



VITAM - Modèle de données

Version 1.4.0

VITAM

juin 26, 2018

Table des matières

1	Introduction	1
1.1	Avertissement	1
1.2	Objectif du document	1
1.3	Création des index	1
1.4	Généralités	2
1.4.1	Cardinalité	2
1.4.2	Nommage des champs	2
1.4.3	Collections et bases	2
1.4.4	Identifiants	2
1.4.5	Dates	2
1.4.6	Limite de caractères acceptés dans les champs	2
1.4.7	Type d'indexation dans elasticSearch	2
2	Base Identity	3
2.1	Collections contenues dans la base	3
2.2	Collection Certificate	3
2.2.1	Utilisation de la collection Certificate	3
2.2.2	Exemple de JSON stocké en base comprenant l'exhaustivité des champs	3
2.2.3	Détail des champs du JSON stocké dans la collection	4
2.3	Collection PersonalCertificate	4
2.3.1	Utilisation de la collection PersonalCertificate	4
2.3.2	Exemple de JSON stocké en base comprenant l'exhaustivité des champs	4
2.3.3	Détail des champs du JSON stocké dans la collection	5
3	Base Logbook	7
3.1	Collections contenues dans la base	7
3.2	Collection LogbookOperation	7
3.2.1	Utilisation de la collection LogbookOperation	7
3.2.2	Exemple de JSON stocké en base comprenant l'exhaustivité des champs de la collection LogbookOperation	8
3.2.3	Détail des champs du JSON stocké dans la collection	9
3.2.4	Champ présents dans les events	14
3.2.5	Détail des champs du JSON stocké en base spécifiques à une opération de sécurisation des journaux d'opération et de cycle de vie	14
3.3	Collection LogbookLifeCycleUnit	16
3.3.1	Utilisation de la collection LogbookLifeCycleUnit	16

3.3.2	Exemple de JSON stocké en base comprenant l'exhaustivité des champs de la collection LogbookLifeCycleUnit	16
3.3.3	Détail des champs du JSON stocké en base	17
3.3.4	Champs présents dans les events	20
3.3.5	Détail des champs du JSON stocké en base spécifiques à une mise à jour	20
3.4	Collection LogbookLifeCycleObjectGroup	21
3.4.1	Utilisation de la collection LogbookLifeCycleObjectGroup	21
3.4.2	Exemple de JSON stocké en base comprenant l'exhaustivité des champs	21
3.4.3	Détail des champs du JSON stocké en base	22
3.4.4	Champs présents dans les events	25
3.5	Collection Offset	26
3.5.1	Utilisation de la collection	26
3.5.2	Exemple de JSON stocké en base comprenant l'exhaustivité des champs	26
3.5.3	Détail des champs	26
4	Base MetaData	27
4.1	Collections contenues dans la base	27
4.2	Collection Unit	27
4.2.1	Utilisation de la collection Unit	27
4.2.2	Exemple de XML en entrée	27
4.2.3	Exemple de Json stocké dans la collection Unit	28
4.2.4	Détail du JSON	30
4.3	Collection ObjectGroup	35
4.3.1	Utilisation de la collection ObjectGroup	35
4.3.2	Exemple de XML	35
4.3.3	Exemple de JSON stocké en base	36
4.3.4	Détail des champs du JSON	38
4.4	Collection Offset	41
4.4.1	Utilisation de la collection	41
4.4.2	Exemple de JSON stocké en base comprenant l'exhaustivité des champs	41
4.4.3	Détail des champs	41
5	Base MasterData	43
5.1	Collections contenues dans la base	43
5.1.1	Collections Cross-tenant	43
5.2	Collection AccessContract	44
5.2.1	Utilisation de la collection AccessContract	44
5.2.2	Exemple d'un fichier d'import de contrat d'accès	44
5.2.3	Exemple de JSON stocké en base comprenant l'exhaustivité des champs de la collection AccesContract	44
5.2.4	Détail des champs	45
5.3	Collection AccessionRegisterDetail	47
5.3.1	Utilisation de la collection AccessionRegisterDetail	47
5.3.2	Exemple de la description dans le XML d'entrée	47
5.3.3	Exemple de JSON stocké en base comprenant l'exhaustivité des champs	47
5.3.4	Détail des champs	48
5.4	Collection AccessionRegisterSummary	52
5.4.1	Utilisation de la collection	52
5.4.2	Exemple de la description dans le bordereau de transfert	52
5.4.3	Exemple de JSON stocké en base comprenant l'exhaustivité des champs	52
5.4.4	Détail des champs	53
5.5	Collection Agencies	55
5.5.1	Utilisation de la collection Agencies	55

5.5.2	Exemple de JSON stocké en base comprenant l'exhaustivité des champs de la collection Agencys	56
5.5.3	Détail des champs	56
5.6	Collection Context	56
5.6.1	Utilisation de la collection	56
5.6.2	Exemple d'un fichier d'import de contexte applicatif	57
5.6.3	Exemple de JSON stocké en base comprenant l'exhaustivité des champs de la collection Context	57
5.6.4	Détail des champs	59
5.7	Collection FileFormat	60
5.7.1	Utilisation de la collection FileFormat	60
5.7.2	Exemple de la description d'un format dans le XML d'entrée	60
5.7.3	Exemple de JSON stocké en base comprenant l'exhaustivité des champs de la collection FileFormat	60
5.7.4	Détail des champs du JSON stocké en base	61
5.8	Collection FileRules	63
5.8.1	Utilisation de la collection FileRules	63
5.8.2	Exemple de JSON stocké en base comprenant l'exhaustivité des champs de la collection FileRules	64
5.8.3	Détail des champs	64
5.9	Collection IngestContract	65
5.9.1	Utilisation de la collection	65
5.9.2	Exemple d'un fichier d'import de contrat	66
5.9.3	Exemple de JSON stocké en base comprenant l'exhaustivité des champs de la collection IngestContract	66
5.9.4	Détail des champs de la collection IngestContract	66
5.10	Collection Profile	68
5.10.1	Utilisation de la collection profile	68
5.10.2	Exemple d'un fichier d'import de notices de Profils d'archivage	68
5.10.3	Exemple de JSON stocké en base comprenant l'exhaustivité des champs de la collection Profile	69
5.10.4	Détail des champs	69
5.11	Collection SecurityProfile	71
5.11.1	Utilisation de la collection	71
5.11.2	Exemple de JSON stocké en base comprenant l'exhaustivité des champs	71
5.11.3	Détail des champs	71
5.12	Collection VitamSequence	72
5.12.1	Utilisation de la collection	72
5.12.2	Exemple de JSON stocké en base comprenant l'exhaustivité des champs	72
5.12.3	Détail des champs	72
5.13	Collection ArchiveUnitProfile	73
5.13.1	Utilisation de la collection	73
5.13.2	Exemple d'un fichier d'import de document type	73
5.13.3	Exemple de JSON stocké en base comprenant l'exhaustivité des champs de la collection ArchiveUnitProfile	74
5.13.4	Détail des champs de la collection ArchiveUnitProfile	74
5.14	Collection Ontology	75
5.14.1	Utilisation de la collection	75
5.14.2	Exemple d'un fichier d'import d'ontology	76
5.14.3	Détail des champs de la collection Ontology	76
6	Annexes	79
6.1	Valeurs possibles pour le champ evType du LogBook Operation	79
6.2	Valeurs possibles pour le champ evType du LogBook LifeCycle	79
6.3	Valeurs possibles pour le champ evTypeProc (type de processus)	79

6.4	Catégories de règles possibles	80
6.5	Valeurs possibles pour le champ Status de la collection AccessionRegisterDetail	80
6.6	Valeurs possibles pour le champ Name de la collection VitamSecquence	80
6.7	Diagramme du modèle de données Vitam	81
6.8	Type d'indexation dans Elasticsearch par collection et par champ	81
6.9	Correspondances des champs spéciaux dans Vitam	89

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 Cardinalité

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

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.

1.4.2 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 ayant avec 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.

1.4.3 Collections et bases

Les bases Mongo sont organisées par bases et collections.

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

1.4.4 Identifiants

Il existe plusieurs types d'identifiant :

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

2.1 Collections contenues dans la base

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

2.2 Collection Certificate

2.2.1 Utilisation de la collection Certificate

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

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

```
{
  "_id": "aeaaaaaaaaahwgpj2aa2fgak7cxqdy6aaaaaq",
  "SubjectDN": "CN=ihm-recette, O=vitam, L=paris, ST=idf, C=fr",
  "ContextId": "CT-000001",
  "SerialNumber": 254,
  "IssuerDN": "CN=ca_intermediate_client-external, OU=authorities, O=vitam, L=paris, C=fr",
  "Certificate": "Q2VydGhmaWNhdGU6CiAgICBEYXRhOgogICAgICAgIFZlcnNpb246IDMgKDB4MikKICA

  [...]

  kbE4KM08yV1dIRlJMWnpQRWZ4eXlxMm1TbVdsauUvUzZUzJVVeswamxobStpbThPa29mZmlLbXlodVpWS3
  S0tRU5EIEINFU1RJRklDQVRFLS0tLS0="
}
```

2.2.3 Détail des champs du JSON stocké dans la collection

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

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

« **SerialNumber** » : Numéro de série du certificat applicatif

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

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

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

« **Certificate** » : Certificat

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

2.3 Collection PersonalCertificate

2.3.1 Utilisation de la collection PersonalCertificate

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

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

```
{
  "_id": "aeaaaaaaaaagcksdyabraialabxzw7aqaaaaq",
  "SubjectDN": "O=VITAM, L=Paris, C=FR",
  "SerialNumber": 2,
  "IssuerDN": "O=VITAM, L=Paris, C=FR",
  "Certificate": "MIIFRjCCAy6gAwIBAgIBAjANBgkqhkiG9w0BAQsFADAtMQswCQYDV...iaA==",
  "Hash": "6088f19bc7d328f301168c064d6fda93a6c4ced9d5c56810c4f70e21e77d841d"
}
```

2.3.3 Détail des champs du JSON stocké dans la collection

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

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

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

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

« **Certificate** » : Certificat au format DER encodé en Base64.

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

« **Hash** » : Hash (SHA256) du certificat

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

3.1 Collections contenues dans la base

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.

3.2 Collection LogbookOperation

3.2.1 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 (implémentée dans la release en cours)
- Données de référence (implémentée dans la release en cours)
- Elimination (non implémentée dans la release en cours)
- Entrée (implémentée dans la release en cours)
- Mise à jour (implémentée dans la release en cours)
- Préservation (non implémentée dans la release en cours)
- Sécurisation (implémentée dans la release en cours)
- Vérification (implémentée dans la release en cours)

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

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

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

```
{
  "_id": "aedqaaaaacec45rhabfy2ak6ox625ciaaaaq",
  "evId": "aedqaaaaacec45rhabfy2ak6ox625ciaaaaq",
  "evParentId": null,
  "evType": "PROCESS_SIP_UNITARY",
  "evDateTime": "2017-09-12T12:08:33.166",
  "evDetData": "{\n  \"EvDetailReq\" : \"Cartes postales (Grande Collecte)\",\n  \
  ↪\"EvDateTimeReq\" : \"2016-10-12T16:28:40\", \n  \"ArchivalAgreement\" : \
  ↪\"ArchivalAgreement0\", \n  \"ServiceLevel\" : null\n}",
  "evIdProc": "aedqaaaaacec45rhabfy2ak6ox625ciaaaaq",
  "evTypeProc": "INGEST",
  "outcome": "STARTED",
  "outDetail": "PROCESS_SIP_UNITARY.STARTED",
  "outMessg": "Début du processus d'entrée du SIP :_\
  ↪aedqaaaaacec45rhabfy2ak6ox625ciaaaaq",
  "agId": "{ \"Name\" : \"vitam-iaas-app-01\", \"Role\" : \"ingest-external\", \"ServerId\
  ↪\":1211004455, \"SiteId\" :1, \"GlobalPlatformId\" :137262631}",
  "agIdApp": "CT-000001",
  "evIdAppSession": "MyApplicationId-ChangeIt",
  "evIdReq": "aedqaaaaacec45rhabfy2ak6ox625ciaaaaq",
  "agIdExt": "{ \"originatingAgency\" : \"Identif0\", \"TransferringAgency\" : \
  ↪\"ARCHIVES DEPARTEMENTALES DE LA VENDEE\", \"ArchivalAgency\" : \"ARCHIVES_\
  ↪DEPARTEMENTALES DE LA VENDEE\"}",
  "rightsStatementIdentifier": "{ \"ArchivalAgreement\" : \"ArchivalAgreement0\"}",
  "obId": "aedqaaaaacec45rhabfy2ak6ox625ciaaaaq",
  "obIdReq": null,
  "obIdIn": "Cartes postales (Grande Collecte)",
  "events": [
    {
      "evId": "aedqaaaaacec45rhabfy2ak6ox625ciaaaaq",
      "evParentId": null,
      "evType": "STP_SANITY_CHECK_SIP.STARTED",
      "evDateTime": "2017-09-12T12:08:33.166",
      "evDetData": null,
      "evIdProc": "aedqaaaaacec45rhabfy2ak6ox625ciaaaaq",
      "evTypeProc": "INGEST",
      "outcome": "OK",
      "outDetail": "STP_SANITY_CHECK_SIP.STARTED.OK",
      "outMessg": "Début du processus des contrôles préalables à l'entrée",
      "agId": "{ \"Name\" : \"vitam-iaas-app-01\", \"Role\" : \"ingest-external\", \
      ↪\"ServerId\" :1211004455, \"SiteId\" :1, \"GlobalPlatformId\" :137262631}",
      "evIdReq": "aedqaaaaacec45rhabfy2ak6ox625ciaaaaq",
      "obId": "aedqaaaaacec45rhabfy2ak6ox625ciaaaaq"
    },
    {
      "evId": "aedqaaaaacec45rhabfy2ak6ox625ciaaaaq",
      "evParentId": null,
      "evType": "STP_SANITY_CHECK_SIP",
      "evDateTime": "2017-09-12T12:08:33.219",
      "evDetData": null,
      "evIdProc": "aedqaaaaacec45rhabfy2ak6ox625ciaaaaq",
      "evTypeProc": "INGEST",
      "outcome": "OK",
    }
  ]
}
```

