contextes applicatifs.

2.1.2. Exemple de JSON stocké en base comprenant l'exhaustivité des champs

```
{
  "_id": "aeaaaaaaaaahkcaqqa4v4alj7kxofsqaaaaq",
  "SubjectDN": "CN=ihm-demo, O=vitam, L=paris, ST=idf, C=fr",
  "ContextId": "CT-000001",
  "SerialNumber": 302,
  "Certificate":
  "Q2VydGhmaWNhdGU6CiAgICBEYXRhOgogICAgICAgIFZlcnNpb246IDMgKDB4MikKICA
  [...]
  kbE4KM08yV1dIRlJMWnpQRWZ4eXlxMm1TbVdsaUUvUzZUbzJVVEswamxobStpbThPa29mZmlLbXlodVpWS3
  S0tRU5EIEIENFU1RJRk1DQVRFLS0tLS0=",
  "IssuerDN": "CN=ca_intermediate_client-external, OU=authorities, O=vitam,
  L=paris, ST=idf, C=fr",
  "Status": "VALID"
}
```

2.1.3. Détail des champs du JSON stocké dans la collection

« **_id** » : identifiant unique du certificat applicatif.

- Il s'agit d'une chaîne de 36 caractères correspondant à un GUID.
- Champ peuplé par la solution logicielle Vitam.
- Cardinalité : 1-1

« **SubjectDN** » : identifiant unique (Distinguished Name) du certificat applicatif.

- Il s'agit d'une chaîne de caractères
- Cardinalité : 1-1

« **ContextId** » : identifiant signifiant (Identifier) du contexte utilisant le certificat applicatif.

- Il s'agit d'une chaîne de caractères correspondant à l'identifiant signifiant d'un contexte applicatif.
- Cardinalité : 1-1

« **SerialNumber** » : numéro de série du certificat applicatif.

- Il s'agit d'un entier.
- Cardinalité : 1-1

« **Certificate** » : certificat.

- Il s'agit d'une chaîne de caractères correspondant au binaire du certificat suivant la norme X509.
- Cardinalité : 1-1

« **IssuerDN** » : identifiant unique (Distinguished Name) de l'autorité de certification.

- Il s'agit d'une chaîne de caractères.
- Cardinalité : 1-1

« **Status** » : statut du certificat.

- Peut être « VALID » si le certificat est valide, ou « REVOKED » s'il est révoqué.
- Cardinalité : 1-1

2.2. Collection PersonalCertificate

2.2.1. Utilisation de la collection PersonalCertificate

La collection PersonalCertificate permet de référencer et décrire unitairement les certificats personnels utilisés pour l'authentification de personae.

2.2.2. Exemple de JSON stocké en base comprenant l'exhaustivité des champs

```
{
  "_id": "aeaaaaaaaaahkcaqqaabxealjtiuxsziaaaaq",
  "SubjectDN": "O=VITAM, L=Paris, C=FR",
  "SerialNumber": 2,
  "Certificate":
  "MIIFRjCCAy6gAwIBAgIBAJANBgkqhkiG9w0BAQsFADAtMQswCQYDVQQGEwJGU[...]jzODUpSkBvDiaA==",
  "IssuerDN": "O=VITAM, L=Paris, C=FR",
  "Status": "VALID",
  "Hash": "6088f19bc7d328f301168c064d6fda93a6c4ced9d5c56810c4f70e21e77d841d"
}
```

2.2.3. Détail des champs du JSON stocké dans la collection

« **_id** » : identifiant unique du certificat personnel.

- Il s'agit d'une chaîne de 36 caractères correspondant à un GUID.
- Champ peuplé par la solution logicielle Vitam.
- Cardinalité : 1-1

« **SubjectDN** » : identifiant unique (Distinguished Name) du certificat personnel.

- Il s'agit d'une chaîne de caractères.
- Cardinalité : 1-1

« **SerialNumber** » : numéro de série du certificat.

- Il s'agit d'un entier.
- Cardinalité : 1-1

« **Certificate** » : certificat.

- Le certificat est au format DER encodé en Base64.
- Il s'agit d'une chaîne de caractères.
- Cardinalité : 1-1

« **IssuerDN** » : identifiant unique (Distinguished Name) de l'autorité de certification.

- Il s'agit d'une chaîne de caractères.
- Cardinalité : 1-1

« **Status** » : statut du certificat.

- Peut être « VALID » si le certificat est valide, ou « REVOKED » s'il est révoqué.
- Cardinalité : 1-1

« **Hash** » : empreinte du certificat.

- le hash utilise l'algorithme SHA256.
- Il s'agit d'une chaîne de caractères.
- Cardinalité : 1-1

3. Base Logbook

La base Logbook contient les collections relatives aux journaux d'opérations et de cycles de vie des unités archivistiques et des groupes d'objets de la solution logicielle Vitam. Une collection technique Offset liée à la reconstruction existe également.

L'ensemble des champs est peuplé automatiquement par la solution logicielle Vitam.

3.1. Collection LogbookOperation

3.1.1. Utilisation de la collection LogbookOperation

La collection LogbookOperation comporte toutes les informations de traitement liées aux opérations effectuées dans la solution logicielle Vitam, chaque opération faisant l'objet d'un enregistrement distinct.

Ces opérations sont :

- Audit
- Données de base
- Élimination
- Entrée
- Export DIP
- Mise à jour des unités archivistiques
- Préservation
- Sécurisation
- Vérification
- Sauvegarde des écritures
- Réorganisation d'arborescence
- Journalisation externe (enregistrement d'opérations extérieures dans la solution logicielle Vitam)

Les valeurs correspondant à ces opérations dans les journaux sont détaillées dans l'annexe 6.3.

3.1.2. Exemple de JSON stocké en base comprenant l'exhaustivité des champs de la collection LogbookOperation

Extrait d'un enregistrement JSON correspondant à une opération d'entrée terminée avec succès.

```
{
  "_id": "aeaaaaabchgzebuaafzaalj4nng5paaaaaq",
  "evId": "aeaaaaabchgzebuaafzaalj4nng5paaaaaq",
  "evParentId": null,
  "evType": "PROCESS_SIP_UNITARY",
  "evDateTime": "2019-04-03T13:19:08.671",
  "evDetData": "{\n  \"EvDetailReq\" : \"2 images de
lac\", \n  \"EvDateTimeReq\" : \"2016-10-
18T14:52:27\", \n  \"ArchivalAgreement\" : \"IC-00001\", \n  \"ServiceLevel\" :
null\n}",
  "evIdProc": "aeaaaaabchgzebuaafzaalj4nng5paaaaaq",
  "evTypeProc": "INGEST",
  "outcome": "STARTED",
}
```

```

    "outDetail": "PROCESS_SIP_UNITARY.STARTED",
    "outMessg": "Début du processus d'entrée du SIP :
aeaaaaaabchgzebuaafzaalj4nng5paaaaaq",
    "agId": "{ \"Name\": \"vitam-env-int-external-01.vitam-env\", \"Role\": \"ingest-
external\", \"ServerId\": 1047302196, \"SiteId\": 1, \"GlobalPlatformId\": 241995828 }",
    "agIdApp": "CT-000001",
    "agIdPers": null,
    "evIdAppSession": "MyApplicationId-ChangeIt",
    "evIdReq": "aeaaaaaabchgzebuaafzaalj4nng5paaaaaq",
    "agIdExt":
"{ \"originatingAgency\": \"Identif4\", \"TransferringAgency\": \"Identif5\", \"Ar
chivalAgency\": \"Identif4\" }",
    "rightsStatementIdentifier": "{ \"ArchivalAgreement\": \"IC-000001\" }",
    "obId": "aeaaaaaabchgzebuaafzaalj4nng5paaaaaq",
    "obIdReq": null,
    "obIdIn": "2 images de lac",
    "events": [
        {
            "evId": "aedqaaaaabchgzebuaafzaalj4nng67yaaaaaq",
            "evParentId": null,
            "evType": "STP_SANITY_CHECK_SIP.STARTED",
            "evDateTime": "2019-04-03T13:19:08.671",
            "evDetData": null,
            "evIdProc": "aeaaaaaabchgzebuaafzaalj4nng5paaaaaq",
            "evTypeProc": "INGEST",
            "outcome": "OK",
            "outDetail": "STP_SANITY_CHECK_SIP.STARTED.OK",
            "outMessg": "Succès du début du processus des contrôles préalables à
l'entrée",
            "agId": "{ \"Name\": \"vitam-env-int-external-01.vitam-
env\", \"Role\": \"ingest-
external\", \"ServerId\": 1047302196, \"SiteId\": 1, \"GlobalPlatformId\": 241995828 }",
            "agIdPers": null,
            "evIdReq": "aeaaaaaabchgzebuaafzaalj4nng5paaaaaq",
            "obId": "aeaaaaaabchgzebuaafzaalj4nng5paaaaaq"
        },
        {
            "evId": "aedqaaaaachfbdnsab3bmalecitge5iaaaaaaq",
            "evParentId": null,
            "evType": "STP_SANITY_CHECK_SIP",
            "evDateTime": "2018-06-18T09:07:42.879",
            "evDetData": null,
            "evIdProc": "aeaaaaaaachfbdnsab3bmalecitgbwqaaaaaq",
            "evTypeProc": "INGEST",
            "outcome": "OK",
            "outDetail": "STP_SANITY_CHECK_SIP.OK",
            "outMessg": "Succès du processus des contrôles préalables à l'entrée",
            "agId": "{ \"Name\": \"vitam-env-itrec-external-01.vitam-
env\", \"Role\": \"ingest-
external\", \"ServerId\": 1045466546, \"SiteId\": 1, \"GlobalPlatformId\": 240160178 }",
            "agIdPers": null,
            "evIdReq": "aeaaaaaaachfbdnsab3bmalecitgbwqaaaaaq",
            "obId": "aeaaaaaaachfbdnsab3bmalecitgbwqaaaaaq"
        },
        {
            "evId": "aedqaaaaachfbdnsab3bmalecitge5iaaaba",
            "evParentId": "aedqaaaaachfbdnsab3bmalecitge5iaaaaaaq",
            "evType": "SANITY_CHECK_SIP",
            "evDateTime": "2018-06-18T09:07:42.879",
            "evDetData": null,
            "evIdProc": "aeaaaaaaachfbdnsab3bmalecitgbwqaaaaaq",
            "evTypeProc": "INGEST",
            "outcome": "OK",
            "outDetail": "SANITY_CHECK_SIP.OK",
            "outMessg": "Succès du contrôle sanitaire du SIP : aucun virus
détecté",
            "agId": "{ \"Name\": \"vitam-env-itrec-external-01.vitam-

```

```

env\","Role\":"ingest-
external\","ServerId\":"1045466546","SiteId\":"1","GlobalPlatformId\":"240160178}",
  "agIdPers": null,
  "evIdReq": "aeaaaaaaaaachfbdnsab3bmalecitgbwqaaaaaq",
  "obId": "aeaaaaaaaaachfbdnsab3bmalecitgbwqaaaaaq"
},
{
  [...]
}
],
  "_tenant": 8,
  "_v": 25,
  "_lastPersistedDate": "2019-04-03T13:19:28.832"
}
    
```

3.1.3. Détail des champs du JSON stocké dans la collection

Chaque enregistrement de cette collection est composé d'une structure auto-imbriquée : la structure possède une première instanciation « incluante » et contient un tableau de n structures identiques, dont seules les valeurs contenues dans les champs changent.

La structure est décrite ci-dessous. Pour certains champs, on indiquera s'il s'agit de la structure incluante ou d'une structure incluse dans celle-ci.

« **_id** » (**identifiant**): identifiant unique donné par le système lors de l'initialisation de l'opération.

- Il s'agit d'une chaîne de 36 caractères correspondant à un GUID.
- La valeur de ce champ peut être réutilisée dans les champs evIdProc, evIdReq, evId et obId pour pouvoir suivre une succession d'opérations déclenchées par une première opération (comme la mise à jour du référentiel des règles de gestion pouvant déclencher une mise à jour des unités archivistiques).
- Cet identifiant constitue la clé primaire de l'opération dans la collection.
- Cardinalité : 1-1
- Ce champ existe uniquement pour la structure incluante.

« **evId** » (**event Identifier**): identifiant de l'événement.

- Il s'agit d'une chaîne de 36 caractères.
- Champ obligatoire peuplé par la solution logicielle Vitam.
- Il identifie l'opération de manière unique dans la collection.
- Cet identifiant doit être l'identifiant d'un événement dans le cadre de l'opération (evIdProc) et doit donc être différent par paire (début/fin).
- Cardinalité : 1-1
- Ce champ existe pour les structures incluantes et incluses.

« **evParentId** » (**event Parent Identifier**): identifiant de l'événement parent.

- Il est constitué d'une chaîne de 36 caractères correspondant à un GUID.
- Il identifie l'événement parent. Par exemple pour le traitement CHECK_SEDA, il s'agit de l'identifiant de l'étape STP_INGEST_CONTROL_SIP.
- Ce champ est toujours à « null » pour la structure incluante et les tâches principales.
- Cardinalité : 1-1
- Ce champ existe pour les structures incluantes et incluses.

« **evType** » (**event Type**): code du type de l'opération.

- Issu de la définition du workflow structuré en JSON (fichier default-workflow.json).
- La liste des valeurs possibles pour ce champ se trouve en annexe. Seul le code est stocké dans ce champ, la traduction se faisant via un fichier properties (vitam-logbook-message-fr.properties).
- Cardinalité : 1-1
- Ce champ existe pour les structures incluant et incluses.

« **evDateTime** » (**event DateTime**): date de lancement de l'opération.

- Il s'agit d'une date au format ISO8601 AAAA-MM-JJ+ »T »+hh:mm:ss:[3 digits de millisecondes]
- Elle est renseignée par le client LogBook.

Exemple : "2016-08-17T08:26:04.227"

- Cardinalité : 1-1
- Ce champ existe pour les structures incluant et incluses.

« **evDetData** » (**event Detail Data**): détails des données l'événement.

- Donne plus de détails sur l'événement ou son résultat.
- Par exemple, pour l'étape ATR_NOTIFICATION, ce champ détaille le nom de l'ArchiveTransferReply, son empreinte et l'algorithme utilisé pour calculer l'empreinte.
- Sur la structure incluant d'une opération d'entrée, il contient un objet JSON composé des champs suivants :
 - EvDetailReq : précisions sur la demande de transfert.
 - Chaîne de caractères.
 - Reprend le champ « Comment » du message ArchiveTransfer.
 - EvDateTimeReq : date de la demande de transfert inscrit dans le champ evDetData.
 - Date au format ISO8601 AAAA-MM-JJ+ »T »+hh:mm:ss:[3 digits de millisecondes].
 - Cardinalité 1-1.
 - ArchivalAgreement : identifiant du contrat d'entrée utilisé.
 - Reprend le champ « ArchivalAgreement » du message ArchiveTransfer.
 - Cardinalité 1-1.
 - ArchiveProfile : identifiant du profil d'archivage utilisé.
 - Reprend le champ « ArchiveProfile » du message ArchiveTransfer.
 - Cardinalité 0-1.
 - ServiceLevel : niveau de service.
 - Chaîne de caractères.
 - Reprend le champ ServiceLevel du message ArchiveTransfer.
 - Cardinalité 0-1.
 - AcquisitionInformation : modalités d'entrée des archives.
 - Chaîne de caractères.
 - Reprend le champ AcquisitionInformation du message ArchiveTransfer.
 - Cardinalité 0-1.
 - LegalStatus : statut des archives échangées.
 - Chaîne de caractères.

- Reprend le champ LegalStatus du message ArchiveTransfer.
- Cardinalité 0-1.

- Cardinalité pour les structures incluant : 1-1
- Cardinalité pour les structures incluses : 0-1
- Ce champ existe pour les structures incluant et incluses.

« **evIdProc** » (**event Identifier Process**): identifiant du processus.

- Il s'agit d'une chaîne de 36 caractères.
- Lorsqu'il s'agit d'une opération indépendante d'autres opérations, tous les événements identiques d'un document dans cette collection reprennent pour ce champ la valeur du champ « _id ». Dans le cas où une opération en déclenche d'autres, les opérations déclenchées utilisent toutes le même evIdProc, qui permet alors de suivre une suite de processus.
- Cardinalité : 1-1
- Ce champ existe pour les structures incluant et incluses.

« **evTypeProc** » (**event Type Process**): type de processus.

- Il s'agit d'une chaîne de caractères.
- Nom du processus, parmi une liste de processus possibles fixée. Cette liste est disponible en annexe 6.3.
- Cardinalité : 1-1
- Ce champ existe pour les structures incluant et incluses.

« **outcome** » : statut de l'événement.

- Il s'agit d'une chaîne de caractères devant correspondre à une valeur de la liste suivante :
 - STARTED (Début de l'événement)
 - OK (Succès de l'événement)
 - KO (Échec de l'événement)
 - WARNING (Succès de l'événement comportant toutefois des alertes)
 - FATAL (Erreur technique)
- Cardinalité : 1-1
- Ce champ existe pour les structures incluant et incluses.

« **outDetail** » (**outcome Detail**): code correspondant au résultat de l'événement.

- Il s'agit d'une chaîne de caractères.
- Il contient le code correspondant au résultat de l'événement, incluant le statut. La liste des valeurs possibles pour ce champ se trouve en annexe. Seul le code doit être stocké dans ce champ, la traduction doit se faire via un fichier properties (vitam-logbook-message-fr.properties)
- Cardinalité : 1-1
- Ce champ existe pour les structures incluant et incluses.

« **outMessg** » (**outcome Detail Message**): détail du résultat de l'événement.

- Il s'agit d'une chaîne de caractères.
- C'est un message intelligible destiné à être lu par un être humain en tant que détail de l'événement.

- Correspond à la traduction du code présent dans outDetail, issue du fichier vitam-logbook-message-fr.properties.
- Cardinalité : 1-1
- Ce champ existe pour les structures incluant et incluses.

« **agId** » (**agent Identifier**): identifiant de l'agent interne réalisant l'évènement.

- Il s'agit de plusieurs chaînes de caractères indiquant le nom, le rôle et l'identifiant du serveur, du site et de la plateforme. Ce champ est calculé par le journal à partir de ServerIdentifier et en s'appuyant sur des fichiers de configuration.

```
Exemple : {"Name\":"vitam-env-itrec-external-01.vitam-env", "Role\":"ingest-external", "ServerId\":"1045466546, "SiteId\":"1, "GlobalPlatformId\":"240160178"},
```

- Cardinalité : 1-1
- Ce champ existe pour les structures incluant et incluses.

« **agIdApp** » (**agent Identifier Application**): identifiant de l'application externe qui appelle la solution logicielle Vitam pour effectuer une opération.

- Cet identifiant est celui du contexte applicatif utilisé par l'application.
- Il s'agit d'une chaîne de caractères.
- Cardinalité : 1-1
- Ce champ existe uniquement pour la structure incluant.

« **agIdPers** » : identifiant personae, issu du certificat personnel.

- Il s'agit d'une chaîne de caractères.
- Peut avoir une valeur null.
- Cardinalité : 1-1
- Ce champ existe pour les structures incluant et incluses.

« **evIdAppSession** » (**event Identifier Application Session**): identifiant de la transaction qui a entraîné le lancement d'une opération dans la solution logicielle Vitam.

- L'application externe est responsable de la gestion de cet identifiant. Il correspond à un identifiant pour une session donnée côté application externe.
- Il s'agit d'une chaîne de caractères.
- Cardinalité : 1-1
- Ce champ existe uniquement pour la structure incluant.

« **evIdReq** » (**event Identifier Request**): identifiant de la requête déclenchant l'opération.

- Il s'agit d'une chaîne de 36 caractères.
- Une requestId est créée pour chaque nouvelle requête http venant de l'extérieur.
- Dans le cas du processus d'entrée, il devrait s'agir du numéro de l'opération (EvIdProc).
- Il s'agit du X-Application-Id.
- Cardinalité : 1-1
- Ce champ existe pour les structures incluant et incluses.

« **agIdExt** » (**agent Identifier External**): identifiant de l'agent externe mentionné dans le message ArchiveTransfer.

- Pour une opération d'INGEST, il s'agit d'un objet JSON comprenant les champs suivants :
 - OriginatingAgency : identifiant du service producteur.
 - Il s'agit d'une chaîne de caractères.
 - Reprend le contenu du champ OriginatingAgencyIdentifieur du message ArchiveTransfer.
 - TransferringAgency : identifiant du service de transfert.
 - Il s'agit d'une chaîne de caractères.
 - Reprend le contenu du champ TransferringAgencyIdentifieur du message ArchiveTransfer.
 - ArchivalAgency : identifiant du service d'archivage.
 - Il s'agit d'une chaîne de caractères.
 - Reprend le contenu du champ ArchivalAgencyIdentifieur du message ArchiveTransfer.
 - SubmissionAgency : identifiant du service versant.
 - Il s'agit d'une chaîne de caractères.
 - Reprend le contenu du champ SubmissionAgencyIdentifieur du message ArchiveTransfer.
- Cardinalité : 1-1
- Ce champ existe uniquement pour la structure incluante.

« **rightsStatementIdentifieur** » : identifiant des données référentielles en vertu desquelles l'opération peut s'exécuter.

- Pour une opération d'INGEST, il s'agit d'un objet JSON comprenant les champs suivants :
 - ArchivalAgreement: identifiant du contrat d'entrée utilisé pour réaliser l'entrée.
 - Il s'agit d'une chaîne de caractères.
 - Reprend le contenu du champ ArchivalAgreement du message ArchiveTransfer.
 - Cardinalité 1-1.
 - Profil : identifiant du profil d'archivage utilisé pour réaliser l'entrée. Il s'agit d'une chaîne de caractères.
 - Reprend le contenu du champ ArchiveProfile du message ArchiveTransfer.
 - Cardinalité 0-1.
- Pour une opération d'UPDATE, il s'agit d'un objet JSON comprenant les champs suivants :
 - AccessContract : identifiant du contrat d'accès utilisé pour réaliser une mise à jour.
- Cardinalité : 1-1
- Ce champ existe pour la structure incluante et certaines structures incluses

« **obId** » (**object Identifieur**): identifiant du lot d'objets auquel s'applique l'opération (lot correspondant à une liste).

- Identifiant peuplé par la solution logicielle Vitam.
- Il s'agit d'une chaîne de 36 caractères.
- Dans le cas d'une opération d'entrée, il s'agit du GUID de l'entrée (evIdProc).
- Dans le cas d'une opération d'audit, il s'agit par exemple du nom d'un lot d'archives prédéfini.
- Dans le cas d'une opération de mise à jour, il s'agit du GUID de l'unité archivistique mise à

jour.

- Dans le cas d'une opération de données de base, il s'agit de l'identifiant de l'opération.
- Cardinalité pour les structures incluant : 1-1
- Cardinalité pour les structures incluses : 0-1
- Ce champ existe pour les structures incluant et incluses.

« **obIdReq** » (**object Identifier Request**): identifiant de la requête caractérisant un lot d'objets auquel s'applique l'opération.

- Identifiant peuplé par la solution logicielle Vitam.
- Ne concerne que les lots d'objets dynamiques, c'est-à-dire obtenus par la présente requête. Ne concerne pas les lots ayant un identifiant défini.
- Actuellement, la valeur est toujours "null".
- Cardinalité : 1-1
- Ce champ existe pour les structures incluant et incluses.

« **obIdIn** » (**Object Identifier Income**): identifiant externe du lot d'objets auquel s'applique l'opération, utilisé pour les opérations d'entrée.

- Il s'agit d'une chaîne de caractères intelligible pour un humain qui permet de comprendre à quel SIP ou quel lot d'archives se rapporte l'événement.
- La structure incluant et les structures incluses reprennent le contenu du champ MessageIdentifier du message ArchiveTransfer.
- Cardinalité pour les structures incluant : 1-1
- Cardinalité pour les structures incluses : 0-1
- Ce champ existe pour les structures incluant et incluses.

« **events** » : tableau de structures listant des événements liés à l'opération.

- Pour la structure incluant, le tableau contient n structures incluses dans l'ordre des événements (date)
- Cardinalité : 1-1
- S'agissant d'un tableau, les structures incluses ont pour cardinalité 1-n.
- Ce champ existe uniquement pour la structure incluant.

« **_tenant** » : identifiant du tenant.

- Il s'agit d'un entier.
- Cardinalité : 1-1
- Ce champ existe uniquement pour la structure incluant.

« **_v** » : version de l'enregistrement décrit

- Il s'agit d'un entier.
- 0 correspond à l'enregistrement d'origine. Si le numéro est supérieur à 0, alors il s'agit du numéro de version de l'enregistrement.
- Cardinalité : 1-1
- Ce champ existe uniquement pour la structure incluant.

« **_lastPersistedDate** » : date technique de sauvegarde en base.

- Il s'agit d'une date au format ISO8601 AAAA-MM-JJ+ »T »+hh:mm:ss:[3 digits de millisecondes]

- Elle est renseignée par le serveur Logbook.
Exemple : "2016-08-17T08:26:04.227"
- Cardinalité : 1-1
- Ce champ existe uniquement pour la structure incluante.

3.1.4. Champs présents dans les events

Les events sont, au minimum, composés des champs suivants :

- evId
- evParentId
- evType
- evDateTime
- evDetData
- evIdProc
- evTypeProc
- outcome
- outDetail
- outMessg
- agId
- AgIdPers
- evIdReq
- obId

D'autres champs peuvent apparaître dans certains events lorsqu'ils mettent à jour le master.

3.1.5. Détail des champs du JSON stocké en base spécifiques à une opération de sécurisation des journaux d'opération et de cycle de vie

Ceci ne concerne aujourd'hui que les sécurisations des journaux d'opération et la sécurisation des journaux de cycle de vie.

Exemple de données stockées par l'opération de sécurisation des journaux d'opération pour le champ evDetData :

"evDetData":

```
{
  "LogType": "OPERATION",
  "StartDate": "2019-04-08T05:55:17.366",
  "EndDate": "2019-04-08T07:55:22.684",
  "Hash": "eeQxnrhZgOwpbCz9m5yT9Z9pQVzJlEJAbZcmGAOs/d1gwU67I69JmaJbkJF3sc0rBtlyWZyItQnaF0+AtENMGw==",
  "TimeStampToken": "MIILITAVAgEAMBAMdk9wZXJhdGlvbiBPa2F5MIILBgYJKoZIhvcNAQcCoIIK9zCCCvMCAQMxDzANBg1ghkgBZQMEAgMFADCBgAYLKoZIhvcNAQkQAQSGcQRvMG0CAQEGASkwUTANBg1ghkgBZQMEAgMFAARAYze0cUub27AyosyVkJ6gSqTmb0ki0AE/VMMjdOkBiHmXp4bt0YhB+7H7tiBxXs8AnCguuZXBNEndCl9xdrBC2QIBARgPMjAxOTA0MDgwODAwMjJaoIIGhzCCBoMwggRroAMCAQICAgDUMA0GCSqGSib3DQEBCwUAMHgxCzAJBgNVBAYTAmZyMQwwCgYDVQQIDANpZGYxDjAMBgNVBAcMBXBhcm1zMQ4wDAYDVQQKDAV2aXRhbTEUMBIGA1UECwwLYXV0aG9yaXRpZXMxJTAjBgNVBAMMHGNhX21udGVybWVkaWF0ZV90aW1lc3RhbXBpbmhcwHhcNMTkwNDA4MDEzNzE2WhcNMjIwNDA3MDEzNzE2WjBUMQswCQYDVQQGEWJmcjEMMAoGA1UECAwDaWRmMQ4wDAYDVQQHDAVwYXJpczEOMAwGA1UECgwFdm10YW0xYzAvBzAVBgNVBAMMDnN1Y3VyZS1sb2dib29rMIIC IjANBgkqhkiG9w0BAQEFAAOCAg8AMIICCgKCAgEA39VhcS+83NFInh4EeXagmh2VHNTN5yRiVT41xX5hNLBZzDwcu0PCPDRUzVLMcGWdLftoFr+J6t0h2cDBx9W3EeXoDX/0faPGw9HFuBX8+7KGZQ2U4Z9SCplT3qLxeCMVLZ9zj1Du0qTyqWuPTFyty1X4ZpQ4ZY6TwlLrhczMvZRKN35+cIgilE28jzyXTobHcZN7MtqXJqDxmZddQTEJw4ohl5I/VpCluDJw0/1dd5HGTvawN6BbcTmYPgvm841w097JXsdcoD90K5ACmOJX9n6bd/a9Bq+DoI
```


« **EndDate** » : date de fin de la période de couverture de l'opération de sécurisation.

- Il s'agit d'une date au format ISO8601 AAAA-MM-JJ+ »T »+hh:mm:ss:[3 digits de millisecondes] (correspond à la date de la dernière opération sécurisée par la précédente sécurisation).

Exemple : "2016-08-17T08:26:04.227"

- Cardinalité : 1-1

« **Hash** » : empreinte racine.

- Il s'agit d'une chaîne de caractères.
- Empreinte de la racine de l'arbre de Merkle.
- Cardinalité : 1-1

« **TimeStampToken** » : tampon d'horodatage.

- Il s'agit d'une chaîne de caractères.
- Tampon d'horodatage sûr du journal sécurisé.
- Cardinalité : 1-1

« **PreviousLogbookTraceabilityDate** » : date de la précédente opération de sécurisation de ce type de journal.

- Il s'agit de la date de début de la précédente opération de sécurisation du même type au format ISO8601 AAAA-MM-JJ+ »T »+hh:mm:ss:[3 digits de millisecondes] (correspond à la date de début de la sécurisation précédente).

Exemple : "2016-08-17T08:26:04.227"

- Cardinalité : 1-1

« **MinusOneMonthLogbookTraceabilityDate** » : date de l'opération de sécurisation passée d'un mois.

- Il s'agit de la date de début de la précédente opération de sécurisation du même type réalisée un mois avant au format ISO8601 AAAA-MM-JJ+ »T »+hh:mm:ss:[3 digits de millisecondes].

Exemple : "2016-08-17T08:26:04.227"

- Cardinalité : 1-1

« **MinusOneYearLogbookTraceabilityDate** » : date de l'opération de sécurisation passée d'un an.

- Il s'agit de la date de début de la précédente opération de sécurisation du même type réalisée un an avant au format ISO8601 AAAA-MM-JJ+ »T »+hh:mm:ss:[3 digits de millisecondes].

Exemple : "2016-08-17T08:26:04.227"

- Cardinalité : 1-1

« **NumberOfElement** » : nombre d'éléments.

- Il s'agit d'un entier.
- Nombre d'opérations sécurisées.
- Cardinalité : 1-1

« **FileName** » : identifiant du fichier.

- Il s'agit d'une chaîne de caractères.
- Nom du fichier sécurisé sur les offres de stockage au format

{tenant}_LogbookOperation_{AAAAMMJJ_HHMMSS}.zip.

Exemple : "0_LogbookOperation_20170127_141136.zip"

- Cardinalité : 1-1

« **Size** » : taille du fichier.

- Il s'agit d'un entier.
- Taille du fichier sécurisé (en octets).
- Cardinalité : 1-1

« **SecurisationVersion** » : version de l'algorithme de sécurisation.

- Il s'agit d'une chaîne de caractères.
- La version est une valeur fixe (v1, v2...)
- Cardinalité : 1-1

« **DigestAlgorithm** » : algorithme de hachage.

- Il s'agit d'une chaîne de caractères.
- Il s'agit du nom de l'algorithme de hachage utilisé pour réaliser le tampon d'horodatage.
- Cardinalité : 1-1

« **MaxEntriesReached** » : permet de savoir si l'ensemble de la sécurisation est terminée.

- Il s'agit d'un booléen « true » ou « false »
- Lorsqu'il y a un ensemble trop grand d'éléments à sécuriser, la solution logicielle Vitam divise le travail pour manipuler des ensembles de fichiers techniquement gérables. Par défaut le nombre maximum d'élément sécurisable dans un lot est de 100.000. Si ce nombre dépasse le seuil, alors la valeur de MaxEntriesReached est « true » et une deuxième sécurisation est lancée là où s'est arrêté la première, puis on regarde à nouveau si le seuil est atteint pour lancer un troisième lot, un quatrième, etc. Lorsque le nombre d'éléments sécurisés est inférieur au seuil, alors la valeur de MaxEntriesReached est « false ».
- Cardinalité : 1-1

3.2. Collection LogbookLifeCycleUnit

3.2.1. Utilisation de la collection LogbookLifeCycleUnit

Le journal du cycle de vie d'une unité archivistique (ArchiveUnit) trace tous les événements qui impactent celle-ci dès sa prise en charge dans le système. Il doit être conservé aussi longtemps que l'unité archivistique est gérée par le système.