(suite sur la page suivante)

(suite de la page précédente)

```

        "outDetail": "STP_SANITY_CHECK_SIP.OK",
        "outMessg": "Début du processus des contrôles préalables à l'entrée",
        "agId": "{ \"Name\": \"vitam-iaas-app-01\", \"Role\": \"ingest-external\", \
↔ \"ServerId\": 1211004455, \"SiteId\": 1, \"GlobalPlatformId\": 137262631 }",
        "evIdReq": "aedqaaaaacec45rhabfy2ak6ox625ciaaaaq",
        "obId": "aedqaaaaacec45rhabfy2ak6ox625ciaaaaq"
    },
    {
        "evId": "aedqaaaaacec45rhabfy2ak6ox625ciaaaaq",
        "evParentId": "aedqaaaaacec45rhabfy2ak6ox625ciaaaaq",
        "evType": "SANITY_CHECK_SIP",
        "evDateTime": "2017-09-12T12:08:33.219",
        "evDetData": null,
        "evIdProc": "aedqaaaaacec45rhabfy2ak6ox625ciaaaaq",
        "evTypeProc": "INGEST",
        "outcome": "OK",
        "outDetail": "SANITY_CHECK_SIP.OK",
        "outMessg": "Succès du contrôle sanitaire",
        "agId": "{ \"Name\": \"vitam-iaas-app-01\", \"Role\": \"ingest-external\", \
↔ \"ServerId\": 1211004455, \"SiteId\": 1, \"GlobalPlatformId\": 137262631 }",
        "evIdReq": "aedqaaaaacec45rhabfy2ak6ox625ciaaaaq",
        "obId": "aedqaaaaacec45rhabfy2ak6ox625ciaaaaq"
    },
    {
        [...]
    }
],
"_tenant": 0,
"_v": 1,
"_lastPersistedDate": "2017-09-12T12:08:33.219"
}

```

3.2.3 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 règle classique est que sa valeur est égale à celle du champ evIdReq. Dans le cas d'une requête déclenchant plusieurs opérations, comme une mise à jour de règles de gestion par exemple, alors ce champ aura pour la première opération la même valeur que le champ evIdReq, puis celle du champ evIdProc pour les suivantes.
- 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.
- Champs 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 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 CHECK_SEDA, il s'agit de STP_INGEST_CONTROL_SIP.
 - Ce champ est toujours à 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 de l'opération
- Issu de la définition du workflow 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 :[3digits 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 JSON composé des champs suivants :
 - evDetDataType : structure impactée. Chaîne de caractères. Doit correspondre à une valeur de l'énumération LogbookEvDetDataType
 - 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 :[3digits de millisecondes].
 - ServiceLevel : niveau de service. Chaîne de caractères. Reprend le champ ServiceLevel du message ArchiveTransfer.
 - AcquisitionInformation : modalités d'entrée des archives. Chaîne de caractères. Reprend le champ AcquisitionInformation du message ArchiveTransfer
 - LegalStatus : statut des archives échangés. Chaîne de caractères. Reprend le champ LegalStatus 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.
- « **evIdProc** » (**event Identifier Process**) : identifiant du processus.
- Il s'agit d'une chaîne de 36 caractères.
 - Toutes les mêmes entrées du journal des opérations partagent la même valeur, qui est celle du champ « _id »

- 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 qui effectue l'action, 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. 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 le PID de l'agent. Ce champ est calculé par le journal à partir de ServerIdentifier. Exemple : `{\"name\": \"ingest-internal_1\", \"role\": \"ingest-internal\", \"pid\": 425367}`
- 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.

« **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 incluante.
- « **evIdReq** » (**event Identifier Request**) : identifiant de la requête déclenchant l'opération.
- Il s'agit d'une chaîne de 36 caractères.
 - Cardinalité : 1-1
 - 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.
 - Ce champ existe pour les structures incluant et incluses.
- « **agIdExt** » (**agent Identifier External**) : identifiant de l'agent externe mentionné dans le message ArchiveTransfer.
- Il s'agit pour un ingest d'un JSON comprenant les champs suivants :
 - OriginatingAgency : identifiant du service producteur. Il s'agit d'une chaîne de caractères. Reprend le contenu du champ OriginatingAgencyIdentifier du message ArchiveTransfer.
 - TransferringAgency : identifiant du service de transfert. Il s'agit d'une chaîne de caractères. Reprend le contenu du champ TransferringAgencyIdentifier du message ArchiveTransfer.
 - ArchivalAgency : identifiant du service d'archivage. Il s'agit d'une chaîne de caractères. Reprend le contenu du champ ArchivalAgencyIdentifier du message ArchiveTransfer.
 - submissionAgency : identifiant du service versant. Il s'agit d'une chaîne de caractères. Reprend le contenu du champ SubmissionAgencyIdentifier du message ArchiveTransfer. Ne contient aucune valeur actuellement
 - Cardinalité : 1-1
 - Ce champ existe uniquement pour la structure incluante.
- « **rightsStatementIdentifier** » : identifiant des données référentielles en vertu desquelles l'opération peut s'exécuter
- Pour une opération d'INGEST, il comprend les champs suivant en JSON :
 - 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.
 - Profil : identifiant du profil utilisé pour réaliser l'entrée.
Il s'agit d'une chaîne de caractères. Reprend le contenu du champ ArchiveProfile du message ArchiveTransfer.
 - Pour une opération d'UPDATE, il comprend les champs suivant en JSON :
 - AccessContract : identifiant du contrat d'accès utilisé pour réaliser une mise à jour.
 - Cardinalité : 1-1
- « **obId** » (**object Identifier**) : identifiant Vitam 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 Masterdata, il s'agit de l'identifiant de l'opération.

- Cardinalité structure incluante : 1-1
- Cardinalité structure incluse : 0-1
- Ce champ existe pour les structures incluantes 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.
- Cardinalité : 1-1
- Actuellement, la valeur est toujours "null".
- Ce champ existe pour les structures incluantes et incluses.

« **obIdIn** » (**Object Identifier Income**) : identifiant externe du lot d'objets auquel s'applique l'opération.

- Chaîne de caractères intelligible pour un humain qui permet de comprendre à quel SIP ou quel lot d'archives se rapporte l'événement.
- Reprend le contenu du champ MessageIdentifier du message ArchiveTransfer.
- Cardinalité structure incluante : 1-1
- Cardinalité structure incluse : 0-1
- Ce champ existe pour les structures incluantes et incluses.

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

- Pour la structure incluante, 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 incluante.

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

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

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

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

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

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

3.2.4 Champ présents dans les events

Les events sont au minimum composés des champs suivants :

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

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

3.2.5 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 :

```
"evDetData":
{
  \"LogType\": \"OPERATION\",
  \"StartDate\": \"2017-06-29T09:22:23.227\",
  \"EndDate\": \"2017-06-29T09:39:08.690\",
  \"Hash\": \\
  ↪ \"HYnFf07gFkar3l0+U2FQ9qkhi9eUMFN5hcH7oU7vrAAL3FAlMm8aJP7+VxkVWhLzmmFolwUEcq6fbS7Km2is5g==\\
  ↪\",
  \"TimeStampToken\": \\
  ↪ \"MIElE1jAVAgEAMBAMdk9wZxJhdGlvbiBPa2F5MIIEewYJKoZIhvcNAQcCoIIEbDCCBGgCAQMxDzANBglghkgBZQMEAgMFADCBgZ
  ↪ rIVKZ74J09qdSDeHw24HHSjw0tAnHjD6ZfUJHjDp8yQSDb6Lf2a6ORPF5JCgsh86CctQ9h93mwIBARgPMjAxNzA2MjkwOTM5MDQ
  ↪ IdBglcs69fsH05yuXOEYuwPhNlyQi jSGEwZAYLKOZIhvcNAQkQAi8xVTBTMFEwTzALBglghkgBZQMEAgMEQGlkJQTJoiVJrGpF
  ↪ /wOcpCmpqIET8w2yUcPlyqQJXYc87YeY1/OWhZiWFqbWXXV9HS4wDQYJKoZIhvcNAQENBQAEggIAV/
  ↪ rdnxIAyhvoGDprIahKAK3TPcriTggh1+gtDjEiD7kGB0KtXwAmPn2gb/2YtOmvIU7/
  ↪ a5KBF1fBR+foIRrc6z52cEdalhSpyHpYgpFuF7SjMFO6Mfso1dwjI9KpZTv6OI6Kplbg6zwK939GpDbPgKaMrXw0EDafk184RQ
  ↪ LbBEOUswGgnfnYG0lo1XbQaI2sM2+YiXHGd/qnl/
  ↪ uAteBayFeaHKXel+gkp8D1ykBFOre46n6fCI5i0OhKHcPAxvxTg8p03M38PrZIwnqSUI1rxFJhk9Hu0JvcQi1EYLBmmyL4IbhX
  ↪ BGTmZmuEksrA4vJr1WEFMUocEFQnL9pOj+iI8U0SusJEDYvjde+yvfnc8ZOGXOsaP9aUuITOMT/
  ↪ wFdrH4RFe8q8Wjxzu5p4lSvJI9P+soSfBbLyzGUjmF2lAi/HdyzjunhmRr/
  ↪ kxHK8P9Bo2CSz77xgN566k2r44ER/
  ↪ lyHFvHme5ITq25CRhJf39kfbPh1Jjku3ulwiqykhnjXX7YGx5bDRNv+z2914tq+AkZqq80+0XY5fLGgauptskhpj+CsfYs0uN
  ↪ rKg=\",
  \"PreviousLogbookTraceabilityDate\": null,
  \"MinusOneMonthLogbookTraceabilityDate\": null,
  \"MinusOneYearLogbookTraceabilityDate\": null,
```

(suite sur la page suivante)

(suite de la page précédente)

```
\\"NumberOfElement\\":379,
\\"FileName\\":\\"0_LogbookOperation_20170629_093907.zip\\",
\\"Size\\":3975227,
\\"DigestAlgorithm\\":\\"SHA512\\"}"
```

Dans le cas de l'événement final d'une opération de sécurisation du LogbookOperation, le champ « **evDetData** » est composé des champs suivants :

« **LogType** » : type de logbook sécurisé.

- Collection faisant l'objet de l'opération de sécurisation

Exemple : "operation"

- La valeur de ce champ est soit OPERATION soit LIFECYCLE.
- Cardinalité : 1-1

« **StartDate** » : date de début 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 :[3digits de millisecondes] (correspond à la date de la première sécurisation)

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

- Cardinalité : 1-1

« **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 :[3digits 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

« **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 :[3digits 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 :[3digits de millisecondes]

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

- Cardinalité : 1-1

« **MinusOneYeaLogbookTraceabilityDate** » : 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 :[3digits de millisecondes]

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

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

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

« **Size** » : Taille du fichier.

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

« **DigestAlgorithm** » : algorithmes 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

3.3 Collection LogbookLifeCycleUnit

3.3.1 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 d'archives est gérée par le système.

- dès la réception d'une unité d'archives, 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.

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

Extrait d'un JSON correspondant à un journal de cycle du vie d'une unité archivistique.

```
{
  "_id": "aeaqaahhbl62nabqkwak3k7qg5tiaaaq",
  "evId": "aedqaaaaghbl62nabqkwak3k7qg5tiaabq",
  "evParentId": null,
  "evType": "LFC.LFC_CREATION",
  "evDateTime": "2017-04-10T12:39:37.933",
  "evIdProc": "aedqaaaaghe45hwabliwak3k7qg7kaaaq",
  "evTypeProc": "INGEST",
  "outcome": "STARTED",
```

(suite sur la page suivante)

(suite de la page précédente)

```

"outDetail": "LFC.LFC_CREATION.STARTED",
"outMessg": "!LFC.LFC_CREATION.STARTED!",
"agId": "{ \"Name\": \"vitam-iaas-app-02\", \"Role\": \"worker\", \"ServerId\"
↪:1041627981, \"SiteId\": 1, \"GlobalPlatformId\": 236321613 }",
"obId": "aeaqaiaaehbl62nabqkwak3k7qg5tiaaaq",
"evDetData": null,
"events": [
  {
    "evId": "aedqaaaaaghl62nabqkwak3k7qg5tiaaabq",
    "evParentId": null,
    "evType": "LFC.CHECK_MANIFEST",
    "evDateTime": "2017-04-10T12:39:37.953",
    "evIdProc": "aedqaaaaaghe45hwabliwak3k7qg7kaaaaaq",
    "evTypeProc": "INGEST",
    "outcome": "OK",
    "outDetail": "LFC.CHECK_MANIFEST.OK",
    "outMessg": "Succès de la vérification de la cohérence du bordereau",
    "agId": "{ \"Name\": \"vitam-iaas-app-02\", \"Role\": \"worker\", \"ServerId\"
↪:1041627981, \"SiteId\": 1, \"GlobalPlatformId\": 236321613 }",
    "obId": "aeaqaiaaehbl62nabqkwak3k7qg5tiaaaq",
    "evDetData": null,
  },
  {
    "evId": "aedqaaaaaghl62n5g8ftak3k7qg5tiaaabq",
    "evParentId": "aedqaaaaaghl62nabqkwak3k7qg5tiaaabq",
    "evType": "LFC.CHECK_MANIFEST.LFC_CREATION",
    "evDateTime": "2017-04-10T12:39:37.953",
    "evIdProc": "aedqaaaaaghe45hwabliwak3k7qg7kaaaaaq",
    "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-iaas-app-02\", \"Role\": \"worker\", \"ServerId\"
↪:1041627981, \"SiteId\": 1, \"GlobalPlatformId\": 236321613 }",
    "obId": "aeaqaiaaehbl62nabqkwak3k7qg5tiaaaq",
    "evDetData": null,
  }, {
    [...]
  }
],
"_tenant": 1,
"_v": 0,
"_lastPersistedDate": "2017-04-10T12:39:37.953"
}

```

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.
- 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 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 LFC.CHECK_MANIFEST.LFC_CREATION, ce champs fera référence au GUID de l'évènement LFC.CHECK_MANIFEST.
- La valeur 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**) : 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 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 :[3digits 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 incluantes et incluses.

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

- Il s'agit d'une chaîne de 36 caractères.
- Toutes les mêmes entrées du 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 incluantes et incluses.

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

- Il s'agit d'une chaîne de caractères.
- Nom du processus qui effectue l'action, parmi une liste de processus possibles fixée. Cette liste est disponible en annexe.
- Cardinalité : 1-1
- Ce champ existe pour les structures incluantes 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 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 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'évènement.
- Il s'agit de 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 ServerIdentifier.
- Exemple : `{\"name\": \"ingest-internal_1\", \"role\": \"ingest-internal\", \"pid\": 425367}`
- 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'évènement.
- Donne plus de détails sur l'évènement.
 - Par exemple, l'historisation de métadonnées lors d'une modification se fait dans ce champ. Dans la structure incluse correspondant à cet événement, il contient, par exemple, composé du champ suivant :
 - diff : contient la différence entre les métadonnées d'origine et les métadonnées modifiées. Chaîne de caractères.
 - En outre, lors de l'historisation de la sauvegarde de l'unité archivistique sur les offres de stockage, on utilise ce champ pour tracer les informations sur le fichier sauvegardé. Il contient, ainsi, un JSON composé comme suit :
 - FileName : Identifiant du fichier. Il s'agit du nom du fichier sauvegardé sur les offres de stockage.
 - Algorithm : Algorithme de hachage. Il s'agit du nom de l'algorithme de hachage.
 - MessageDigest : Empreinte du fichier. Il s'agit d'une chaîne de caractères contenant l'empreinte du fichier.
 - Offers : Offres de stockage. Il s'agit des offres de stockage utilisées pour la sauvegarde du fichier.
 - 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 incluante.

« **_tenant** » : identifiant du tenant

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

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

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

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

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

3.3.4 Champs présents dans les events

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

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

Exemple de données stockées :

```
"evDetData": "{\\"diff\\":\\"- Title : Recommandation de 2012 du CCSDS for Space Data_\n↪System Practices - Reference Model for an Open Archival Information System_\n↪(OAIS)\\n+ Title : Recommandation de 2012 du CCSDS for Space Data System Practices_\n↪- Reference Model for an Open Archival Information System (OAIS) 222\\n-\n↪#operations : [ aedqaaaaacaam7mxabxecakz3jbfwpaaaaaq \\n+ #operations : [\n↪aedqaaaaacaam7mxabxecakz3jbfwpaaaaaq, aecaaaaacaam7mxabjssak2dzsjniyaaaaaq \\"}"
```

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)

Exemple : -Titre: Discours du Roi \n+Titre: Discours du Roi Louis XVI \n-Description: Etat Généraux du 5 mai 1789 \n+Description: Etat Généraux du 5 mai 1789 au Château de Versailles

3.4 Collection LogbookLifeCycleObjectGroup

3.4.1 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.

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

```
{
  "_id": "aeaaaaaaaaam7mxaap44akyf7hurgaaaaba",
  "evId": "aedqaaaaacaam7mxaap44akyf7hurgaaaabq",
  "evParentId": null,
  "evType": "CHECK_CONSISTENCY",
  "evDateTime": "2016-11-04T14:47:43.512",
  "evIdProc": "aedqaaaaacaam7mxaau56akyf7hr45qaaaaq",
  "evTypeProc": "INGEST",
  "outcome": "STARTED",
  "outDetail": "STARTED",
  "outMessg": "Début de la vérification de la cohérence entre objets/groupes d'objets_
↪ et ArchiveUnit.",
  "agId": "{ \"Name\": \"vitam-iaas-worker-01\", \"Role\": \"worker\", \"PlatformId\"
↪ : 425367 }",
  "obId": "aeaaaaaaaaam7mxaap44akyf7hurgaaaaba",
  "evDetData": null,
  "events": [
    {
      "evId": "aedqaaaaacaam7mxaap44akyf7hurgaaaabq",
      "evParentId": null,
      "evType": "CHECK_CONSISTENCY",
      "evDateTime": "2016-11-04T14:47:43.515",
      "evIdProc": "aedqaaaaacaam7mxaau56akyf7hr45qaaaaq",
      "evTypeProc": "INGEST",
      "outcome": "OK",
```

(suite sur la page suivante)

```

        "outDetail": "OK",
        "outMessg": "Objet/groupe dobjct référencé par un ArchiveUnit.",
        "agId": "{ \"Name\": \"vitam-iaas-app-02\", \"Role\": \"worker\", \"ServerId\"
↪\": 773928267, \"SiteId\": 1, \"GlobalPlatformId\": 237057355 }",
        "obId": "aeaaaaaaaaam7mxaap44akyf7hurgaaaaba",
        "evDetData": null,
    },
    {
        "evId": "\"aeaaaaaaaaam7mxaap44akyf7hurgaaaaba\"",
        "evParentId": null,
        "evType": "CHECK_DIGEST",
        "evDateTime": "2016-11-04T14:47:45.132",
        "evIdProc": "aedqaaaaacaam7mxaau56akyf7hr45qaaaaq",
        "evTypeProc": "INGEST",
        "outcome": "STARTED",
        "outDetail": "STARTED",
        "outMessg": "Début de la vérification de lempeinte.",
        "agId": "{ \"Name\": \"vitam-iaas-worker-01\", \"Role\": \"worker\", \"
↪PlatformId\": 425367 }",
        "obId": "aeaaaaaaaaam7mxaap44akyf7hurgaaaaba",
        "evDetData": "{ \"MessageDigest\": \"
↪0f1de441a7d44a277e265eb741e748ea18c96a59c8c0385f938b9768a42e375716dfa3b20cc125905636
5aa0d3541f6128389ad60c8effbdc63b94df9a2e02bb\", \"Algorithm\": \"SHA512\", \"
↪SystemMessageDigest\": \"SHA-512\", \"SystemAlgorithm\": \"
↪0f1de441a7d44a277e265eb741e748ea18c96a59c8c0385f938b9768a42e375716dfa3b20cc125905636
5aa0d3541f6128389ad60c8effbdc63b94df9a2e02bb\" } ",
    },
    {
        [...]
    }
],
"_tenant": 0,
"_v": 0,
"_lastPersistedDate": "2016-11-04T14:47:45.132"
}

```

3.4.3 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 LFC.CHECK_MANIFEST.LFC_CREATION, ce champs 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 :[3digits 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 incluantes et incluses.

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

- Il s'agit d'une chaîne de 36 caractères.
- Toutes les mêmes entrées 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 qui effectue l'action, parmi une liste de processus possibles fixée. Cette liste est disponible en annexe.
- Cardinalité : 1-1
- Ce champ existe pour les structures incluantes et incluses

« outcome » : statut de l'événement.

- Il s'agit d'une chaîne de caractères devant correspondre 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 incluantes 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 du code présent dans outDetail.
- Cardinalité : 1-1
- Ce champ existe pour les structures incluant et incluses.

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

- Il s'agit de 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 ServerIdentifier.

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

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

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

- Si l'évènement 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.
- Par exemple, pour l'événement LFC.CHECK_DIGEST, lorsque l'empreinte d'un objet inscrite dans le bordereau n'est pas calculée en SHA512, 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 JSON composé des champs suivants :
 - MessageDigest : empreinte de l'objet dans le bordereau. Chaîne de caractères, reprenant le champ « MessageDigest » du message ArchiveTransfer.
 - Algorithm : algorithme de hachage utilisé dans le bordereau. 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.
- En outre, pour l'événement LFC.OBJ_STORAGE, on utilise ce champ pour tracer les informations sur l'objet (fichier binaire) sauvegardé. Il contient un JSON composé comme suit :
 - FileName : Identifiant du fichier. Il s'agit du nom du fichier sauvegardé sur les offres de stockage.
 - Algorithm : Algorithme de hachage. Il s'agit du nom de l'algorithme de hachage.
 - MessageDigest : Empreinte de l'objet. Il s'agit d'une chaîne de caractères contenant l'empreinte de l'objet.
 - Offers : Offres de stockage. Il s'agit des offres de stockage utilisées pour la sauvegarde de l'objet.
- Pour l'événement LFC.OG_METADATA_STORAGE, on utilise ce champ pour tracer les informations sur le fichier (métadonnées) sauvegardé. Il contient un JSON composé comme suit :
 - FileName : Identifiant du fichier. Il s'agit du nom du fichier sauvegardé sur les offres de stockage.

- Algorithm : Algorithme de hachage. Il s'agit du nom de l'algorithme de hachage.
- MessageDigest : Empreinte du fichier. Il s'agit d'une chaîne de caractères contenant l'empreinte du fichier.
- Offers : Offres de stockage. Il s'agit des offres de stockage utilisées pour la sauvegarde du fichier.

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

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

- Pour la structure incluante, 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 incluante.

« **_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.
- Cardinalité : 1-1
- Ce champ existe uniquement pour la structure incluante.

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

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

3.4.4 Champs présents dans les events

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

3.5 Collection Offset

3.5.1 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
- OBJETGROUP_GRAPH

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

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

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

3.5.3 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** » : Il s'agit de 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 seule valeur possible est *logbook*

« **_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

4.1 Collections contenues dans la base

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.