- dès la réception d'une unité archivistique, l'ensemble des opérations qui lui sont appliquées est tracé.
- les journaux du cycle de vie sont « committés » une fois le stockage des objets et l'indexation des métadonnées effectués sans échec, avant l'envoi d'une notification au service versant.

Chaque unité archivistique possède une et une seule entrée dans la collection LogbookLifeCycleUnit.

3.2.2. Exemple de JSON stocké en base comprenant l'exhaustivité des champs de la collection LogbookLifeCycleUnit

Extrait d'un enregistrement JSON correspondant au journal de cycle de vie d'une unité archivistique.

```

{
  "_id": "aeaqaaaabahf4qxrab2nualjtkuyd6yaaabq",
  "evId": "aedqaaaabchf4qxrab2nualjtkuyfaaaaabq",
  "evParentId": null,
  "evType": "LFC.LFC_CREATION",
  "evDateTime": "2019-03-20T10:33:14.112",
  "evIdProc": "aeaaaaaabchgzebuaaeckaljtkuxtjqaaaaq",
  "evTypeProc": "INGEST",
  "outcome": "OK",
  "outDetail": "LFC.LFC_CREATION.OK",
  "outMessg": "Succès de l'alimentation du journal du cycle de vie",
  "agId": "{\"Name\":\"vitam-env-int-worker-01.vitam-
env\", \"Role\":\"worker\", \"ServerId\":1046364913, \"SiteId\":1, \"GlobalPlatformId\":
241058545}",
  "obId": "aeaqaaaabahf4qxrab2nualjtkuyd6yaaabq",
  "evDetData": null,
  "events": [
    {
      "evId": "aedqaaaabchf4qxrab2nualjtkuyfaaaaaca",
      "evParentId": null,
      "evType": "LFC.CHECK_MANIFEST",
      "evDateTime": "2019-03-20T10:33:14.112",
      "evIdProc": "aeaaaaaabchgzebuaaeckaljtkuxtjqaaaaq",
      "evTypeProc": "INGEST",
      "outcome": "OK",
      "outDetail": "LFC.CHECK_MANIFEST.OK",
      "outMessg": "Succès de la vérification de la cohérence du bordereau de
transfert",
      "agId": "{\"Name\":\"vitam-env-int-worker-01.vitam-
env\", \"Role\":\"worker\", \"ServerId\":1046364913, \"SiteId\":1, \"GlobalPlatformId\":
241058545}",
      "obId": "aeaqaaaabahf4qxrab2nualjtkuyd6yaaabq",
      "evDetData": "{ }",
      "_lastPersistedDate": "2019-03-20T10:33:29.151"
    },
    [...]
  ]
},
{
  "_tenant": 8,
  "_v": 10,
  "_lastPersistedDate": "2019-04-02T14:58:15.820"
}

```

3.2.3. Détail des champs du JSON stocké en base

« **_id** » : identifiant donné par le système lors de l'initialisation du journal du cycle de vie.

- Il est constitué d'une chaîne de 36 caractères correspondant à un GUID.
- Cet identifiant constitue la clé primaire du journal du cycle de vie de l'unité archivistique. Il reprend la valeur du champ `_id` d'une unité archivistique enregistré dans la collection Unit.
- Cardinalité : 1-1
- Ce champ existe uniquement pour la structure incluante.

« **evId** » (**event Identifier**): identifiant de l'événement.

- Il est constitué d'une chaîne de 36 caractères correspondant à un GUID.

- Il identifie l'événement de manière unique dans la base.
- Cardinalité : 1-1
- Ce champ existe pour les structures incluant et incluses.

« **evParentId** » (**event Parent Identifier**): identifiant de l'événement parent.

- Il est constitué d'une chaîne de 36 caractères correspondant à un GUID.
- Il identifie l'événement parent. Par exemple pour l'événement LFC.CHECK_MANIFEST.LFC_CREATION, ce champ fera référence au GUID de l'événement LFC.CHECK_MANIFEST.
- La valeur est toujours « null » pour la structure incluant et les tâches principales.
- Cardinalité : 1-1
- Ce champ existe pour les structures incluant et incluses.

« **evType** » (**event Type**): code du type d'événement.

- Il s'agit d'une chaîne de caractères.
- La liste des valeurs possibles pour ce champ se trouve en annexe. Seul le code est stocké dans ce champ, la traduction se fait via un fichier properties (vitam-logbook-message-fr.properties).
- Cardinalité : 1-1
- Ce champ existe pour les structures incluant et incluses.

« **evDateTime** » (**event DateTime**): date de l'événement.

- Il s'agit d'une date au format ISO8601 AAAA-MM-JJ+ »T »+hh:mm:ss:[3 digits de millisecondes]
- Exemple : "2016-08-17T08:26:04.227"
- Ce champ est positionné par le client LogBook.
- Cardinalité : 1-1
- Ce champ existe pour les structures incluant et incluses.

« **evIdProc** » (**event Identifier Process**): identifiant du processus.

- Il s'agit d'une chaîne de 36 caractères.
- Toutes les occurrences de ce champ pour un même document dans le journal du cycle de vie partagent la même valeur, qui est celle du champ « **_id** » d'une opération enregistrée dans la collection LogbookOperation.
- Cardinalité : 1-1
- Ce champ existe pour les structures incluant et incluses.

« **evTypeProc** » (**event Type Process**): type de processus.

- Il s'agit d'une chaîne de caractères.
- Nom du processus parmi une liste de processus possibles fixée. Cette liste est disponible en annexe.
- Cardinalité : 1-1
- Ce champ existe pour les structures incluant et incluses.

« **outcome** » : statut de l'événement.

- Il s'agit d'une chaîne de caractères devant correspondre à une valeur de la liste suivante :
 - STARTED (Début de l'événement)

- OK (Succès de l'événement)
- KO (Echec de l'événement)
- WARNING (Succès de l'événement comportant des alertes)
- FATAL (Erreur technique)
- Cardinalité : 1-1
- Ce champ existe pour les structures incluant et incluses.

« **outDetail** » (**outcome Detail**): code correspondant au résultat de l'événement.

- Il s'agit d'une chaîne de caractères.
- Il contient le code correspondant au résultat de l'événement, incluant le statut. La liste des valeurs possibles pour ce champ se trouve en annexe. Seul le code est stocké dans ce champ, la traduction se fait via le fichier properties (vitam-logbook-message-fr.properties).
- Cardinalité : 1-1
- Ce champ existe pour les structures incluant et incluses.

« **outMessg** » (**outcome Detail Message**): détail du résultat de l'événement.

- Il s'agit d'une chaîne de caractères.
- C'est un message intelligible destiné à être lu par un être humain en tant que détail de l'événement.
- Traduction du code présent dans outDetail issue du fichier vitam-logbook-message-fr.properties.
- Cardinalité : 1-1
- Ce champ existe pour les structures incluant et incluses.

« **agId** » (**agent Identifier**): identifiant de l'agent réalisant l'événement.

- Il s'agit d'un objet JSON, comprenant une suite de chaînes de caractères indiquant le nom, le rôle et le PID de l'agent. Ce champ est calculé par le journal à partir de ServerIdentifier.
Exemple : `"agId": "{ \"Name\": \"vitam-env-int-worker-01.vitam-env\", \"Role\": \"worker\", \"ServerId\": 1044139788, \"SiteId\": 1, \"GlobalPlatformId\": 238833420 }"`
- Cardinalité : 1-1
- Ce champ existe pour les structures incluant et incluses.

« **obId** » (**object Identifier**): identifiant de la solution logicielle Vitam correspondant au GUID de l'unité archivistique sur laquelle s'applique l'opération.

- Il s'agit d'une chaîne de 36 caractères correspondant à un GUID.
- Cardinalité : 1-1
- Ce champ existe pour les structures incluant et incluses

« **evDetData** » (**event Detail Data**): détails des données de l'événement.

- Il s'agit d'un objet JSON pouvant contenir des informations ou être vide.
- Donne plus de détails sur l'événement ou son résultat.
- Par exemple, l'historisation de métadonnées lors d'une modification se fait dans ce champ. Dans la structure incluse correspondant à cet événement, il est, par exemple, composé du champ suivant :
 - diff : contient la différence entre les métadonnées d'origine et les métadonnées

modifiées. Chaîne de caractères.

- Cardinalité : 1-1
- Ce champ existe pour les structures incluant et incluses.

« **events** » : tableau de structure listant les événements liés à une opération.

- Pour la structure incluant, le tableau contient n structures incluses dans l'ordre des événements (date).
- Cardinalité : 1-1
- S'agissant d'un tableau, les structures incluses ont pour cardinalité 1-n.
- Ce champ existe uniquement pour la structure incluant.

« **_tenant** » : identifiant du tenant.

- Il s'agit d'un entier.
- Cardinalité : 1-1
- Ce champ existe uniquement pour la structure incluant.

« **_v** » : version de l'enregistrement décrit.

- Il s'agit d'un entier.
- 0 correspond à l'enregistrement d'origine. Si le numéro est supérieur à 0, alors il s'agit du numéro de version de l'enregistrement.
- Cardinalité : 1-1
- Ce champ existe uniquement pour la structure incluant.

« **_lastPersistedDate** » : date technique de sauvegarde en base.

- Il s'agit d'une date au format ISO8601 AAAA-MM-JJ+ »T »+hh:mm:ss:[3 digits de millisecondes].
- Elle est renseignée par le serveur Logbook.

Exemple : "2016-08-17T08:26:04.227"

- Cardinalité : 1-1
- Ce champ existe pour les structures incluant et incluses.

3.2.4. Champs présents dans les events

- evId
- evParentId
- evType
- evDateTime
- evIdProc
- evTypeProc
- outcome
- outDetail
- outMessg
- agId
- obId
- evDetData

- lastPersistedDate

3.2.5. Détail des champs du JSON stocké en base spécifiques à une mise à jour

Exemple de données stockées dans le champ evDetData :

```
"evDetData": "{ \"diff\" : \"- Title : Steffi Graf\\n+ Title : Stefanie Maria Graf dite Steffi Graf\\n- StartDate : 2005-07-14T18:55:32\\n+ StartDate : 1988-06-13T22:00:00\\n- #operations : [ aeeaaaaabchgzebuabfkgalj7trrhdiiaaaq ]\\n+ #operations : [ aeeaaaaabchgzebuabfkgalj7trrhdiiaaaq, aeeaaaaabchgzebuaa4dealj7tskvfiaaaq ]\\n- #version : 0\\n+ #version : 1\" }",
```

Dans le cas d'une mise à jour de métadonnées d'une unité archivistique (ArchiveUnit), le champ « **evDetData** » de l'événement final est composé du champ suivant :

« **diff** » : historisation des modifications de métadonnées.

- Son contenu doit respecter la forme suivante : les anciennes valeurs sont précédées d'un « - » (-champ1 : valeur1) et les nouvelles valeurs sont précédées d'un « + » (+champ1 : valeur2). Le changement d'un champ entraîne forcément l'ajout d'une nouvelle opération (le champ _ops de l'unité est modifié) et d'une nouvelle version de l'unité (le champ _v est modifié). Ces changements apparaissent également dans le « diff ».

3.3. Collection LogbookLifeCycleObjectGroup

3.3.1. Utilisation de la collection LogbookLifeCycleObjectGroup

Le journal du cycle de vie du groupe d'objets (ObjectGroup) trace tous les événements qui impactent le groupe d'objets (et les objets associés) dès sa prise en charge dans le système. Il doit être conservé aussi longtemps que les objets sont gérés dans le système.

- Dès la réception des objets, on trace les opérations effectuées sur les groupes d'objets et objets qui sont dans le SIP.
- Les journaux du cycle de vie sont « committés » une fois le stockage des objets effectué et l'indexation des métadonnées effectuée, avant l'envoi d'une notification au service versant.

Chaque groupe d'objets possède une et une seule entrée dans la collection LogbookLifeCycleObjectGroup.

3.3.2. Extrait d'un JSON stocké en base comprenant l'exhaustivité des champs

```
{
  "_id": "aebaaaaabahf4qxrab2nualjtkuydyyaaaaq",
  "evId": "aedqaaaabchf4qxrab2nualjtkuyetyaaaaq",
  "evParentId": null,
  "evType": "LFC.LFC_CREATION",
  "evDateTime": "2019-03-20T10:33:14.063",
  "evIdProc": "aeeaaaaabchgzebuaaeckaljtkuxtjqaaaaq",
  "evTypeProc": "INGEST",
  "outcome": "OK",
  "outDetail": "LFC.LFC_CREATION.OK",
  "outMessg": "Succès de l'alimentation du journal du cycle de vie",
  "agId": "{ \"Name\": \"vitam-env-int-worker-01.vitam-env\\n\", \"Role\": \"worker\\n\", \"ServerId\": \"1046364913\", \"SiteId\": \"1\", \"GlobalPlatformId\": \"241058545\" }",
  "obId": "aebaaaaabahf4qxrab2nualjtkuydyyaaaaq",
  "evDetData": null,
  "events": [
    {

```

```

        "evId": "aedqaaaabchf4qxrab2nualjtkuyetyaaaba",
        "evParentId": null,
        "evType": "LFC.CHECK_MANIFEST",
        "evDateTime": "2019-03-20T10:33:14.063",
        "evIdProc": "aeaaaaabchgzebuaaeckaljtuxtjqaaaq",
        "evTypeProc": "INGEST",
        "outcome": "OK",
        "outDetail": "LFC.CHECK_MANIFEST.OK",
        "outMessg": "Succès de la vérification de la cohérence du bordereau de
transfert",
        "agId": "{ \"Name\": \"vitam-env-int-worker-01.vitam-
env\", \"Role\": \"worker\", \"ServerId\": 1046364913, \"SiteId\": 1, \"GlobalPlatformId\":
241058545 }",
        "obId": "aebaaaaabahf4qxrab2nualjtkuydyaaaaq",
        "evDetData": null,
        "_lastPersistedDate": "2019-03-20T10:33:28.028"
    },
    {
        "evId": "aedqaaaabchf4qxrab2nualjtkuyetyaaabq",
        "evParentId": "aedqaaaabchf4qxrab2nualjtkuyetyaaaba",
        "evType": "LFC.CHECK_MANIFEST.LFC_CREATION",
        "evDateTime": "2019-03-20T10:33:14.063",
        "evIdProc": "aeaaaaabchgzebuaaeckaljtuxtjqaaaq",
        "evTypeProc": "INGEST",
        "outcome": "OK",
        "outDetail": "LFC.CHECK_MANIFEST.LFC_CREATION.OK",
        "outMessg": "Succès de la création du journal du cycle de vie",
        "agId": "{ \"Name\": \"vitam-env-int-worker-01.vitam-
env\", \"Role\": \"worker\", \"ServerId\": 1046364913, \"SiteId\": 1, \"GlobalPlatformId\":
241058545 }",
        "obId": "aebaaaaabahf4qxrab2nualjtkuydyaaaaq",
        "evDetData": null,
        "_lastPersistedDate": "2019-03-20T10:33:28.028"
    },
    {
        "evId": "aedqaaaachf4qxrab2nualjtkuygqaaaaaq",
        "evParentId": null,
        "evType": "LFC.CHECK_CONSISTENCY",
        "evDateTime": "2019-03-20T10:33:14.304",
        "evIdProc": "aeaaaaabchgzebuaaeckaljtuxtjqaaaq",
        "evTypeProc": "INGEST",
        "outcome": "OK",
        "outDetail": "LFC.CHECK_CONSISTENCY.OK",
        "outMessg": "Succès de la vérification de la cohérence entre objets,
groupes d'objets et unités archivistiques",
        "agId": "{ \"Name\": \"vitam-env-int-worker-01.vitam-
env\", \"Role\": \"worker\", \"ServerId\": 1046364913, \"SiteId\": 1, \"GlobalPlatformId\":
241058545 }",
        "obId": "aebaaaaabahf4qxrab2nualjtkuydyaaaaq",
        "evDetData": null,
        "_lastPersistedDate": "2019-03-20T10:33:28.028"
    },
    [...]
],
    "_tenant": 8,
    "_v": 5,
    "_lastPersistedDate": "2019-03-20T10:33:28.028"

```

3.3.3. Détail des champs du JSON stocké en base

« **_id** » : identifiant donné par le système lors de l’initialisation du journal du cycle de vie.

- Il est constitué d’une chaîne de 36 caractères correspondant à un GUID. Il reprend la valeur du champ **_id** du groupe d’objets enregistré dans la collection ObjectGroup.

- Cet identifiant constitue la clé primaire du journal du cycle de vie du groupe d'objets.
- Cardinalité : 1-1
- Ce champ existe uniquement pour la structure incluante.

« **evId** » (**event Identifier**): identifiant de l'événement.

- Il est constitué d'une chaîne de 36 caractères correspondant à un GUID.
- Il identifie l'événement de manière unique dans la base.
- Cardinalité : 1-1
- Ce champ existe pour les structures incluantes et incluses.

« **evParentId** » (**event Parent Identifier**): identifiant de l'événement parent.

- Il est constitué d'une chaîne de 36 caractères correspondant à un GUID.
- Il identifie l'événement parent. Par exemple pour l'événement LFC.CHECK_MANIFEST.LFC_CREATION, ce champ fera référence au GUID de l'évènement LFC.CHECK_MANIFEST.
- La valeur du champ est toujours « null » pour la structure incluante et les tâches principales.
- Cardinalité : 1-1
- Ce champ existe pour les structures incluantes et incluses.

« **evType** » (**event Type**): nom de l'événement.

- Il s'agit d'une chaîne de caractères.
- La liste des valeurs possibles pour ce champ se trouve en annexe. Seul le code doit être stocké dans ce champ, la traduction doit se faire via le fichier properties (vitam-logbook-message-fr.properties).
- Cardinalité : 1-1
- Ce champ existe pour les structures incluantes et incluses.

« **evDateTime** » (**event DateTime**): date de l'événement.

- Il s'agit d'une date au format ISO8601 AAAA-MM-JJ+ »T »+hh:mm:ss:[3 digits de millisecondes]
- Ce champ est positionné par le client LogBook.
- Cardinalité : 1-1
- Ce champ existe pour les structures incluantes et incluses.

Exemple : "2016-08-17T08:26:04.227".

« **evIdProc** » (**event Identifier Process**): identifiant du processus.

- Il s'agit d'une chaîne de 36 caractères.
- Toutes les occurrences de ce champ pour un même document du journal du cycle de vie partagent la même valeur, qui est celle du champ « _id » de l'opération enregistrée dans la collection LogbookOperation.
- Cardinalité : 1-1
- Ce champ existe pour les structures incluantes et incluses.

« **evTypeProc** » (**event Type Process**): type de processus.

- Il s'agit d'une chaîne de caractères.
- Nom du processus parmi une liste de processus possibles fixée. Cette liste est disponible en

annexe.

- Cardinalité : 1-1
- Ce champ existe pour les structures incluant et incluses

« **outcome** » : statut de l'événement.

- Il s'agit d'une chaîne de caractères devant correspondre à une valeur de la liste suivante :
 - STARTED (Début de l'événement)
 - OK (Succès de l'événement)
 - KO (Échec de l'événement)
 - WARNING (Succès de l'événement comportant des alertes)
 - FATAL (Erreur technique)
- Cardinalité : 1-1
- Ce champ existe pour les structures incluant et incluses.

« **outDetail** » (**outcome Detail**): code correspondant à l'erreur.

- Il s'agit d'une chaîne de caractères.
- Il contient le code fin de l'événement, incluant le statut. La liste des valeurs possibles pour ce champ se trouve en annexe. Seul le code est stocké dans ce champ, la traduction doit se faire via le fichier properties (vitam-logbook-message-fr.properties)
- Cardinalité : 1-1
- Ce champ existe pour les structures incluant et incluses.

« **outMessg** » (**outcome Detail Message**): détail du résultat de l'événement.

- Il s'agit d'une chaîne de caractères.
- C'est un message intelligible destiné à être lu par un être humain en tant que détail du résultat de l'événement.
- Traduction du code présent dans outDetail, issue du fichier vitam-logbook-message-fr.properties.
- Cardinalité : 1-1
- Ce champ existe pour les structures incluant et incluses.

« **agId** » (**agent Identifiant**): identifiant de l'agent réalisant l'événement.

- Il s'agit d'un objet JSON contenant plusieurs chaînes de caractères indiquant le nom, le rôle et le PID de l'agent.
- Ce champ est calculé par le journal à partir de ServerIdentifiant.
- Cardinalité : 1-1
- Ce champ existe pour les structures incluant et incluses.

Exemple : `{"Name\":\"vitam-iaas-app-01\", \"Role\":\"ingest-external\", \"ServerId\":1514166061, \"SiteId\":1, \"GlobalPlatformId\":171988781}`

« **obId** » (**object Identifiant**): identifiant de la solution logicielle Vitam du lot d'objets auquel s'applique l'opération (lot correspondant à une liste).

- Si l'événement touche tout le groupe d'objets, alors le champ contiendra l'identifiant de ce groupe d'objets. S'il ne touche qu'un seul objet du groupe d'objets, alors il ne contiendra que celui de l'objet en question.

- Cardinalité : 1-1
- Ce champ existe pour les structures incluant et incluses.

« **evDetData** » (**event Detail Data**): détails des données de l'événement.

- Donne plus de détails sur l'événement ou son résultat.
- Par exemple, pour l'événement LFC.CHECK_DIGEST, lorsque l'empreinte d'un objet inscrite dans le bordereau de transfert n'est pas calculée en SHA-512, ce champ précise l'empreinte d'origine et celle réalisée ensuite par la solution logicielle Vitam. Dans la structure incluse correspondant à cet événement, il contient un objet JSON composé des champs suivants :
 - MessageDigest : empreinte de l'objet dans le bordereau de transfert. Chaîne de caractères, reprenant le champ « MessageDigest » du message ArchiveTransfer.
 - Algorithm : algorithme de hachage utilisé dans le bordereau de transfert. Chaîne de caractères, reprenant l'attribut de champ « MessageDigest » du message ArchiveTransfer.
 - SystemMessageDigest : empreinte de l'objet réalisée par la solution logicielle Vitam. Chaîne de caractères.
 - SystemAlgorithm : algorithme de hachage utilisé par la solution logicielle Vitam. Chaîne de caractères.
- Cardinalité : 1-1
- Ce champ existe pour les structures incluant et incluses.

« **events** » : tableau de structure.

- Pour la structure incluant, le tableau contient n structures incluses dans l'ordre des événements (date).
- Cardinalité : 1-1
- S'agissant d'un tableau, les structures incluses ont pour cardinalité 1-n.
- Ce champ existe uniquement pour la structure incluant.

« **_tenant** » : identifiant du tenant.

- Il s'agit d'un entier.
- Cardinalité : 1-1

« **_v** » : version de l'enregistrement décrit.

- Il s'agit d'un entier.
- Si le numéro est supérieur à 0, alors il s'agit du numéro de version de l'enregistrement.
- Cardinalité : 1-1
- Ce champ existe pour les structures incluant et incluses.

« **_lastPersistedDate** » : date technique de sauvegarde en base.

- Il s'agit d'une date au format ISO8601 AAAA-MM-JJ+ »T »+hh:mm:ss:[3 digits de millisecondes]
- Elle est renseignée par le serveur Logbook.
Exemple : "2016-08-17T08:26:04.227"
- Cardinalité : 1-1
- Ce champ existe uniquement pour la structure incluant.

3.3.4. Champs présents dans les events

- evId
- evParentId
- evType
- evDateTime
- evIdProc
- evTypeProc
- outcome
- outDetail
- outMessg
- agId
- obId
- evDetData
- LastPersistedDate

3.4. Collection Offset

3.4.1. Utilisation de la collection

Cette collection, optionnelle, permet de persister les offsets des dernières données reconstruites des offres de stockage lors de la reconstruction au fil de l'eau pour les collections :

- LogbookOperation
- Unit
- ObjetGroup
- UNIT_GRAPH
- OBJETGROUP_GRAPH

Il y a une valeur d'offset par couple tenant/collection.

3.4.2. Exemple de JSON stocké en base comprenant l'exhaustivité des champs

```
{
  "_id": ObjectId("507f191e810c19729de860ea"),
  "offset": 1357,
  "collection": "logbook",
  "_tenant": 1
}
```

3.4.3. Détail des champs

« **_id** » : identifiant unique mongo.

- Il s'agit d'un champ de type mongo composé comme suit : ObjectId(<hexadecimal>).
- Champ peuplé par la solution logicielle Vitam.
- Cardinalité : 1-1

« **offset** » : la valeur de l'offset.

- Il s'agit d'un entier encodé 64 bits.

- Champ peuplé par la solution logicielle Vitam.
- Cardinalité : 1-1

« **collection** » : collection impactée.

- La seule valeur possible est *logbook*.

« **_tenant** » : identifiant du tenant.

- Il s'agit de l'identifiant du tenant utilisant l'enregistrement.
- Il s'agit d'un entier.
- Champ peuplé par la solution logicielle Vitam.
- Cardinalité : 1-1

4. Base MetaData

La base Metadata contient les collections relatives aux métadonnées des unités archivistiques (collection Unit) et des groupes d'objets (collection ObjectGroup). Une collection technique Offset liée à la reconstruction existe également.

4.1. Collection Unit

4.1.1. Utilisation de la collection Unit

La collection Unit contient les informations relatives aux unités archivistiques.

4.1.2. Exemple de XML en entrée

Ci-après, la portion d'un bordereau de transfert (manifest.xml) utilisée pour compléter les champs du JSON. Il s'agit des informations situées entre les balises <ArchiveUnit>.

```
<?xml version="1.0" encoding="UTF-8"?>
<ArchiveUnit id="ID3">
  <Management>
    <AccessRule>
      <Rule>ACC-00002</Rule>
      <StartDate>2015-11-19</StartDate>
    </AccessRule>
  </Management>
  <Content>
    <DescriptionLevel>RecordGrp</DescriptionLevel>
    <Title>Liste des Métros de Tokyo</Title>
    <Description>Le métro de Tokyo (Tōkyō no chikatetsu) est un des
    systèmes de transport en commun desservant l'agglomération de Tokyo au Japon. Il se
    compose des lignes de deux compagnies : Tokyo Metro et Toei.
    La première ligne du métro de Tokyo, la ligne Ginza, a été
    ouverte en 1927.</Description>
    <AcquiredDate>2015-12-04T09:02:25</AcquiredDate>
    <StartDate>2015-12-04T09:02:25</StartDate>
    <EndDate>2016-06-02T15:32:29</EndDate>
    <Event>
      <EventDateTime>2013-11-04T09:02:25</EventDateTime>
    </Event>
  </Content>
  <ArchiveUnit id="ID5">
    <ArchiveUnitRefId>ID4</ArchiveUnitRefId>
  </ArchiveUnit>
</ArchiveUnit>
```

4.1.3. Exemple de JSON stocké dans la collection Unit

Les champs présentés dans l'exemple ci-après ne font pas état de l'exhaustivité des champs disponibles dans le SEDA. Ceux-ci sont référencés dans la documentation SEDA disponible au lien suivant : https://redirect.francearchives.fr/seda/api_v2-1/seda-2.1-main.html

```
{
  "_id": "aeaqaamaamhad455abcwsalep4lzf2iaaaaa",
  "_og": "aebaaaaamhad455abcwsalep4lzfvaaaaa",
  "_mgt": {
    "AccessRule": {
      "Rules": [
        {
          "Rule": "ACC-00002",
          "StartDate": "2000-01-01",

```

```

        "EndDate": "2025-01-01"
      }
    ]
  }
},
"DescriptionLevel": "Item",
"Title": "Stalingrad.txt",
"TransactedDate": "2017-04-04T08:07:06",
"SedaVersion": "2.1",
"ImplementationVersion": "1.7.0-SNAPSHOT",
"_storage": {
  "strategyId": "default"
},
"_sps": [
  "RATP"
],
"_sp": "RATP",
"_ops": [
  "aeaaaaaaaaohi422caa4paalep4lxwoyaaaaaq",
  "aeaaaaaaaaohi422caaieaaalesqjo5hqaaaaaq",
  "aeaaaaaaaaohi422caaieaaalesqkbhnaaaaaaq",
  "aeaaaaaaaaohi422caaieaaalesqml2vyaaaaaq"
],
"_opi": "aeaaaaaaaaohi422caa4paalep4lxwoyaaaaaq",
"_unitType": "INGEST",
"_up": [
  "aeaqaamaamhad455abcwsalep4lzf2iaaada"
],
"_us": [
  "aeaqaamaamhad455abcwsalep4lzf2aaaaeq",
  "aeaqaamaamhad455abcwsalep4lzf2iaaada",
  "aeaqaamaamhad455abcwsalep4lzf2iaaabq"
],
"_graph": [
"aeaqaamaamhad455abcwsalep4lzf2iaaabq/aeaqaamaamhad455abcwsalep4lzf2aaaaeq",
"aeaqaamaamhad455abcwsalep4lzf2iaaade/aeaqaamaamhad455abcwsalep4lzf2iaaada",
  "aeaqaamaamhad455abcwsalep4lzf2iaaada/aeaqaamaamhad455abcwsalep4lzf2iaaabq"
],
"_uds": {
  "1": [
    "aeaqaamaamhad455abcwsalep4lzf2iaaada"
  ],
  "2": [
    "aeaqaamaamhad455abcwsalep4lzf2iaaabq"
  ],
  "3": [
    "aeaqaamaamhad455abcwsalep4lzf2aaaaeq"
  ]
},
"_us_sp": {
  "RATP": [
    "aeaqaamaamhad455abcwsalep4lzf2aaaaeq",
    "aeaqaamaamhad455abcwsalep4lzf2iaaada",
    "aeaqaamaamhad455abcwsalep4lzf2iaaabq"
  ]
},
"_min": 1,
"_max": 4,
"_glpd": "2018-07-09T12:50:30.733",
"_v": 3,
"_av": 0,
"_tenant": 3,
"Description": "",
"_history": [
  {

```


- ClassificationRule (durée de classification)
- DisseminationRule (durée de diffusion)
- ReuseRule (durée de réutilisation)
- StorageRule (durée d'utilité courante)

Cardinalité : 0-1, pour chaque catégorie.

Chaque catégorie peut contenir :

- « **Rules** » : tableau de règles de gestion.
 - Il s'agit d'un tableau d'objets
 - Cardinalité : 0-1
 - Chacune des règles de ce tableau est elle-même composée de plusieurs informations :
 - « **Rule** » : identifiant de la règle
 - Correspond à une valeur du champ RuleId de la collection FileRules.
 - Cardinalité : 0-1
 - « **StartDate** » : date de début du calcul de l'échéance.
 - Il s'agit d'une date.
 - Cette date est déclarée dans le message ArchiveTransfer ou ajoutée *a posteriori* par une modification de l'unité archivistique.
 - Cardinalité : 0-1
 - « **EndDate** » : date de fin d'application de la règle.
 - Il s'agit d'une date.
 - Cette valeur est issue d'un calcul réalisé par la solution logicielle Vitam. Celui-ci consiste en l'ajout du délai correspondant à la règle dans la collection FileRules à la valeur du champ startDate (EndDate = StartDate + Durée)
 - Cardinalité : 0-1
- Des données spécifiques aux catégories :
 - Pour les catégories « StorageRule » et « AppraisalRule » uniquement :
 - « **FinalAction** » : sort final des règles dans ces catégories.
 - Cardinalité : 1-1
 - La valeur contenue dans le champ peut être :
 - Pour StorageRule : « Transfer », « Copy » ou « RestrictAccess » (énumération issue du FinalActionStorageCodeType du SEDA 2.1)
 - Pour AppraisalRule : « Keep » ou « Destroy » (énumération issue du FinalActionAppraisalCodeType du SEDA 2.1)
 - Pour la catégorie ClassificationRule uniquement :
 - « **ClassificationLevel** » : niveau de classification.
 - Il s'agit d'une chaîne de caractères, dont les valeurs sont paramétrables au niveau de la plateforme.
 - Champ obligatoire et systématiquement renseigné
 - Cardinalité : 1-1
 - « **ClassificationOwner** » : propriétaire de la classification.