4.2 Collection Unit

4.2.1 Utilisation de la collection Unit

La collection Unit contient les informations relatives aux unités archivistiques.

4.2.2 Exemple de XML en entrée

Ci-après, la portion d'un bordereau (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="ID44">
  <Management>
    <DisseminationRule>
      <Rule>DIS-00002</Rule>
      <StartDate>2000-01-01</StartDate>
      <RefNonRuleId>DIS-00001</RefNonRuleId>
    </DisseminationRule>
  </Management>
  <Content>
    <DescriptionLevel>RecordGrp</DescriptionLevel>
    <Title>Gare du Nord</Title>
    <Description>Cette unité de description hérite de son parent de la règle ACC-
    →00003 avec pour StartDate 01/01/2000, bloque l'héritage de la règle DIS-00001 mais
    (suite sur la page suivante)
    →déclare la règle DIS-00002 avec pour StartDate 01/01/2000</Description>
```

```

    <StartDate>2017-04-05T08:11:56</StartDate>
    <EndDate>2017-04-05T08:11:56</EndDate>
  </Content>
  <ArchiveUnit id="ID75">
    <ArchiveUnitRefId>ID32</ArchiveUnitRefId>
  </ArchiveUnit>
</ArchiveUnit>

```

4.2.3 Exemple de Json stocké dans la collection Unit

Les champs présentés dans l'exemple ci-après ne fait 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/doc.html

```

{
  "_id": "aeaqaahccnklabtgyak7pkvypgyaaacq",
  "_mgt": {
    "DisseminationRule": {
      "Rules": [
        {
          "Rule": "DIS-00002",
          "StartDate": "2000-01-01",
          "EndDate": "2075-01-01"
        }
      ],
      "Inheritance": {
        "PreventRulesId": [
          "DIS-00001"
        ]
      }
    },
    "OriginatingAgency": "RATP"
  },
  "DescriptionLevel": "RecordGrp",
  "Title": "Gare du Nord",
  "Titles": {
    "fr": "Gare du Nord"
  },
  "Description": "Cette unité de description hérite de son parent de la règle ACC-
↳00003 avec pour StartDate 01/01/2000, bloque l'héritage de la règle DIS-00001 mais
↳déclare la règle DIS-00002 avec pour StartDate 01/01/2000",
  "Descriptions": {
    "fr": "Cette unité de description hérite de son parent de la règle ACC-00003
↳avec pour StartDate 01/01/2000, bloque l'héritage de la règle DIS-00001 mais
↳déclare la règle DIS-00002 avec pour StartDate 01/01/2000"
  },
  "StartDate": "2017-04-05T08:11:56",
  "EndDate": "2017-04-05T08:11:56",
  "_storage": {
    "_nbc": 2,
    "strategyId": "default",
    "offerIds": [
      "vitam-iaas-app-02.int",
      "vitam-iaas-app-03.int"
    ]
  }
}

```

(suite sur la page suivante)

(suite de la page précédente)

```

    },
    "_sps": [
        "RATP"
    ],
    "_sp": "RATP",
    "_ops": [
        "aedqaaaaacfeavznabdrgak7pkvyhgiaaaaq"
    ],
    "_opi": "aedqaaaaacfeavznabdrgak7pkvyhgiaaaaq",
    "_unitType": "INGEST",

    "_max": 4,
    "_min": 1,
    "_up": [
        "aeaqaaaaahccnklabtgyak7pkvypgyaaaba"
    ],
    "_nbc": 1,
    "_us": [
        "aeaqaaaaahccnklabtgyak7pkvypgqaaaba",
        "aeaqaaaaahccnklabtgyak7pkvypgqaaacq",
        "aeaqaaaaahccnklabtgyak7pkvypgyaaaba"
    ],
    "_uds": {
        "1": ["aeaqaaaaahccnklabtgyak7pkvypgyaaaba"],
        "2": ["aeaqaaaaahjgl36aazigaldnxdkimiaaabq"],
        "3": ["aeaqaaaaahccnklabtgyak7pkvypgqaaaba"],
    },
    ],
    "_v": 1,
    "_tenant": 0
}

```

Voici un autre extrait de détail JSON :

```

{ .....
"_sp": "SP1",
"_up": [
    "aeaqaaaaahjgl36aazigaldnxdkivyaaabq",
    "aeaqaaaaahjgl36aazigaldnxdkivaaaaba"
],
"_us": [
    "aeaqaaaaahjgl36aazigaldnxdkivyaaabq",
    "aeaqaaaaahjgl36aazigaldnxdkimiaaabq",
    "aeaqaaaaahjgl36aazigaldnxdkivaaaaba"
],
"_sps": [
    "SP1",
    "SP2"
],
"_graph": [
    "aeaqaaaaahjgl36aazigaldnxdkivyaaabq/aeaqaaaaahjgl36aazigaldnxdkimiaaabq",
    "aeaqaaaaahjgl36aazigaldnxdkivaaaaba/aeaqaaaaahjgl36aazigaldnxdkivaaaaba",
    "aeaqaaaaahjgl36aazigaldnxdkivaaaaba/aeaqaaaaahjgl36aazigaldnxdkivyaaabq",
    "aeaqaaaaahjgl36aazigaldnxdkivaaaaba/aeaqaaaaahjgl36aazigaldnxdkimiaaabq"
],

```

(suite sur la page suivante)

```

"_uds": {
  "1": [
    "aeaqaahhahjgl36aazigaldnxdkivaaaaba",
    "aeaqaahhahjgl36aazigaldnxdkivyaaabq"
  ],
  "2": [
    "aeaqaahhahjgl36aazigaldnxdkimiaaaba"
  ]
},
"_us_sp": {
  "SP1": [
    "aeaqaahhahjgl36aazigaldnxdkivaaaaba",
    "aeaqaahhahjgl36aazigaldnxdkivyaaabq"
  ],
  "SP2": [
    "aeaqaahhahjgl36aazigaldnxdkimiaaaba",
  ]
},
"_min": 1,
"_max": 3,
"_glpd": "2018-05-17T11:05:38.705"
}

```

4.2.4 Détail du JSON

La structure de la collection Unit est composée de la transposition JSON de toutes les balises XML contenues dans la balise <DescriptiveMetadata> du bordereau de transfert conforme au standard SEDA v.2.1., c'est-à-dire toutes les balises se rapportant aux unités archivistiques.

Cette transposition se fait comme suit :

« **_id** » : 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

« **_og** » (**objectGroup**) : identifiant du groupe d'objets représentant cette unité archivistique.

- Il s'agit d'une chaîne de 36 caractères correspondant au GUID du champ `_id` du groupe d'objets de la collection `objectGroup`.
- Cardinalité : 0-1

« **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 : 1-1

« **Titles** » : 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, la valeur le titre. Par exemple : « fr » : « Ceci est un titre. »
 - Cardinalité : 0-1
- « **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
- « **Descriptions** » : description de l'unité archivistique par langue.
- Il s'agit d'un tableau JSON
 - Les descriptions sont organisées sous la forme de clef / valeur, la clef étant l'indicatif de la langue, la valeur la description. Par exemple : « fr » : « Ceci est une description. »
 - Cardinalité : 0-n
- « **XXXXX** » : des champs facultatifs peuvent être contenus dans le JSON lorsqu'ils sont renseignés dans le bordereau au niveau du Contenu 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.
- « **_storage** » : contient trois champs qui permettent d'identifier les offres de stockage.
- Il s'agit d'un JSON constitué des champs suivants :
 - « strategyId » : identifiant de la stratégie de stockage.
 - « offerIds » : liste des offres de stockage pour une stratégie donnée
 - « _nbc » : nombre d'offres.
 - Ne peut être vide
 - Cardinalité : 1-1
- « **_sps** » : services producteurs liés à l'unité archivistique suite à un rattachement et ayant des droits d'accès sur celle-ci.
- 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.
 - Ne peut être vide. Il comprend au minimum le service versant déclaré dans le bordereau de transfert.
 - Cardinalité : 1-1
- « **_sp** » : service producteur d'origine déclaré lors de la prise en charge de l'unité archivistique par la solution logicielle Vitam.
- Il s'agit du service producteur inscrit dans le bordereau lié au transfert de l'unité archivistique et déclaré via une extension du schéma <OtherManagementAbstract>, la balise <OriginatingAgencyIdentifier>.
 - Il s'agit d'une chaîne de caractères.
 - Cardinalité : 1-1
- « **_ops** » (operations) : tableau contenant les identifiants d'opérations auxquelles cette unité archivistique a participé.
- Il s'agit d'une chaîne de 36 caractères correspondant au GUID du champ _id 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 GUID du champs _id de 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

« **_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

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

- 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

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

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

« **_nbc** » : nombre d'enfants immédiats de l'unité archivistique.

- Il s'agit d'un entier.
- Champ peuplé par la solution logicielle Vitam.
- 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 de chaînes de 36 caractères.
- Champ peuplé par la solution logicielle Vitam.
- Ne peut être vide
- Cardinalité : 1-1

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

- Ces informations sont réunies dans le tableau sous la forme de clef/valeur. Exemple [{GUID1 : depth1}, {GUID2 : depth2}, ...]].
- Il s'agit d'un tableau de JSON.
- Champ peuplé par la solution logicielle Vitam.
- Cardinalité : 1-1

« **_us_sp** » : Liste des AU parents concernant tous les niveaux de parentalité

- Il s'agit d'un tableau contenant les identifiants de tous les services producteurs de tous les parents liées à l'unité archivistique.
- Il s'agit d'un tableau de chaînes de caractères.
- Ne peut être vide. Il comprend au minimum le service versant déclaré dans le bordereau de transfert.
- Cardinalité : 1-1

« **_glpd** » : Date de la dernière modification de la partie `_graph`

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

_profil : Profil d'archivage utilisé lors de l'entrée.

- Correspond à ArchiveProfile, le profil d'archivage utilisé lors de l'entrée. Sa valeur correspond à l'identifiant métier d'un profil enregistré dans la collection ArchiveProfil.
- Chaîne de caractères.
- Cardinalité : 0-1

« **_mgt** » : contient les balises contenues dans le bloc <Management> du bordereau de transfert pour cette unité archivistique.

Une catégorie de règles de gestion appliquées à cette unité archivistique. Ces catégories sont, exhaustivement (cardinalité 0-1 pour chaque catégorie) :

- 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)

Chaque catégorie peut contenir :

1. Un tableau de règles de gestion (tableau d'objets, cardinalité 0-1)

Chacune des règles de ce tableau est elle-même composée de plusieurs informations :

- « **Rule** » : identifiant de la règle, qui correspond à une valeur du champ RuleId de la collection FileRules. (cardinalité 1-1)
- « **StartDate** » : « StartDate » : date de début du calcul de l'échéance. Cette date est déclarée dans le message ArchiveTransfer ou ajoutée *a posteriori* par une modification de l'unité archivistique. (cardinalité 1-1)
- « **EndDate** » : date de fin d'application de la règle. 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 1-1)

2. Des données spécifiques aux catégories :

- Pour les catégories « StorageRule » et « AppraisalRule » uniquement : le champ « **FinalAction** » décrit le 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 en SEDA 2.1)
 - Pour AppraisalRule : « Keep » ou « Destroy » (énumération issue du FinalActionAppraisalCodeType en SEDA 2.1)
- Pour ClassificationRule uniquement :
 - « **ClassificationLevel** » : niveau de classification, obligatoire et systématiquement renseigné (cardinalité 1-1)
 - « **ClassificationOwner** » : propriétaire de la classification, obligatoire et systématiquement renseigné (cardinalité 1-1)
 - « **ClassificationAudience** » : permet de gérer les questions de « diffusion restreinte », « spécial France » et « Confidentiel Industrie », champ optionnel (cardinalité 0-1)
 - « **ClassificationReassessingDate** » : date de réévaluation de la classification, optionnelle. (cardinalité 0-1)
 - « **NeedReassessingAuthorization** » : indique si une autorisation humaine est nécessaire pour réévaluer la classification, optionnel (cardinalité 0-1)

3. Des paramètres de gestion d'héritage de règles, dans un objet nommé « **Inheritance** » (cardinalité 0-1). Cet objet peut avoir comme valeur :

- « **PreventInheritance** » : « true » ou « false », utilisé pour bloquer l'héritage de toutes les règles de gestion de la même catégorie (cardinalité 0-1)
- « **PreventRulesId** » : est un tableau d'identifiants de règles de gestion qui ne doivent pas être héritées d'un parent (cardinalité 0-1)

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