- Il s'agit d'une chaîne de caractères.
- Champ obligatoire et systématiquement renseigné
- Cardinalité : 1-1
- « **ClassificationAudience** » : permet de gérer les mentions additionnelles de limitation du champ de diffusion (exemple : « spécial France »)
 - Il s'agit d'une chaîne de caractères.
 - Champ optionnel
 - Cardinalité : 0-1
- « **ClassificationReassessingDate** » : date de réévaluation de la classification.
 - Il s'agit d'une date.
 - Champ optionnel.
 - Cardinalité : 0-1
- « **NeedReassessingAuthorization** » : indique si une autorisation humaine est nécessaire pour réévaluer la classification.
 - Il s'agit d'un booléen. Si la valeur est à « true », une autorisation humaine sera nécessaire pour réévaluer la classification.
 - Champ optionnel
 - Cardinalité : 0-1
- Des paramètres de gestion d'héritage de règles.
 - « **Inheritance** »
 - Il s'agit d'un objet.
 - Cardinalité 0-1
 - Cet objet peut avoir comme valeur :
 - « **PreventInheritance** » : utilisé pour bloquer l'héritage de toutes les règles de gestion de la même catégorie
 - Il s'agit d'un booléen, dont la valeur peut être « true » ou « false »,
 - Cardinalité : 1-1 à partir du moment où le champ Inheritance existe
 - « **PreventRulesId** » : règle(s) de gestion qui ne doivent pas être héritées d'un parent.
 - Il s'agit d'un tableau d'identifiants de règles de gestion.
 - A l'entrée, il s'agit de la valeur de la balise <RefNonRuleId> du SEDA
 - Cardinalité : 1-1 à partir du moment où le champ Inheritance existe.

Extrait d'une unité archivistique ayant un bloc « _mgt » possédant des règles de gestion :

```
"_mgt": {
  "AppraisalRule": {
    "Rules": [
      {
        "Rule": "APP-00001",
        "StartDate": "2015-01-01",
        "EndDate": "2095-01-01"
      },
      {
        "Rule": "APP-00002"
      }
    ],
    "Inheritance": {
```

```
        "PreventInheritance": true,
        "PreventRulesId": []
    },
    "FinalAction": "Keep"
},
"AccessRule": {
    "Rules": [
        {
            "Rule": "ACC-00001",
            "StartDate": "2016-06-03",
            "EndDate": "2016-06-03"
        }
    ]
},
"DisseminationRule": {
    "Inheritance": {
        "PreventInheritance": true,
        "PreventRulesId": []
    }
},
"ReuseRule": {
    "Inheritance": {
        "PreventRulesId": [
            "REU-00001", "REU-00002"
        ]
    }
},
"ClassificationRule": {
    "ClassificationLevel": "Secret Défense",
    "ClassificationOwner": "Projet_Vitam",
    "Rules": [
        {
            "ClassificationReassessingDate": "2025-06-03",
            "NeedReassessingAuthorization": true,
            "Rule": "CLASS-00001"
        }
    ]
}
},
```

« **DescriptionLevel** » : niveau de description archivistique de l'unité archivistique.

- Il s'agit d'une chaîne de caractères.
- Ce champ est renseigné avec les valeurs situées entre les balises <DescriptionLevel> présentes dans le bordereau de transfert.
- Cardinalité : 1-1

« **Title** » : titre de l'unité archivistique.

- Il s'agit d'une chaîne de caractères.
- Ce champ est renseigné avec les valeurs situées entre les balises <Title> dans le bordereau de transfert.
- Cardinalité : 0-1, le modèle d'une unité archivistique doit comporter au moins un champ Title et/ou au moins un champ Title_

« **Title_** » : titres de l'unité archivistique par langue

- Il s'agit d'un tableau JSON.
- Les titres sont organisés sous la forme de clef : valeur, la clef étant l'indicatif de la langue en xml:lang et la valeur le titre. Par exemple : « fr » : « Ceci est un titre. »
- Cardinalité : 0-1, le modèle d'une unité archivistique doit comporter au moins un champ Title et/ou au moins un champ Title_

```
{  
  "fr": "FrenchMySIP",  
  "en": "EnglishMySIP"  
},
```

« **Description** » : description de l'unité archivistique.

- Il s'agit d'une chaîne de caractères.
- Ce champ est renseigné avec les informations situées entre les balises <Description> de l'unité archivistique concernée dans le bordereau de transfert.
- Cardinalité : 0-1, le modèle d'une unité archivistique doit comporter au moins un champ Description et/ou au moins un champ Description_

« **Description_** » : description de l'unité archivistique par langue.

- Il s'agit d'un tableau JSON
- Les titres sont organisés sous la forme de clef : valeur, la clef étant l'indicatif de la langue en xml:lang et la valeur la description. Par exemple : « fr » : « Ceci est une description. »
- Cardinalité : 0-1, le modèle d'une unité archivistique doit comporter au moins un champ Description et/ou au moins un champ Description_

```
"Description_": {  
  "fr": "Une autre description",  
  "en": "another description"  
},
```

« **XXXXXX** » : des champs facultatifs peuvent être contenus dans l'enregistrement JSON lorsqu'ils sont renseignés dans le bordereau de transfert au niveau du Content de chaque unité archivistique.

- Se reporter à la documentation descriptive du SEDA 2.1 et notamment le schéma ontology.xsd pour connaître la liste des métadonnées facultatives.

ArchiveUnitProfile: profil d'archivage de l'unité archivistique utilisé lors de l'entrée.

- Correspond à l'identifiant du profil d'archivage associé à l'unité archivistique
- Chaîne de caractères.
- Cardinalité : 0-1

« **_sedaVersion** » : version du SEDA utilisé lors de l'entrée de cette unité archivistique.

- Champ peuplé par la solution logicielle Vitam.
- Exemple de valeur : « 2.1 »
- Cardinalité : 1-1

« **_implementationVersion** » : version du modèle de donnée actuellement utilisé par l'unité archivistique.

- Champ peuplé par la solution logicielle Vitam.
- Exemple de valeur : « 1.7.0-SNAPSHOT »
- Cardinalité : 1-1

« **_history** » : données historiques de l'unité archivistique

- Champ peuplé par la solution logicielle Vitam au moment d'une mise à jour d'une unité archivistique, uniquement si la mise à jour déclenche une historisation.
- Cardinalité : 0-1
- Ce champ contient les clés suivantes :

- « **ud** » : date du changement de la métadonnée.
 - Champ peuplé par la solution logicielle Vitam.
 - Il s’agit d’une date au format ISO 8601 `YYY-MM-DD + “T” + hh:mm:ss.millisecondes « + » timezone hh:mm`.
- Exemple : `2016-08-19T16:36:07.942+02:00`
- Cardinalité : 1-1
- « **data** » : données historisées.
 - Il s’agit d’un objet, peuplé par la solution logicielle Vitam.
 - Cardinalité : 1-1
 - Le champ « data » contient les champs suivants :
 - Il s’agit d’un entier, peuplé par la solution logicielle Vitam.
 - « **_v** » : version de l’enregistrement de l’unité archivistique avant modification. Ce champ est repris du champ « **_v** » à la racine du modèle de données de l’unité archivistique.
 - « **_mgt** » : règle de gestion historicisée.
 - Ce champ reprend le contenu de la version précédemment modifiée d’une règle de gestion.
 - Dans l’exemple ci-dessous, on constate qu’au 25 juillet 2018, l’unité archivistique a historisé une règle de classification située dans le bloc Management (**_mgt**) de son modèle.
 - Peut être vide.
 - Cardinalité : 1-1.

```

_history": [
{
  "ud": "2018-07-25T15:28:49.040",
  "data": {
    "_v": 0,
    "_mgt": {
      "ClassificationRule": {
        "ClassificationAudience": "ClassificationAudience0",
        "ClassificationLevel": "Secret Défense",
        "ClassificationOwner": "ClassificationOwner0",
        "ClassificationReassessingDate": "2016-06-03",
        "NeedReassessingAuthorization": true,
        "Rules": [
          {
            "Rule": "CLASS-00001",
            "StartDate": "2015-06-03",
            "EndDate": "2025-06-03"
          }
        ]
      }
    }
  }
}
]

```

Le champ **_history** peut également être créé depuis les données contenues dans un bordereau de transfert, contenues dans le bloc Content d’une unité archivistique :

```

<History>
  <UpdateDate>2018-08-02T14:06:23.374</UpdateDate>
  <Data>

```

```
        <Version>0</Version>
        <Management>
          <ClassificationRule>
            <ClassificationLevel>Secret Défense</ClassificationLevel>
        </ClassificationRule>
    </ClassificationOwner>ClassificationOwner0</ClassificationOwner>
        </ClassificationRule>
    </Management>
    </Data>
  </History>
  <History>
    <UpdateDate>2018-08-02T14:30:20.137</UpdateDate>
    <Data>
      <Version>1</Version>
      <Management>
        <ClassificationRule>
          <ClassificationLevel>Confidentiel
Défense</ClassificationLevel>
        </ClassificationRule>
    </Management>
  </Data>
</History>
```

Le mapping est le suivant :

- La balise <History> du bordereau devient le tableau « _history » dans la base de données
- <Data> devient « data »
- <Version> devient « _v »
- <Management> devient « _mgt »

« **_storage** » : contient les champs qui permettent d'identifier les offres de stockage.

- Il s'agit d'un objet constitué du champ :
 - « **strategyId** » : identifiant de la stratégie de stockage.
 - Il s'agit d'une chaîne de caractère.
- Ne peut être vide
- Cardinalité : 1-1

« **_sps** » : services producteurs auxquels l'unité archivistique a été rattachée (au titre de leurs fonds symboliques)

- Il s'agit d'un tableau contenant les identifiants de tous les services producteurs référençant l'unité archivistique.
- Il s'agit d'un tableau de chaînes de caractères.
- Correspond à une valeur valide du champ « Identifier » de la collection Agencies.
- Ne peut être vide.
- Cardinalité : 1-1

« **_sp** » : service producteur responsable de l'unité archivistique, qui appartient à son fond propre.

- Il s'agit du service producteur inscrit dans le bordereau de transfert lié au transfert de l'unité archivistique et déclaré dans la balise <OriginatingAgencyIdentifier> du message ArchiveTransfer.
- Il s'agit d'une chaîne de caractères.
- Correspond à une valeur valide du champ « Identifier » de la collection Agencies.

- Cardinalité : 1-1

« **_ops** » (**operations**): tableau contenant les identifiants d'opérations auxquelles cette unité archivistique a participé.

- Il s'agit d'un tableau contenant une à plusieurs chaînes de 36 caractères correspondant au champ `_id` de l'opération ou GUID de l'opération, enregistré dans la collection `LogbookOperation`.
- Ne peut être vide.
- Cardinalité : 1-1

« **_opi** » : identifiant de l'opération à l'origine de la création de cette unité archivistique.

- Il s'agit d'une chaîne de 36 caractères correspondant au champ `_id` de l'opération d'entrée ou GUID de cette opération, enregistré dans la collection `LogbookOperation`.
- Ne peut être vide.
- Cardinalité : 1-1

« **_unitType** » : champ indiquant le type d'unité archivistique concerné.

- Il s'agit d'une chaîne de caractères.
- La valeur contenue doit être conforme à l'énumération `UnitType`. Celle-ci peut être :
 - `INGEST` : unité archivistique issue d'un SIP
 - `FILING_UNIT` : unité archivistique issue d'un plan de classement
 - `HOLDING_UNIT` : unité archivistique issue d'un arbre de positionnement
- Cardinalité : 1-1

« **_up** » (**unit up**): tableau recensant les `_id` des unités archivistiques parentes (parents immédiats).

- Il s'agit d'une chaîne de 36 caractères correspondant à un GUID. Valeur du champ `_id` d'une unité archivistique (ou GUID) enregistré dans la collection `Unit`.
- Champ peuplé par la solution logicielle Vitam.
- Tableau pouvant être vide.
- Cardinalité : 1-1

« **_us** » : tableau contenant la parentalité, c'est-à-dire l'ensemble des unités archivistiques parentes, indexé de la manière suivante : [`GUID1, GUID2...`].

- Tableau listant une à plusieurs chaînes de 36 caractères correspondant à un GUID. Valeur du champ `_id` d'une unité archivistique (ou GUID) enregistré dans la collection `Unit`.
- Champ peuplé par la solution logicielle Vitam.
- Tableau pouvant être vide pour l'unité archivistique racine uniquement
- Cardinalité : 1-1

« **_graph** » : Tableau des chemins de l'unité archivistique

- Il s'agit d'un tableau contenant tous les chemins pour accéder à l'unité archivistique depuis les racines. Ces chemins sont composés sous la forme `id1/id2/id3/.../idn` Où chaque `id` est un identifiant d'unité archivistique. `id1` étant l'unité courante et où `idn` est l'identifiant de l'unité de plus haut niveau.
- Tableau pouvant être vide.
- Cardinalité 1-1

« **_us_sp** » : Liste des unités archivistique parentes concernant tous les niveaux de parentalité

- Il s'agit d'un objet contenant les identifiants de tous les services producteurs de tous les parents liées à l'unité archivistique.
- Il s'agit d'un objet contenant une liste de chaînes de caractères.
- Vide uniquement si l'unité archivistique n'a pas de parents
- Cardinalité : 1-1

```
"_us_sp": {
  "FRAN_NP_050632": [
    "aeaqaabaahad7diaat24allvtywa4aaaaba",
    "aeaqaabaahad7diaat24allvtywa4yaaaba"
  ],
  "FRAN_NP_050634": [
    "aeaqaabaahducypaasryallvmupmaaaada"
  ],
  "FRAN_NP_051587": [
    "aeaqaabaahducypaasryallvmo4sqaaaba",
    "aeaqaabaahducypaasryallvmo4ryaaaba"
  ]
}
```

« **_uds** » : objet contenant la parentalité, c'est-à-dire l'ensemble des unités archivistiques parentes, ainsi que le niveau de profondeur relative.

- Il s'agit d'un objet contenant une liste de tableaux JSON.
- Ces informations sont réunies dans cet objet sous la forme de clef/valeur, la clé étant la profondeur du parent (de type entier), la valeur étant elle-même un tableau d'identifiant d'unité archivistique. Exemple d'une unité qui a un parent direct, lui-même ayant deux parents.
- Champ peuplé par la solution logicielle Vitam.
- Peut être vide.
- Cardinalité : 1-1

```
"1": [
  "aeaqaabaamhad455abcwsalep4lzf2iaaada"
],
"2": [
  "aeaqaabaamhad455abcwsalep4lzf2iaaabq",
  "aeaqaabaamhad455abcwsalep4lzf2iaaabq"
],
```

« **_min** » : profondeur minimum de l'unité archivistique par rapport à une racine.

- Il s'agit d'un entier.
- Calculée, cette profondeur correspond au minimum des profondeurs, quels que soient les racines concernées et les chemins possibles.
- Champ peuplé par la solution logicielle Vitam.
- Cardinalité : 1-1

« **_max** » : profondeur maximale de l'unité archivistique par rapport à une racine.

- Il s'agit d'un entier.

- Calculée, cette profondeur correspond au maximum des profondeurs, quels que soient les racines concernées et les chemins possibles.
- Champ peuplé par la solution logicielle Vitam.
- Cardinalité : 1-1

« **_glpd** » : Date de la dernière modification du graph dont l'unité dépend

- Il s'agit d'une date au format ISO 8601 YYYY-MM-DD + "T" + hh:mm:ss.millisecondes « + » timezone hh:mm.

Exemple : 2016-08-19T16:36:07.942+02:00

- Champ peuplé par la solution logicielle Vitam.
- Cardinalité : 1-1

« **_av** » : version atomique de l'enregistrement décrit, incrémentée automatiquement en cas de modification de tout champ de la collection.

- Il s'agit d'un entier.
- Champ peuplé par la solution logicielle Vitam.
- 0 correspond à l'enregistrement d'origine. Si le numéro est supérieur à 0, alors il s'agit du numéro de version de l'enregistrement.
- Cardinalité : 1-1

« **_v** » : version de l'enregistrement décrit, incrémentée dans le seul cas de modification d'un champ descriptif.

- Il s'agit d'un entier.
- Champ peuplé par la solution logicielle Vitam.
- 0 correspond à l'enregistrement d'origine. Si le numéro est supérieur à 0, alors il s'agit du numéro de version de l'enregistrement.
- Cardinalité : 1-1

« **_tenant** » : identifiant du tenant.

- Il s'agit d'un entier.
- Champ peuplé par la solution logicielle Vitam.
- Cardinalité : 1-1

« **_elimination** » : tableau contenant les résultats pour l'unité archivistique lorsqu'une opération d'analyse d'élimination a été lancée.

- Il s'agit d'un tableau, pouvant référencer plusieurs opérations d'analyse d'élimination.
- Champ peuplé par la solution logicielle Vitam au moment d'une indexation réalisée lors d'une phase d'analyse d'élimination.
- Cardinalité : 1-1
- Ce bloc contient les clés suivantes :
 - « **OperationId** » : identifiant de l'opération d'élimination
 - Il s'agit d'une chaîne de 36 caractères
 - Ne peut être vide
 - Cardinalité : 1-1
 - « **GlobalStatus** » : indique le statut de l'unité archivistique lors de son indexation

- les valeurs ne peuvent être que DESTROY ou CONFLICT
- Ne peut être vide.
- Cardinalité : 1-1
- « **DestroyableOriginatingAgencies** » : Service(s) producteur(s) pour le(s)quel(s) l'unité archivistique est éliminable
 - Il s'agit d'un tableau pouvant contenir une à plusieurs chaînes de caractères, correspondant à l'identifiant d'un service agent référencé dans la collection « Agencies ».
 - Peut être vide.
 - Cardinalité : 1-1
- « **NonDestroyableOriginatingAgencies** » : Service(s) producteur(s) pour le(s)quel(s) l'unité archivistique n'est pas éliminable
 - Il s'agit d'un tableau pouvant contenir une à plusieurs chaînes de caractères, correspondant à l'identifiant d'un service agent référencé dans la collection « Agencies ».
 - Peut être vide.
 - Cardinalité : 1-1
- « **ExtendedInfo** » : tableau donnant des informations complémentaires dans les cas de CONFLICT
 - Il s'agit d'un tableau d'objets.
 - Peut être vide.
 - Cardinalité : 1-1
 - Ce champ peut contenir une liste d'objets comprenant les éléments suivants :
 - « **ExtendedInfoType** » : ce champ indique les situations impliquant un CONFLICT.
 - Il s'agit d'une chaîne de caractères.
 - Cardinalité : 0-1
- Les valeurs attendues dans ce champ sont :
 - « **KEEP_ACCESS_SP** » : l'unité archivistique n'est pas éliminable car l'accès est conservé pour un service producteur autre que le service producteur principal.
 - « **ACCESS_LINK_INCONSISTENCY** » : l'unité archivistique n'est pas éliminable, car sa suppression occasionnerait une incohérence dans le fonds d'archives.
 - « **FINAL_ACTION_INCONSISTENCY** » : l'unité archivistique a par héritage deux sorts finaux différents pour un même service producteur.
- « **ExtendedInfoDetails** » : détails concernant les situations impliquant un CONFLICT.
 - Il s'agit d'un objet.
 - Cet objet est présent dans les cas de **ACCESS_LINK_INCONSISTENCY** et de **FINAL_ACTION_INCONSISTENCY**.
 - Cardinalité : 0-1
 - Pour chaque cas de **ACCESS_LINK_INCONSISTENCY**, l'unité parente est obligatoirement spécifiée avec son GUID, ainsi que le service producteur

concerné.

- « **ParentUnitId** » : identifiant de l'unité archivistique parente.
 - Il s'agit d'une chaîne de caractères, correspondant au GUID de l'unité archivistique parente.
 - Cardinalité : 1-1
- « **DestroyableOriginatingAgencies** » : Service(s) producteur(s) pour le(s)quel(s) l'unité archivistique est éliminable.
 - Il s'agit d'un tableau pouvant contenir une à plusieurs chaînes de caractères, correspondant à l'identifiant d'un service agent référencé dans la collection « Agencies ».
 - Cardinalité : 1-1
- « **NonDestroyableOriginatingAgencies** » : Service(s) producteur(s) pour le(s)quel(s) l'unité archivistique n'est pas éliminable.
 - Il s'agit d'un tableau pouvant contenir une à plusieurs chaînes de caractères, correspondant à l'identifiant d'un service agent référencé dans la collection « Agencies ».
 - Cardinalité : 1-1
- Pour chaque cas de « **FINAL_ACTION_INCONSISTENCY** », la solution logicielle indexe également le service producteur concerné.
 - « **OriginatingAgenciesInConflict** » : Service(s) producteur(s) pour le(s)quel(s) l'unité archivistique ne peut être éliminée.
 - Il s'agit d'un tableau pouvant contenir une à plusieurs chaînes de caractères, correspondant à l'identifiant d'un service agent référencé dans la collection « Agencies ».
 - Cardinalité : 0-1

```
"_elimination": [
  {
    "OperationId": "aeaaaaaabgho3tftabgyiallvvnhq2aaaaaq",
    "GlobalStatus": "CONFLICT",
    "DestroyableOriginatingAgencies": [
      "FRAN_NP_050634"
    ],
    "NonDestroyableOriginatingAgencies": [
      "FRAN_NP_051587"
    ],
    "ExtendedInfo": [
      {
        "ExtendedInfoType": "KEEP_ACCESS_SP"
      },
      {
        "ExtendedInfoType": "ACCESS_LINK_INCONSISTENCY",
        "ExtendedInfoDetails": {
          "ParentUnitId": "aeaqaabaabehducypaasryallvvmupmaaaada",
          "DestroyableOriginatingAgencies": [
            "FRAN_NP_050634"
          ],
          "NonDestroyableOriginatingAgencies": [
            "FRAN_NP_051587"
          ]
        }
      }
    ]
  }
]
```

« **_computedInheritedRule** » :

- une liste de catégories de règles de gestion appliquées à cette unité archivistique.

Les catégories pouvant être incluses dans cet objet sont, exhaustivement :

- AccessRule (délai de communicabilité)
- AppraisalRule (durée d'utilité administrative)
- ClassificationRule (durée de classification)
- DisseminationRule (durée de diffusion)
- ReuseRule (durée de réutilisation)
- StorageRule (durée d'utilité courante)

Cardinalité : 0-1, pour chaque catégorie.

Chaque catégorie peut contenir :

- Des données communes :
 - « MaxEndDate »
 - « EndDates » : Les dates de fin par règle de gestion hérités identifiés.
 - En clé il y a l'identifiant de la règle de gestion défini au niveau de l'unité archivistique ou hérités des unités archivistiques parents.
 - En valeur, il y a la date de fin de validité la plus longue en fonction des règles de gestion applicable.
- Des données spécifiques aux catégories :
 - Pour les catégories « StorageRule » et « AppraisalRule » uniquement :
 - « **FinalAction** » : sort final des règles dans ces catégories.
 - Cardinalité : 1-1
 - La valeur contenue dans le champ peut être :
 - Pour StorageRule : « Transfer », « Copy » ou « RestrictAccess » (énumération issue du FinalActionStorageCodeType du SEDA 2.1)
 - Pour AppraisalRule : « Keep » ou « Destroy » (énumération issue du FinalActionAppraisalCodeType du SEDA 2.1)
 - Pour la catégorie ClassificationRule uniquement :
 - « **ClassificationLevel** » : niveau de classification.
 - Il s'agit d'une chaîne de caractères, dont les valeurs sont paramétrables au niveau de la plateforme.
 - Champ obligatoire et systématiquement renseigné
 - Cardinalité : 1-1
 - « **ClassificationOwner** » : propriétaire de la classification.
 - Il s'agit d'une chaîne de caractères.
 - Champ obligatoire et systématiquement renseigné
 - Cardinalité : 1-1
 - « **ClassificationAudience** » : permet de gérer les mentions additionnelles de limitation du champ de diffusion (exemple : « spécial France »)
 - Il s'agit d'une chaîne de caractères.

- Champ optionnel
- Cardinalité : 0-1
- « **ClassificationReassessingDate** » : date de réévaluation de la classification.
 - Il s'agit d'une date.
 - Champ optionnel.
 - Cardinalité : 0-1
- « **NeedReassessingAuthorization** » : indique si une autorisation humaine est nécessaire pour réévaluer la classification.
 - Il s'agit d'un booléen. Si la valeur est à « true », une autorisation humaine sera nécessaire pour réévaluer la classification.
 - Champ optionnel
 - Cardinalité : 0-1

« **_inheritedRulesAPIOutput** » :

Il contient les règles de gestion calculée applicable en prenant en compte les différents héritages de règles de gestion provenant des différents services producteurs.

- Cardinalité : 0-1
- Peut être vide.
- Il contient :
 - « **Global Properties** » : tableau contenant une liste de propriété
 - Cardinalité : 0-1
 - Chaque propriété contient l'ensemble suivant d'informations :
 - « **UnitId** » : identifiant unique de l'unité archivistique.
 - Il s'agit d'une chaîne de 36 caractères correspondant à un GUID.
 - Champ peuplé par la solution logicielle Vitam.
 - Cardinalité : 1-1
 - « **OriginatingAgency** » : identifiant du service producteur
 - Correspond à une valeur valide du champ « Identifier » de la collection Agencies.
 - Il s'agit d'une chaîne de caractères.
 - Cardinalité : 1-1
 - « **Paths** » : tableau des chemins d'héritage
 - Il s'agit d'un tableau de chaîne de caractères.
 - Cardinalité : 1-n
 - « **PropertyName** » : Nom d'une propriété
 - Il s'agit d'une chaîne de caractères.
 - Cardinalité : 1-1
 - « **PropertyValue** » : Valeur de la propriété nommé ci-dessus.
 - Le type varie en fonction de la propriété héritée.
 - Cardinalité : 1 -1
 - une liste de catégories de règles de gestion appliquées à cette unité archivistique. Les catégories pouvant être incluses dans cet objet sont, exhaustivement :
 - AccessRule (délai de communicabilité)

- AppraisalRule (durée d'utilité administrative)
- ClassificationRule (durée de classification)
- DisseminationRule (durée de diffusion)
- ReuseRule (durée de réutilisation)
- StorageRule (durée d'utilité courante)

Cardinalité : 0-1, pour chaque catégorie.

Chaque catégorie peut contenir :

- Rules : tableau de règle de gestion.
 - Il s'agit d'un tableau d'objets
 - Cardinalité : 0-1
 - Chacun des objets de ce tableau est elle-même composée de plusieurs informations :
 - « **UnitId** » : identifiant unique de l'unité archivistique.
 - Il s'agit d'une chaîne de 36 caractères correspondant à un GUID.
 - Champ peuplé par la solution logicielle Vitam.
 - Cardinalité : 1-1
 - « **OriginatingAgency** » : identifiant du service producteur
 - Correspond à une valeur valide du champ « Identifier » de la collection Agencies.
 - Il s'agit d'une chaîne de caractères.
 - Cardinalité : 1-1
 - « **Paths** » : tableau des chemins d'héritage
 - Il s'agit d'un tableau de chaîne de caractères.
 - Cardinalité : 1-n
 - « **Rule** » : identifiant de la règle
 - Correspond à une valeur du champ RuleId de la collection FileRules.
 - Cardinalité : 0-1
 - « **StartDate** » : date de début du calcul de l'échéance.
 - Il s'agit d'une date.
 - Cette date est déclarée dans le message ArchiveTransfer ou ajoutée *a posteriori* par une modification de l'unité archivistique.
 - Cardinalité : 0-1
 - « **EndDate** » : date de fin d'application de la règle.
 - Il s'agit d'une date.
 - Cette valeur est issue d'un calcul réalisé par la solution logicielle Vitam. Celui-ci consiste en l'ajout du délai correspondant à la règle dans la collection FileRules à la valeur du champ startDate (EndDate = StartDate + Durée)
 - Cardinalité : 0-1
- Properties : tableau contenant une liste de propriété
 - Cardinalité : 1-n

- Chaque propriété contient l'ensemble suivant d'informations :
 - « **UnitId** » : identifiant unique de l'unité archivistique.
 - Il s'agit d'une chaîne de 36 caractères correspondant à un GUID.
 - Champ peuplé par la solution logicielle Vitam.
 - Cardinalité : 1-1
 - « **OriginatingAgency** » : identifiant du service producteur
 - Correspond à une valeur valide du champ « Identifier » de la collection Agencies.
 - Il s'agit d'une chaîne de caractères.
 - Cardinalité : 1-1
 - « **Paths** » : tableau des chemins d'héritage
 - Il s'agit d'un tableau de chaîne de caractères.
 - Cardinalité : 1-n
 - « **PropertyName** » : Nom d'une propriété
 - Il s'agit d'une chaîne de caractères.
 - Cardinalité : 1-1
 - « **PropertyValue** » : Valeur de la propriété nommé ci-dessus.
 - Le type varie en fonction de la propriété inhérite.
 - Cardinalité : 1 -1
 - « **_validComputedInheritedRules** » : validité des règles de gestion applicables sur ces unités archivistiques.
 - Il s'agit d'un booléen. Si la valeur est à « true », les règles de gestion applicables sur cette unité archivistique était encore d'actualité.
 - Cardinalité : 1-1

```

"inheritedRulesAPIOutput": {
  "GlobalProperties": [],
  "StorageRule": {
    "Rules": [],
    "Properties": []
  },
  "AppraisalRule": {
    "Rules": [],
    "Properties": [
      {
        "UnitId": "aeaqaaaabaheaypqaahnia1m6ht7jeqaaada",
        "OriginatingAgency": "RATP",
        "Paths": [
          [
            "aeaqaaaabaheaypqaahnia1m6ht7jeyaaadq",
            "aeaqaaaabaheaypqaahnia1m6ht7jeyaaacq",
            "aeaqaaaabaheaypqaahnia1m6ht7jeyaaaba",
            "aeaqaaaabaheaypqaahnia1m6ht7jeyaaafa",
            "aeaqaaaabaheaypqaahnia1m6ht7jeqaaada"
          ],
          [
            "aeaqaaaabaheaypqaahnia1m6ht7jeyaaadq",
            "aeaqaaaabaheaypqaahnia1m6ht7jeyaaacq",
            "aeaqaaaabaheaypqaahnia1m6ht7jeyaaaba",
            "aeaqaaaabaheaypqaahnia1m6ht7jeqaaaeq",
            "aeaqaaaabaheaypqaahnia1m6ht7jeqaaada"
          ]
        ]
      }
    ]
  }
}

```

```

        ],
        "PropertyName": "FinalAction",
        "PropertyValue": "Keep"
    }
]
},
"DisseminationRule": {
    "Rules": [
        {
            "UnitId": "aeaqaaaabaheaypqaahniaalm6ht7jeyaaaada",
            "OriginatingAgency": "RATP",
            "Paths": [
                [
                    "aeaqaaaabaheaypqaahniaalm6ht7jeyaaadq",
                    "aeaqaaaabaheaypqaahniaalm6ht7jeyaaacq",
                    "aeaqaaaabaheaypqaahniaalm6ht7jeyaaaba",
                    "aeaqaaaabaheaypqaahniaalm6ht7jeyaaafa",
                    "aeaqaaaabaheaypqaahniaalm6ht7jeyaaaada"
                ],
                [
                    "aeaqaaaabaheaypqaahniaalm6ht7jeyaaadq",
                    "aeaqaaaabaheaypqaahniaalm6ht7jeyaaacq",
                    "aeaqaaaabaheaypqaahniaalm6ht7jeyaaaba",
                    "aeaqaaaabaheaypqaahniaalm6ht7jeyaaaecq",
                    "aeaqaaaabaheaypqaahniaalm6ht7jeyaaaada"
                ]
            ],
            "Rule": "DIS-00001",
            "StartDate": "2000-01-01",
            "EndDate": "2025-01-01"
        }
    ],
    "Properties": []
},
"ReuseRule": {
    "Rules": [],
    "Properties": []
},
"ClassificationRule": {
    "Rules": [],
    "Properties": []
},
"AccessRule": {
    "Rules": [
        {
            "UnitId": "aeaqaaaabaheaypqaahniaalm6ht7jeyaaaba",
            "OriginatingAgency": "RATP",
            "Paths": [
                [
                    "aeaqaaaabaheaypqaahniaalm6ht7jeyaaadq",
                    "aeaqaaaabaheaypqaahniaalm6ht7jeyaaacq",
                    "aeaqaaaabaheaypqaahniaalm6ht7jeyaaaba"
                ]
            ],
            "Rule": "ACC-00003",
            "StartDate": "2002-01-01",
            "EndDate": "2027-01-01"
        },
        {
            "UnitId": "aeaqaaaabaheaypqaahniaalm6ht7jeyaaafa",
            "OriginatingAgency": "RATP",
            "Paths": [
                [
                    "aeaqaaaabaheaypqaahniaalm6ht7jeyaaadq",
                    "aeaqaaaabaheaypqaahniaalm6ht7jeyaaacq",
                    "aeaqaaaabaheaypqaahniaalm6ht7jeyaaaba",
                    "aeaqaaaabaheaypqaahniaalm6ht7jeyaaafa"
                ]
            ]
        }
    ]
}

```

```
    ],
    "Rule": "ACC-00001",
    "StartDate": "2000-01-01",
    "EndDate": "2000-01-01"
  },
  {
    "UnitId": "aeaqaabaheaypqaahniaalm6ht7jeyaaaeq",
    "OriginatingAgency": "RATP",
    "Paths": [
      [
        "aeaqaabaheaypqaahniaalm6ht7jeyaaadq",
        "aeaqaabaheaypqaahniaalm6ht7jeyaaacq",
        "aeaqaabaheaypqaahniaalm6ht7jeyaaaba",
        "aeaqaabaheaypqaahniaalm6ht7jeyaaaeq"
      ]
    ],
    "Rule": "ACC-00036",
    "StartDate": "2000-01-01",
    "EndDate": "2999-01-01"
  }
],
"Properties": []
},
"indexationDate": "2019-09-02"
},
"_validComputedInheritedRules": true
}
```

« **_opts** » (**operations de transfert**): tableau contenant les identifiants d'opérations de transfert auxquelles cette unité archivistique est associée.

- Il s'agit d'un tableau contenant une à plusieurs chaînes de 36 caractères correspondant au champ `_id` de l'opération ou GUID de l'opération, enregistré dans la collection `LogbookOperation`.
- Champ peuplé par la solution logicielle Vitam au moment d'une indexation réalisée lors d'une phase de transfert.
- Ne peut être vide.
- Cardinalité : 0-1

4.2. Collection ObjectGroup

4.2.1. Utilisation de la collection ObjectGroup

La collection ObjectGroup contient les informations relatives aux groupes d'objets.

4.2.2. Exemple de XML

Ci-après, un extrait d'un bordereau de transfert (manifest.xml) utilisé pour compléter les champs du JSON correspondant à un groupe d'objets.

```
<DataObjectGroupId id="ID0009">
  <PhysicalDataObject id="ID109">
    <DataObjectVersion>PhysicalMaster</DataObjectVersion>
    <PhysicalId>1 Num 1/191-3</PhysicalId>
    <PhysicalDimensions>
      <Height unit="centimetre">10.5</Height>
      <Length unit="centimetre">14.8</Length>
      <Thickness unit="micrometre">350</Thickness>
      <Weight unit="gram">3</Weight>
    </PhysicalDimensions>
    <Extent>1 carte imprimée</Extent>
    <Dimensions>10,5cm x 14,8cm</Dimensions>
    <Color>Noir et blanc</Color>
    <Framing>Paysage</Framing>
    <Technique>Phototypie</Technique>
  </PhysicalDataObject>
  <BinaryDataObject id="ID9">
    <DataObjectVersion>BinaryMaster</DataObjectVersion>
    <Uri>Content/1NUM_9.JPG</Uri>
    <MessageDigest algorithm="SHA-
512">0e0cec05a1d72ee5610eaa5afbc904c012d190037cbc827d08272102cdecf0226efcad122b86e7
699f767c661c9f3702379b8c2cb01c4f492f69deb200661bb9</MessageDigest>
    <Size>7702</Size>
    <FormatIdentification>
      <FormatLiteral>JPEG File Interchange Format</FormatLiteral>
      <MimeType>image/jpeg</MimeType>
      <FormatId>fmt/43</FormatId>
    </FormatIdentification>
    <FileInfo>
      <Filename>1NUM_9.JPG</Filename>
    </FileInfo>
    <Metadata>
      <Image>
        <Dimensions>117x76</Dimensions>
        <Width>117px</Width>
        <Height>76px</Height>
        <VerticalResolution>96ppp</VerticalResolution>
        <HorizontalResolution>96ppp</HorizontalResolution>
        <ColorDepth>24</ColorDepth>
      </Image>
    </Metadata>
  </BinaryDataObject>
</DataObjectGroupId>
```

4.2.3. Exemple de JSON stocké en base

Les champs présentés dans l'exemple ci-après ne font pas état de l'exhaustivité des champs disponibles dans le SEDA. Ceux-ci sont référencés dans la documentation SEDA disponible au lien suivant : https://redirect.francearchives.fr/seda/api_v2-1/seda-2.1-main.html

```
{
```

```

    "_id": "aebaaaaaaafgsz3wabcugak7ube6dxyaaabq",
    "_tenant": 0,
    "_profil": "Image",
    "FileInfo": {
      "Filename": "1NUM_9.JPG"
    },
    "_qualifiers": [
      {
        "qualifier": "PhysicalMaster",
        "_nbc": 1,
        "versions": [
          {
            "_id": "aeaaaaaaafgsz3wabcugak7ube6dzqaaaca",
            "DataObjectGroupId": "aebaaaaaaafgsz3wabcugak7ube6dxyaaabq",
            "DataObjectVersion": "PhysicalMaster_1",
            "PhysicalId": "1 Num 1/191-3",
            "PhysicalDimensions": {
              "Height": {
                "unit": "centimetre",
                "dValue": 10.5
              },
              "Length": {
                "unit": "centimetre",
                "dValue": 14.8
              },
              "Thickness": {
                "unit": "micrometre",
                "dValue": 350
              },
              "Weight": {
                "unit": "gram",
                "dValue": 3
              }
            },
            "Extent": "1 carte imprimée",
            "Dimensions": "10,5cm x 14,8cm",
            "Color": "Noir et blanc",
            "Framing": "Paysage",
            "Technique": "Phototypie",
            "_opi": "aeaaaaaashi422cab3gyalenej2kcyaaaaq"
          }
        ]
      },
      {
        "qualifier": "BinaryMaster",
        "_nbc": 1,
        "versions": [
          {
            "_id": "aeaaaaaaafgsz3wabcugak7ube6dxyaaaba",
            "DataObjectGroupId": "aebaaaaaaafgsz3wabcugak7ube6dxyaaabq",
            "DataObjectVersion": "BinaryMaster_1",
            "FormatIdentification": {
              "FormatLitteral": "JPEG File Interchange Format",
              "MimeType": "image/jpeg",
              "FormatId": "fmt/43"
            },
            "FileInfo": {
              "Filename": "1NUM_9.JPG"
            },
            "Metadata": {
              "Image": {
                "Dimensions": "117x76",
                "Width": "117px",
                "Height": "76px",
                "VerticalResolution": "96ppp",
                "HorizontalResolution": "96ppp",
                "ColorDepth": 24
              }
            }
          }
        ]
      }
    ]
  }
}

```

```

    }
    },
    "_opi": "aeaaaaaashi422cab3gyalenej2kcyaaaaq",
    "Size": 7702,
    "Uri": "Content/1NUM_9.JPG",
    "MessageDigest":
"0e0cec05a1d72ee5610eaa5afbc904c012d190037cbc827d08272102cdecf0226efcad122b86e7699f
767c661c9f3702379b8c2cb01c4f492f69deb200661bb9",
    "Algorithm": "SHA-512",
    "_storage": {
      "strategyId": "default"
    }
  }
]
}
],
"_up": [
  "aeaqaaaaafgsz3wabcugak7ube6d4qaaaaq"
],
"_nbc": 0,
"_ops": [
  "aedqaaaaachxqyktaai4aak7ube557iaaaaq"
],
"_opi": "aedqaaaaachxqyktaai4aak7ube557iaaaaq",
"_sp": "Vitam",
"_sps": [
  "Vitam"
],
"_storage": {
  "strategyId": "default"
},
"_v": 1,
"_av": 1,
"_glpd": "2018-07-05T13:55:39.779",
"_us": [
  "aeaqaaaaahducypaasryallvnrahwaaaabq",
  "aeaqaaaaahducypaasryallvnrai5aaaaaq"
]
}

```

4.2.4. Détail des champs du JSON

« **_id** » : identifiant du groupe d'objets.

- Il s'agit d'une chaîne de 36 caractères correspondant à un GUID.
- Champ peuplé par la solution logicielle Vitam.
- Cardinalité : 1-1

« **_tenant** » : identifiant du tenant.

- Il s'agit d'un entier.
- Champ peuplé par la solution logicielle Vitam.
- Cardinalité : 1-1

« **_profil** » : catégorie de l'objet.

- Repris du nom de la balise présente dans le bloc Metadata du DataObjectPackage présent dans le bordereau de transfert au niveau du BinaryMaster. Attention, il s'agit d'une reprise de la balise et non pas des valeurs à l'intérieur.
- Les valeurs possibles pour ce champ sont : Audio, Document, Text, Image et Video. Des

extensions seront possibles (Database, Plan3D, ...).

- Peut être vide.
- Cardinalité : 1-1

« **FileInfo** » : informations sur le fichier constituant l'objet-données numérique de référence.

- Reprend le bloc FileInfo du BinaryMaster présent dans le bordereau de transfert.
- L'objet de ce bloc est de pouvoir conserver les informations initiales du premier BinaryMaster pour en faciliter la recherche et augmenter la qualité des résultats en cas de recherche multi critères ne portant que sur les BinaryMaster.
- Peut être vide ou contenir la valeur « null ».
- Cardinalité : 1-1

« **_qualifiers** » : tableau de structures décrivant les objets inclus dans ce groupe d'objets.

- Cardinalité : 1-1
- Une structure est composée comme suit :
 - « **qualifier** » : usage de l'objet.
 - Correspond à la valeur contenue dans le champ <DataObjectVersion> du bordereau de transfert. Par exemple pour <DataObjectVersion>BinaryMaster_1</DataObjectVersion>, c'est la valeur « BinaryMaster » qui est reportée.
 - Cardinalité : 1-1
 - « **_nbc** » : nombre d'objets correspondant à cet usage.
 - Il s'agit d'un entier.
 - Champ peuplé par la solution logicielle Vitam.
 - Cardinalité : 1-1
 - « **versions** » : tableau des objets par version (une version = une entrée dans le tableau).
 - « **_id** » : identifiant de l'objet.
 - Il s'agit d'une chaîne de 36 caractères correspondant à un GUID, généré par la solution logicielle Vitam.
 - Cardinalité : 1-n
 - « **DataObjectGroupId** » : identifiant du groupe d'objets
 - Il s'agit d'une chaîne de 36 caractères correspondant à un GUID, généré par la solution logicielle Vitam.
 - Cardinalité : 1-1
 - « **DataObjectVersion** » : version de l'objet par rapport à son usage.
 - Il s'agit d'une chaîne de caractères.
 - Par exemple, si on a *BinaryMaster* sur l'usage, on aura au moins un objet *BinaryMaster_1*. Ces champs sont renseignés si possible avec les valeurs récupérées dans les balises <DataObjectVersion> du bordereau de transfert. Chaque ajout d'un objet du même usage incrémente de un le numéro de la version, même si le bordereau de transfert indique une information contraire. Par exemple s'il existe un groupe d'objets avec deux objets : *BinaryMaster_1* et *BinaryMaster_2*, lorsqu'un nouveau SIP ajoute un objet déclaré comme un « *BinaryMaster_6* » dans le bordereau de transfert, celui-ci sera enregistré comme « *BinaryMaster_3* ».

- Cardinalité : 1-1
- « **FormatIdentification** » : contient trois champs qui permettent d'identifier le format du fichier.
 - Une vérification de la cohérence entre ce qui est déclaré dans le XML, ce qui existe dans le référentiel PRONOM et les valeurs que porte le document est faite.
 - Cardinalité : 1-1
 - Cet objet contient les champs suivants :
 - « **FormatLitteral** » : nom du format.
 - C'est une reprise de la valeur située entre les balises <FormatLitteral> du message ArchiveTransfer.
 - Cardinalité : 1-1
 - « **MimeType** » : type Mime.
 - C'est une reprise de la valeur située entre les balises <MimeType> du message ArchiveTransfer ou des valeurs correspondant au format tel qu'identifié par la solution logicielle Vitam.
 - Cardinalité : 1-1
 - « **FormatId** » : PUID du format de l'objet.
 - Il est défini par la solution logicielle Vitam à l'aide du référentiel PRONOM maintenu par The National Archives (UK) et correspondant à la valeur du champ PUID de la collection FileFormat.
 - Cardinalité : 1-1
- « **FileInfo** » : contient les informations sur le fichier.
 - « **Filename** » : nom de l'objet.
 - Ce champ est renseigné avec la métadonnée correspondante portée par le message ArchiveTransfer.
 - Cardinalité : 0-1
 - « **CreatingApplicationName** » : nom de l'application avec laquelle l'objet a été créé.
 - Ce champ est renseigné avec la métadonnée correspondante portée par le message ArchiveTransfer.
 - Cardinalité : 0-1
 - « **CreatingApplicationVersion** » : numéro de version de l'application avec laquelle le document a été créé.
 - Ce champ est renseigné avec la métadonnée correspondante portée par le message ArchiveTransfer.
 - Cardinalité : 0-1
 - « **CreatingOs** » : système d'exploitation avec lequel l'objet a été créé.
 - Ce champ est renseigné avec la métadonnée correspondante portée par le message ArchiveTransfer.
 - Cardinalité : 0-1
 - « **CreatingOsVersion** » : Version du système d'exploitation avec lequel l'objet a été créé.
 - Ce champ est renseigné avec la métadonnée correspondante portée par le message ArchiveTransfer.

- Cardinalité : 0-1
- « **LastModified** » : date de dernière modification de l'objet.
 - Il s'agit d'une date au format ISO 8601 YYYY-MM-DD + "T" + hh:mm:ss.millisecondes « + » timezone hh:mm.
 - Exemple : 2016-08-19T16:36:07.942+02:00
 - Ce champ est optionnel, et est renseigné avec la métadonnée correspondante portée par le fichier.
 - Cardinalité : 0-1
- « **OtherMetadata** » : autres métadonnées techniques, non référencées dans le SEDA 2.1.
 - Il s'agit d'un objet.
 - Ce champ correspond à la balise <OtherMetadata>, extension du schéma SEDA du message ArchiveTransfer.
 - Cardinalité 0-1.
 - Il s'agit d'un objet, pouvant contenir :
 - des champs issus du bordereau de transfert au niveau du bloc BinaryMaster.
 - Cardinalité : 0-n
 - des champs enregistrés sous forme de tableaux, issus d'une opération d'extraction de métadonnées.
 - Cardinalité : 1-n
 - un champ « RawMetadata », correspondant à un tableau contenant l'ensemble des métadonnées issues d'une opération d'extraction de métadonnées.
 - Cardinalité : 0-1

```
"OtherMetadata": {
    "geometry": [
        {
            "width": [
                "800"
            ],
            "RawMetadata": [
                "[{ \"image\": { \"geometry\": { \"width\": 800}}]"
            ]
        }
    ]
}
```

- « **_opi** » : identifiant de l'opération à l'origine de la création de cet objet.
 - Il s'agit d'une chaîne de 36 caractères correspondant au GUID contenu dans le champ `_id` de la collection `LogbookOperation`.
 - Champ peuplé par la solution logicielle Vitam.
 - Ne peut être vide
 - Cardinalité : 1-1
- « **Size** » : taille de l'objet (en octet).
 - Il s'agit d'un entier.

- Cardinalité : 0 (objet physique) ou 1 (objet binaire)
- « **PhysicalDimensions** » : contient les différentes informations concernant un objet physique (DataObjectVersion = PhysicalMaster). Il pourra donner des informations sur la taille, le poids, etc... de l'objet.
 - Cardinalité : 0 (objet physique) ou 1 (objet binaire)
 - Il comprend les champs suivants :
 - « **Width** » : largeur de l'objet. Ce champ contient 2 sous champs : « unit » (string) et « dValue » (double)
 - « **Height** » : hauteur de l'objet. Ce champ contient 2 sous champs : « unit » (string) et « dValue » (double)
 - « **Depth** » : profondeur de l'objet. Ce champ contient 2 sous champs : « unit » (string) et « dValue » (double)
 - « **Diameter** » : diamètre de l'objet. Ce champ contient 2 sous champs : « unit » (string) et « dValue » (double)
 - « **Length** » : longueur de l'objet. Ce champ contient 2 sous champs : « unit » (string) et « dValue » (double)
 - « **Thickness** » : épaisseur de l'objet. Ce champ contient 2 sous champs : « unit » (string) et « dValue » (double)
 - « **Weight** » : poids de l'objet. Ce champ contient 2 sous champs : « unit » (string) et « dValue » (double)
 - « **Shape** » : forme de l'objet. Ce champ contient une chaîne de caractères.
- « **Uri** » : localisation du fichier correspondant à l'objet dans le SIP.
 - Chaîne de caractères
 - Cardinalité : 0 (objet physique) ou 1 (objet binaire)
- « **MessageDigest** » : empreinte du fichier correspondant à l'objet.
 - Il s'agit d'une chaîne de caractères.
 - La valeur est calculée par la solution logicielle Vitam.
 - Cardinalité : 0 (objet physique) ou 1 (objet binaire)
- « **Algorithm** » : algorithme utilisé pour réaliser l'empreinte du fichier correspondant à l'objet.
 - Chaîne de caractères
 - Cardinalité 0 (objet physique) ou 1 (objet binaire)
- « **_storage** » : contient les champs qui permettent d'identifier les offres de stockage.
 - Ne peut être vide.
 - Cardinalité : 0 (objet physique) ou 1 (objet binaire)
 - Il s'agit d'un objet constitué des champs suivants :
 - « **strategyId** » : identifiant de la stratégie de stockage.
 - Il s'agit d'une chaîne de caractère.

« **_up** » (**unit up**): tableau identifiant les unités archivistiques représentées par ce groupe d'objets.

- Il s'agit d'un tableau de chaînes de 36 caractères correspondant au GUID contenu dans le champ `_id` des unités archivistiques enregistrées dans la collection Unit.
- Champ peuplé par la solution logicielle Vitam.
- Ne peut être vide

- Cardinalité : 1-1

« **_nbc** » : nombre d'objets dans le groupe d'objets.

- Il s'agit d'un entier.
- Champ peuplé par la solution logicielle Vitam.
- Cardinalité : 1-1

« **_ops** » (**operations**): tableau des identifiants d'opérations auxquelles ce groupe d'objets a participé.

- Il s'agit d'un tableau de chaînes de 36 caractères correspondant au GUID contenu dans le champ `_id` d'opération enregistré dans la collection `LogBookOperation`.
- Champ peuplé par la solution logicielle Vitam.
- Ne peut être vide
- Cardinalité : 1-1

« **_opi** » : identifiant de l'opération à l'origine de la création de ce groupe d'objets.

- Il s'agit d'une chaîne de 36 caractères correspondant au GUID contenu dans le champ `_id` de la collection `LogbookOperation`.
- Champ peuplé par la solution logicielle Vitam.
- Ne peut être vide
- Cardinalité : 1-1

« **_sp** » : service producteur responsable du groupe d'objets, qui appartient à son fond propre. Il s'agit de la valeur de la balise `OriginatingAgencyIdentifier` dans le message `ArchiveTransfer`.

- Il s'agit d'une chaîne de caractères.
- Correspond à une valeur valide du champ « Identifier » de la collection `Agencies`.
- Champ peuplé par la solution logicielle Vitam.
- Cardinalité : 1-1

« **_sps** » : services producteurs auxquels le groupe d'objets techniques a été rattaché (au titre de leurs fonds symboliques).

- Il s'agit d'un tableau contenant tous les services producteurs référençant le groupe d'objets.
- Il s'agit d'un tableau de chaînes de caractères.
- Correspond à une valeur valide du champ « Identifier » de la collection `Agencies`.
- Champ peuplé par la solution logicielle Vitam.
- Ne peut être vide
- Cardinalité : 1-1

« **_storage** » : contient trois champs qui permettent d'identifier les offres de stockage.

- Ne peut être vide.
- Cardinalité : 0 (objet physique) ou 1 (objet binaire)
- Il s'agit d'un objet constitué des champs suivants :
 - « **strategyId** » : identifiant de la stratégie de stockage.
 - Il s'agit d'une chaîne de caractères.
 - « **offerIds** » : liste des offres de stockage pour une stratégie donnée
 - Il s'agit d'un tableau.

- « **_nbc** » : nombre d’offres.
 - Il s’agit d’un entier.

« **_v** » : version de l’enregistrement décrit.

- Il s’agit d’un entier.
- 0 correspond à l’enregistrement d’origine. Si le numéro est supérieur à 0, alors il s’agit du numéro de version de l’enregistrement.
- Champ peuplé par la solution logicielle Vitam.
- Cardinalité : 1-1

« **_av** » : version atomique de l’enregistrement décrit, incrémentée automatiquement en cas de modification de tout champ de la collection.

- Il s’agit d’un entier.
- Champ peuplé par la solution logicielle Vitam.
- 0 correspond à l’enregistrement d’origine. Si le numéro est supérieur à 0, alors il s’agit du numéro de version de l’enregistrement.
- Cardinalité : 1-1

« **_glpd** » : Date de la dernière modification du graph dont l’objet dépend

- Il s’agit d’une date au format ISO 8601 YYYY-MM-DD + “T” + hh:mm:ss.millisecondes « + » timezone hh:mm.

Exemple : 2016-08-19T16:36:07.942+02:00.

- Champ peuplé par la solution logicielle Vitam.
- Cardinalité : 1-1

« **_us** » : Reprend l’union de tous les champs **_us** de toutes les unités archivistiques possédant le groupe d’objets (parentalité).

- Tableau de chaînes de 36 caractères.
- Champ peuplé par la solution logicielle Vitam.
- Cardinalité : 1-1

4.3. Collection Offset

4.3.1. Utilisation de la collection

Cette collection permet de persister les offsets des dernières données reconstruites des offres de stockage lors de la reconstruction au fil de l'eau pour les collections :

- LogbookOperation
- Unit
- ObjetGroup
- UNIT_GRAPH
- OBJETGROUPE_GRAPH

Il y a une valeur d'offset par couple tenant/collection.

4.3.2. Exemple de JSON stocké en base comprenant l'exhaustivité des champs

```
{
  "_id": ObjectId("507f191e810c19729de860ea"),
  "offset": 1357,
  "collection": "UNIT",
  "strategyId": "default",
  "_tenant": 1
}
```

4.3.3. Détail des champs

« **_id** » : identifiant unique mongo.

- Il s'agit d'un champ de type mongo : ObjectId(<hexadecimal>).
- Champ peuplé par la solution logicielle Vitam.
- Cardinalité : 1-1

« **offset** » : la valeur de l'offset.

- Il s'agit d'un entier encodé 64 bits.
- Champ peuplé par la solution logicielle Vitam.
- Cardinalité : 1-1

« **collection** » : collection impactée.

- les valeurs possibles sont *UNIT* et *OBJECTGROUPE*.

« **strategyId** » : identifiant de la stratégie de stockage.

- Il s'agit d'une chaîne de caractère.

« **_tenant** » : identifiant du tenant.

- Il s'agit d'un entier.
- Champ peuplé par la solution logicielle Vitam.
- Cardinalité : 1-1

5. Base MasterData

La base Masterdata contient les collections relatives aux référentiels utilisés par la solution logicielle Vitam. Ceux-ci sont :

- AccessContract
- AccessionRegisterDetail
- AccessionRegisterSummary
- AccessionRegisterSymbolic
- Agencies
- ArchiveUnitProfile
- Context
- FileFormat
- FileRules
- Griffin
- IngestContract
- ManagementContract
- Ontology
- PreservationScenario
- Profile
- SecurityProfile
- VitamSequence

Certaines collections sont enregistrées sur un tenant et utilisables pour tous les tenants. Elles sont qualifiées de « Cross-tenant ». Il s’agit des collections suivantes :

- Context
- FileFormat
- Griffin
- Ontology
- SecurityProfile

Elles sont enregistrées sur le tenant d’administration.

5.1. Collection AccessContract

5.1.1. Utilisation de la collection AccessContract

La collection AccessContract permet de référencer et de décrire unitairement les contrats d’accès.

5.1.2. Exemple d’un fichier d’import de contrat d’accès

Les contrats d’accès sont importés dans la solution logicielle Vitam sous la forme d’un fichier JSON.

```
[
  {
    "Name": "ContratTNR",
```

```

    "Identifiant": "AC-000034",
    "Description": "Contrat permettant de faire des opérations pour tous les
services producteurs et sur tous les usages",
    "Status": "ACTIVE",
    "CreationDate": "2016-12-10T00:00:00.000",
    "LastUpdate": "2017-11-07T07:57:10.581",
    "ActivationDate": "2016-12-10T00:00:00.000",
    "DeactivationDate": "2016-12-10T00:00:00.000",
    "DataObjectVersion": [
      "PhysicalMaster",
      "BinaryMaster",
      "Dissemination",
      "Thumbnail",
      "TextContent"
    ],
    "WritingPermission": true,
    "EveryOriginatingAgency": true,
    "EveryDataObjectVersion": false
  }
]

```

Les champs à renseigner obligatoirement à la création d'un contrat sont :

- Name
- Identifiant (selon la configuration du tenant : Identifiant n'est obligatoire que si l'identifiant du contrat d'accès n'est pas généré par la solution logicielle Vitam)

Un fichier d'import peut décrire plusieurs contrats.

5.1.3. Exemple de JSON stocké en base comprenant l'exhaustivité des champs de la collection AccesContract

```

{
  "_id": "aefqaaaabahcd5ayaabdaaljtkulb2aaaaaq",
  "Name": "Contrat d'accès - Vitam",
  "Identifiant": "AC-000001",
  "Description": "Contrat permettant de faire des opérations pour tous les services
producteurs et sur tous les usages",
  "Status": "ACTIVE",
  "CreationDate": "1976-02-12T00:00:00.000",
  "LastUpdate": "2019-04-09T12:16:19.997",
  "ActivationDate": "2019-03-20T10:32:20.451",
  "WritingPermission": true,
  "WritingRestrictedDesc": true,
  "EveryOriginatingAgency": false,
  "EveryDataObjectVersion": false,
  "AccessLog": "ACTIVE",
  "_tenant": 8,
  "_v": 6,
  "OriginatingAgencies": [
    "FRAN_NP_051587_elim"
  ],
  "DataObjectVersion": [
    "BinaryMaster",
    "Thumbnail"
  ],
  "RootUnits": [
    "aeaqaaaabahf4qxrab4cialj7trrxzaaaaaaq"
  ],
  "ExcludedRootUnits": [
    "aeaqaaaabahf4qxrab4cialj7trrxzaaaaaaq"
  ]
}

```

5.1.4. Détail des champs

« **_id** » : identifiant unique du contrat pour un tenant donné.

- Il s'agit d'une chaîne de 36 caractères correspondant à un GUID.
- Champ peuplé par la solution logicielle Vitam.
- Cardinalité : 1-1

« **Name** » : nom du contrat d'accès.

- Il s'agit d'une chaîne de caractères.
- Cardinalité : 1-1

« **Identifiant** » : identifiant signifiant donné au contrat.

- Il est constitué du préfixe « AC- » suivi d'une suite de 6 chiffres s'il est peuplé par la solution logicielle Vitam. Par exemple : AC-001223. Si le référentiel est en position esclave, cet identifiant peut être géré par l'application à l'origine du contrat et est unique sur le tenant.
- Il s'agit d'une chaîne de caractères.
- Cardinalité : 1-1

« **Description** » : description du contrat d'accès.

- Il s'agit d'une chaîne de caractères.
- Cardinalité : 0-1

« **Status** » : statut du contrat.

- Il s'agit d'une chaîne de caractères.
- Peut avoir comme valeur : « ACTIVE » ou « INACTIVE ».
- Cardinalité : 1-1

« **CreationDate** » : date de création du contrat.

- La date est au format ISO 8601 et prend la forme suivante :

"CreationDate": "2017-04-10T11:30:33.798"

- Champ peuplé par la solution logicielle Vitam.
- Cardinalité : 1-1

« **LastUpdate** » : date de dernière mise à jour du contrat

- La date est au format ISO 8601 et prend la forme suivante :

"LastUpdate": "2017-04-10T11:30:33.798"

- Champ peuplé par la solution logicielle Vitam.
- Cardinalité : 1-1

« **ActivationDate** » : date d'activation du contrat.

- La date est au format ISO 8601 et prend la forme suivante :

"ActivationDate": "2017-04-10T11:30:33.798"

- Champ peuplé par la solution logicielle Vitam.
- Cardinalité : 1-1

« **DeactivationDate** » : date de désactivation du contrat.

- La date est au format ISO 8601 et prend la forme suivante :

"DeactivationDate": "2017-04-10T11:30:33.798"

- Champ peuplé par la solution logicielle Vitam.
- Cardinalité : 1-1

« **DataObjectVersion** » : type d'usages des groupes d'objets auxquels le détenteur du contrat a accès.

- Il s'agit d'un tableau de chaînes de caractères.
- Peut avoir comme valeur : « PhysicalMaster », « BinaryMaster », « Dissemination », « Thumbnail », « TextContent ».
- Peut être vide.
- Cardinalité : 0-1

« **OriginatingAgencies** » : services producteurs dont le détenteur du contrat peut consulter les archives.

- Il s'agit d'un tableau de chaînes de caractères.
- Correspond à une valeur valide du champ « Identifier » de la collection Agencies.
- Peut être vide.
- Cardinalité : 0-1

« **WritingPermission** » : droit d'écriture.

- Il s'agit d'un booléen. Si la valeur est à « true », le détenteur du contrat peut effectuer des mises à jour.
- Cardinalité : 1-1

« **WritingRestrictedDesc** » : droit de modification des métadonnées descriptives seulement.

- Il s'agit d'un booléen. Si la valeur est à « true », le détenteur du contrat peut effectuer des mises à jour seulement sur les métadonnées descriptives. Si la valeur est à « false », le détenteur du contrat peut effectuer des mises à jour sur les métadonnées descriptives, ainsi que sur les métadonnées de gestion.
- Cardinalité : 1-1

« **EveryOriginatingAgency** » : droit de consultation sur tous les services producteurs.

- Il s'agit d'un booléen.
- Si la valeur est à « true », alors le détenteur du contrat peut accéder aux archives de tous les services producteurs.
- Cardinalité : 1-1

« **EveryDataObjectVersion** » : droit de consultation sur tous les usages.

- Il s'agit d'un booléen.
- Si la valeur est à « true », alors le détenteur du contrat peut accéder à tous les types d'usages.
- Cardinalité : 1-1

« **AccessLog** » : enregistrement des accès.

- Il s'agit d'une chaîne de caractères.
- Peut avoir comme valeur : « ACTIVE » ou « INACTIVE »
- Si la valeur est à « ACTIVE », alors les téléchargements des objets sont enregistrés dans un fichier de log

- Cardinalité : 1-1

« **_tenant** » : identifiant du tenant.

- Il s'agit d'un entier.
- Cardinalité : 1-1

« **_v** » : version de l'enregistrement décrit

- Il s'agit d'un entier.
- Champ peuplé par la solution logicielle Vitam.
- Cardinalité : 1-1
- 0 correspond à l'enregistrement d'origine. Si le numéro est supérieur à 0, alors il s'agit du numéro de version de l'enregistrement.

« **RootUnits** » : Liste des nœuds de consultation auxquels le détenteur du contrat a accès.

- Si aucun nœud n'est spécifié, alors l'utilisateur a accès à tous les nœuds.
- Il s'agit d'un tableau de chaînes de caractères.
- Peut être vide.
- Cardinalité : 0-1

« **ExcludedRootUnits** » : Liste des nœuds de consultation à partir desquels le détenteur du contrat n'a pas accès.

- Si aucun nœud n'est spécifié, alors l'utilisateur a accès à tous les nœuds.
- Il s'agit d'un tableau de chaînes de caractères.
- Peut être vide.
- Cardinalité : 0-1

« **RuleCategoryToFilter** » : Liste de catégories de règles pour lesquelles le détenteur du contrat n'a pas accès si ces règles ne sont pas échues.

- Si aucune catégorie de règle n'est spécifiée, alors l'utilisateur a accès à toutes les archives, que leurs règles de gestion soient échues ou non.
- Il s'agit d'un tableau de chaînes de caractères.
- Peut être vide.
- Cardinalité : 0-1

5.2. Collection AccessionRegisterDetail

5.2.1. Utilisation de la collection AccessionRegisterDetail

Cette collection a pour vocation de référencer l'ensemble des informations sur les opérations d'entrée ou de préservation réalisées pour un service producteur. À ce jour, il y a autant d'enregistrements que d'opérations d'entrées effectuées pour ce service producteur, mais des évolutions sont d'ores et déjà prévues. Cette collection reprend les éléments du bordereau de transfert, ainsi que les éléments correspondant à des opérations d'élimination, de transfert ou de préservation.