```
"_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"
      }
    ]
  }
}
```

(suite sur la page suivante)

(suite de la page précédente)

```

    }
  ],
  "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"
      }
    ]
  }
},
},

```

4.3 Collection ObjectGroup

4.3.1 Utilisation de la collection ObjectGroup

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

4.3.2 Exemple de XML

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

```

<PhysicalDataObject id="ID109">
  <DataObjectGroupReferenceId>ID0009</DataObjectGroupReferenceId>
  <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>

```

(suite sur la page suivante)

```

    <Color>Noir et blanc</Color>
    <Framing>Paysage</Framing>
    <Technique>Phototypie</Technique>
</PhysicalDataObject>
<BinaryDataObject id="ID9">
  <DataObjectGroupId>ID0009</DataObjectGroupId>
  <DataObjectVersion>BinaryMaster</DataObjectVersion>
  <Uri>Content/1NUM_9.JPG</Uri>
  <MessageDigest algorithm="SHA-512">
→ 0e0cec05a1d72ee5610eaa5afbc904c012d190037cbc827d08272102cdecf0226efcad122b86e7699f767c661c9f370237
→ </MessageDigest>
  <Size>7702</Size>
  <FormatIdentification>
    <FormatLitteral>JPEG File Interchange Format</FormatLitteral>
    <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>

```

4.3.3 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/doc.html

```

{
  "_id": "aebaaaaaaaafgsz3wabcugak7ube6dxyaaabq",
  "_tenant": 0,
  "_profil": "Image",
  "FileInfo": {
    "Filename": "1NUM_9.JPG"
  },
  "_qualifiers": [
    {
      "qualifier": "PhysicalMaster",
      "_nbc": 1,
      "versions": [
        {
          "_id": "aeaaaaaaaafgsz3wabcugak7ube6dzqaaaca",
          "DataObjectGroupId": "aebaaaaaaaafgsz3wabcugak7ube6dxyaaabq",
          "DataObjectVersion": "PhysicalMaster_1",
          "PhysicalId": "1 Num 1/191-3",

```

(suite sur la page suivante)

(suite de la page précédente)

```

    "PhysicalDimensions": {
      "Height": {
        "unit": "centimetre",
        "value": 10.5
      },
      "Length": {
        "unit": "centimetre",
        "value": 14.8
      },
      "Thickness": {
        "unit": "micrometre",
        "value": 350
      },
      "Weight": {
        "unit": "gram",
        "value": 3
      }
    },
    "Extent": "1 carte imprimée",
    "Dimensions": "10,5cm x 14,8cm",
    "Color": "Noir et blanc",
    "Framing": "Paysage",
    "Technique": "Phototypie",
    "_storage": {
      "_nbc": 0,
      "offerIds": [],
      "strategyId": "default"
    }
  }
]
},
{
  "qualifier": "BinaryMaster",
  "_nbc": 1,
  "versions": [
    {
      "_id": "aeaaaaaaaaafgsz3wabcugak7ube6dxyaaaba",
      "DataObjectGroupId": "aebaaaaaaaaafgsz3wabcugak7ube6dxyaaabq",
      "DataObjectVersion": "BinaryMaster_1",
      "FormatIdentification": {
        "FormatLiteral": "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
        }
      }
    }
  ]
}

```

(suite sur la page suivante)

```

        "Size": 7702,
        "Uri": "Content/1NUM_9.JPG",
        "MessageDigest":
↪ "0e0cec05a1d72ee5610eaa5afbc904c012d190037cbc827d08272102cdecf0226efcad122b86e7699f767c661c9f37023"
↪ ",
        "Algorithm": "SHA-512",
        "_storage": {
            "_nbc": 2,
            "offerIds": [
                "vitam-iaas-app-02.int",
                "vitam-iaas-app-03.int"
            ],
            "strategyId": "default"
        }
    ]
}
],
"_up": [
    "aeaqaafgsz3wabcugak7ube6d4qaaaq"
],
"_nbc": 0,
"_ops": [
    "aedqaaaachxqyktai4aak7ube557iaaaaq"
],
"_opi": "aedqaaaachxqyktai4aak7ube557iaaaaq",
"_sp": "Vitam",
"_sps": [
    "Vitam"
],
"_storage": {
    "_nbc": 2,
    "offerIds": [
        "vitam-iaas-app-02.int",
        "vitam-iaas-app-03.int"
    ],
    "strategyId": "default"
},
"_v": 1
}

```

4.3.4 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, ...).

- 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 (version de création), au cas où cette version serait éliminée (en application de règle de gestion).
- Cardinalité : 1-1

« **_qualifiers** » : tableau de structures décrivant les objets inclus dans ce groupe d'objets. Il est composé comme suit :

- « qualifier » : usage de l'objet.
Ceci correspond à la valeur contenue dans le champ <DataObjectVersion> du bordereau. Par exemple pour <DataObjectVersion>BinaryMaster_1</DataObjectVersion>, c'est la valeur « BinaryMaster » qui est reportée.
- « _nbc » : nombre d'objets correspondant à cet usage.
- « 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.
 - « DataObjectGroupId » : identifiant du groupe d'objets, composé d'une chaîne de 36 caractères.
 - « DataObjectVersion » : version de l'objet par rapport à son usage.

Par exemple, si on a *BinaryMaster* sur l'usage, on aura au moins un objet *BinaryMaster_1*. Ces champs sont renseignés avec les valeurs récupérées dans les balises <DataObjectVersion> du bordereau de transfert.

- « 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.
 - « FormatLiteral » : nom du format. C'est une reprise de la valeur située entre les balises <FormatLiteral> du message ArchiveTransfer.
 - « 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.
 - « 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.
- « FileInfo » : Contient les informations sur les fichiers.
 - « Filename » : nom de l'objet.
 - « 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. *Ce champ est facultatif et n'est pas présent systématiquement.*
 - « 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. *Ce champ est facultatif et n'est pas présent systématiquement.*
 - « CreatingOs » : système d'exploitation avec lequel l'objet a été créé. Ce champ est renseigné avec la métadonnée correspondante portée par le message ArchiveTransfer. *Ce champ est facultatif et n'est pas présent systématiquement.*

- « 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. *Ce champ est facultatif et n'est pas présent systématiquement.*
 - « LastModified » : date de dernière modification de l'objet 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.
 - « Size » : taille de l'objet (en octet). Ce champ contient un nombre entier.
 - « OtherMetadata » : Ce champ est renseigné avec les valeurs contenues entre les balises <OtherMetadata>, de l'une extension du schéma SEDA du message ArchiveTransfer.
 - « Uri » : localisation du fichier correspondant à l'objet dans le SIP.
Chaîne de caractères
 - « MessageDigest » : empreinte du fichier correspondant à l'objet. La valeur est calculée par la solution logicielle Vitam.
Chaîne de caractères
 - « Algorithm » : algorithme utilisé pour réaliser l'empreinte du fichier correspondant à l'objet.
Chaîne de caractères
 - « _storage » : contient trois champs qui permettent d'identifier les offres de stockage.
 - « strategyId » : identifiant de la stratégie de stockage.
 - « offerIds » : liste des offres de stockage pour une stratégie donnée
 - « _nbc » : nombre d'offres.
- « **_up** » (**unit up**) : tableau identifiant les unités archivistiques représentée 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** » (**nbjects**) : 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 GOT.
- Il s'agit d'une chaînes de 36 caractères correspondant au GUID contenue 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 déclaré dans le message ArchiveTransfer (OriginatingAgencyIdentifier)
- Il s'agit d'une chaîne de caractères.

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

« **_sps** » : service producteur d'origine déclaré lors de la prise en charge du groupe d'objets par la solution logicielle Vitam.

- 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.
- Champ peuplé par la solution logicielle Vitam.
- Ne peut être vide
- 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.

4.4 Collection Offset

4.4.1 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 :

- Unit
- ObjectGroup

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

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

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

4.4.3 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** » : Il s'agit de 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*

« **_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

5.1 Collections contenues dans la base

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

- AccessContract
- AccessionRegisterDetail
- AccessionRegisterSummary
- Agencies
- Context
- FileFormat
- FileRules
- IngestContract
- Profile
- SecurityProfile
- VitamSequence

5.1.1 Collections Cross-tenant

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
- SecurityProfile

Elles sont enregistrées sur le tenant d'administration.

5.2 Collection AccessContract

5.2.1 Utilisation de la collection AccessContract

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

5.2.2 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
- Description

Un fichier d'import peut décrire plusieurs contrats.

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

```
{
  "_id": "aefqaaaaahbl62nabkzgak3k6qtf3aaaaaq",
  "Name": "SIA archives nationales",
  "Identifiant": "AC-000009",
  "Description": "Contrat d'accès - SIA archives nationales",
  "Status": "ACTIVE",
  "CreationDate": "2017-04-10T11:30:33.798",
  "LastUpdate": "2017-04-10T11:30:33.798",
  "ActivationDate": "2017-04-10T11:30:33.798",
}
```

(suite sur la page suivante)

(suite de la page précédente)

```

"DeactivationDate": null,
"DataObjectVersion": ["PhysicalMaster", "BinaryMaster", "Dissemination", "Thumbnail",
↔ "TextContent"],
"OriginatingAgencies": ["FRA-56", "FRA-47"],
"RootUnits": [
  "aeaqaahxunbaabg3yak6urend2yaaaq",
  "aeaqaahxunbaabg3yak6urendoqaaaq"
],
"WritingPermission": true,
"EveryOriginatingAgency": false,
"EveryDataObjectVersion": true,
"_tenant": 0,
"_v": 0,
"RootUnits": [
  "aeaqaahxunbaabg3yak6urend2yaaaq",
  "aeaqaahxunbaabg3yak6urendoqaaaq"
]
"ExcludedRootUnits": ["aeaqaahxunbaabg3yak6urendoqaaaq"],
}

```

5.2.4 Détail des champs

« **_id** » : identifiant unique par tenant par contrat.

- 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 de l'identifiant du tenant.
- 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.
- 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.

- Peut être 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 dans la collection AccesContrat.
- 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** » : usages d'un groupe d'objets auxquels le détenteur du contrat a accès.
- Il s'agit d'un tableau de chaînes de caractères.
 - 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.
 - Peut être vide
 - Cardinalité : 0-1
- « **WritingPermission** » : droit d'écriture.
- PII s'agit d'un booléen. Si la valeur est à true, le détenteur du contrat peut effectuer des mises à jour.
 - 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
- « **_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 noeuds de consultation auxquels le détenteur du contrat a accès. Si aucun noeud n'est spécifié, alors l'utilisateur a accès à tous les noeuds.

- Il s'agit d'un tableau de chaînes de caractères.
- Peut être vide
- Cardinalité : 0-1

« **ExcludedRootUnits** » : Liste des noeuds de consultation à partir desquels le détenteur du contrat n'a pas accès. Si aucun noeud n'est spécifié, alors l'utilisateur a accès à tous les noeuds.

- Il s'agit d'un tableau de chaînes de caractères.
- Peut être vide
- Cardinalité : 0-1

5.3 Collection AccessionRegisterDetail

5.3.1 Utilisation de la collection AccessionRegisterDetail

Cette collection a pour vocation de référencer l'ensemble des informations sur les opérations d'entrée réalisées pour un service producteur. A 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.

5.3.2 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>
```

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

```
{
  "_id": "aedqaaaaakhpuaosabkcgak4ebd7deiaaaaq",
  "OriginatingAgency": "FRAN_NP_009734",
  "Identifiant": "aedqaaaaakhpuaosabkcgak4ebd7deiaaaaq",
  "OperationGroup": "aedqaaaaakhpuaosabkcgak4ebd7deiaaaaq",
  "SubmissionAgency": "FRAN_NP_009734",
  "AcquisitionInformation": "AcquisitionInformation0"
  "LegalStatus": "Private Archive"
  "ArchivalAgreement": "ArchivalAgreement0",
  "EndDate": "2017-05-19T12:36:52.572+02:00",
  "StartDate": "2017-05-19T12:36:52.572+02:00",
```

(suite sur la page suivante)

```

"Symbolic": true,
>Status": "STORED_AND_COMPLETED",
>LastUpdate": "2017-05-19T12:36:52.572+02:00",
>TotalObjectGroups": {
  "ingested": 0,
  "deleted": 0,
  "remained": 0,
  "attached": 0,
  "detached": 0,
  "symbolicRemained": 0
},
>TotalUnits": {
  "ingested": 11,
  "deleted": 0,
  "remained": 11,
  "attached": 0,
  "detached": 0,
  "symbolicRemained": 0
},
>TotalObjects": {
  "ingested": 0,
  "deleted": 0,
  "remained": 0,
  "attached": 0,
  "detached": 0,
  "symbolicRemained": 0
},
>ObjectSize": {
  "ingested": 0,
  "deleted": 0,
  "remained": 0,
  "attached": 0,
  "detached": 0,
  "symbolicRemained": 0
},
>OperationIds": [
  "aedqaaaaakhpuaosabkcgak4ebd7deiaaaaq"
],
>_v": 5,
>_tenant": 2
}

```

5.3.4 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

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

- Il est issu du le bloc <OriginatinAgencyIdentifier>
- Correspond au champ Identifier de la collection Agencies.