5.2.2. Exemple de la description dans le XML d'entrée

Les seuls éléments issus du message ArchiveTransfer utilisés ici sont ceux correspondant à la

déclaration des identifiants du service producteur et du service versant. Ils sont placés dans le bloc <ManagementMetadata>

```
<ManagementMetadata>
  <OriginatingAgencyIdentifier>FRAN_NP_051314</OriginatingAgencyIdentifier>
  <SubmissionAgencyIdentifier>FRAN_NP_005761</SubmissionAgencyIdentifier>
</ManagementMetadata>
```

5.2.3. Exemple de JSON stocké en base comprenant l'exhaustivité des champs

```
{
  "_id": "aeheaaaaaa4haffe6ab7yialiplg6plqaaaaaq",
  "ObjectSize": {
    "ingested": 77256,
    "deleted": 77256,
    "remained": 0
  },
  "OriginatingAgency": "Vitam",
  "SubmissionAgency": "Vitam",
  "ArchivalAgreement": "IC-000001",
  "EndDate": "2019-01-23T13:02:21.102",
  "StartDate": "2019-01-23T13:02:21.102",
  "LastUpdate": "2019-01-23T13:12:24.200",
  "Status": "UNSTORED",
  "TotalObjectGroups": {
    "ingested": 2,
    "deleted": 2,
    "remained": 0
  },
  "TotalUnits": {
    "ingested": 3,
    "deleted": 3,
    "remained": 0
  },
  "TotalObjects": {
    "ingested": 2,
    "deleted": 2,
    "remained": 0
  },
  "Opc": "aeheaaaaaa6hfj4pcaaot2aliplg3kkiaaaaaaq",
  "Opi": "aeheaaaaaa6hfj4pcaaot2aliplg3kkiaaaaaaq",
  "OpType": "INGEST",
  "Events": [
    {
      "Opc": "aeheaaaaaa6hfj4pcaaot2aliplg3kkiaaaaaaq",
      "OpType": "INGEST",
      "Gots": 2,
      "Units": 3,
      "Objects": 2,
      "ObjSize": 77256,
      "CreationDate": "2019-01-23T13:02:21.102"
    },
    {
      "Opc": "aeheaaaaaa6hfj4pcaafkkalip11qjxiaaaaaaq",
      "OpType": "ELIMINATION",
      "Gots": -2,
      "Units": -3,
      "Objects": -2,
      "ObjSize": -77256,
      "CreationDate": "2019-01-23T13:12:24.200"
    }
  ],
  "OperationIds": [
    "aeheaaaaaa6hfj4pcaaot2aliplg3kkiaaaaaaq"
  ],
  "_tenant": 7,
```

```
"_v": 1  
}
```

5.2.4. Détail des champs

« **_id** » : identifiant unique.

- Il s'agit d'une chaîne de 36 caractères correspondant à un GUID.
- Champ peuplé par la solution logicielle Vitam.
- Cardinalité : 1-1

« **ObjectSize** » : Contient la répartition du volume total des fichiers du fonds par état pour l'opération journalisée (ingested, deleted et remained) :

- « ingested » : volume en octet des fichiers pris en charge dans le cadre de l'enregistrement concerné. La valeur contenue dans le champ est un entier.
- « deleted » : volume en octet des fichiers supprimés ou sortis du système pour l'enregistrement concerné. La valeur contenue dans ce champ est un entier.
- « remained » : volume en octet des fichiers conservés dans le système pour l'enregistrement concerné. La valeur contenue dans ce champ est un entier.
- Champ peuplé par la solution logicielle Vitam.
- Cardinalité : 1-1

« **OriginatingAgency** » : identifiant du service producteur.

- Il reprend la valeur du champ <OriginatingAgencyIdentifieur> du manifeste
- Correspond à une valeur valide du champ « Identifier » de la collection Agencies.

Par exemple :

```
<OriginatingAgencyIdentifieur>FRAN_NP_051314</OriginatingAgencyIdentifieur>
```

On récupère la valeur FRAN_NP_051314

- Il s'agit d'une chaîne de caractères.
- Cardinalité : 1-1

« **SubmissionAgency** » : contient l'identifiant du service versant.

- Il reprend la valeur du champ <SubmissionAgencyIdentifieur> du manifeste
- Correspond à une valeur valide du champ « Identifier » de la collection Agencies.

Par exemple

```
<SubmissionAgencyIdentifieur>FRAN_NP_005761</SubmissionAgencyIdentifieur>
```

On récupère la valeur FRAN_NP_005761.

- Il s'agit d'une chaîne de caractère.
- Cardinalité : 1-1

Ce champ est facultatif dans le bordereau. S'il est absent ou vide, alors la valeur contenue dans le champ <OriginatingAgencyIdentifieur> est reportée dans ce champ.

« **ArchivalAgreement** » :

- Contient le contrat d'entrée utilisé pour réaliser l'entrée.
- Il reprend la valeur du champ <ArchivalAgreement> du manifeste

- Il correspond à une valeur valide du champ « Identifier » de la collection IngestContract.

Par exemple pour

```
<ArchivalAgreement>IC-000001</ArchivalAgreement>
```

On récupère la valeur IC-000001.

- Il s'agit d'une chaîne de caractère.
- Cardinalité : 1-1

« **AcquisitionInformation** » :

- Contient les modalités d'entrée des archives
- Il reprend la valeur du champ <AcquisitionInformation> du manifeste
- Il s'agit d'une chaîne de caractères.
- Cardinalité : 0-1

« **LegalStatus** » :

- Contient le statut juridique des archives échangés
- Il reprend la valeur du champ <LegalStatus> du manifeste
- Cardinalité : 0-1

« **EndDate** » : date de la dernière opération d'entrée pour l'enregistrement concerné.

- La date est au format ISO 8601
- Champ peuplé par la solution logicielle Vitam.
- Cardinalité : 1-1

```
"EndDate": "2017-04-10T11:30:33.798"
```

« **StartDate** » : date de la première opération d'entrée pour l'enregistrement concerné.

- La date est au format ISO 8601
- Champ peuplé par la solution logicielle Vitam.
- Cardinalité : 1-1

```
"StartDate": "2017-04-10T11:30:33.798"
```

« **LastUpdate** » : Date de la dernière mise à jour pour l'enregistrement concerné.

- La date est au format ISO 8601
- Champ peuplé par la solution logicielle Vitam.
- Cardinalité : 1-1

```
"LastUpdate": "2017-04-10T11:30:33.798"
```

« **Status** » : état des archives concernées par l'enregistrement.

- Il s'agit d'une chaîne de caractères
- Peut avoir comme valeur : STORED_AND_COMPLETED, STORED_AND_UPDATED, UNSTORED
- Champ peuplé par Vitam.
- Cardinalité : 1-1

« **TotalObjectGroups** » : Contient la répartition du nombre de groupes d'objets du fonds par état pour l'opération journalisée (ingested, deleted et remained) :

- « ingested » : nombre de groupes d'objets pris en charge dans le cadre de l'enregistrement concerné. La valeur contenue dans le champ est un entier.
- « deleted » : nombre de groupes d'objets supprimés ou sortis du système pour l'enregistrement concerné. La valeur contenue dans ce champ est un entier.
- « remained » : nombre de groupes d'objets conservés dans le système pour l'enregistrement concerné. La valeur contenue dans ce champ est un entier.
- Champ peuplé par la solution logicielle Vitam.
- Cardinalité : 1-1

« **TotalUnits** » : Il contient la répartition du nombre d'unités archivistiques du fonds par état pour l'opération journalisée :

- « ingested » : nombre d'unités archivistiques prises en charge dans le cadre de l'enregistrement concerné. La valeur contenue dans le champ est un entier.
- « deleted » : nombre d'unités archivistiques supprimées ou sorties du système pour l'enregistrement concerné. La valeur contenue dans ce champ est un entier.
- « remained » : nombre d'unités archivistiques conservées dans le système pour l'enregistrement concerné. La valeur contenue dans ce champ est un entier.
- Champ peuplé par la solution logicielle Vitam.
- Cardinalité : 1-1

« **TotalObjects** » : Contient la répartition du nombre d'objets du fonds par état pour l'opération journalisée :

- « ingested » : nombre d'objets pris en charge dans le cadre de l'enregistrement concerné. La valeur contenue dans le champ est un entier.
- « deleted » : nombre d'objets supprimés ou sorties du système pour l'enregistrement concerné. La valeur contenue dans ce champ est un entier.
- « remained » : nombre d'objets conservés dans le système pour l'enregistrement concerné. La valeur contenue dans ce champ est un entier.
- Champ peuplé par la solution logicielle Vitam.
- Cardinalité : 1-1
- « **Opc** » : identifiant d'une opération courante
- Il s'agit d'une chaîne de 36 caractères correspondant à un GUID.
- Il s'agit soit de l'identifiant de l'opération ayant provoqué la prise en charge dans le système des archives recensées dans ce détail du registre des fonds, soit d'une opération ayant modifié le fonds d'une opération d'ingest. Exemple : l'opération de l'élimination
- Cardinalité 1-1

« **Opi** » : identifiant de l'opération d'entrée ayant provoqué la prise en charge dans le système des archives recensées dans ce détail du registre des fonds

- Il s'agit d'une chaîne de 36 caractères correspondant à un GUID.
- Dans le cas de SIP faisant des rattachements (par exemple une nouvelle unité archivistique à une unité archivistique existante), il s'agira toujours de l'identifiant de l'opération de l'entrée en cours (celle générant ces documents Mongo)
- Cardinalité 1-1

« **OpType** » : type d'opération ayant provoqué la création de l'enregistrement (INGEST, PRESERVATION...)

- Il s'agit d'une chaîne de caractères.
- Cardinalité 1-1

« **Events** » : les détails des registres des fonds ayant modifié un lot d'ingest existant.

- Le premier événement contient les remained de l'opération d'ingest.
- Les événements suivants concernent les opérations ayant modifié un lot d'ingest existant (Elimination, Transfer...)
- Cardinalité : 1-n

« **Events.Opc** » : identifiant de l'opération courante.

- Il s'agit d'une chaîne de 36 caractères correspondant à un GUID.
- Dans le cas d'un ingest, opc égale à l'id de l'opération d'ingest.
- Cardinalité : 1-1

« **Events.OpType** » : Le type de l'opération (INGEST, ELIMINATION, TRANSFER)

- Il s'agit d'une chaîne de caractères.
- Cardinalité : 1-1

« **Events.Gots** » : Nombre total de groupe d'objets impactés par l'opération de l'événement

- Il s'agit d'un entier.
- Cardinalité : 1-1

« **Events.Units** » : Nombre total d'unités archivistiques impactées par l'opération de l'événement

- Il s'agit d'un entier.
- Cardinalité : 1-1

« **Events.Objects** » : Nombre total d'objets impactés par l'opération de l'événement

- Il s'agit d'un entier.
- Cardinalité : 1-1

« **Events.ObjSize** » : Le poids total de tous les objets impactés par l'opération de l'événement.

- Il s'agit d'un entier.
- Dans le cas d'un ingest, opc égale à l'id de l'opération d'ingest.
- Cardinalité : 1-1

« **Events.CreationDate** » : La date de l'événement.

- La date est au format ISO 8601
- Cardinalité : 1-1

« **OperationIds** » : opérations d'entrée concernées

- Il s'agit d'un tableau.
- Ne peut être vide
- Champ peuplé par la solution logicielle Vitam.
- Cardinalité : 1-n

« **_tenant** » : correspondant à l'identifiant du tenant.

- Il s'agit d'une chaîne de caractères.
- Champ peuplé par la solution logicielle Vitam.

- **Cardinalité : 1-1**

« **_v** » : version de l'enregistrement décrit.

- Il s'agit d'un entier.
- Champ peuplé par la solution logicielle Vitam.
- 0 correspond à l'enregistrement d'origine. Si le numéro est supérieur à 0, alors il s'agit du numéro de version de l'enregistrement.
- Cardinalité : 1-1

5.3. Collection AccessionRegisterSummary

5.3.1. Utilisation de la collection

Cette collection contient une vue macroscopique des fonds pris en charge dans la solution logicielle Vitam. Chaque service producteur possède un et un seul document le concernant dans cette collection. Ce document est **calculé** à partir des données enregistrées dans la collection AccessionRegisterDetail pour ce service producteur.

5.3.2. Exemple de JSON stocké en base comprenant l'exhaustivité des champs

```
{
  "_id": "aefaaaaaaahlpvjiaablwalj7y632fyaaaaaq",
  "OriginatingAgency": "FRAN_NP_009913",
  "TotalObjects": {
    "ingested": 71,
    "deleted": 0,
    "remained": 71
  },
  "TotalObjectGroups": {
    "ingested": 68,
    "deleted": 0,
    "remained": 68
  },
  "TotalUnits": {
    "ingested": 205,
    "deleted": 0,
    "remained": 205
  },
  "ObjectSize": {
    "ingested": 2406907,
    "deleted": 0,
    "remained": 2406907
  },
  "CreationDate": "2019-04-08T18:37:32.823",
  "_v": 25,
  "_tenant": 0
}
```

5.3.3. Détail des champs

« **_id** » : identifiant unique du fonds.

- Il s'agit d'une chaîne de 36 caractères correspondant à un GUID.
- Champ peuplé par la solution logicielle Vitam.
- Cardinalité : 1-1

« **OriginatingAgency** » : identifiant d'un service producteur.

- la valeur de ce champ est une chaîne de caractères.
- Ce champ est la clef primaire pour un enregistrement dans le registre des fonds. Il permet l'agrégation de tous les documents de la collection AccessionRegisterDetail pour ce service producteur. Cette valeur correspond nécessairement à une valeur valide du champ « Identifier » de la collection Agencies.
- Cardinalité : 1-1

« **TotalObjects** » : Contient la répartition du nombre d'objets du service producteur par état

- « ingested » : nombre total d'objets pris en charge dans le système pour ce service producteur. La valeur contenue dans ce champ est un entier.
- « deleted » : nombre d'objets supprimés ou sortis du système. La valeur contenue dans ce champ est un entier.
- « remained » : nombre actualisé d'objets conservés dans le système. La valeur contenue dans ce champ est un entier.
- Champ peuplé par la solution logicielle Vitam.
- Cardinalité : 1-1

« **TotalObjectGroups** » : Contient la répartition du nombre de groupes d'objets du service producteur par état

- « ingested » : nombre total de groupes d'objets pris en charge dans le système pour ce service producteur. La valeur contenue dans le champ est un entier.
- « deleted » : nombre de groupes d'objets supprimés ou sortis du système. La valeur contenue dans ce champ est un entier.
- « remained » : nombre actualisé de groupes d'objets conservés dans le système. La valeur contenue dans ce champ est un entier.
- Il s'agit d'un JSON
- Champ peuplé par la solution logicielle Vitam.
- Cardinalité : 1-1

« **TotalUnits** » : Contient la répartition du nombre d'unités archivistiques du service producteur par état.

- « ingested » : nombre total d'unités archivistiques prises en charge dans le système pour ce service producteur. La valeur contenue dans le champ est un entier.
- « deleted » : nombre d'unités archivistiques supprimées ou sorties du système. La valeur contenue dans ce champ est un entier.
- « remained » : nombre actualisé d'unités archivistiques conservées. La valeur contenue dans ce champ est un entier.
- Champ peuplé par la solution logicielle Vitam.
- Cardinalité : 1-1

« **ObjectSize** » : Contient la répartition du volume total des fichiers du service producteur par état.

- « ingested » : volume total en octet des fichiers pris en charge dans le système pour ce service producteur. La valeur contenue dans le champ est un entier.
- « deleted » : volume total en octet des fichiers supprimés ou sortis du système. La valeur contenue dans ce champ est un entier.
- « remained » : volume actualisé en octet des fichiers conservés dans le système. La valeur contenue dans ce champ est un entier.

- Champ peuplé par la solution logicielle Vitam.
- Cardinalité : 1-1

« **CreationDate** » : Date du dernier calcul de ce document dans la collection.

- La date est au format ISO 8601
- Cardinalité : 1-1

```
"CreationDate": "2017-04-10T11:30:33.798"
```

« **_v** » : version de l'enregistrement décrit.

- Il s'agit d'un entier.
- Champ peuplé par la solution logicielle Vitam.
- 0 correspond à l'enregistrement d'origine. Si le numéro est supérieur à 0, alors il s'agit du numéro de version de l'enregistrement.
- Cardinalité : 1-1

« **_tenant** » : correspondant à l'identifiant du tenant.

- Il s'agit d'une chaîne de caractères.
- Champ peuplé par la solution logicielle Vitam.
- Cardinalité : 1-1

5.4. Collection AccessionRegisterSymbolic

5.4.1. Utilisation de la collection

Cette collection contient une vue macroscopique des fonds relatifs aux services producteurs symboliques. Ces documents sont calculés périodiquement à partir des métadonnées renseignées dans les unités archivistiques et les groupes d'objets.

Chaque document représente un instantané (snapshot) du stock symbolique pour un producteur, conservé pour l'historisation des fonds de ce dernier. Un nouveau document est donc créé à chaque fois que le registre des fonds symboliques est calculé.

```
{
  "_id": "aefaaaaaaaae2tauiaak6ualgbn5dp5aaaaaq",
  "CreationDate": "2018-09-24T14:07:31.053",
  "_tenant": 0,
  "OriginatingAgency": "RATP",
  "ArchiveUnit": 1,
  "ObjectGroup": 1,
  "BinaryObject": 1,
  "BinaryObjectSize": 6,
  "_v": 0
}
```

5.4.2. Détail des champs

« **_id** » : identifiant unique du fonds.

- Il s'agit d'une chaîne de 36 caractères correspondant à un GUID.
- Champ peuplé par la solution logicielle Vitam.
- Cardinalité : 1-1

« **CreationDate** » : Date de calcul de ce document.

- La date est au format ISO 8601
- Cardinalité : 1-1

Exemple : "CreationDate": "2017-04-10T11:30:33.798"

« **OriginatingAgency** » : identifiant du service producteur symbolique.

- La valeur de ce champ est une chaîne de caractères.
- Cardinalité : 1-1

« **ArchiveUnit** » : Nombre actualisé d'unités archivistiques conservées.

- Il s'agit d'un entier
- Champ peuplé par la solution logicielle Vitam.
- Cardinalité : 1-1

« **ObjectGroup** » : Nombre actualisé de groupes d'objets conservés dans le système.

- Il s'agit d'un JSON
- Champ peuplé par la solution logicielle Vitam.
- Cardinalité : 0-1

« **BinaryObject** » : nombre actualisé d'objets conservés dans le système.

- Il s'agit d'un entier
- Champ peuplé par la solution logicielle Vitam.
- Cardinalité : 0-1

« **BinaryObjectSize** » : Volume actualisé en octet des fichiers conservés dans le système. La valeur contenue dans ce champ est un entier.

- Il s'agit d'un entier
- Champ peuplé par la solution logicielle Vitam.
- Cardinalité : 0-1

« **_v** » : version de l'enregistrement décrit.

- Il s'agit d'un entier.
- Champ peuplé par la solution logicielle Vitam.
- 0 correspond à l'enregistrement d'origine. Si le numéro est supérieur à 0, alors il s'agit du numéro de version de l'enregistrement. Un document dans le registre des fonds symbolique n'est pas censé être modifié et donc avoir une version supérieure à 0
- Cardinalité : 1-1

« **_tenant** » : correspondant à l'identifiant du tenant.

- Il s'agit d'une chaîne de caractères.
- Champ peuplé par la solution logicielle Vitam.
- Cardinalité : 1-1

5.5. Collection ArchiveUnitProfile

5.5.1. Utilisation de la collection

La collection ArchiveUnitProfile permet de référencer et décrire unitairement les profils d'unité archivistique.

5.5.2. Exemple d'un fichier d'import de profils d'unité archivistique

Les profils d'unité archivistique sont importés dans la solution logicielle Vitam sous la forme d'un fichier JSON.

```
{
  "Name": "Facture",
  "Description": "profil d'unité archivistique d'une facture associée à un dossier de marché",
  "Identifiant": "AUP_IDENTIFIEUR_0",
  "Statut": "ACTIVE",
  "ControlSchema": "{}",
  "LastUpdate": "10/12/2016",
  "CreationDate": "10/12/2016",
  "ActivationDate": "10/12/2016",
  "DeactivationDate": "10/12/2016"
}
```

Les champs à renseigner obligatoirement à l'import d'un profil d'unité archivistique sont :

- Name
- Description
- ControlSchema (même si le champ est vide)

Un fichier JSON peut décrire plusieurs profils d'unité archivistique.

5.5.3. Exemple de JSON stocké en base comprenant l'exhaustivité des champs de la collection ArchiveUnitProfile

```
{
  "_id": "aegaaaaabmhdh434aapnqalcd7mufiyaaaaq",
  "Identifiant": "AUP_IDENTIFIEUR_0",
  "Name": "Facture",
  "Description": "profil d'unité archivistique d'une facture associée à un dossier de marché",
  "Statut": "ACTIVE",
  "ControlSchema": "{}",
  "Fields": [],
  "LastUpdate": "10/12/2016",
  "CreationDate": "10/12/2016",
  "ActivationDate": "10/12/2016",
  "DeactivationDate": "10/12/2016",
  "_tenant": 11,
  "_v": 0
}
```

5.5.4. Détail des champs de la collection ArchiveUnitProfile

« **_id** » : identifiant unique du profil d'unité archivistique.

- Il s'agit d'une chaîne de 36 caractères correspondant à un GUID.
- Champ peuplé par la solution logicielle Vitam.
- Cardinalité : 1-1

« **Name** » : Nom du profil d'unité archivistique.

- Il s'agit d'une chaîne de caractères.
- Cardinalité : 1-1

« **Identifiant** » : identifiant signifiant donné au profil d'unité archivistique.

- Il est constitué du préfixe « AUP- » suivi d'une suite de 6 chiffres dans le cas où la solution

logicielle Vitam peuple l'identifiant. Par exemple : AUP-007485. Si le référentiel est en position esclave, cet identifiant peut être géré par l'application à l'origine du profil d'unité archivistique.

- Il s'agit d'une chaîne de caractères.
- Cardinalité : 1-1

« **Description** » : description du profil d'unité archivistique.

- Il s'agit d'une chaîne de caractères.
- Cardinalité : 1-1

« **Status** » : statut du profil d'unité archivistique.

- Il s'agit d'une chaîne de caractères.
- Peut avoir comme valeur : « ACTIVE » ou « INACTIVE »
- Cardinalité : 1-1

« **CreationDate** » : date de création du profil d'unité archivistique.

- La date est au format ISO 8601
- Cardinalité : 1-1

Exemple : "CreationDate": "2017-04-10T11:30:33.798"

« **LastUpdate** » : date de dernière mise à jour du profil d'unité archivistique dans la collection ArchiveUnitProfile.

- La date est au format ISO 8601
- Cardinalité : 1-1

Exemple : "LastUpdate": "2017-04-10T11:30:33.798"

« **ActivationDate** » : date d'activation du profil d'unité archivistique.

- La date est au format ISO 8601
- Cardinalité : 0-1

Exemple : "ActivationDate": "2017-04-10T11:30:33.798"

« **DeactivationDate** » : date de désactivation du profil d'unité archivistique.

- La date est au format ISO 8601
- Cardinalité : 0-1

Exemple : "DeactivationDate": "2017-04-10T11:30:33.798"

« **ControlSchema** » : schéma de contrôle du profil d'unité archivistique

- Il s'agit d'un bloc JSON.
- Peut être vide
- Cardinalité : 1-1

```
{
  "_id": "aegaaaaaehk2lclaaaf5ialisr6tklaaaaaq",
  "Identifiant": "AUP_CUSTOM_SCHEMA",
  "Name": "ArchiveUnitProfileWithCustomSchema",
  "Description": "Test d'import d'un document type avec schéma",
  "Status": "ACTIVE",
  "CreationDate": "2016-12-10T00:00:00.000",
  "LastUpdate": "2019-01-28T12:44:20.135",
  "ActivationDate": "2016-12-10T00:00:00.000",
  "DeactivationDate": "2016-12-10T00:00:00.000",
  "ControlSchema": "{\r\n  \"$schema\": \"http://vitam-json-schema.org/draft-
```

```
04/schema#",\r\n  \_id\": \"http://example.com/root.json\", \r\n  \_type\": \"object\", \r\n  \_additionalProperties\": true, \r\n  \_properties\": {\r\n    \_id\": {\r\n      \_type\": \"string\", \r\n    }, \r\n    \_og\": {\r\n      \_type\": \"string\", \r\n    }, \r\n    \"DescriptionLevel\": {\r\n      \_type\": \"string\", \r\n      \_enum\": [\r\n        \"Item\", \r\n        \"SubGrp\", \r\n        \"File\" \r\n      ], \r\n      \_Title\": {\r\n        \_description\": \"All TitleGroup\", \r\n        \_type\": [\r\n          \"string\", \r\n          \"array\", \r\n          \"number\" \r\n        ], \r\n        \_minLength\": 1, \r\n        \_minItems\": 1 \r\n      ] \r\n    }, \r\n    \"Fields\": [\r\n      \_id,\r\n      \_og,\r\n      \"DescriptionLevel\", \r\n      \"Title\" \r\n    ], \r\n    \_tenant\": 1, \r\n    \_v\": 0 \r\n  } \r\n}
```

« **Fields** » : liste des champs déclarés dans le schéma de contrôle

- Il s’agit d’un tableau de chaînes de caractères
- Champ peuplé automatiquement par la solution logicielle Vitam
- Cardinalité 0-1

« **_tenant** » : identifiant du tenant.

- Il s’agit d’un entier.
- Champ peuplé par la solution logicielle Vitam.
- Cardinalité : 1-1

« **_v** » : version de l’enregistrement décrit.

- Il s’agit d’un entier.
- Champ peuplé par la solution logicielle Vitam.
- Cardinalité : 1-1
- 0 correspond à l’enregistrement d’origine. Si le numéro est supérieur à 0, alors il s’agit du numéro de version de l’enregistrement.

5.6. Collection Agencies

5.6.1. Utilisation de la collection Agencies

La collection Agencies permet de référencer et décrire unitairement les services agents.

Cette collection est alimentée par l’import d’un fichier CSV contenant l’ensemble des services agents. Celui doit être structuré comme ceci :

<i>Identifiant</i>	<i>Name</i>	<i>Description</i>
Identifiant du service agent	Nom du service agent	Description du service agent

Le fichier .csv doit avoir comme séparateur de champ la virgule.

5.6.2. Exemple de JSON stocké en base comprenant l’exhaustivité des champs de la collection Agencies

```
{
  \"_id\": \"aeaaaaaaaaevq6lcaamxsak7psyd2uyaaadq\",
  \"Identifiant\": \"Identifiant5\",
```

```
"Name": "Identifier5",  
"Description": "une description de service agent",  
"_tenant": 2,  
"_v": 1  
}
```

5.6.3. Détail des champs

« **_id** » : identifiant unique du service agent.

- Il s'agit d'une chaîne de 36 caractères correspondant à un GUID.
- Champ peuplé par la solution logicielle Vitam.
- Cardinalité : 1-1

« **Name** » : nom du service agent.

- Il s'agit d'une chaîne de caractères.
- Cardinalité : 1-1

« **Description** » : description du service agent.

- Il s'agit d'une chaîne de caractères.
- Cardinalité : 0-1

« **Identifiant** » : identifiant signifiant donné au service agent.

- Le contenu de ce champ est obligatoirement renseigné dans le fichier CSV permettant de créer le service agent. En aucun cas la solution logicielle Vitam ne peut être maître sur la création de cet identifiant comme cela peut être le cas pour d'autres données référentielles.
- Il s'agit d'une chaîne de caractères.
- Cardinalité : 1-1

« **_tenant** » : information sur le tenant.

- Il s'agit de l'identifiant du tenant utilisant l'enregistrement
- Il s'agit d'un entier.
- Champ peuplé par la solution logicielle Vitam.
- Cardinalité : 1-1

« **_v** » : version de l'enregistrement décrit.

- Il s'agit d'un entier.
- Champ peuplé par la solution logicielle Vitam.
- Cardinalité : 1-1
- 0 correspond à l'enregistrement d'origine. Si le numéro est supérieur à 0, alors il s'agit du numéro de version de l'enregistrement.

5.7. Collection Context

5.7.1. Utilisation de la collection

La collection Context permet de référencer et décrire unitairement les contextes applicatifs.

5.7.2. Exemple d'un fichier d'import de contexte applicatif

Les contextes applicatifs sont importés dans la solution logicielle Vitam sous la forme d'un fichier JSON.

```
{
  "Name": "My_Context_5",
  "Status": "ACTIVE",
  "SecurityProfile": "admin-security-profile",
  "Permissions": [
    {
      "tenant": 1,
      "AccessContracts": [
        "AccessContracts_1",
        "AccessContracts_2"
      ],
      "IngestContracts": [
        "IngestContracts_1",
        "IngestContracts_2"
      ]
    },
    {
      "tenant": 0,
      "AccessContracts": [
        "AccessContracts_5",
        "AccessContracts_6"
      ],
      "IngestContracts": [
        "IngestContracts_9",
        "IngestContracts_10"
      ]
    }
  ]
}
```

5.7.3. Exemple de JSON stocké en base comprenant l'exhaustivité des champs de la collection Context

```
{
  "_id": "aegqaaaaaevq6lcaamxsak7psqdcmqaaaaq",
  "Name": "admin-context",
  "Status": "ACTIVE",
  "EnableControl": false,
  "Identifier": "CT-000001",
  "SecurityProfile": "admin-security-profile",
  "Permissions": [
    {
      "tenant": 0,
      "AccessContracts": [],
      "IngestContracts": []
    },
    {
      "tenant": 1,
      "AccessContracts": [],
      "IngestContracts": []
    },
    {
      "tenant": 2,
      "AccessContracts": [],
      "IngestContracts": []
    },
    {
      "tenant": 3,
      "AccessContracts": [],
      "IngestContracts": []
    }
  ]
}
```

```

    },
    {
      "tenant": 4,
      "AccessContracts": [],
      "IngestContracts": []
    },
    {
      "tenant": 5,
      "AccessContracts": [],
      "IngestContracts": []
    },
    {
      "tenant": 6,
      "AccessContracts": [],
      "IngestContracts": []
    },
    {
      "tenant": 7,
      "AccessContracts": [],
      "IngestContracts": []
    },
    {
      "tenant": 8,
      "AccessContracts": [],
      "IngestContracts": []
    },
    {
      "tenant": 9,
      "AccessContracts": [],
      "IngestContracts": []
    }
  ],
  "CreationDate": "2017-11-02T12:06:34.034",
  "LastUpdate": "2017-11-02T12:06:34.036",
  "_v": 0
}

```

Il est possible de mettre plusieurs contextes applicatifs dans un même fichier, sur le même modèle que les contrats d'entrée ou d'accès par exemple. On pourra noter que le contexte est multi-tenant et définit chaque tenant de manière indépendante. Il doit être enregistré dans le tenant d'administration.

Les champs à renseigner obligatoirement à la création d'un contexte applicatif sont :

- Name
- Permissions. La valeur de Permissions peut cependant être vide : « Permissions : [] »

5.7.4. Détail des champs

« **_id** » : identifiant unique du contexte applicatif.

- Il s'agit d'une chaîne de 36 caractères correspondant à un GUID.
- Champ peuplé par la solution logicielle Vitam.
- Cardinalité : 1-1

« **Name** » : nom du contexte applicatif.

- Il s'agit d'une chaîne de caractères.
- Cardinalité : 1-1

« **Status** » : statut du contexte applicatif.

- Il s'agit d'une chaîne de caractères.

- Peut avoir pour valeur : ACTIVE ou INACTIVE
- Cardinalité : 1-1

« **Identifiant** » : identifiant signifiant donné au contexte applicatif.

- Il est constitué du préfixe « CT- » suivi d'une suite de 6 chiffres. Par exemple : CT-001573.
- Il s'agit d'une chaîne de caractères.
- Cardinalité : 1-1

« **SecurityProfile** » : nom du profil de sécurité utilisé par le contexte applicatif.

- Il s'agit d'une chaîne de caractères correspondant à une valeur valide du champ « _id » de la collection SecurityProfile.
- Cardinalité : 1-1

« **Permissions** » : début du bloc appliquant les permissions à chaque tenant.