Par exemple :

```
<OriginatingAgencyIdentifier>FRAN_NP_051314</OriginatingAgencyIdentifier>
```

On récupère la valeur FRAN_NP_051314

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

« **Identifier** » : identifiant d'une opération d'entrée

- Il s'agit soit de l'identifiant de l'opération ayant versé les archives recensées dans ce détail du registre des fonds, soit d'une opération d'entrée précédente concernée par l'entrée. Par exemple en cas de rattachement d'une nouvelle unité archivistique AU1 à une autre AU2 déjà existante dans le système, on aura dans ce champ l'identifiant correspondant à l'opération de l'AU1 dans un des documents Mongo et l'identifiant de l'entrée de l'AU2 dans un autre.
- Cardinalité 1-1

« **OperationGroup** » : : identifiant de l'opération d'entrée ayant versé les archives recensées dans ce détail du registre des fonds

- Dans le cas de SIP faisant des rattachements (par exemple une nouvelle AU à une AU existante), il s'agira toujours de l'identifiant de l'opération de l'entrée en cours (celle générant ces documents Mongo)
- Cardinalité 1-1

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

- Il est contenu entre les balises <SubmissionAgencyIdentifier>
- Correspond au champ Identifier de la collection Agencies.

Par exemple

```
<SubmissionAgencyIdentifier>FRAN_NP_005761</SubmissionAgencyIdentifier>
```

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 <OriginatingAgencyIdentifier> est reportée dans ce champ.

« **ArchivalAgreement** » :

- Contient le contrat utilisé pour réaliser l'entrée.
- Il est contenu entre les balises <ArchivalAgreement>
- Correspond à la valeur contenue dans le 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 est contenu entre les balises <AcquisitionInformation>
- Cardinalité : 1-1

« **LegalStatus** » :

- Contient le statut des archives échangés

- Il est contenu entre les balises <LegalStatus>
- Cardinalité : 1-1

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

- La date est au format ISO 8601
- "EndDate": "2017-04-10T11:30:33.798"
- Champ peuplé par la solution logicielle Vitam.
 - Cardinalité : 1-1

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

- La date est au format ISO 8601
- "StartDate": "2017-04-10T11:30:33.798"
- Champ peuplé par la solution logicielle Vitam.
 - Cardinalité : 1-1

Symbolic : Indique si le fonds concerné est propre au service producteur ou s'il lui est rattaché symboliquement. Si le champ correspond à la valeur true, il s'agit de liens symboliques.

- Il s'agit d'un booléen
- Cardinalité : 1-1

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

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

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

- La date est au format ISO 8601
- "LastUpdate": "2017-04-10T11:30:33.798"
- Champ peuplé par la solution logicielle Vitam.
 - Cardinalité : 1-1

« **TotalObjectGroups** » : Contient la répartition du nombre de groupes d'objets du fonds par état pour l'opération journalisée (

- « 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.
- « attached » : nombre de groupes d'objets rattachés symboliquement de ce service producteur pour l'enregistrement concerné. La valeur contenue dans le champ est un entier.
- « detached » : nombre de groupes d'objets détachés symboliquement de ce service producteur. La valeur contenue dans ce champ est un entier.
- « symbolicRemained » : nombre actualisé de groupes d'objets attachés symboliquement de ce service producteur pour l'enregistrement concerné et conservés dans la solution logicielle Vitam. 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 » : 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.
- « attached » : nombre d'unités archivistiques rattachées symboliquement de ce service producteur pour l'enregistrement concerné. La valeur contenue dans le champ est un entier.
- « detached » : nombre d'unités archivistiques détachées symboliquement de ce service producteur. La valeur contenue dans ce champ est un entier.
- « symbolicRemained » : nombre actualisé d'unités archivistiques attachées symboliquement de ce service producteur pour l'enregistrement concerné et conservées dans la solution logicielle Vitam. 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

« 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.
- « attached » : nombre d'objets rattachés symboliquement de ce service producteur pour l'enregistrement concerné. La valeur contenue dans le champ est un entier.
- « detached » : nombre d'objets détachés symboliquement de ce service producteur. La valeur contenue dans ce champ est un entier.
- « symbolicRemained » : Nombre actualisé d'objets attachés symboliquement de ce service producteur pour l'enregistrement concerné et conservés dans la solution logicielle Vitam. 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

« ObjectSize » : Contient la répartition du volume total des fichiers du fonds par état pour l'opération journalisée (ingested, del

- « 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.
- « attached » : volume en octet des fichiers rattachés symboliquement de ce service producteur pour l'enregistrement concerné. La valeur contenue dans le champ est un entier.

- « detached » : volume en octet des fichiers détachés symboliquement de ce service producteur. La valeur contenue dans ce champ est un entier.
- « symbolicRemained » : Volume actualisé en octets des fichiers attachés symboliquement de ce service producteur pour l'enregistrement concerné et conservés dans la solution logicielle Vitam. 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

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

- Il s'agit d'un tableau.
- Ne peut être vide
- 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 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.

5.4 Collection AccessionRegisterSummary

5.4.1 Utilisation de la collection

Cette collection contient une vue macroscopique des fonds pris en charge dans la solution logicielle Vitam. Elle est constitué à partir des éléments du bordereau de transfert.

5.4.2 Exemple de la description dans le bordereau de transfert

Les seuls éléments issus du message ArchiveTransfer, utilisés ici sont ceux correspondants 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>
```

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

```
{
  "_id": "aefaaaaaaaaed4nrpaas4uak7cxykxiaaaaaq",
  "OriginatingAgency": "Vitam",
  "TotalObjects": {
    "ingested": 292,
    "deleted": 0,
  }
}
```

(suite sur la page suivante)

(suite de la page précédente)

```

    "remained": 292,
    "attached": 12,
    "detached": 0,
    "symbolicRemained": 12
  },
  "TotalObjectGroups": {
    "ingested": 138,
    "deleted": 0,
    "remained": 138,
    "attached": 14,
    "detached": 0,
    "symbolicRemained": 14
  },
  "TotalUnits": {
    "ingested": 201,
    "deleted": 0,
    "remained": 201,
    "attached": 37,
    "detached": 0,
    "symbolicRemained": 37
  },
  "ObjectSize": {
    "ingested": 35401855,
    "deleted": 0,
    "remained": 35401855,
    "attached": 917440,
    "detached": 0,
    "symbolicRemained": 917440
  },
  "creationDate": "2017-04-12T17:01:11.764",
  "_v": 1,
  "_tenant": 0
}

```

5.4.4 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 et sert de concaténation pour toutes les entrées effectuées sur ce producteur d'archives. Récupère la valeur contenue dans le bloc <OriginatingAgencyIdentifier> du message ArchiveTransfer. Cette valeur doit également correspondre au champ Identifier de la collection Agencies.
- Cardinalité : 1-1

Par exemple pour

```
<OriginatingAgencyIdentifier>FRAN_NP_051314</OriginatingAgencyIdentifier>
```

On récupère la valeur FRAN_NP_051314.

« **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 le 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.
- « attached » : nombre total d'objets attachés symboliquement de ce service producteur. La valeur contenue dans le champ est un entier.
- « detached » : nombre d'objets détachés symboliquement de ce service producteur. La valeur contenue dans ce champ est un entier.
- « symbolicRemained » : nombre actualisé d'objets attachés symboliquement de ce service producteur et conservés dans la solution logicielle Vitam. 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

« **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.
- « attached » : nombre de groupes d'objets attachés symboliquement de ce service producteur. La valeur contenue dans le champ est un entier.
- « detached » : nombre de groupes d'objets détachés symboliquement de ce service producteur. La valeur contenue dans ce champ est un entier.
- « symbolicRemained » : nombre actualisé de groupes d'objets rattachés symboliquement de ce service producteur et conservés dans la solution logicielle Vitam. 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.
- « attached » : nombre total d'unités archivistiques attachées symboliquement de ce service producteur. La valeur contenue dans le champ est un entier.
- « detached » : nombre d'unités archivistiques détachées symboliquement de ce service producteur. La valeur contenue dans ce champ est un entier.

- « symbolicRemained » : Nombre actualisé d'unités archivistiques attachés symboliquement de ce service producteur. 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
- « **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.
 - « attached » : volume total en octet des fichiers attachés symboliquement de ce service producteur. La valeur contenue dans le champ est un entier.
 - « detached » : volume total en octet des fichiers détachés symboliquement de ce service producteur. La valeur contenue dans ce champ est un entier.
 - « symbolicRemained » : volume actualisé en octet des fichiers rattachés symboliquement de ce service producteur et conservés dans la solution logicielle Vitam. 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
- « **creationDate** » : Date d'inscription du service producteur concerné dans le registre des fonds.
- La date est au format ISO 8601
"CreationDate": "2017-04-10T11:30:33.798",
 - 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.
- « **_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 Agencies

5.5.1 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 :

Identifiant	Name	Description
Identifiant du service agent	Nom du service agent	Description du service agent

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

```
{
  "_id": "aeaaaaaaaaevq6lcaamxsak7psyd2uyaaadq",
  "Identifiant": "Identifiant5",
  "Name": "Identifiant5",
  "Description": "une description de service agent",
  "_tenant": 2,
  "_v": 1
}
```

5.5.3 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 champs est obligatoirement renseignée dans le fichier CSV permettant de créer le service agent. En aucun cas la solution logicielle Vitam 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.

5.6 Collection Context

5.6.1 Utilisation de la collection

La collection Context permet de stocker unitairement les contextes applicatifs.

5.6.2 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": true,
  "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"
      ]
    }
  ]
}
```

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

```
{
  "_id": "aegqaaaaaaevq6lcaamxsak7psqdcmqaaaaq",
  "Name": "admin-context",
  "Status": true,
  "Identifiant": "CT-000001",
  "SecurityProfile": "admin-security-profile",
  "Permissions": [
    {
      "_tenant": 0,
      "AccessContracts": [],
      "IngestContracts": []
    },
    {
      "_tenant": 1,
      "AccessContracts": [],
      "IngestContracts": []
    }
  ]
}
```

(suite sur la page suivante)

```
        "_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,  
"EnableControl": false  
}
```

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 : [] »

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

« **Identifier** » : 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.

- Ce nom doit correspondre à celui d'un profil de sécurité enregistré dans la collection SecurityProfile.
- Il s'agit d'une chaîne de caractères
- 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

« **AccessContracts** » : tableau d'identifiants de contrats d'accès appliqués sur le tenant.

- Il s'agit d'un tableau de chaînes de caractères
- 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
- Peut être vide
- Cardinalité : 0-1

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

- Il s'agit d'une date au format ISO 8601
"CreationDate": "2017-04-10T11:30:33.798",
- Cardinalité : 1-1

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

- Il s'agit d'une date au format ISO 8601
"LastUpdate": "2017-04-10T11:30:33.798",

- Cardinalité : 1-1

« **ActivationDate** » : date d'activation du contexte applicatif.

- La date est au format ISO 8601

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

- Cardinalité : 0-1

« **DeactivationDate** » : date de désactivation du contexte applicatif.

- La date est au format ISO 8601

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

- Cardinalité : 0-1

« **_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.

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

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

5.7 Collection FileFormat

5.7.1 Utilisation de la collection FileFormat

La collection FileFormat permet de référencer et décrire 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 enregistré sur le tenant d'administration.

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

Ci-après, la portion d'un fichier de signature (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>
```

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

```
{
  "_id": "aeaaaaaaaaahbl62nabduoak3jc2zqciaadiq",
  "CreatedDate": "2016-09-27T15:37:53",
  "VersionPronom": "88",
  "PUID": "fmt/961",
  "Version": "2",
  "Name": "Mobile eXtensible Music Format",
  "Extension": [
    "mxmf"
  ],
  "HasPriorityOverFileFormatID": [
    "fmt/714"
  ],
  "MIMEType": "audio/mobile-xmf",
  "Group": "",
  "Alert": false,
  "Comment": "",
  "_v": 0
}
```

5.7.4 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 initialiser la collection.

- 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"

- Cardinalité : 1-1

« **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">
```

« **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.
- 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 signature.