- C'est un mot clé qui n'a pas de valeur associée.
- Il s'agit d'un tableau.
- Peut être vide.
- Cardinalité : 1-1
- pour un tenant donné, il contient un objet JSON contenant les champs suivants :
 - « **tenant** » : tenant sur lequel sont appliquées les permissions.
 - Il s'agit d'un entier.
 - Cardinalité : 1-1
 - « **AccessContracts** » : tableau d'identifiants de contrats d'accès appliqués sur le tenant.
 - Il s'agit d'un tableau de chaînes de caractères.
 - Les identifiants valides correspondent au champ Identifiant des documents de la collection AccessContract.
 - Peut être vide.
 - Cardinalité : 0-1
 - « **IngestContracts** » : tableau d'identifiants de contrats d'entrées appliqués sur le tenant.
 - Il s'agit d'un tableau de chaînes de caractères.
 - Les identifiants valides correspondent au champ Identifiant des documents de la collection IngestContract.
 - Peut être vide.
 - Cardinalité : 0-1

« **CreationDate** » : date de création du contexte applicatif.

- Il s'agit d'une date au format ISO 8601.
- Cardinalité : 1-1

Exemple : "CreationDate": "2017-04-10T11:30:33.798",

« **LastUpdate** » : date de dernière modification du contexte applicatif.

- Il s'agit d'une date au format ISO 8601.
- Cardinalité : 1-1

Exemple : "LastUpdate": "2017-04-10T11:30:33.798",

« **_v** » : version de l'enregistrement décrit.

- Il s'agit d'un entier.
- Champ peuplé par la solution logicielle Vitam.
- 0 correspond à l'enregistrement d'origine. Si le numéro est supérieur à 0, alors il s'agit du numéro de version de l'enregistrement.
- Cardinalité : 1-1

« **EnableControl** » : activation des contrôles sur les tenants.

- Il s'agit d'un booléen
- Il peut avoir pour valeur « true » ou « false » et a la valeur par défaut : « false ».
 - « true » : le contrôle est actif
 - « false » : le contrôle est inactif
- Cardinalité : 1-1

5.8. Collection FileFormat

5.8.1. Utilisation de la collection FileFormat

La collection FileFormat permet de référencer et décrire unitairement les différents formats de fichiers ainsi que leur description. La collection est initialisée à partir de l'import du fichier de signature PRONOM, mis à disposition par The National Archive (UK).

Cette collection est commune à tous les tenants. Elle est enregistrée sur le tenant d'administration.

5.8.2. Exemple de la description d'un format dans le XML d'entrée

Ci-après, la portion d'un fichier de signatures (DROID_SignatureFile_VXX.xml) utilisée pour renseigner les champs du JSON.

```
<FileFormat ID="105" MimeType="application/msword" Name="Microsoft Word for
Macintosh Document" PUID="x-fmt/64" Version="4.0">
  <InternalSignatureID>486</InternalSignatureID>
  <Extension>mcw</Extension>
</FileFormat>
```

5.8.3. Exemple de JSON stocké en base comprenant l'exhaustivité des champs de la collection FileFormat

```
{
  "_id": "aeaaaaaaaaahbl62nabduoak3jc2zqciaadiq",
  "CreateDate": "2016-09-27T15:37:53",
  "VersionPronom": "88",
  "UpdateDate": "2016-09-27T15:37:53",
  "PUID": "fmt/961",
  "Version": "2",
  "Name": "Mobile eXtensible Music Format",
  "Extension": [
    "mxf"
  ],
  "HasPriorityOverFileFormatID": [
    "fmt/714"
  ],
  "MimeType": "audio/mobile-mxf",
  "Group": "",
  "Alert": false,
```

```
"Comment": "",  
"_v": 0  
}
```

5.8.4. Détail des champs du JSON stocké en base

« **_id** » : identifiant unique du format.

- Il s'agit d'une chaîne de 36 caractères correspondant à un GUID.
- Champ peuplé par la solution logicielle Vitam.
- Cardinalité : 1-1

« **CreatedDate** » : date de création de la version du fichier de signatures PRONOM utilisé pour créer l'enregistrement.

- Il s'agit d'une date au format ISO 8601 YYYY-MM-DD + "T" + hh:mm:ss.millisecondes « + » timezone hh:mm.
- La date de création est à l'origine déclarée dans le fichier de signatures au niveau de la balise <FFSignatureFile> au niveau de l'attribut « DateCreated » .
- Cardinalité : 1-1

Exemple : "2016-08-19T16:36:07.942+02:00"

« **VersionPronom** » : numéro de version du fichier de signatures PRONOM utilisé pour créer l'enregistrement.

- Il s'agit d'un entier.
- Le numéro de version de PRONOM est à l'origine déclaré dans le fichier de signature au niveau de la balise <FFSignatureFile> au niveau de l'attribut « Version » .
- Cardinalité : 1-1

Dans cet exemple, le numéro de version est 88 :

```
<FFSignatureFile DateCreated="2016-09-27T15:37:53" Version="88"  
xmlns="http://www.nationalarchives.gov.uk/pronom/SignatureFile">
```

« **UpdateDate** » : date de mise à jour de la version du fichier de signatures PRONOM utilisé pour mettre à jour la collection.

- Il s'agit d'une date au format ISO 8601 YYYY-MM-DD + "T" + hh:mm:ss.millisecondes « + » timezone hh:mm.
- Champ peuplé par la solution logicielle Vitam.
- Cardinalité : 1-1

Exemple : "2016-08-19T16:36:07.942+02:00"

« **PUID** » : identifiant unique du format au sein du référentiel PRONOM.

- Il s'agit d'une chaîne de caractères.
- Il est issu du champ « PUID » de la balise <FileFormat>. La valeur est composée du préfixe « fmt » ou « x-fmt », puis d'un nombre correspondant au numéro d'entrée du format dans le référentiel PRONOM. Les deux éléments sont séparés par un « / ».
- Cardinalité : 1-1

Par exemple :

```
x-fmt/64
```

Les PUID comportant un préfixe « x-fmt » indiquent que ces formats sont en cours de validation

par The National Archives (UK). Ceux possédant un préfixe « fmt » sont validés.

« **Version** » : version du format.

- Il s'agit d'une chaîne de caractères.
- Peut être vide.
- Cardinalité : 1-1

Exemples de formats :

```
Version="3D Binary Little Endian 2.0"  
Version="2013"  
Version="1.5"
```

L'attribut « version » n'est pas obligatoire dans la balise <FileFormat> du fichier de signatures.

« **Name** » : nom du format.

- Il s'agit d'une chaîne de caractères.
- Le nom du format est issu de la valeur de l'attribut « Name » de la balise <FileFormat> du fichier de signature.
- Cardinalité : 1-1

« **MimeType** » : Type MIME correspondant au format de fichier.

- Il s'agit d'une chaîne de caractères.
- Peut être vide.
- Il est renseigné avec le contenu de l'attribut « MimeType » de la balise <FileFormat>. Cet attribut est facultatif dans le fichier de signatures.
- Cardinalité : 1-1

« **HasPriorityOverFileFormatID** » : liste des PUID des formats sur lesquels le format a la priorité.

- Il s'agit d'un tableau de chaînes de caractères.
- Peut être vide.
- Cardinalité : 1-1

```
<HasPriorityOverFileFormatID>1121</HasPriorityOverFileFormatID>
```

Cet identifiant est ensuite utilisé dans Vitam pour retrouver le PUID correspondant.

S'il existe plusieurs balises <HasPriorityOverFileFormatID> dans le fichier XML initial pour un format donné, alors les PUID seront stockés dans le JSON sous la forme suivante :

```
"HasPriorityOverFileFormatID": [  
  "fmt/714",  
  "fmt/715",  
  "fmt/716"  
],
```

« **Extension** » : extension(s) du format.

- Il s'agit d'un tableau de chaînes de caractères.
- Peut être vide.
- Il contient les valeurs situées entre les balises <Extension> elles-mêmes encapsulées entre les balises <FileFormat>.
- Cardinalité : 1-1

Le champ <Extension> peut-être multivalué. Dans ce cas, les différentes valeurs situées entre les différentes balises <Extension> sont placées dans le tableau et séparées par une virgule.

Par exemple, pour le format dont le PUID est fmt/918 la représentation XML est la suivante :

```
<FileFormat ID="1723" Name="AmiraMesh" PUID="fmt/918" Version="3D ASCII 2.0">
  <InternalSignatureID>1268</InternalSignatureID>
  <Extension>am</Extension>
  <Extension>amiramesh</Extension>
  <Extension>hx</Extension>
</FileFormat>
```

Les valeurs des balises <Extension> seront stockées de la façon suivante dans le JSON :

```
"Extension": [
  "am",
  "amiramesh",
  "hx"
],
```

« **Group** » : champ permettant d'indiquer le nom d'une famille de formats.

- Il s'agit d'une chaîne de caractères.
- Peut être vide.
- C'est un champ propre à la solution logicielle Vitam.
- Cardinalité : 1-1

« **Alert** » : alerte sur l'obsolescence du format.

- Il s'agit d'un booléen dont la valeur est par défaut placée à false.
- Cardinalité : 1-1

« **Comment** » : commentaire.

- Il s'agit d'une chaîne de caractères.
- Peut être vide.
- C'est un champ propre à la solution logicielle Vitam.
- Cardinalité : 1-1

« **_v** » : version de l'enregistrement décrit.

- Il s'agit d'un entier.
- Champ peuplé par la solution logicielle Vitam.
 - 0 correspond à l'enregistrement d'origine.
 - Si le numéro est supérieur à 0, alors il s'agit du numéro de version de l'enregistrement.
- Cardinalité : 1-1

5.9. Collection FileRules

5.9.1. Utilisation de la collection FileRules

La collection FileRules permet de référencer et décrire unitairement les différentes règles de gestion utilisées dans la solution logicielle Vitam pour calculer les échéances associées aux unités archivistiques.

Cette collection est alimentée par l'import d'un fichier CSV contenant l'ensemble des règles. Celui-ci doit être structuré comme ceci :

RuleId	RuleType	RuleValue	RuleDescription	RuleDuration	RuleMeasurement
Id de la règle	Type de règle	Intitulé de la règle	Description de la règle	Durée de la règle	Unité de mesure de la durée de la règle

Le fichier .csv doit avoir comme séparateur de champs la virgule.

La liste des types de règle disponibles est en annexe.

Les valeurs renseignées dans la colonne unité de mesure doivent correspondre à une valeur de l'énumération RuleMeasurementEnum, à savoir :

- MONTH
- DAY
- YEAR

5.9.2. Exemple de JSON stocké en base comprenant l'exhaustivité des champs de la collection FileRules

```
{
  "_id": "aeaaaaaaaaahbl62nabduoak3jc4avsyaaha",
  "RuleId": "ACC-00011",
  "RuleType": "AccessRule",
  "RuleValue": "Communicabilité des informations portant atteinte au secret de la
défense nationale",
  "RuleDescription": "Durée de communicabilité applicable aux informations portant
atteinte au secret de la défense nationale\nL'échéance est calculée à partir de la
date du document ou du document le plus récent inclus dans le dossier",
  "RuleDuration": "50",
  "RuleMeasurement": "YEAR",
  "CreationDate": "2017-11-02T13:50:28.922",
  "UpdateDate": "2017-11-06T09:11:54.062",
  "_v": 0,
  "_tenant": 0
}
```

5.9.3. Détail des champs

« **_id** » : identifiant unique.

- Il s'agit d'une chaîne de 36 caractères correspondant à un GUID.
- Champ peuplé par la solution logicielle Vitam.
- Cardinalité : 1-1

« **RuleId** » : identifiant unique par tenant de la règle dans le référentiel utilisé.

- Il s'agit d'une chaîne de caractères.
- La valeur est reprise du champ RuleId du fichier d'import. Par commodité, les exemples sont composés d'un préfixe puis d'un nombre, séparés par un tiret, mais ce formalisme n'est pas obligatoire.
- Cardinalité : 1-1

Par exemple :

```
ACC-00027
```

« **RuleType** » : type de règle.

- Il s'agit d'une chaîne de caractères.

- Il correspond à la valeur située dans la colonne RuleType du fichier d'import. Les valeurs possibles pour ce champ sont indiquées en annexe.
- Cardinalité : 1-1

« **RuleValue** » : intitulé de la règle.

- Il s'agit d'une chaîne de caractères.
- Elle correspond à la valeur de la colonne RuleValue du fichier d'import.
- Cardinalité : 1-1

« **RuleDescription** » : description de la règle.

- Il s'agit d'une chaîne de caractères.
- Elle correspond à la valeur de la colonne RuleDescription du fichier d'import.
- Cardinalité : 1-1

« **RuleDuration** » : durée de la règle.

- Il s'agit d'un entier compris entre 0 et 999.
- Associé à la valeur indiquée dans RuleMeasurement, il permet de décrire la durée d'application de la règle de gestion. Il correspond à la valeur de la colonne RuleDuration du fichier d'import.
- Cardinalité : 1-1

« **RuleMeasurement** » : unité de mesure de la durée décrite dans la colonne RuleDuration du fichier d'import.

- Il s'agit d'une chaîne de caractères devant correspondre à une valeur de l'énumération RuleMeasurementEnum, à savoir :
 - MONTH
 - DAY
 - YEAR
- Cardinalité : 1-1

« **CreationDate** » : date de création de la règle dans la collection FileRules.

- La date est au format ISO 8601.
 - Cardinalité : 1-1
- Exemple : "2017-11-02T13:50:28.922"

« **UpdateDate** » : Date de dernière mise à jour de la règle dans la collection FileRules.

- La date est au format ISO 8601.
 - Champ peuplé par la solution logicielle Vitam.
 - Cardinalité : 1-1
- Exemple : "2017-11-02T13:50:28.922"

« **_v** » : version de l'enregistrement décrit

- Il s'agit d'un entier.
- Champ peuplé par la solution logicielle Vitam.
- Cardinalité : 1-1
- 0 correspond à l'enregistrement d'origine. Si le numéro est supérieur à 0, alors il s'agit du numéro de version de l'enregistrement.

« **_tenant** » : identifiant du tenant.

- Il s'agit d'un entier.
- Champ peuplé par la solution logicielle Vitam.
- Cardinalité : 1-1

5.10. Collection Griffin

5.10.1. Utilisation de la collection Griffin

La collection Griffin permet de référencer et décrire unitairement les griffons utilisés pour mettre en œuvre les opérations de préservation.

5.10.2. Exemple d'un fichier d'import de griffon

Le référentiel des griffons est importé dans la solution logicielle Vitam sous la forme d'un fichier JSON.

```
[
  {
    "Identifiant": "GRI-000004",
    "Name": "Griffon ImageMagick",
    "Description": "A griffin for griff the griffins",
    "CreationDate": "10/12/2016",
    "ExecutableName": "imagemagick-griffin",
    "ExecutableVersion": "V1.0.0"
  },
  {
    "Identifiant": "GRI-000005",
    "Name": "Griffon Jhove",
    "Description": "A jhove griffin",
    "CreationDate": "2018-11-16T15:55:30.721",
    "ExecutableName": "jhove-griffin",
    "ExecutableVersion": "V1.0.0"
  }
]
```

Les champs à renseigner obligatoirement à l'import d'un griffon sont :

- Name
- Identifiant
- ExecutableName
- ExecutableVersion

Un fichier d'import peut décrire plusieurs griffons.

5.10.3. Exemple de JSON stocké en base comprenant l'exhaustivité des champs de la collection Agencies

```
{
  "_id": "aeaaaaaaaaahlopljab2wualhmuydxiaaaaaaq",
  "Name": "imgmagic",
  "Identifiant": "GRIFFIN1",
  "Description": "Griffon IMG",
  "CreationDate": "2016-12-10T00:00:00.000",
  "LastUpdate": "2018-12-07T04:25:57.510",
  "ExecutableName": "imagemagick-griffin",
  "ExecutableVersion": "V1",
  "_tenant": 1,
}
```

```
"_v": 13  
}
```

5.10.4. Détail des champs

« **_id** » : identifiant unique faisant référence à un exécutable et à sa version.

- Il s'agit d'une chaîne de 36 caractères correspondant à un GUID.
- Champ peuplé par la solution logicielle Vitam.
- Cardinalité : 1-1

« **Name** » : nom du griffon.

- Il s'agit d'une chaîne de caractères.
- Cardinalité : 1-1

« **Identifiant** » : identifiant signifiant donné au griffon.

- Il s'agit d'une chaîne de caractères.
- Cardinalité : 1-1

« **Description** » : description du griffon.

- Il s'agit d'une chaîne de caractères.
- Cardinalité : 0-1

« **CreationDate** » : date de création du griffon.

- La date est enregistrée au format ISO 8601.
- S'il n'est pas renseigné dans le fichier d'import, champ peuplé par la solution logicielle Vitam.
- Cardinalité : 1-1

```
"CreationDate": "2017-04-10T11:30:33.798"
```

« **LastUpdate** » : date de dernière mise à jour du griffon dans la collection Griffin.

- La date est au format ISO 8601.
- Champ peuplé par la solution logicielle Vitam.
- Cardinalité : 1-1

```
"LastUpdate": "2017-04-10T11:30:33.798"
```

« **ExecutableName** » : nom technique du griffon utilisé pour lancer l'exécutable sur le système.

- Il s'agit d'une chaîne de caractères.
- Cardinalité : 1-1

« **ExecutableVersion** » : version du griffon utilisé.

- Un même exécutable (ExecutableName) peut être associé à plusieurs versions.
- Cardinalité : 1-1

```
"ExecutableName": "imagemagick-griffin"
```

```
"ExecutableVersion": "v1.0.0"
```

« **_tenant** » : information sur le tenant.

- Il s'agit d'un entier.
- Champ peuplé par la solution logicielle Vitam.

- Cardinalité : 1-1

« **_v** » : version de l'enregistrement décrit.

- Il s'agit d'un entier.
- Champ peuplé par la solution logicielle Vitam.
- Cardinalité : 1-1
- 0 correspond à l'enregistrement d'origine. Si le numéro est supérieur à 0, alors il s'agit du numéro de version de l'enregistrement.

5.11. Collection IngestContract

5.11.1. Utilisation de la collection

La collection IngestContract permet de référencer et décrire unitairement les contrats d'entrée.

5.11.2. Exemple d'un fichier d'import de contrat

Les contrats d'entrée sont importés dans la solution logicielle Vitam sous la forme d'un fichier JSON.

```
[
  {
    "Name": "Contrat Archives Départementales",
    "Description": "Test entrée - Contrat Archives Départementales",
    "Status" : "ACTIVE",
  },
  {
    "Name": "SIA archives nationales",
    "Description": "Contrat d'accès - SIA archives nationales",
    "Status" : "INACTIVE",
    "ArchiveProfiles": [
      "ArchiveProfile8"
    ],
    "LinkParentId" : "aeaqaaaaaagbcaacaax56ak35rpo6zqaaaaq"
  }
]
```

Les champs à renseigner obligatoirement à l'import d'un contrat sont :

- Name
- Identifiant (selon la configuration du tenant : Identifiant n'est obligatoire que si l'identifiant du contrat d'entrée n'est pas généré par la solution logicielle Vitam)

Un fichier d'import peut décrire plusieurs contrats.

5.11.3. Exemple de JSON stocké en base comprenant l'exhaustivité des champs de la collection IngestContract

```
{
  "_id": "aefqaaaaaahbl62nabkzgak3k6qtf3aaaaaq",
  "Name": "SIA archives nationales",
  "Identifiant": "IC-000012",
  "Description": "Contrat d'accès - SIA archives nationales",
  "Status": "INACTIVE",
  "CreationDate": "2017-04-10T11:30:33.798",
  "LastUpdate": "2017-04-10T11:30:33.798",
  "ActivationDate": "2017-04-10T11:30:33.798",
  "DeactivationDate": null,
  "MasterMandatory": true,
}
```

```
"EveryDataObjectVersion":false,
"DataObjectVersion":"PhysicalMaster",
"ArchiveProfiles": [
  "ArchiveProfile8"
],
"CheckParentLink": "ACTIVE",
"LinkParentId":
  "aeaqaaaaaagbcaacaax56ak35rpo6zqaaaaq",
"FormatUnidentifiedAuthorized":true,
"EveryFormatType":false,
"FormatType":["fmt/17","fmt/12"],
"_tenant": 0,
"_v": 0 }
```

5.11.4. Détail des champs de la collection IngestContract

« **_id** » : identifiant unique du contrat.

- Il s'agit d'une chaîne de 36 caractères correspondant à un GUID.
- Champ peuplé par la solution logicielle Vitam.
- Cardinalité : 1-1

« **Name** » : nom du contrat d'entrée.

- Il s'agit d'une chaîne de caractères.
- Cardinalité : 1-1

« **Identifiant** » : identifiant signifiant donné au contrat.

- Il s'agit d'une chaîne de caractères.
- Il est constitué du préfixe « IC- » suivi d'une suite de 6 chiffres dans le cas où la solution logicielle Vitam peuple l'identifiant. Par exemple : IC-007485. Si le référentiel est en position esclave, cet identifiant peut être géré par l'application à l'origine du contrat. Cardinalité : 1-1

« **Description** » : description du contrat d'entrée.

- Il s'agit d'une chaîne de caractères.
- Cardinalité : 0-1

« **Status** » : statut du contrat.

- Il s'agit d'une chaîne de caractères.
- Peut avoir comme valeur : ACTIVE ou INACTIVE
- Cardinalité : 1-1

« **CreationDate** » : date de création du contrat.

- La date est au format ISO 8601.
- Cardinalité : 1-1

Exemple : "CreationDate": "2017-04-10T11:30:33.798"

« **LastUpdate** » : date de dernière mise à jour du contrat dans la collection IngestContract.

- La date est au format ISO 8601.
- Cardinalité : 1-1

Exemple : "LastUpdate": "2017-04-10T11:30:33.798"

« **ActivationDate** » : date d'activation du contrat.

- La date est au format ISO 8601.
- Cardinalité : 0-1

Exemple : "ActivationDate": "2017-04-10T11:30:33.798"

« **DeactivationDate** » : date de désactivation du contrat.

- La date est au format ISO 8601.
- Cardinalité : 0-1

Exemple : "DeactivationDate": "2017-04-10T11:30:33.798"

« **CheckParentLink** » : option permettant de définir le comportement à propos des rattachements déclarés dans les SIP.

- Si le SIP déclare un rattachement alors que cette déclaration n'est pas autorisée, l'entrée est en KO. De même, si le SIP ne déclare pas de rattachement alors que cette déclaration est obligatoire, l'entrée est en KO.
- Peut avoir comme valeur : « AUTHORIZED » (la déclaration d'un rattachement est autorisée, mais pas requise), « REQUIRED » (la déclaration d'un rattachement est obligatoire), « UNAUTHORIZED » (la déclaration d'un rattachement n'est pas autorisée).
- Dans le fichier JSON du contrat à importer, ce champ peut être absent. Dans ce cas, il sera enregistré avec la valeur « AUTHORIZED » en base de données lors de l'import.
- Cardinalité : 1-1

« **ComputedInheritedRulesAtIngest** » : activation de l'enregistrement automatique des règles de gestion héritées en base de données.

- Il s'agit d'un booléen.
- Cardinalité : 0-1

« **LinkParentId** » : point de rattachement automatique des SIP en application du contrat d'entrée correspondant à l'identifiant d'une unité archivistique standard, de plan de classement ou d'arbre de positionnement.

- Il s'agit d'une chaîne de 36 caractères correspondant à un GUID et à une valeur valide du champ `_id` d'un enregistrement de la collection Unit.
- Cardinalité : 0-1

« **CheckParentId** » : option permettant d'activer un contrôle sur les nœuds de rattachement déclarés dans le SIP.

- Le nœud déclaré dans un SIP utilisant un contrat ayant cette variable doit impérativement être le nœud déclaré dans ce paramètre ou un de ses fils.
- Il s'agit d'une ou plusieurs chaînes de 36 caractères correspondant à un GUID et à une valeur valide du champ `_id` d'un enregistrement de la collection Unit.
- Cardinalité : 0-n

« **MasterMandatory** » : option qui rend obligatoire la présence d'un objet dont l'usage est de type Master (Physical ou Binary).

- Il s'agit d'un booléen.
- Peut avoir comme valeur : « true » ou « false ».
- Dans le fichier JSON du contrat à importer, ce champ peut être absent. Dans ce cas, il sera enregistré avec la valeur « true » en base de données lors de l'import.
- Cardinalité : 1-1

« **EveryDataObjectVersion** » : option qui permet de préciser que tous les types d’usages sont autorisés lors de l’entrée d’un SIP procédant à des rattachements d’objets à des groupes d’objets techniques déjà existants.

- Il s’agit d’un booléen.
- Peut avoir comme valeur : « true » ou « false ».
- Si le champ a pour valeur « false », alors le champ DataObjectVersion sera utilisé. S’il a pour valeur « true », « DataObjectVersion » sera ignoré.
- Dans le fichier JSON du contrat à importer, ce champ peut être absent. Dans ce cas, il sera enregistré avec la valeur « INACTIVE » en base de données lors de l’import.
- Cardinalité : 1-1

« **DataObjectVersion** » : liste les types d’usages autorisés lors de l’entrée d’un SIP procédant à des rattachements d’objets à des groupes d’objets techniques déjà existants. Les usages des objets rattachés n’étant pas dans cette liste provoqueront une entrée en KO des SIP.

- Il s’agit d’un tableau de chaîne de caractères.
- Peut avoir comme valeur : « Dissemination », « TextContent », « PhysicalMaster », « BinaryMaster », « Thumbnail »
- Dans le fichier JSON du contrat à importer, ce champ peut être absent. Si le champ « EveryDataObjectVersion » a pour valeur « true », ce champ sera ignoré.
- Cardinalité : 0-1

« **FormatUnidentifiedAuthorized** » : option autorisant ou non l’entrée d’objets dont le format n’est pas identifié par la solution logicielle Vitam

- Il s’agit d’un booléen.
- Peut avoir comme valeur : « true » ou « false ».
- Dans le fichier JSON du contrat à importer, ce champ peut être absent. Dans ce cas, il sera enregistré avec la valeur « false » en base de données lors de l’import.
- Cardinalité : 1-1

« **EveryFormatType** » : option autorisant ou non l’entrée d’objets sans restriction de formats.

- Il s’agit d’un booléen.
- Peut avoir comme valeur : « true » ou « false ».
- Si ce champ a comme valeur « false », alors le champ « FormatType » sera utilisé. S’il a comme valeur « true », alors le champ « FormatType » sera ignoré.
- Dans le fichier JSON du contrat à importer, ce champ peut être absent. Dans ce cas, il sera enregistré avec la valeur « false » en base de données lors de l’import.
- Cardinalité : 1-1

« **FormatType** » : liste de PUID de formats de fichiers autorisés lors de l’entrée d’un objet. Les objets n’étant pas dans cette liste de format provoqueront une entrée KO de leurs SIP

- Il s’agit d’un tableau de chaîne de caractères correspondant à des identifiants valides du champ PUID de la collection FileFormat.
- Si la variable EveryFormatType est à « true », ce champ sera ignoré.
- Cardinalité : 0-1

« **ManagementContractId** » : définition d’une stratégie de stockage dans le contrat d’entrée.

- Il s’agit d’une chaîne de caractères, correspondant à l’identifiant du contrat de gestion

associé au contrat d'entrée.

- Cardinalité : 0-1

« **ArchiveProfiles** » : profil(s) d'archivage associé(s) contrat d'entrée.

- Il s'agit d'un tableau pouvant contenir une à plusieurs chaînes de caractères, correspondant à l'identifiant de profil(s) d'archivage défini dans le référentiel des profils d'archivage.
- Cardinalité : 0-1

« **ComputedInheritedRulesAtIngest** » : paramètre permettant d'activer l'enregistrement automatique des règles de gestion héritées par une unité archivistique en base de données dans un champ spécifique.

- Il s'agit d'un booléen
- Valeur par défaut : « false »
- Cardinalité : 1-1

« **_tenant** » : identifiant du tenant.

- Il s'agit d'un entier.
- Champ peuplé par la solution logicielle Vitam.
- Cardinalité : 1-1

« **_v** » : version de l'enregistrement décrit.

- Il s'agit d'un entier.
- Champ peuplé par la solution logicielle Vitam.
- 0 correspond à l'enregistrement d'origine. Si le numéro est supérieur à 0, alors il s'agit du numéro de version de l'enregistrement.
- Cardinalité : 1-1

5.12. Collection ManagementContract

5.12.1. Utilisation de la collection ManagementContract

La collection ManagementContract permet de référencer et de décrire unitairement les contrats de gestion.

5.12.2. Exemple d'un fichier d'import de contrat de gestion

Les contrats de gestion sont importés dans la solution logicielle Vitam sous la forme d'un fichier JSON.

```
[
  {
    "Name": "Contrat de gestion avec stockage",
    "Identifiant": "MCDefaultStorageAll",
    "Description": "Contrat de gestion valide déclarant pas de surcharge pour le
stockage avec la stratégie par défaut",
    "Status": "ACTIVE",
    "Storage": {
      "UnitStrategy": "default",
      "ObjectGroupStrategy": "default",
      "ObjectStrategy": "default"
    }
  }
]
```

Les champs à renseigner obligatoirement à la création d'un contrat sont :

- Name
- Identifiant (selon la configuration du tenant : Identifiant n'est obligatoire que si l'identifiant du contrat d'accès n'est pas généré par la solution logicielle Vitam)

Un fichier d'import peut décrire plusieurs contrats.

5.12.3. Exemple de JSON stocké en base comprenant l'exhaustivité des champs de la collection ManagementContract

```
{
  "_id": "aefqaaaaahjy6rtaapocalm6uiw5oqaaaaq",
  "Name": "Contrat de gestion avec stockage",
  "Identifiant": "MCDefaultStorageAll",
  "Description": "Contrat de gestion valide déclarant pas de surcharge pour le
stockage avec la stratégie par défaut",
  "Status": "ACTIVE",
  "CreationDate": "2016-12-10T00:00:00.000",
  "LastUpdate": "2019-09-03T03:00:56.115",
  "ActivationDate": "2016-12-10T00:00:00.000",
  "DeactivationDate": "2016-12-10T00:00:00.000",
  "Storage": {
    "UnitStrategy": "default",
    "ObjectGroupStrategy": "default",
    "ObjectStrategy": "default"
  },
  "_tenant": 0,
  "_v": 0
}
```

5.12.4. Détail des champs

« **_id** » : identifiant unique du contrat pour un tenant donné.

- Il s'agit d'une chaîne de 36 caractères correspondant à un GUID.
- Champ peuplé par la solution logicielle Vitam.
- Cardinalité : 1-1

« **Name** » : nom du contrat de gestion.

- Il s'agit d'une chaîne de caractères.
- Cardinalité : 1-1

« **Identifiant** » : identifiant signifiant donné au contrat.

- Il est constitué du préfixe « MC- » suivi d'une suite de 6 chiffres s'il est peuplé par la solution logicielle Vitam. Par exemple : MC-001223. Si le référentiel est en position esclave, cet identifiant peut être géré par l'application à l'origine du contrat et est unique sur le tenant.
- Il s'agit d'une chaîne de caractères.
- Cardinalité : 1-1

« **Description** » : description du contrat de gestion.

- Il s'agit d'une chaîne de caractères.
- Cardinalité : 0-1

« **Status** » : statut du contrat.

- Il s'agit d'une chaîne de caractères.
- Peut avoir comme valeur : « ACTIVE » ou « INACTIVE ».
- Cardinalité : 1-1

« **CreationDate** » : date de création du contrat.

- La date est au format ISO 8601 et prend la forme suivante :

"CreationDate": "2017-04-10T11:30:33.798"

- Champ peuplé par la solution logicielle Vitam.
- Cardinalité : 1-1

« **LastUpdate** » : date de dernière mise à jour du contrat

- La date est au format ISO 8601 et prend la forme suivante :

"LastUpdate": "2017-04-10T11:30:33.798"

- Champ peuplé par la solution logicielle Vitam.
- Cardinalité : 1-1

« **ActivationDate** » : date d'activation du contrat.

- La date est au format ISO 8601 et prend la forme suivante :

"ActivationDate": "2017-04-10T11:30:33.798"

- Champ peuplé par la solution logicielle Vitam.
- Cardinalité : 1-1

« **DeactivationDate** » : date de désactivation du contrat.