« **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.
- Il est renseigné avec le contenu de l'attribut « MIMEType » de la balise <FileFormat>. Cet attribut est facultatif dans le fichier de signature.
- Cardinalité : 0-1

« **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 : 0-1

```
<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"  
],
```

« **Group** » : Champ permettant d'indiquer le nom d'une famille de format.

- Il s'agit d'une chaîne de caractères.
- C'est un champ propre à la solution logicielle Vitam.
- Cardinalité : 0-1

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

- Il s'agit d'un booléen dont la valeur est par défaut placée à false.
- Cardinalité : 0-1

« **Comment** » : commentaire.

- Il s'agit d'une chaîne de caractères.
- C'est un champ propre à la solution logicielle Vitam.
- Cardinalité : 0-1

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

- Il s'agit d'un tableau de chaînes de caractères.
- Il contient les valeurs situées entre les balises <Extension> elles-mêmes encapsulées entre les balises <File-Format>. 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.
- Cardinalité : 0-1

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"
],
```

« **_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.

5.8 Collection FileRules

5.8.1 Utilisation de la collection FileRules

La collection FileRules permet de stocker 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

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 Rule-MeasurementEnum, à savoir :

- MONTH
- DAY
- YEAR

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

```
{
  "_id": "aeaaaaaaaaahbl62nabduoak3jc4avsyaaaha",
  "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
}
```

5.8.3 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
```

Les préfixes indiquent le type de règle dont il s'agit. La liste des valeurs pouvant être utilisées comme préfixes ainsi que les types de règles auxquelles elles font référence sont disponibles en annexe.

« **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 FileRule.

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

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

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

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

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

5.9 Collection IngestContract

5.9.1 Utilisation de la collection

La collection IngestContract permet de référencer et décrire unitairement les contrats d'entrée.

5.9.2 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" : "aeaqaataaagbcaacaax56ak35rpo6zqaaaq"
  }
]
```

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

- Name
- Description

Un fichier d'import peut décrire plusieurs contrats.

5.9.3 Exemple de JSON stocké en base comprenant l'exhaustivité des champs de la collection IngestContract

```
{
  "_id": "aefqaaaaahbl62nabkzgak3k6qtf3aaaaaq",
  "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":
    "aeaqaataaagbcaacaax56ak35rpo6zqaaaq",
  "_tenant": 0,
  "_v": 0 }
}
```

5.9.4 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

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

- 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.
- Il s'agit d'une chaîne de caractères.
- 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 être ACTIVE ou INACTIVE
- Cardinalité : 1-1

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

- La date est au format ISO 8601

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

- Cardinalité : 1-1

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

- La date est au format ISO 8601

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

- Cardinalité : 1-1

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

- La date est au format ISO 8601

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

- Cardinalité : 0-1

DeactivationDate : date de désactivation du contrat.

- La date est au format ISO 8601

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

- Cardinalité : 0-1

MasterIsMandatory : Option qui rend obligatoire la présence d'un objet dont l'usage est de type Master (Physical ou Binary)

- True ou false
- Peut être vide mais sera à enregistrer à true
- Cardinalité : 0-1

DataObjectVersion : Option qui permet de préciser les types d'usages autorisés lors des imports.

- Liste des valeurs autorisées : Dissemination, Text Content, Physical Master, Binary Master, Thumbnail
- Peut être vide
- Cardinalité : 0-1

EveryDataObjectVersion : Option qui permet de préciser que tous les types d'usages sont autorisés lors des imports.

- Liste des valeurs autorisées : true, false
- Peut être vide
- Cardinalité : 0-1

« **CheckParentLink** » :

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

« **LinkParentId** » : point de rattachement automatique des SIP en application de ce contrat correspondant à l'identifiant d'une unité archivistique d'un plan de classement ou d'un arbre de positionnement.

- Il s'agit d'une chaîne de 36 caractères correspondant à un GUID et issue du champ _id d'un enregistrement de la collection Unit.
- Cardinalité : 0-1

L'unité archivistique concernée doit être de type FILING_UNIT ou HOLDING afin que l'opération aboutisse

« **_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.

5.10 Collection Profile

5.10.1 Utilisation de la collection profile

La collection Profile permet de référencer et décrire unitairement les notices de profil d'archivage.

5.10.2 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",
  }
]
```

(suite sur la page suivante)

(suite de la page précédente)

```

    "Format": "RNG"
  }
]

```

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

- Name
- Description
- Format

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

```

{
  "_id": "aegaaaaaaehlfs7waax4iak4f52mzriaaaaq",
  "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_aegaaaaaaehlfs7waax4iak4f52mzriaaaaq_20170522_092333.xsd"
}

```

5.10.4 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** » : indique l'identifiant significatif 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** » : indique le 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.

- Peut être 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** » : **Champ obligatoire** Indiquant le 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
"CreationDate": "2017-04-10T11:30:33.798",
- Champ peuplé par la solution logicielle Vitam.
- Cardinalité : 1-1

« **LastUpdate** » : date de dernière mise à jour de la notice du profil d'archivage dans la collection Profile.

- La date est au format ISO 8601
"LastUpdate": "2017-04-10T11:30:33.798"
- Champ peuplé par la solution logicielle Vitam.
- Cardinalité : 1-1

« **ActivationDate** » : date d'activation de la notice du profil d'archivage.

- La date est au format ISO 8601
"ActivationDate": "2017-04-10T11:30:33.798"
- Champ peuplé par la solution logicielle Vitam.
- Cardinalité : 1-1

« **DeactivationDate** » : date de désactivation de la notice du profil d'archivage.

- La date est au format ISO 8601
"DeactivationDate": "2017-04-10T11:30:33.798"
- 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.

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

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

« **Path** » : Champ contribué par Vitam lors d'un import de fichier XSD ou RNG.

- Indiquant le chemin pour accéder au fichier du profil d'archivage.
- Chaîne de caractères.
- Le format de fichier doit correspondre à celui qui est décrit dans le champ Format.
- Cardinalité : 0-1

5.11 Collection SecurityProfile

5.11.1 Utilisation de la collection

Cette collection contient les profils de sécurité mobilisés par les contextes applicatifs.

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

```
{
  "_id": "aegqaaaaaaaaeucszwabglyak64gjmgyaaaba",
  "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
}
```

5.11.3 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.
- Il s'agit d'une chaîne de 36 caractères correspondant à un GUID.
- 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

« **_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.

5.12 Collection VitamSequence

5.12.1 Utilisation de la collection

Cette collection permet de générer des identifiants significatifs pour les enregistrements des collections suivantes :

- IngestContract
- AccesContract
- Context
- Profile
- FileRules
- SecurityProfile
- Agencies

Ces identifiants sont 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 Identifier des collections concernées.

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

```
{
  "_id": "aeaaaaaaaaahkwxukabqteak4q5mtmdyaaaaq",
  "Name": "AC",
  "Counter": 44,
  "_tenant": 1,
  "_v": 0
}
```

5.12.3 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** » : Il s'agit du 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 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.
- 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 ArchiveUnitProfile

5.13.1 Utilisation de la collection

La collection ArchiveUnitProfile permet de référencer et décrire unitairement les documents type.

5.13.2 Exemple d'un fichier d'import de document type

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

```
{
  "Name": "Facture",
  "Description": "Document type d'une facture associée à un dossier de marché",
  "Identifiant": "AUP_IDENTIFIEUR_0",
  "Status": "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 document type sont :

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

Un fichier JSON peut décrire plusieurs documents type.

5.13.3 Exemple de JSON stocké en base comprenant l'“exhaustivité des champs de la collection ArchiveUnitProfile

```
{
  "_id": "aegaaaaabmhdh434aapnqalcd7mufiyaaaaq",
  "Identifier": "AUP_IDENTIFIER_0",
  "Name": "Facture",
  "Description": "Document type d'une facture associée à un dossier de marché",
  "Status": "ACTIVE",
  "ControlSchema": "{}",
  "LastUpdate": "10/12/2016",
  "CreationDate": "10/12/2016",
  "ActivationDate": "10/12/2016",
  "DeactivationDate": "10/12/2016"
  "_tenant": 11,
  "_v": 0
}
```

5.13.4 Détail des champs de la collection ArchiveUnitProfile

« **id** » : identifiant unique du Document type.

- 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 Document type.

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

« **Identifier** » : Identifiant signifiant donné au Document type.

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

« **Description** » : description du Document type.

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

« **Status** » : statut du Document type.

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

« **ControlSchema** » : Schema de contrôle du document type

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