- La date est au format ISO 8601 et prend la forme suivante :

"DeactivationDate": "2017-04-10T11:30:33.798"

- Champ peuplé par la solution logicielle Vitam.
- Cardinalité : 0-1

« **Storage** » : définition d'une stratégie de stockage pouvant être appliquée aux unités archivistiques, aux groupes d'objets techniques et/ou aux objets techniques.

- Cardinalité : 0-1
- Cet objet peut contenir les champs suivants :
 - « **UnitStrategy** » : stratégie de stockage définie pour les métadonnées correspondant aux unités archivistiques .
 - Il s'agit d'une chaîne de caractères.
 - Cardinalité : 0-1
 - « **ObjectGroupStrategy** » : stratégie de stockage définie pour les métadonnées correspondant aux groupes d'objets techniques.
 - Il s'agit d'une chaîne de caractères.
 - Cardinalité : 0-1
 - « **ObjectStrategy** » : stratégie de stockage pour les objets techniques.
 - Il s'agit d'une chaîne de caractères.
 - Cardinalité : 0-1

« **_tenant** » : identifiant du tenant.

- Il s'agit d'un entier.

- Cardinalité : 1-1

« **_v** » : version de l'enregistrement décrit

- Il s'agit d'un entier.
- Champ peuplé par la solution logicielle Vitam.
- Cardinalité : 1-1
- 0 correspond à l'enregistrement d'origine. Si le numéro est supérieur à 0, alors il s'agit du numéro de version de l'enregistrement.

5.13. Collection Ontology

5.13.1. Utilisation de la collection

La collection Ontology permet de référencer et décrire unitairement les champs définissant l'ontologie utilisée dans la solution logicielle Vitam.

5.13.2. Exemple d'un fichier d'import d'ontology

L'ontologie est importée dans la solution logicielle Vitam sous la forme d'un fichier JSON.

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

Les champs à renseigner obligatoirement pour chaque définition de champ dans l'ontologie sont :

- Identifiant
- Type
- Origin
- Collections

Un fichier JSON décrit la totalité des champs de l'ontologie (interne et externe).

5.13.3. Détail des champs de la collection Ontology

« **_id** » : identifiant unique d'un vocabulaire de l'ontologie.

- Il s'agit d'une chaîne de 36 caractères correspondant à un GUID.

- Champ peuplé par la solution logicielle Vitam.
- Cardinalité : 1-1

« **Identifiant** » : identifiant signifiant du vocabulaire de l'ontologie.

- Il s'agit d'une chaîne de caractères.
- Cardinalité : 1-1

« **SedaField** » : nom du vocabulaire dans la nomenclature SEDA.

- Il s'agit d'une chaîne de caractères.
- Peut être vide.
- Cardinalité : 1-1

« **Description** » : description du vocabulaire de l'ontologie.

- Il s'agit d'une chaîne de caractères.
- Cardinalité : 0-1

« **Type** » : type d'indexation d'un vocabulaire de l'ontologie.

- Il s'agit d'une chaîne de caractères.
- Peut avoir comme valeur : DATE, TEXT, KEYWORD, BOOLEAN, LONG, DOUBLE, ENUM, GEO_POINT.
- Cardinalité : 1-1

« **Origin** » : origine d'un vocabulaire de l'ontologie.

- Il s'agit d'une chaîne de caractères.
- Peut avoir comme valeur : INTERNAL ou EXTERNAL
- Cardinalité : 1-1

« **ShortName** » : traduction signifiante du vocabulaire.

- Il s'agit d'une chaîne de caractères.
- Cardinalité : 0-1

« **Collections** » : collections concernées par un vocabulaire de l'ontologie.

- Il s'agit d'une liste de chaînes de caractères.
- Cardinalité : 1-n

« **_tenant** » : information sur le tenant.

- Il s'agit d'un entier.
- Champ peuplé par la solution logicielle Vitam.
- Cardinalité : 1-1

« **_v** » : version de l'enregistrement décrit.

- Il s'agit d'un entier.
- Champ peuplé par la solution logicielle Vitam.
- Cardinalité : 1-1
- 0 correspond à l'enregistrement d'origine. Si le numéro est supérieur à 0, alors il s'agit du numéro de version de l'enregistrement.

« **ApiField** » : identifiant d'un vocabulaire de l'ontologie qui sera retourné via le DSL.

- Il s'agit d'une chaîne de caractères.
- Peut être vide.
- Cardinalité : 1-1

5.14. Collection PreservationScenario

5.14.1. Utilisation de la collection PreservationScenario

La collection PreservationScenario permet de référencer et décrire unitairement les scénarios de préservation utilisés pour lancer des opérations de préservation.

5.14.2. Exemple d'un fichier d'import de scénario de préservation

Les scénarios de préservation sont importés dans la solution logicielle Vitam sous la forme d'un fichier JSON.

```
[
  {
    "Identifiant": "PSC-000002",
    "Name": "Transformation en GIF MINI",
    "Description": "Ce scenario transforme une image JPEG en GIF mini",
    "ActionList": [
      "GENERATE"
    ],
    "GriffinByFormat": [
      {
        "FormatList": ["fmt/41", "fmt/43"],
        "GriffinIdentifiant": "GRI-000001",
        "TimeOut": 20,
        "MaxSize": 10000000,
        "Debug": true,
        "ActionDetail": [
          {
            "Type": "GENERATE",
            "Values": {
              "Extension": "GIF",
              "Args": [
                "-thumbnail",
                "100x100"
              ]
            }
          }
        ]
      }
    ]
  }
]
```

Les champs à renseigner obligatoirement à l'import d'un scénario de préservation sont :

- Name ;
- Identifiant ;
- ActionList ;
- GriffinByFormat, avec les champs :
 - FormatList,
 - GriffinIdentifiant,
 - Timeout,
 - MaxSize,

- Debug,
- ActionDetail,
- Type.

Un fichier d'import peut décrire plusieurs scénarios de préservation.

5.14.3. Exemple de JSON stocké en base comprenant l'exhaustivité des champs de la collection PreservationScenario

```
{
  "_id": "aefqaaaabahn6dttabew6alha45dfgqaaaaaq",
  "Identifiant": "PSC-000023",
  "Name": "Normalisation d'entrée",
  "Description": "Ce scénario permet de faire une validation des formats et de créer une version de diffusion en PDF. Il est en général appliqué au contenu d'une entrée pour donner un retour de la qualité du versement et préparer une consultation fréquente.",
  "CreationDate": "2018-11-16T15:55:30.721",
  "LastUpdate": "2018-11-20T15:34:21.542",
  "ActionList": ["ANALYSE", "GENERATE"],
  "GriffinByFormat": [{
    "FormatList": ["fmt/136", "fmt/137", "fmt/138", "fmt/139", "fmt/290", "fmt/294", "fmt/292", "fmt/296", "fmt/291", "fmt/295", "fmt/293", , "fmt/297"],
    "GriffinIdentifiant": "GRI-0000023",
    "TimeOut": 20,
    "MaxSize": 10000000,
    "ActionDetail": [{
      "Action": "ANALYSE",
      "Values": {
        "Args": ["-strict"]
      }
    }, {
      "Action": "GENERATE",
      "Values": {
        "Extension": "pdf",
        "Args": ["-f", "pdf", "-e", "SelectedPdfVersion=1"]
      }
    }
  ]
}, {
  "FormatList": ["fmt/41", "fmt/42", "x-fmt/398", "x-fmt/390", "x-fmt/391", "fmt/645", "fmt/43", "fmt/44", "fmt/112", "fmt/11", "fmt/12", "fmt/13", "fmt/935", "fmt/152", "fmt/399", "fmt/388", "fmt/387", "fmt/155", "fmt/353", "fmt/154", "fmt/153", "fmt/156", "x-fmt/392", "x-fmt/178", "fmt/408", "fmt/568", "fmt/567", "fmt/566"],
  "GriffinIdentifiant": "GRI-0000012",
  "TimeOut": 10,
  "MaxSize": 10000000,
  "ActionDetail": [{
    "Action": "ANALYSE"
  }, {
    "Action": "GENERATE",
    "Values": {
      "Extension": "pdf",
      "Args": ["-quality", "90"]
    }
  }
]
},
  "GriffinDefault": {
```

```
"GriffinIdentifiant": "GRI-0000005",
"TimeOut": 10,
"MaxSize": 10000000,
"ActionDetail": [{
  "Action": "ANALYSE",
  "Values": {
    "Args": ["-strict"]
  }
}]
},
"_tenant": 3,
"_v": 2
}
...
```

5.14.4. Détail des champs

« **_id** » : identifiant unique du scénario de préservation.

- Il s'agit d'une chaîne de 36 caractères correspondant à un GUID.
- Champ peuplé par la solution logicielle Vitam.
- Cardinalité : 1-1

« **Name** » : nom du scénario de préservation.

- Il s'agit d'une chaîne de caractères.
- Cardinalité : 1-1

« **Identifiant** » : identifiant signifiant donné au scénario de préservation.

- Il s'agit d'une chaîne de caractères.
- Cardinalité : 1-1

« **Description** » : description du scénario de préservation.

- Il s'agit d'une chaîne de caractères.
- Cardinalité : 0-1

« **CreationDate** » : date de création du scénario de préservation.

- La date est enregistrée au format ISO 8601.
- S'il n'est pas renseigné dans le fichier d'import, champ peuplé par la solution logicielle Vitam.
- Cardinalité : 1-1

```
"CreationDate": "2017-04-10T11:30:33.798"
```

« **LastUpdate** » : date de dernière de mise à jour du scénario de préservation.

- La date est au format ISO 8601.
- Champ peuplé par la solution logicielle Vitam.
- Cardinalité : 1-1

```
"LastUpdate": "2017-04-10T11:30:33.798"
```

« **ActionList** » : liste des actions prévues par le scénario de préservation.

- Il s'agit d'un tableau de chaînes de caractères.
- Il peut avoir comme valeurs : ANALYSE, GENERATE, IDENTIFY, EXTRACT, EXTRACT_AU.

- Cardinalité : 1-1

« **GriffinByFormat** » : description des actions à effectuer pour une liste de formats.

- Il s'agit d'un tableau d'objets.
- Cardinalité : 1-1
- Ce tableau est composé des champs suivants :
 - « **FormatList** » : identifiants des formats de fichiers sur lesquels l'action est effectuée. Ces identifiants doivent correspondre à des identifiants valides de la collection FileFormat.
 - Il s'agit d'un tableau de chaînes de caractères.
 - Cardinalités : 1-1
 - « **GriffinIdentifieur** » : identifiant du griffon qui effectue l'action pour les objets identifiés par un format du champ FormatList. Cet identifiant doit correspondre à un identifiant valide de la collection Griffin.
 - Il s'agit d'une chaîne de caractères.
 - Cardinalité : 1-1
 - « **Timeout** » : temps en minutes au bout duquel la solution logicielle Vitam, en l'absence de réponse du griffon, arrêtera l'action de préservation.
 - Il s'agit d'un entier.
 - Cardinalités : 1-1
 - « **MaxSize** » : taille maximale en octet des objets sur lesquels l'action de préservation peut être effectuée en utilisant ce scénario de préservation.
 - Il s'agit d'un entier.
 - Cardinalités : 1-1
 - « **Debug** » : debug.
 - Il s'agit d'un booléen. Si la valeur est « true », les erreurs rencontrées sont remontées dans les logs de la solution logicielle Vitam.
 - Cardinalité : 1-1
 - « **ActionDetail** » : tableau d'objets permettant de décrire les commandes techniques associées à chaque action de préservation.
 - Cardinalité : 1-1
 - Cet objet est composé des champs suivants :
 - « **Type** » : action de préservation.
 - Ce champ doit avoir une chaîne de caractères faisant partie des valeurs autorisées pour le champ ActionList.
 - Cardinalité : 1-1
 - « **Values** » : valeurs précisant les commandes passées par le scénario de préservation au griffon.
 - Cardinalité : 0-1
 - pour les actions ANALYSE, GENERATE, EXTRACT et EXTRACT_AU, ce champ a pour valeur « null » ou peut être absent.
 - pour l'action GENERATE, c'est un objet possédant deux champs :
 - « **Extension** » : chaîne de caractère servant à rajouter une extension aux fichiers générés (ex : .pdf).

- Cardinalité : 0-1
- « **Args** » : liste d'arguments utilisés lors de la commande système qu'effectue le griffon sur les objets concernés.
 - Cardinalité : 0-1
- pour l'action EXTRACT, c'est un objet possédant un champ :
 - « **FilteredExtractedObjectGroupData** » : liste de métadonnées à extraire.
 - Il s'agit d'un tableau de chaînes de caractères.
 - Ce champ peut contenir les valeurs suivantes : « ALL_METADATA », « RAW_METADATA » et/ou une liste de métadonnées internes en particulier (ex : resolution, compression, geometry).
 - Cardinalité : 0-1

« **GriffinDefault** » : description de l'action par défaut à effectuer si aucun format ne correspond à ceux attendus dans les objets de GriffinByFormat

- Il s'agit d'un tableau d'objets reprenant la structure de ceux de GriffinByFormat.
- S'il n'y a pas d'action par défaut à effectuer, ce champ peut être "null".
- Cardinalité : 0-1

« **_tenant** » : information sur le tenant.

- Il s'agit de l'identifiant du tenant utilisant le scénario de préservation.
- Il s'agit d'un entier.
- Champ peuplé par la solution logicielle Vitam.
- Cardinalité : 1-1

« **_v** » : version de l'enregistrement décrit.

- Il s'agit d'un entier.
- Champ peuplé par la solution logicielle Vitam.
- Cardinalité : 1-1
- 0 correspond à l'enregistrement d'origine. Si le numéro est supérieur à 0, alors il s'agit du numéro de version de l'enregistrement.

5.15. Collection Profile

5.15.1. Utilisation de la collection Profile

La collection Profile permet de référencer et décrire unitairement les notices de profil d'archivage.

5.15.2. Exemple d'un fichier d'import de notices de Profils d'archivage

Un fichier d'import peut décrire plusieurs notices de profil d'archivage.

```
[
  {
    "Name": "ArchiveProfile0",
    "Description": "Description of the Profile",
    "Status": "ACTIVE",
    "Format": "XSD"
  }
]
```

```
},
{
  "Name": "ArchiveProfile1",
  "Description": "Description of the profile 2",
  "Status": "ACTIVE",
  "Format": "RNG"
}
]
```

Les champs à renseigner obligatoirement à la création d'un profil d'archivage sont :

- Name
- Format

5.15.3. Exemple de JSON stocké en base comprenant l'exhaustivité des champs de la collection Profile

```
{
  "_id": "aegaaaaaehlfs7waax4iak4f52mzriaaaaq",
  "Identifiant": "PR-000003",
  "Name": "ArchiveProfile0",
  "Description": "Description of the Profile",
  "Status": "ACTIVE",
  "Format": "XSD",
  "CreationDate": "2016-12-10T00:00",
  "LastUpdate": "2017-05-22T09:23:33.637",
  "ActivationDate": "2016-12-10T00:00",
  "DeactivationDate": "2016-12-10T00:00",
  "_v": 1,
  "_tenant": 1,
  "Path": "1_profile_aegaaaaaehlfs7waax4iak4f52mzriaaaaq_20170522_092333.xsd"
}
```

5.15.4. Détail des champs

« **_id** » : identifiant unique de la notice de profil d'archivage.

- Il s'agit d'une chaîne de 36 caractères correspondant à un GUID.
- Champ peuplé par la solution logicielle Vitam.
- Cardinalité : 1-1

« **Identifiant** » : identifiant signifiant de la notice de profil d'archivage.

- Si Vitam est maître dans la création de cet identifiant, il est alors constitué du préfixe « PR- » suivi d'une suite de 6 chiffres. Par exemple : PR-001573. Si le référentiel est en position esclave, cet identifiant peut être géré par l'application à l'origine de la notice du profil d'archivage.
- Il s'agit d'une chaîne de caractères.
- Cardinalité : 1-1

« **Name** » : nom de la notice du profil d'archivage.

- Il s'agit d'une chaîne de caractères.
- Cardinalité : 1-1

« **Description** » : description du profil d'archivage.

- Il s'agit d'une chaîne de caractères.
- Cardinalité : 0-1

« **Status** » : statut du profil d'archivage.

- Il s'agit d'une chaîne de caractères.
- Peut avoir comme valeur : ACTIVE ou INACTIVE
- Si ce champ n'est pas défini lors de la création de l'enregistrement, alors il est par défaut INACTIVE.
- Cardinalité : 1-1

« **Format** » : format attendu pour le fichier décrivant les règles du profil d'archivage.

- Il s'agit d'une chaîne de caractères devant correspondre à l'énumération ProfileFormat.
- Ses valeurs sont soit RNG, soit XSD.
- Cardinalité : 1-1

« **CreationDate** » : date de création de la notice du profil d'archivage.

- La date est au format ISO 8601.
- Champ peuplé par la solution logicielle Vitam.
- Cardinalité : 1-1

Exemple : "CreationDate": "2017-04-10T11:30:33.798",

« **LastUpdate** » : date de dernière mise à jour de la notice du profil d'archivage.

- La date est au format ISO 8601
- Champ peuplé par la solution logicielle Vitam.
- Cardinalité : 1-1

Exemple : "LastUpdate": "2017-04-10T11:30:33.798"

« **ActivationDate** » : date d'activation de la notice du profil d'archivage.

- La date est au format ISO 8601
- Champ peuplé par la solution logicielle Vitam.
- Cardinalité : 1-1

Exemple : "ActivationDate": "2017-04-10T11:30:33.798"

« **DeactivationDate** » : date de désactivation de la notice du profil d'archivage.

- La date est au format ISO 8601
- Champ peuplé par la solution logicielle Vitam.
- Cardinalité : 1-1

Exemple : "DeactivationDate": "2017-04-10T11:30:33.798"

« **_tenant** » : information sur le tenant.

- Il s'agit d'un entier.
- Champ peuplé par la solution logicielle Vitam.
- Cardinalité : 1-1

« **_v** » : version de l'enregistrement décrit

- Il s'agit d'un entier.
- Champ peuplé par la solution logicielle Vitam.
- Cardinalité : 1-1
- 0 correspond à l'enregistrement d'origine. Si le numéro est supérieur à 0, alors il s'agit du

numéro de version de l'enregistrement.

« **Path** » : champ contribué par la solution logicielle Vitam lors d'un import de fichier XSD ou RNG.

- Indique le chemin pour accéder au fichier du profil d'archivage.
- Il s'agit d'une chaîne de caractères.
- Le format de fichier doit correspondre à celui qui est décrit dans le champ Format.
- Cardinalité : 0-1

5.16. Collection SecurityProfile

5.16.1. Utilisation de la collection

Cette collection référence et décrit les profils de sécurité mobilisés par les contextes applicatifs.

5.16.2. Exemple de JSON stocké en base comprenant l'exhaustivité des champs

```
{
  "_id": "aegqaaaaaeucszwabglyak64gjmgyaaaba",
  "Identifiant": "SEC_PROFILE-000002",
  "Name": "demo-security-profile",
  "FullAccess": false,
  "Permissions": [
    "securityprofiles:create",
    "securityprofiles:read",
    "securityprofiles:id:read",
    "securityprofiles:id:update",
    "accesscontracts:read",
    "accesscontracts:id:read",
    "contexts:id:update"
  ],
  "_v": 1
}
```

5.16.3. Détail des champs

« **_id** » : identifiant unique du profil de sécurité.

- Il s'agit d'une chaîne de 36 caractères correspondant à un GUID.
- Champ peuplé par la solution logicielle Vitam.
- Cardinalité : 1-1

« **Identifiant** » : identifiant signifiant donné au profil de sécurité.

- Il est constitué du préfixe « SEC_PROFILE- » suivi d'une suite de 6 chiffres tant qu'il est défini par la solution logicielle Vitam. Par exemple : SEC_PROFILE-001573. Si le référentiel est en position esclave, cet identifiant peut être géré par l'application à l'origine du profil de sécurité.
- Cardinalité : 1-1

« **Name** » : nom du profil de sécurité.

- Il s'agit d'une chaîne de caractères.
- Cardinalité : 1-1

« **FullAccess** » : mode super-administrateur donnant toutes les permissions.

- Il s’agit d’un booléen.
- S’il est à « false », le mode super-administrateur n’est pas activé et les valeurs du champ permission sont utilisées. S’il est à « true », le champ permission doit être vide.
- Cardinalité : 1-1

« **Permissions** » : décrit l’ensemble des permissions auxquelles le profil de sécurité donne accès. Chaque API externe contient un verbe OPTION qui retourne la liste des services avec leur description et permissions associées.

- Il s’agit d’un tableau de chaînes de caractères.
- Peut être vide
- Cardinalité : 0-1

accesscontracts:create:json accesscontracts:id:read accesscontracts:id:update accesscontracts:read accessionregisters:id:accessionregister details:read accessionregisters:read accessionregistersymbolic:read agencies:create agencies:id:read agencies:read agenciesfile:check agenciesreferential:id:read archiveunitprofiles:create:binary archiveunitprofiles:create:json archiveunitprofiles:id:read:json archiveunitprofiles:id:update:json archiveunitprofiles:read audits:create batchreport:id:read contexts:create:json contexts:id:read contexts:id:update contexts:read dipexport:create dipexport:id:dip:read	distributionreport:id:read elimination:action elimination:analysis evidenceaudit:check forcepause:check formats:create formats:id:read formats:read formatsfile:check griffin:read griffins:create griffins:read ingestcontracts:create:json ingestcontracts:id:read ingestcontracts:id:update ingestcontracts:read ingests:create ingests:id:archivetransfertreply:read ingests:id:manifests:read ingests:local:create logbookobjects:lifecycles:id:read logbookoperations:create logbookoperations:id:read logbookoperations:read logbookunitlifecycles:id:read	objects:read ontologies:create:json ontologies:id:read:json ontologies:read operations:id:delete operations:id:read operations:id:read:status operations:id:update operations:read preservation:update preservationScenario:read preservationScenarios:create preservationScenarios:read probativevalue:create profiles:create:binary profiles:create:json profiles:id:read:binary profiles:id:read:json profiles:id:update:binair profiles:id:update:json profiles:read reclassification:update rectificationaudit:check removeforcepause:check rules:create rules:id:read rules:read	rulesfile:check rulesreferential:id:read rulesreport:id:read securityprofiles:create:json securityprofiles:id:read securityprofiles:id:update securityprofiles:read storageaccesslog:read:binary traceability:id:read traceabilitychecks:create units:id:objects:read:binary units:id:objects:read:json units:id:read:json units:id:update units:read units:rules:update units:update unitsWithInheritedRules:read workflows:read
--	---	--	---

« **_v** » : version de l’enregistrement décrit.

- Il s’agit d’un entier.
- Champ peuplé par la solution logicielle Vitam.
- Cardinalité : 1-1
- 0 correspond à l’enregistrement d’origine. Si le numéro est supérieur à 0, alors il s’agit du numéro de version de l’enregistrement.

5.17. Collection VitamSequence

5.17.1. Utilisation de la collection

Cette collection permet de générer des identifiants significatifs pour les enregistrements des collections suivantes :

- AccesContract
- AccessionRegisterDetail
- AccessionRegisterSymbolic
- Agencies
- ArchiveUnitProfile
- Context
- FileFormat
- FileRules
- Griffin
- IngestContract
- PreservationScenario
- Profile
- SecurityProfile

Ces identifiants sont généralement composés d'un préfixe de deux lettres, d'un tiret et d'une suite de six chiffres. Par exemple : IC-027593. Il sont reportés dans les champs Identifiant des collections concernées.

5.17.2. Exemple de JSON stocké en base comprenant l'exhaustivité des champs

```
{
  "_id": "aeaaaaaaaaahkwxukabqteak4q5mtmdyaaaaq",
  "Name": "AC",
  "Counter": 44,
  "_tenant": 1,
  "_v": 0
}
```

5.17.3. Détail des champs

« **_id** » : identifiant unique.

- Il s'agit d'une chaîne de 36 caractères correspondant à un GUID.
- Champ peuplé par la solution logicielle Vitam.
- Cardinalité : 1-1

« **Name** » : préfixe utilisé pour générer un identifiant significatif.

- La valeur contenue dans ce champ doit correspondre à la table de concordance du service VitamCounterService.java. La liste des valeurs possibles est détaillée en annexe.
- Il s'agit d'une chaîne de caractères.
- Champ peuplé par la solution logicielle Vitam.
- Cardinalité : 1-1

« **Counter** » : numéro incrémental.

- Il s'agit du dernier numéro utilisé pour générer un identifiant significatif.
- Il s'agit d'un entier.

- Champ peuplé par la solution logicielle Vitam.
- Cardinalité : 1-1

« **_tenant** » : information sur le tenant.

- Il s'agit de l'identifiant du tenant utilisant l'enregistrement.
- Il s'agit d'un entier.
- Champ peuplé par la solution logicielle Vitam.
- Cardinalité : 1-1

« **_v** » : version de l'enregistrement décrit

- Il s'agit d'un entier.
- Champ peuplé par la solution logicielle Vitam.
- 0 correspond à l'enregistrement d'origine. Si le numéro est supérieur à 0, alors il s'agit du numéro de version de l'enregistrement.
- Cardinalité : 1-1

5.18. Collection Offset

5.18.1. Utilisation de la collection

Cette collection permet de persister les offsets des dernières données reconstruites des offres de stockage lors de la reconstruction au fil de l'eau pour les collections de la base Masterdata.

Il y a une valeur d'offset par couple tenant/collection.

5.18.2. Exemple de JSON stocké en base comprenant l'exhaustivité des champs

```
{
  "_id": ObjectId("507f191e810c19729de860ea"),
  "offset": 1357,
  "collection": "PROFILE",
  "strategyId": "default",
  "_tenant": 1
}
```

5.18.3. Détail des champs

« **_id** » : identifiant unique mongo.

- Il s'agit d'un champ de type mongo : ObjectId(<hexadecimal>).
- Champ peuplé par la solution logicielle Vitam.
- Cardinalité : 1-1

« **offset** » : la valeur de l'offset.

- Il s'agit d'un entier encodé 64 bits.
- Champ peuplé par la solution logicielle Vitam.
- Cardinalité : 1-1

« **collection** » : collection impactée.

- les valeurs possibles sont *UNIT* et *OBJECTGROUP*.

« **strategyId** » : identifiant de la stratégie de stockage.

- Il s'agit d'une chaîne de caractère.

« **_tenant** » : identifiant du tenant.

- Il s'agit d'un entier.
- Champ peuplé par la solution logicielle Vitam.
- Cardinalité : 1-1

6. Base Report

La base Report contient des collections remplies temporairement, utilisées pour construire des rapports d'opérations dans la solution logicielle Vitam. Ces collections sont :

- AuditObjectGroup
- EliminationActionUnit
- EliminationActionObjectGroup
- InvalidUnit
- PreservationReport
- UpdateUnitReport

6.1. Collection AuditObjectGroup

6.1.1. Utilisation de la collection

La collection AuditObjectGroup permet à la solution logicielle Vitam de construire des rapports d'audit. Les données de cette collection sont temporaires et sont supprimées dès que les rapports correspondants sont créés. Il est donc possible de trouver la collection vide.

6.1.2. Détail des champs

« **_id** » : identifiant unique de l'enregistrement

- Il s'agit d'une chaîne de 36 caractères correspondant à un GUID.
- Champ peuplé par la solution logicielle Vitam.
- Cardinalité : 1-1

« **processId** » : identifiant de l'opération d'audit.

- Il s'agit d'une chaîne de caractères.
- Champ peuplé par la solution logicielle Vitam.
- Cardinalité : 1-1

« **_metadata** » objet contenant une liste de paramètres concernant les métadonnées du groupe d'objets. Il est composé comme suit :

- « **id** » : identifiant du groupe d'objets.
- « **status** » : statut de l'action d'audit pour ce groupe d'objets. Il s'agit d'une chaîne de caractères qui peut avoir comme valeurs : OK, WARNING, KO.
- « **opi** » : identifiant de l'opération d'entrée du groupe d'objets.
- « **originatingAgency** » : identifiant du service producteur du groupe d'objets.
- « **parentUnitIds** » : identifiant des unités archivistiques parentes du groupe d'objets ayant été audités.
- « **objectIds** » : identifiants des objets du groupe d'objets.

« **_tenant** » : information sur le tenant

- Il s'agit de l'identifiant du tenant.
- Champ peuplé par la solution logicielle Vitam.

- Cardinalité : 1-1

« **creationDateTime** » : date d'enregistrement du document.

- Il s'agit d'une date.
- Champ peuplé par la solution logicielle Vitam.
- Cardinalité : 1-1

6.2. Collection EliminationActionUnit

6.2.1. Utilisation de la collection

La collection EliminationActionUnit permet à la solution logicielle Vitam de construire des rapports d'élimination d'unités archivistiques. Les données de cette collection sont temporaires et sont supprimées dès que les rapports correspondants sont créés. Il est donc possible de trouver la collection vide.

6.2.2. Détail des champs

« **_id** » : identifiant unique de l'enregistrement.

- Il s'agit d'une chaîne de 36 caractères correspondant à un GUID.
- Champ peuplé par la solution logicielle Vitam.
- Cardinalité : 1-1

« **processId** » : identifiant de l'opération d'élimination.

- Il s'agit d'une chaîne de caractères.
- Champ peuplé par la solution logicielle Vitam.
- Cardinalité : 1-1

_metadata objet contenant une liste de paramètres concernant les métadonnées de l'unité archivistiques. Il est composé comme suit :

- « **id** » : identifiant unique de l'enregistrement.
- « **status** » : statut de l'action d'élimination pour cette unité archivistique. Il s'agit d'une chaîne de caractères qui peut avoir comme valeurs :
 - DELETED,
 - NON_DESTROYABLE_HAS_CHILD_UNITS,
 - GLOBAL_STATUS_KEEP,
 - GLOBAL_STATUS_CONFLICT.
- « **opi** » : identifiant de l'opération d'entrée de cette unité archivistique.
- « **originatingAgency** » : identifiant du service producteur de cette unité archivistique.
- « **objectGroupId** » : identifiant du groupe d'objets attaché à cette unité archivistique.

« **_tenant** » : information sur le tenant.

- Il s'agit de l'identifiant du tenant.
- Champ peuplé par la solution logicielle Vitam.
- Cardinalité : 1-1

« **creationDateTime** » : date d'enregistrement du document.

- Il s'agit d'une date.
- Champ peuplé par la solution logicielle Vitam.
- Cardinalité : 1-1

6.3. Collection EliminationActionObjectGroup

6.3.1. Utilisation de la collection

La collection EliminationActionObjectGroup permet à la solution logicielle Vitam de construire des rapports d'élimination des groupes d'objets techniques. Les données de cette collection sont temporaires et sont supprimées dès que les rapports correspondants sont créés. Il est donc possible de trouver la collection vide.

6.3.2. Détail des champs

« **_id** » : identifiant unique de l'enregistrement.

- Il s'agit d'une chaîne de 36 caractères correspondant à un GUID.
- Champ peuplé par la solution logicielle Vitam.
- Cardinalité : 1-1

« **processId** » : identifiant de l'opération d'élimination.

- Il s'agit d'une chaîne de caractères.
- Champ peuplé par la solution logicielle Vitam.
- Cardinalité : 1-1

« **metadata** » objet contenant une liste de paramètres concernant les métadonnées du groupe d'objets. Il est composé comme suit :

- « **id** » : identifiant du groupe d'objets
- « **status** » : statut de l'action d'élimination pour ce groupe d'objets. Il s'agit d'une chaîne de caractères qui peut avoir comme valeurs : DELETED, PARTIAL_DETACHMENT.
- « **opi** » : identifiant de l'opération d'entrée du groupe d'objets.
- « **originatingAgency** » : identifiant du service producteur du groupe d'objets.
- « **deletedParentUnitIds** » : identifiant des unités archivistiques parentes du groupe d'objets et ayant été supprimées.
- « **objectIds** » : identifiants des objets du groupe d'objets.

« **_tenant** » : information sur le tenant.

- Il s'agit de l'identifiant du tenant.
- Champ peuplé par la solution logicielle Vitam.
- Cardinalité : 1-1

« **creationDate** » : date d'enregistrement du document

- Il s'agit d'une date.
- Champ peuplé par la solution logicielle Vitam.
- Cardinalité : 1-1

6.4. Collection PreservationReport

6.4.1. Utilisation de la collection

La collection PreservationReport permet à la solution logicielle Vitam de construire des rapports de préservation des groupes d'objets techniques. Les données de cette collection sont temporaires et sont supprimées dès que les rapports correspondants sont créés. Il est donc possible de trouver la collection vide.

6.4.2. Détail des champs

« **_id** » : identifiant unique de l'enregistrement.

- Il s'agit d'une chaîne de 36 caractères correspondant à un GUID.
- Champ peuplé par la solution logicielle Vitam.
- Cardinalité : 1-1

« **processId** » : identifiant de l'opération de préservation.

- Il s'agit d'une chaîne de caractères.
- Champ peuplé par la solution logicielle Vitam.
- Cardinalité : 1-1

« **_tenant** » : information sur le tenant.

- Il s'agit de l'identifiant du tenant.
- Champ peuplé par la solution logicielle Vitam.
- Cardinalité : 1-1

« **_action** » : action mise en œuvre dans le cadre de l'opération de préservation.

- Il s'agit d'une chaîne de caractères.
- Peut avoir comme valeurs : ANALYSE, GENERATE, IDENTIFY, EXTRACT.
- Champ peuplé par la solution logicielle Vitam.
- Cardinalité : 1-1

« **analyseResult** » : statut de l'action de préservation pour ce groupe d'objets tel que renvoyé par le griffin.

- Il s'agit d'une chaîne de caractères.
- Champ peuplé par la solution logicielle Vitam.
- Cardinalité : 1-1

« **creationDateTime** » : date d'enregistrement du document.

- Il s'agit d'une date.
- Champ peuplé par la solution logicielle Vitam.
- Cardinalité : 1-1

« **inputName** » : identifiant unique de l'objet ayant servi de source à l'opération de préservation.

- Il s'agit d'une chaîne de 36 caractères correspondant à un GUID.
- Champ peuplé par la solution logicielle Vitam.
- Cardinalité : 1-1

« **objectGroupId** » : identifiant unique du groupe d'objets ayant fait l'objet de l'opération de préservation.

- Il s'agit d'une chaîne de 36 caractères correspondant à un GUID.

- Champ peuplé par la solution logicielle Vitam.
- Cardinalité : 1-1

« **outputName** » : identifiant unique de l'objet créé lors de l'opération de préservation.

- Il s'agit d'une chaîne de 36 caractères correspondant à un GUID.
- Champ peuplé par la solution logicielle Vitam.
- Cardinalité : 1-1

« **status** » : statut de l'action de préservation pour ce groupe d'objets.

- Il s'agit d'une chaîne de caractères.
- Peut avoir comme valeurs : OK, KO, FATAL, WARNING.
- Champ peuplé par la solution logicielle Vitam.
- Cardinalité : 1-1

« **unitId** » : identifiant unique de l'unité archivistique déclarant le groupe d'objets ayant fait l'objet de l'opération de préservation.

- Il s'agit d'une chaîne de 36 caractères correspondant à un GUID.
- Champ peuplé par la solution logicielle Vitam.
- Cardinalité : 1-1

7. Annexes

7.1. Valeurs possibles pour le champ evType du LogBook Operation

L'ensemble des étapes, tâches et traitements sont détaillés dans la documentation Modèle de workflow.

7.2. Valeurs possibles pour le champ evType du LogBook LifeCycle

L'ensemble des étapes, tâches et traitements sont détaillées dans la documentation Modèle de workflow.

7.3. Valeurs possibles pour le champ evTypeProc (type de processus)

<i>Process Type</i>	<i>Valeur</i>	<i>Description</i>
Archive Transfer process	ARCHIVE_TRANSFER	Transfert
Audit type process	AUDIT	Audit
Check type process	CHECK	Vérification
ComputedInheritedRules process	COMPUTE_INHERITED_RULES	Calcul des règles de gestion applicables
Migration	DATA_MIGRATION	Migration
Destruction type process	ELIMINATION	Élimination
Evidence Audit type process	EVIDENCEAUDIT	Audit de traçabilité
DIP export	EXPORT_DIP	Export de DIP
Evidence probativevalue export	EXPORT_PROBATIVE_VALUE	Export d'un relevé de valeur probante
External	EXTERNAL	Opération Externe à VITAM
Filing scheme type process	FILINGSCHEME	Import de plan de classement
Holding scheme type process (tree)	HOLDINGSCHEME	Entrée de plan
Ingest type process	INGEST	Entrée
Ingest test type process	INGEST_TEST	Entrée à blanc
Mass update of archive units	MASS_UPDATE	Modification de masse
Rules Manager process	MASTERDATA	Données de base
Preservation type process	PRESERVATION	Préservation
Reclassification process (attachment/detachment)	RECLASSIFICATION	Modification d'arborescence
Storage Backup type process	STORAGE_BACKUP	Enregistrement du backup »
Storage logbook type process	STORAGE_LOGBOOK	Enregistrement des journaux

<i>Process Type</i>	<i>Valeur</i>	<i>Description</i>
Storage Agencies type process	STORAGE_RULE	Enregistrement du référentiel des services agents
Traceability type process	TRACEABILITY	Sécurisation
Update process	UPDATE	Mise à jour

7.4. Catégories de règles possibles

<i>Prefixe (Peut être modifié)</i>	<i>Type de règle correspondante</i>	<i>Description du type de règle</i>
ACC	AccessRule	Règle d'accès / délai de communicabilité
APP	Appraisal	Règle correspondant à la durée d'utilité administrative (DUA)/ Durée de rétention / conservation
CLASS	ClassificationRule	Règle de classification
DIS	DisseminationRule	Règle de diffusion
REU	ReuseRule	Règle de réutilisation
STO	StorageRule	Durée d'utilité courante / durée de conservation au sens de la loi Informatique et Libertés

7.5. Valeurs possibles pour le champ Status de la collection AccessionRegisterDetail

<i>Status type</i>	<i>Valeur</i>
Le fonds est complet et sauvegardé	STORED_AND_COMPLETED
Le fonds est mis à jour et sauvegardé	STORED_AND_UPDATED
Le fonds n'est pas sauvegardé	UNSTORED

7.6. Valeurs possibles pour le champ Name de la collection VitamSequence

<i>Prefixe</i>	<i>Type de collection correspondante</i>	<i>Description</i>
AC	AccessContract	Contrats d'accès
AG	Agencies	Services agents
AUP	ArchiveUnitProfile	Profil d'unité archivistique
CT	Context	Contextes applicatifs
GR	Griffin	Griffons
IC	IngestContract	Contrats d'entrée
FORMATS	FileFormats	Formats
MC	ManagementContract	Contrats de gestion

Prefixe	Type de collection correspondante	Description
ON	Ontology	Ontologie
PR	Profile	Profils d'archivage
PSC	PreservationScenario	Scénarios de préservation
REGISTER_DETAIL	AccessionRegisterDetail	Détail du registre des fonds
REGISTER_SYMBOLIC	AccessionRegisterSymbolic	Registre des fonds symboliques
RULE	FileRules	Règles de gestion
SEC_PROFILE	SecurityProfiles	Profils de sécurité

7.7. Type d'indexation des chaînes de caractères dans Elasticsearch par collection et par champ

Collection AccessContract

<i>Champ</i>	<i>Type d'indexation</i>
AccessLog	Non analysé
DataObjectVersion	Non analysé
Description	Analysé
Identifiant	Non analysé
Name	Analysé
OriginatingAgencies	Non analysé
RootUnits	Non analysé
Status	Non analysé
ExcludedRootUnits	Non analysé

Collection AccessionRegisterDetail

<i>Champ</i>	<i>Type d'indexation</i>
ArchivalAgreement	Non analysé
OperationIds	Non analysé
OriginatingAgency	Non analysé
Status	Non analysé
SubmissionAgency	Non analysé
Opc	Non analysé
Opi	Non analysé
OpType	Non analysé
LegalStatus	Analysé
AcquisitionInformation	Non analysé
Archive Profile	Non analysé

Collection AccessionRegisterSummary

<i>Champ</i>	<i>Type d'indexation</i>
OriginatingAgency	Non analysé

Collection AccessionRegisterSymbolic

<i>Champ</i>	<i>Type d'indexation</i>
OriginatingAgency	Non analysé

Collection Agencies

<i>Champ</i>	<i>Type d'indexation</i>
Description	Analysé
Identifiant	Non analysé
Name	Analysé

Collection Context

<i>Champ</i>	<i>Type d'indexation</i>
Identifiant	Non analysé
Name	Analysé
Permissions.AccessContracts	Non analysé
Permissions.IngestContracts	Non analysé
SecurityProfile	Non analysé
Status	Non analysé

Collection FileFormat

<i>Champ</i>	<i>Type d'indexation</i>
Comment	Analysé
Extension	Non analysé
Group	Analysé
HasPriorityOverFileFormatID	Non analysé
MimeType	Analysé
Name	Analysé
PUID	Non analysé
Version	Non analysé
VersionPrenom	Non analysé

Collection FileRule

<i>Champ</i>	<i>Type d'indexation</i>
RuleDescription	Analysé
RuleDuration	Non analysé
RuleId	Non analysé
RuleMeasurement	Non analysé
RuleType	Non analysé
RuleValue	Analysé

Collection Griffin

Champ	Type d'indexation
Name	Analysé
Description	Analysé
ExecutableVersion	Non analysé
ExecutableName	Non analysé
Identifiant	Non analysé

Collection IngestContract

Champ	Type d'indexation
ArchiveProfiles	Non analysé
Description	Analysé
Identifiant	Non analysé
LinkParentId	Non analysé
CheckParentLink	Non analysé
Name	Analysé
Status	Non analysé
DataObjectVersion	Non analysé
FormatType	Non analysé

Collection LogbookOperation

Champ	Type d'indexation
events.evDetData.evDetDataType	Non analysé
events.evDetData.LogType	Non analysé
events.evDetData.Hash	Non analysé
events.evDetData.TimeStampToken	Non analysé
events.evDetData.FileName	Analysé
events.evDetData.EvDetailReq	Non analysé
events.evDetData.AgIfTrans	Non analysé
events.evDetData.ArchivalAgreement	Non analysé
events.evDetData.ServiceLevel	Non analysé
events.evDetData.DigestAlgorithm	Non analysé
events.evDetData.SecurisationVersion	Analysé
events.evDetData.validateUnitReport.loadingURI	Analysé
events.evDetData.validateUnitReport.loadingURI	Analysé
events.agIdExt.originatingAgency	Non analysé

Champ	Type d'indexation
events.agIdExt.TransferringAgency	Non analysé
events.agIdExt.ArchivalAgency	Non analysé
events.rightsStatementIdentifier.ArchivalAgreement	Analysé
events.evTypeProc	Non analysé
events.evType	Non analysé
events.outcome	Non analysé
events.outDetail	Non analysé
events.outMessg	Analysé
events.agId	Analysé
events.obId	Non analysé
evId	Non analysé
evIdProc	Non analysé
evIdReq	Non analysé
evParentId	Non analysé
evTypeProc	Non analysé
evType	Non analysé
outcome	Non analysé
outMessg	Analysé
agId	Analysé
outMessg	Analysé
LegalStatus	Non analysé
obId	Analysé

Collection ObjectGroup

Champ	Type d'indexation
FileInfo.CreatingApplicationName	Analysé
FileInfo.CreatingApplicationVersion	Analysé
FileInfo.CreatingOs	Analysé
FileInfo.CreatingOsVersion	Analysé
FileInfo.Filename	Analysé
_glpd	Non analysé
_opi	Non analysé
_ops	Non analysé
_profil	Non analysé

Champ	Type d'indexation
_qualifiers.qualifier	Non analysé
_qualifiers.versions.Algorithm	Non analysé
_qualifiers.versions.DataObjectGroupId	Non analysé
_qualifiers.versions.DataObjectVersion	Non analysé
_qualifiers.versions.FileInfo.CreatingApplicationName	Analysé
_qualifiers.versions.FileInfo.CreatingApplicationVersion	Analysé
_qualifiers.versions.FileInfo.CreatingOs	Analysé
_qualifiers.versions.FileInfo.CreatingOsVersion	Analysé
_qualifiers.versions.FileInfo.FileName	Analysé
_qualifiers.versions.FormatIdentification.Encoding	Non analysé
_qualifiers.versions.FormatIdentification.FormatId	Non analysé
_qualifiers.versions.FormatIdentification.FormatLiteral	Non analysé
_qualifiers.versions.FormatIdentification.MimeType	Non analysé
_qualifiers.versions.MessageDigest	Non analysé
_qualifiers.versions.PhysicalDimensions.Depth.unit	Non analysé
_qualifiers.versions.PhysicalDimensions.Diameter.unit	Non analysé
_qualifiers.versions.PhysicalDimensions.Height.unit	Non analysé
_qualifiers.versions.PhysicalDimensions.Length.unit	Non analysé
_qualifiers.versions.PhysicalDimensions.Shape	Non analysé
_qualifiers.versions.PhysicalDimensions.Thickness.unit	Non analysé
_qualifiers.versions.PhysicalDimensions.Weight.unit	Non analysé
_qualifiers.versions.PhysicalDimensions.Width.unit	Non analysé
_qualifiers.versions.PhysicalId	Non analysé
_qualifiers.versions.Uri	Non analysé
_qualifiers.versions._id	Non analysé
_qualifiers.versions._storage.offerIds	Non analysé
_qualifiers.versions._storage.strategyId	Non analysé
_score	notIndexed
_sp	Non analysé
_sps	Non analysé
_storage.offerIds	Non analysé
_storage.strategyId	Non analysé
_up	Non analysé

<i>Champ</i>	<i>Type d'indexation</i>
_us	Non analysé

Collection Ontology

<i>Champ</i>	<i>Type d'indexation</i>
ApiField	Non analysé
SedaField	Non analysé
Identifieur	Non analysé
Description	Analysé
Type	Non analysé
Origin	Non analysé
ShortName	Non analysé
Collections	Non analysé

Collection PreservationScenario

<i>Champ</i>	<i>Type d'indexation</i>
Name	Analysé
Identifieur	Non analysé
Description	Analysé
ActionList	Non analysé
FormatList	Non analysé
GriffinIdentifieur	Non analysé
Type	Non analysé
Extension	Non analysé
Args	Analysé

Collection Profile

<i>Champ</i>	<i>Type d'indexation</i>
Description	Analysé
Format	Non analysé
Identifieur	Non analysé
Name	Analysé
Path	Non analysé
Status	Non analysé

Collection Unit

Programme Vitam – Modèle de données – v 8.0.

Champ	Type d'indexation
Addressee.BirthName	Analysé

Champ	Type d'indexation
Addressee.BirthPlace.Address	Analysé
Addressee.BirthPlace.City	Analysé
Addressee.BirthPlace.Country	Analysé
Addressee.BirthPlace.Geogname	Analysé
Addressee.BirthPlace.PostalCode	Non analysé
Addressee.BirthPlace.Region	Analysé
Addressee.Corpname	Analysé
Addressee.DeathPlace.Address	Analysé
Addressee.DeathPlace.City	Analysé
Addressee.DeathPlace.Country	Analysé
Addressee.DeathPlace.Geogname	Analysé
Addressee.DeathPlace.PostalCode	Non analysé
Addressee.DeathPlace.Region	Analysé
Addressee.FirstName	Analysé
Addressee.Gender	Analysé
Addressee.GivenName	Analysé
Addressee.Identifier	Non analysé
Addressee.Nationality	Analysé
ArchivalAgencyArchiveUnitIdentifier	Non analysé
ArchiveUnitProfile	Non analysé
AuthorizedAgent.BirthName	Analysé
AuthorizedAgent.BirthPlace.Address	Analysé
AuthorizedAgent.BirthPlace.City	Analysé
AuthorizedAgent.BirthPlace.Country	Analysé
AuthorizedAgent.BirthPlace.Geogname	Analysé
AuthorizedAgent.BirthPlace.PostalCode	Non analysé
AuthorizedAgent.BirthPlace.Region	Analysé
AuthorizedAgent.Corpname	Analysé
AuthorizedAgent.DeathPlace.Address	Analysé
AuthorizedAgent.DeathPlace.City	Analysé
AuthorizedAgent.DeathPlace.Country	Analysé
AuthorizedAgent.DeathPlace.Geogname	Analysé
AuthorizedAgent.DeathPlace.PostalCode	Non analysé

Champ	Type d'indexation
AuthorizedAgent.DeathPlace.Region	Analysé
AuthorizedAgent.FirstName	Analysé
AuthorizedAgent.Gender	Analysé
AuthorizedAgent.GivenName	Analysé
AuthorizedAgent.Identifier	Non analysé
AuthorizedAgent.Nationality	Analysé
Coverage.Jurisdictional	Analysé
Coverage.Spatial	Analysé
Coverage.Temporal	Analysé
CustodialHistory.CustodialHistoryFile.DataObjectGroupReferenceId	Non analysé
CustodialHistory.CustodialHistoryItem	Analysé
Description	Analysé
DescriptionLanguage	Non analysé
DescriptionLevel	Non analysé
DocumentType	Analysé
Event.EventDetail	Analysé
Event.EventIdentifier	Non analysé
Event.EventType	Analysé
FilePlanPosition	Non analysé
Gps.GpsAltitude	Non analysé
Gps.GpsAltitudeRef	Non analysé
Gps.GpsDateStamp	Non analysé
Gps.GpsLatitude	Non analysé
Gps.GpsLatitudeRef	Non analysé
Gps.GpsLongitude	Non analysé
Gps.GpsLongitudeRef	Non analysé
Gps.GpsVersionID	Non analysé
Non analysé.Non analyséContent	Non analysé
Non analysé.Non analyséReference	Non analysé
Non analysé.Non analyséType	Non analysé
Language	Non analysé
OriginatingAgency.Identifier	Non analysé
OriginatingAgencyArchiveUnitIdentifier	Non analysé

Champ	Type d'indexation
OriginatingSystemId	Non analysé
Recipient.BirthName	Analysé
Recipient.BirthPlace.Address	Analysé
Recipient.BirthPlace.City	Analysé
Recipient.BirthPlace.Country	Analysé
Recipient.BirthPlace.Geogname	Analysé
Recipient.BirthPlace.PostalCode	Non analysé
Recipient.BirthPlace.Region	Analysé
Recipient.Corpname	Analysé
Recipient.DeathPlace.Address	Analysé
Recipient.DeathPlace.City	Analysé
Recipient.DeathPlace.Country	Analysé
Recipient.DeathPlace.Geogname	Analysé
Recipient.DeathPlace.PostalCode	Non analysé
Recipient.DeathPlace.Region	Analysé
Recipient.FirstName	Analysé
Recipient.Gender	Analysé
Recipient.GivenName	Analysé
Recipient.Identifier	Non analysé
Recipient.Nationality	Analysé
RelatedObjectReference.IsPartOf.ArchiveUnitRefId	Non analysé
RelatedObjectReference.IsPartOf.DataObjectReference.DataObjectGroupReferenceId	Non analysé
RelatedObjectReference.IsPartOf.DataObjectReference.DataObjectReferenceId	Non analysé
RelatedObjectReference.IsPartOf.RepositoryArchiveUnitPID	Non analysé
RelatedObjectReference.IsPartOf.RepositoryObjectPID	Non analysé
RelatedObjectReference.IsVersionOf.ArchiveUnitRefId	Non analysé
RelatedObjectReference.IsVersionOf.DataObjectReference.DataObjectGroupReferenceId	Non analysé
RelatedObjectReference.IsVersionOf.DataObjectReference.DataObjectReferenceId	Non analysé
RelatedObjectReference.IsVersionOf.RepositoryArchiveUnitPID	Non analysé
RelatedObjectReference.IsVersionOf.RepositoryObjectPID	Non analysé
RelatedObjectReference.References.ArchiveUnitRefId	Non analysé
RelatedObjectReference.References.DataObjectReference.DataObjectGroupReferenceId	Non analysé
RelatedObjectReference.References.DataObjectReference.DataObjectReferenceId	Non analysé

Champ	Type d'indexation
RelatedObjectReference.References.RepositoryArchiveUnitPID	Non analysé
RelatedObjectReference.References.RepositoryObjectPID	Non analysé
RelatedObjectReference.Replaces.ArchiveUnitRefId	Non analysé
RelatedObjectReference.Replaces.DataObjectReference.DataObjectGroupReferenceId	Non analysé
RelatedObjectReference.Replaces.DataObjectReference.DataObjectReferenceId	Non analysé
RelatedObjectReference.Replaces.ExternalReference	Analysé
RelatedObjectReference.Replaces.RepositoryArchiveUnitPID	Non analysé
RelatedObjectReference.Replaces.RepositoryObjectPID	Non analysé
RelatedObjectReference.Requires.ArchiveUnitRefId	Non analysé
RelatedObjectReference.Requires.DataObjectReference.DataObjectGroupReferenceId	Non analysé
RelatedObjectReference.Requires.DataObjectReference.DataObjectReferenceId	Non analysé
RelatedObjectReference.Requires.RepositoryArchiveUnitPID	Non analysé
RelatedObjectReference.Requires.RepositoryObjectPID	Non analysé
Sender.Activity	Non analysé
Sender.BirthName	Analysé
Sender.BirthPlace.Address	Analysé
Sender.BirthPlace.City	Analysé
Sender.BirthPlace.Country	Analysé
Sender.BirthPlace.Geogname	Analysé
Sender.BirthPlace.PostalCode	Non analysé
Sender.BirthPlace.Region	Analysé
Sender.DeathPlace.Address	Analysé
Sender.DeathPlace.City	Analysé
Sender.DeathPlace.Country	Analysé
Sender.DeathPlace.Geogname	Analysé
Sender.DeathPlace.PostalCode	Non analysé
Sender.DeathPlace.Region	Analysé
Sender.FirstName	Analysé
Sender.Function	Non analysé
Sender.Gender	Analysé
Sender.GivenName	Analysé
Sender.Identifier	Non analysé

Champ	Type d'indexation
Sender.Mandate	Analysé
Sender.Nationality	Analysé
Sender.Position	Analysé
Sender.Role	Analysé
Signature.Masterdata.Value	Non analysé
Signature.ReferencedObject.SignedObjectDigest.Algorithm	Non analysé
Signature.ReferencedObject.SignedObjectDigest.MessageDigest	Non analysé
Signature.ReferencedObject.SignedObjectId	Non analysé
Signature.Signer.Activity	Non analysé
Signature.Signer.BirthName	Analysé
Signature.Signer.BirthPlace.Address	Analysé
Signature.Signer.BirthPlace.City	Analysé
Signature.Signer.BirthPlace.Country	Analysé
Signature.Signer.BirthPlace.Geogname	Analysé
Signature.Signer.BirthPlace.PostalCode	Non analysé
Signature.Signer.BirthPlace.Region	Analysé
Signature.Signer.Corpname	Analysé
Signature.Signer.DeathPlace.Address	Analysé
Signature.Signer.DeathPlace.City	Analysé
Signature.Signer.DeathPlace.Country	Analysé
Signature.Signer.DeathPlace.Geogname	Analysé
Signature.Signer.DeathPlace.PostalCode	Non analysé
Signature.Signer.DeathPlace.Region	Analysé
Signature.Signer.FirstName	Analysé
Signature.Signer.Fullname	Analysé
Signature.Signer.Function	Non analysé
Signature.Signer.Gender	Analysé
Signature.Signer.GivenName	Analysé
Signature.Signer.Identifier	Non analysé
Signature.Signer.Nationality	Analysé
Signature.Signer.Position	Analysé
Signature.Signer.Role	Analysé
Signature.Validator.Activity	Non analysé

Champ	Type d'indexation
Signature.Validator.BirthName	Analysé
Signature.Validator.BirthPlace.Address	Analysé
Signature.Validator.BirthPlace.City	Analysé
Signature.Validator.BirthPlace.Country	Analysé
Signature.Validator.BirthPlace.Geogname	Analysé
Signature.Validator.BirthPlace.PostalCode	Non analysé
Signature.Validator.BirthPlace.Region	Analysé
Signature.Validator.Corpname	Analysé
Signature.Validator.DeathPlace.Address	Analysé
Signature.Validator.DeathPlace.City	Analysé
Signature.Validator.DeathPlace.Country	Analysé
Signature.Validator.DeathPlace.Geogname	Analysé
Signature.Validator.DeathPlace.PostalCode	Non analysé
Signature.Validator.DeathPlace.Region	Analysé
Signature.Validator.FirstName	Analysé
Signature.Validator.FullName	Analysé
Signature.Validator.Function	Non analysé
Signature.Validator.Gender	Analysé
Signature.Validator.GivenName	Analysé
Signature.Validator.Identifier	Non analysé
Signature.Validator.Nationality	Analysé
Signature.Validator.Position	Analysé
Signature.Validator.Role	Analysé
Source	Analysé
Status	Non analysé
SubmissionAgency.Identifier	Non analysé
SystemId	Non analysé
Tag	Non analysé
Title	Analysé
TransferringAgencyArchiveUnitIdentifier	Non analysé
Transmitter.Activity	Non analysé
Transmitter.BirthName	Analysé
Transmitter.BirthPlace.Address	Analysé

Champ	Type d'indexation
Transmitter.BirthPlace.City	Analysé
Transmitter.BirthPlace.Country	Analysé
Transmitter.BirthPlace.Geogname	Analysé
Transmitter.BirthPlace.PostalCode	Non analysé
Transmitter.BirthPlace.Region	Analysé
Transmitter.DeathPlace.Address	Analysé
Transmitter.DeathPlace.City	Analysé
Transmitter.DeathPlace.Country	Analysé
Transmitter.DeathPlace.Geogname	Analysé
Transmitter.DeathPlace.PostalCode	Non analysé
Transmitter.DeathPlace.Region	Analysé
Transmitter.FirstName	Analysé
Transmitter.Function	Non analysé
Transmitter.Gender	Analysé
Transmitter.GivenName	Analysé
Transmitter.Identifier	Non analysé
Transmitter.Nationality	Analysé
Transmitter.Position	Analysé
Transmitter.Role	Analysé
Type	Non analysé
Version	Non analysé
Writer.Activity	Non analysé
Writer.BirthName	Analysé
Writer.BirthPlace.Address	Analysé
Writer.BirthPlace.City	Analysé
Writer.BirthPlace.Country	Analysé
Writer.BirthPlace.Geogname	Analysé
Writer.BirthPlace.PostalCode	Non analysé
Writer.BirthPlace.Region	Analysé
Writer.DeathPlace.Address	Analysé
Writer.DeathPlace.City	Analysé
Writer.DeathPlace.Country	Analysé
Writer.DeathPlace.Geogname	Analysé

Champ	Type d'indexation
Writer.DeathPlace.PostalCode	Non analysé
Writer.DeathPlace.Region	Analysé
Writer.FirstName	Analysé
Writer.Function	Non analysé
Writer.Gender	Analysé
Writer.GivenName	Analysé
Writer.Identifiant	Non analysé
Writer.Nationality	Analysé
Writer.Position	Analysé
Writer.Role	Analysé
_elimination.DestroyableOriginatingAgencies	Non analysé
_elimination.ExtendedInfo.ExtendedInfoDetails.DestroyableOriginatingAgencies	Non analysé
_elimination.ExtendedInfo.ExtendedInfoDetails.NonDestroyableOriginatingAgencies	Non analysé
_elimination.ExtendedInfo.ExtendedInfoDetails.ParentUnitId	Non analysé
_elimination.ExtendedInfo.ExtendedInfoType	Non analysé
_elimination.GlobalStatus	Non analysé
_elimination.NonDestroyableOriginatingAgencies	Non analysé
_elimination.OperationId	Non analysé
_history.data._mgt.ClassificationRule.ClassificationAudience	Non analysé
_history.data._mgt.ClassificationRule.ClassificationLevel	Non analysé
_history.data._mgt.ClassificationRule.ClassificationOwner	Analysé
_history.data._mgt.ClassificationRule.Inheritance.PreventRulesId	Non analysé
_history.data._mgt.ClassificationRule.Rules.Rule	Non analysé
_implementationVersion	Non analysé
_mgt.AccessRule.Inheritance.PreventRulesId	Non analysé
_mgt.AccessRule.Rules.Rule	Non analysé
_mgt.AppraisalRule.FinalAction	Non analysé
_mgt.AppraisalRule.Inheritance.PreventRulesId	Non analysé
_mgt.AppraisalRule.Rules.Rule	Non analysé
_mgt.ClassificationRule.ClassificationAudience	Non analysé
_mgt.ClassificationRule.ClassificationLevel	Non analysé
_mgt.ClassificationRule.ClassificationOwner	Analysé
_mgt.ClassificationRule.Inheritance.PreventRulesId	Non analysé

Champ	Type d'indexation
_mgt.ClassificationRule.Rules.ClassificationAudience	Analysé
_mgt.ClassificationRule.Rules.Rule	Non analysé
_mgt.DisseminationRule.Inheritance.PreventRulesId	Non analysé
_mgt.DisseminationRule.Rules.Rule	Non analysé
_mgt.ReuseRule.Inheritance.PreventRulesId	Non analysé
_mgt.ReuseRule.Rules.Rule	Non analysé
_mgt.StorageRule.FinalAction	Non analysé
_mgt.StorageRule.Inheritance.PreventRulesId	Non analysé
_mgt.StorageRule.Rules.Rule	Non analysé
_og	Non analysé
_opi	Non analysé
_ops	Non analysé
_sedaVersion	Non analysé
_sp	Non analysé
_sps	Non analysé
_storage.offerIds	Non analysé
_storage.strategyId	Non analysé
_unitType	Non analysé
_up	Non analysé
_us	Non analysé

Collection SecurityProfile

Champ	Type d'indexation
Identifiant	Non analysé
Name	Analysé
Permissions	Non analysé

7.8. Correspondances des champs spéciaux dans Vitam

Les champs dont le nom est préfixé d'un « _ » ne sont pas accessibles directement, une correspondance est nécessaire pour y accéder.

Collection AccessContract

Champ	Champ interne
#id	_id
#tenant	_tenant

<i>Champ</i>	<i>Champ interne</i>
#version	_v

Collection AccessionRegisterDetail

<i>Champ</i>	<i>Champ interne</i>
#id	_id
#tenant	_tenant

Collection AccessionRegisterSummary

<i>Champ</i>	<i>Champ interne</i>
#id	_id
#tenant	_tenant
#version	_v

Collection AccessionRegisterSymbolic

<i>Champ</i>	<i>Champ interne</i>
#id	_id
#tenant	_tenant
#version	_v

Collection Agencies

<i>Champ</i>	<i>Champ interne</i>
#id	_id
#tenant	_tenant
#version	_v

Collection Context

<i>Champ</i>	<i>Champ interne</i>
#id	_id
#version	_v

Collection FileFormat

<i>Champ</i>	<i>Champ interne</i>
#id	_id
#version	_v

Collection FileRule

<i>Champ</i>	<i>Champ interne</i>
#id	_id

<i>Champ</i>	<i>Champ interne</i>
#tenant	_tenant
#version	_v

Collection Griffin

<i>Champ</i>	<i>Champ interne</i>
#id	_id
#tenant	_tenant
#version	_v

Collection IngestContract

<i>Champ</i>	<i>Champ interne</i>
#id	_id
#tenant	_tenant
#version	_v

Collection LogbookLifeCycle

<i>Champ</i>	<i>Champ interne</i>
#id	_id
#tenant	_tenant
#version	_v

Collection LogbookOperation

<i>Champ</i>	<i>Champ interne</i>
#id	_id
#tenant	_tenant

Collection ObjectGroup

<i>Champ</i>	<i>Champ interne</i>
#id	_id
#profil	_profil
#qualifiers	_qualifiers
#size	_qualifiers.versions.size
#nbojects	_nbc
#originating_agency	_sp
#originating_agencies	_sps
#unitups	_up

Champ	Champ interne
#storage	_storage
#operations	_ops
#opi	_opi
#score	_score
#version	_v
#tenant	_tenant

Collection Profile

Champ	Champ interne
#id	_id
#tenant	_tenant
#version	_v

Collection Ontology

Champ	Champ interne
#id	_id
#tenant	_tenant
#version	_v

Collection PreservationScenario

Champ	Champ interne
#id	_id
#tenant	_tenant
#version	_v

Collection Unit

Le champs « _uds » n'est pas accessible en externe.

Champ	Champ interne
#id	_id
#management	_mgt
#min	_min
#max	_max
#nbunits	_nbc
#object	_og
#originating_agency »	_sp
#originating_agencies	_sps

Programme Vitam – Modèle de données – v 8.0.

Champ	Champ interne
#unitups	_up
#allunitups	_us
#nbunits	_nbc
#unitType	_unitType
#storage	_storage
#operations	_ops
#opi	_opi
#opts	_opts
#sedaVersion	_sedaVersion
#implementationVersion	_implementationVersion
#score	_score
#version	_v
#tenant	_tenant