```

{
  "$schema": "http://vitam-json-schema.org/draft-04/schema#",
  "id": "http://example.com/root.json",
  "type": "object",
  "additionalProperties": true,
  "properties": {
    "DescriptionLevel": {
      "type": "string",
      "enum": [
        "RecordGrp",
        "SubGrp",
        "File"
      ]
    }
  }
}

```

« **CreationDate** » : date de création du Document type.

- La date est au format ISO 8601

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

- Cardinalité : 1-1

« **LastUpdate** » : date de dernière mise à jour du Document type dans la collection ArchiveUnitProfile.

- La date est au format ISO 8601

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

- Cardinalité : 1-1

« **ActivationDate** » : date d'activation du Document type.

- La date est au format ISO 8601

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

- Cardinalité : 0-1

« **DeactivationDate** » : date de désactivation du Document type.

- La date est au format ISO 8601

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

- 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.

5.14 Collection Ontology

5.14.1 Utilisation de la collection

La collection Ontology permet de référencer et décrire unitairement les champs définissant l'ontologie VITAM.

5.14.2 Exemple d'un fichier d'import d'ontology

Les ontologies sont importées 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'ontology :

- Identifiant
- SedaField
- ApiField
- Description
- Type
- Origin
- ShortName
- Collections

Un fichier JSON décrit la totalité des champs de l'ontology (interne et externe).

5.14.3 Détail des champs de la collection Ontology

« **_id** » : identifiant unique du champ de l'ontology

- 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 unique du champ de l'ontology.

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

« **SedaField** » : Identifiant dans la nomenclature SEDA du champ de l'ontology

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

« **ApiField** » : Identifiant du champ de l'ontology qui sera retourné via le DSL

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

« **Description** » : description du champ de l'ontologie.

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

« **Type** » : type du champ 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 du champ de l'ontologie

- Peut avoir comme valeur : INTERNAL ou EXTERNAL
- Cardinalité : 1-1

« **ShortName** » : Identifiant technique pour traduction

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

« **Collections** » : Collections concernées par le champ de l'ontologie

- Il s'agit d'une liste de chaînes de caractères.
- Cardinalité : 1-n

6.1 Valeurs possibles pour le champ evType du LogBook Operation

L'ensemble des étapes, tâches et traitements sont détaillés dans la documentation Modèle de workflow.

6.2 Valeurs possibles pour le champ evType du LogBook LifeCycle

L'ensemble des étapes, tâches et traitements sont détaillées dans la documentation Modèle de workflow.

6.3 Valeurs possibles pour le champ evTypeProc (type de processus)

Process Type	Valeur	Description
Audit type process	AUDIT	Audit
Check type process	CHECK	Vérification
Destruction type process (v2)	DESTRUCTION	Élimination
Ingest type process	INGEST	Entrée
Preservation type process	PRESERVATION	Préservation
Rules Manager process	MASTERDATA	Données de base
Traceability type process	TRACEABILITY	Sécurisation
Update process	UPDATE	Mise à jour
Evidence Audit type process	EVIDENCEAUDIT	Audit de traçabilité

6.4 Catégories de règles possibles

Prefixe (Peut être modifié)	Type de règle correspondante	Description du type de règle
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

6.5 Valeurs possibles pour le champ Status de la collection AccessionRegisterDetail

Status type	Valeur
Le fonds est complet sauvegardé	STORED_AND_COMPLETED
Le fonds est mis à jour et sauvegardé	STORED_AND_UPDATED
Le fonds n'est pas sauvegardé	UNSTORED

6.6 Valeurs possibles pour le champ Name de la collection VitamSecquence

Prefixe	Type de collection correspondante	Description
AC	AccessContract	Contrats d'accès
IC	IngestContract	Contrats d'entrée
PR	Profile	Profils d'archivage
CT	Context	Contextes applicatifs
AG	Agencies	Services agents
RULE	FileRules	Règles de gestion
SEC_PROFILE	SecurityProfiles	Profils de sécurité

6.7 Diagramme du modèle de données Vitam

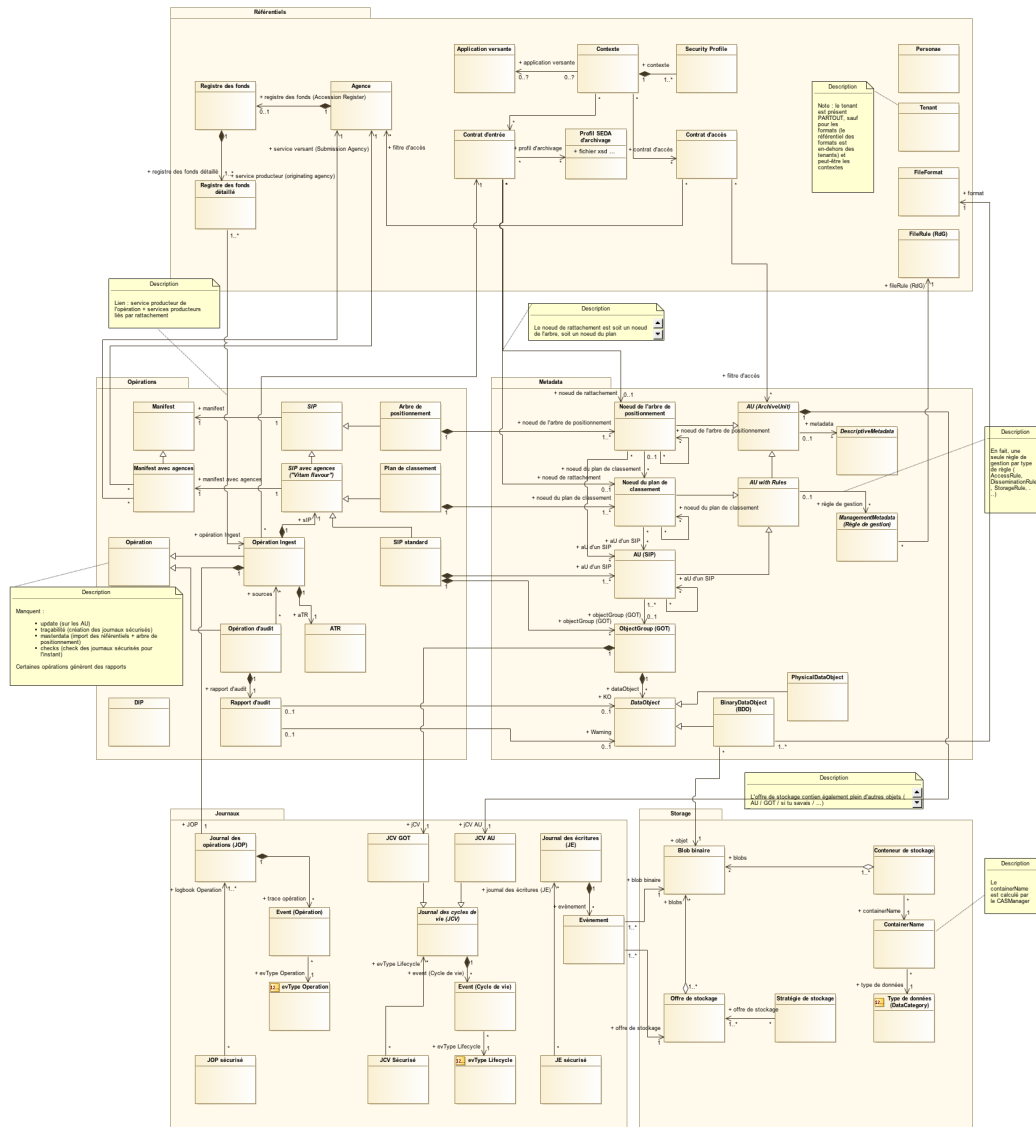


Fig. 1 – Modèle de données métier : vue d'ensemble.

6.8 Type d'indexation dans Elasticsearch par collection et par champ

Collection AccessContract

Champ	Type d'indexation
DataObjectVersion	Non analysé
Description	Analysé
Identifiant	Non analysé
Name	Analysé
OriginatingAgencies	Non analysé
RootUnits	Non analysé
Status	Non analysé

Collection AccessionRegisterDetail

Champ	Type d'indexation
ArchivalAgreement	Non analysé
OperationIds	Non analysé
OriginatingAgency	Non analysé
Status	Non analysé
SubmissionAgency	Non analysé

Collection AccessionRegisterSummary

Champ	Type d'indexation
OriginatingAgency	Non analysé

Collection Agencies

Champ	Type d'indexation
Description	Analysé
Identifiant	Non analysé
Name	Analysé

Collection Context

Champ	Type d'indexation
Identifiant	Non analysé
Name	Analysé
Permissions.AccessContracts	Non analysé
Permissions.IngestContracts	Non analysé
SecurityProfile	Non analysé

Collection FileFormat

Champ	Type d'indexation
Comment	Analysé
Extension	Non analysé
Group	Analysé
HasPriorityOverFileFormatID	Non analysé
MIMEType	Analysé
Name	Analysé
PUID	Non analysé
Version	Non analysé
VersionPronom	Non analysé

Collection FileRule

Champ	Type d'indexation
RuleDescription	Analysé
RuleDuration	Non analysé
RuleId	Non analysé
RuleMeasurement	Non analysé
RuleType	Non analysé
RuleValue	Analysé

Collection IngestContract

Champ	Type d'indexation
ArchiveProfiles	Non analysé
Description	Analysé
Identifiant	Non analysé
LinkParentId	Non analysé
Name	Analysé
Status	Non analysé

Collection Profile

Champ	Type d'indexation
Description	Analysé
Format	Non analysé
Identifiant	Non analysé
Name	Analysé
Path	Non analysé
Status	Non analysé

Collection ObjectGroup

Champ	Type d'indexation
FileInfo.CreatingApplicationName	Analysé
FileInfo.CreatingApplicationVersion	Analysé
FileInfo.CreatingOs	Analysé
FileInfo.CreatingOsVersion	Analysé
FileInfo.FileName	Analysé

Suite sur la page suivante

Tableau 1 – suite de la page précédente

Champ	Type d'indexation
Metadata.	Analysé
OtherMetadata	Analysé
_ops	Non analysé
_opi	Non analysé
_profil	Non analysé
_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	Analysé
_qualifiers.versions.FormatIdentification.FormatId	Non analysé
_qualifiers.versions.FormatIdentification.FormatLiteral	Analysé
_qualifiers.versions.FormatIdentification.MimeType	Analysé
_qualifiers.versions.MessageDigest	Non analysé
_qualifiers.versions.PhysicalDimensions.Depth.unit	Non analysé
_qualifiers.versions.PhysicalDimensions.Depth.value	Non analysé
_qualifiers.versions.PhysicalDimensions.Diameter.unit	Non analysé
_qualifiers.versions.PhysicalDimensions.Diameter.value	Non analysé
_qualifiers.versions.PhysicalDimensions.Height.unit	Non analysé
_qualifiers.versions.PhysicalDimensions.Height.value	Non analysé
_qualifiers.versions.PhysicalDimensions.Length.unit	Non analysé
_qualifiers.versions.PhysicalDimensions.Length.value	Non analysé
_qualifiers.versions.PhysicalDimensions.Shape	Non analysé
_qualifiers.versions.PhysicalDimensions.Thickness.unit	Non analysé
_qualifiers.versions.PhysicalDimensions.Thickness.value	Non analysé
_qualifiers.versions.PhysicalDimensions.Weight.unit	Non analysé
_qualifiers.versions.PhysicalDimensions.Weight.value	Non analysé
_qualifiers.versions.PhysicalDimensions.Width.unit	Non analysé
_qualifiers.versions.PhysicalDimensions.Width.value	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
_sp	Non analysé
_sps	Non analysé
_storage.offerIds	Non analysé
_storage.strategyId	Non analysé
_up	Non analysé
_us	Non analysé

Collection Unit

Champ	Type d'indexation
Addressee.BirthName	Analysé
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
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	Analysé
DescriptionLevel	Non analysé
Descriptions.*	Analysé
DocumentType	Analysé

Suite sur la page suivante

Tableau 2 – suite de la page précédente

Champ	Type d'indexation
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
Keyword.KeywordContent	Non analysé
Keyword.KeywordReference	Non analysé
Keyword.KeywordType	Non analysé
Language	Analysé
OriginatingAgency.Identifier	Non analysé
OriginatingAgencyArchiveUnitIdentifier	Non analysé
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.Corporname	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

Suite sur la page suivante

Tableau 2 – suite de la page précédente

Champ	Type d'indexation
RelatedObjectReference.References.DataObjectReference.DataObjectGroupReferenceId	Non analysé
RelatedObjectReference.References.DataObjectReference.DataObjectReferenceId	Non analysé
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.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
Signature.Masterdata	Non analysé
Signature.ReferencedObject.SignedObjectDigest.Algorithm	Non analysé
Signature.ReferencedObject.SignedObjectDigest.Value	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.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
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

Suite sur la page suivante

Tableau 2 – suite de la page précédente

Champ	Type d'indexation
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.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
Titles.*	Analysé
TransferringAgencyArchiveUnitIdentifier	Non 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
Writer.DeathPlace.PostalCode	Non analysé
Writer.DeathPlace.Region	Analysé
Writer.FirstName	Analysé
Writer.Function	Non analysé
Writer.Gender	Analysé
Writer.GivenName	Analysé
Writer.Identifier	Non analysé
Writer.Nationality	Analysé
Writer.Position	Analysé
Writer.Role	Analysé
_mgt.AccessRule.Inheritance.PreventRulesId	Non analysé
_mgt.AccessRule.Rules.Rule	Non analysé

Suite sur la page suivante

Tableau 2 – suite de la page précédente

Champ	Type d'indexation
_mgt.AppraisalRule.FinalAction	Non analysé
_mgt.AppraisalRule.Inheritance.PreventRulesId	Non analysé
_mgt.AppraisalRule.Rules.Rule	Non analysé
_mgt.ClassificationRule.Inheritance.PreventRulesId	Non analysé
_mgt.ClassificationRule.Rules.ClassificationLevel	Non analysé
_mgt.ClassificationRule.Rules.ClassificationOwner	Analysé
_mgt.ClassificationRule.Rules.Rule	Non analysé
_mgt.DisseminationRule.Inheritance.PreventRulesId	Non analysé
_mgt.DisseminationRule.Rules.Rule	Non analysé
_mgt.OriginatingAgency	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
_ops	Non analysé
_opi	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
Identifier	Non analysé
Name	Analysé
Permissions	Non analysé

6.9 Correspondances des champs spéciaux dans Vitam

Les champs dont le nom est préfixé d'un « _ » ne sont pas accessible directement, une correspondance est nécessaire pour y accéder.

Collection LogbookOperation

Champ	Champ interne
#id	_id
#tenant	_tenant

Collection AccessContract

Champ	Champ interne
#id	_id
#tenant	_tenant

Collection AccessionRegisterDetail

Champ	Champ interne
#id	_id
#tenant	_tenant

Collection AccessionRegisterSummary

Champ	Champ interne
#id	_id
#tenant	_tenant

Collection Agencies

Champ	Champ interne
#id	_id
#tenant	_tenant

Collection Context

Champ	Champ interne
#id	_id

Collection FileFormat

Champ	Champ interne
#id	_id

Collection FileRule

Champ	Champ interne
#id	_id
#tenant	_tenant
#version	_v

Collection IngestContract

Champ	Champ interne
#id	_id
#tenant	_tenant

Collection Profile

Champ	Champ interne
#id	_id
#tenant	_tenant

Collection Unit

Avertissement : 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
BLOQUÉ	_uds
#unitups	_up
#allunitups	_us
#nbunits	_nbc
#unitType	_unitType
#storage	_storage
#operations	_ops
#opi	_opi
#score	_score
#version	_v
#tenant	_tenant

Collection ObjectGroup

Champ	Champ interne
#id	_id
#profil	_profil
#qualifiers	_qualifiers
#size	_qualifiers.versions.size
#nbojects	_nbc
#originating_agency	_sp
#originating_agencies	_sps
#unitups	_up
#storage	_storage
#operations	_ops
#opi	_opi
#score	_score
#version	_v
#tenant	_tenant