Abilian
CS LAB

OPEN INTEROP · S4

Profil S4 : scénarios de test du CTK

Stefane Fermigier (Abilian SAS) · 4 novembre 2026 · DRAFT

Version 0.5
Licence CC BY 4.0

Cette note spécifie les scénarios du *Compliance Test Kit* (CTK) du profil S4, consacré à la réversibilité du stockage objet compatible S3. Chaque scénario vérifie des contrats comportementaux et alimente des critères de la scorecard. La note fixe aussi les jeux de données de référence, les verdicts, le format du rapport et celui du manifeste d'export.

Les scénarios CTK-01 à CTK-06 ont été retenus par l'atelier Open Interop du 9 octobre 2026 ; ils restent soumis à la validation du groupe de travail. Une implémentation, le CTK S4, couvre déjà CTK-01 à CTK-05. Chaque scénario se termine ci-dessous par une courte ligne d'état ; la note sur le CTK S4 décrit l'implémentation, son usage et les résultats des premiers essais.

1. Principes

- **Une exigence vaut par son test.** Chaque scénario renvoie aux contrats et aux critères de la scorecard qu'il évalue.
- **Structure commune** : préconditions, action, données, résultat attendu, mesure, seuil, verdict.
- **Une preuve reproductible.** L'acheteur peut exécuter, ou faire exécuter, un test versionné sur un jeu de données connu, puis obtenir un rapport vérifiable. Le rapport identifie le service, sa version et les fonctions revendiquées, le profil et la version du CTK, le jeu de données et les conditions d'exécution, les résultats, écarts et limites, ainsi que la commande qui rejoue le test (§6).
- **Un transfert par un outil indépendant.** La migration passe par un outil qui n'appartient ni au fournisseur source (A) ni au fournisseur cible (B), par copie directe ou par un export intermédiaire. Le rapport décrit le chemin suivi.
- **Une source figée.** Les écritures sont suspendues pendant le test. La migration pendant les écritures relève d'un périmètre ultérieur.
- **Assembler l'existant** : s3-tests (Ceph) pour les tests unitaires de conformité, rclone ou un client de référence équivalent pour les transferts, enfin les composants propres au CTK S4 (générateur de données, manifeste, comparateur).
- **Des cibles de référence** pour le réimport : Ceph RGW et Garage, déployés dans des conteneurs aux versions fixées. Les essais croisés visent en outre au moins deux fournisseurs indépendants.

Le CTK produit une preuve technique, au sens de la doctrine proposée pour Open Interop (§5.3) : autoévaluation quand le fournisseur l'exécute lui-même, vérification quand un tiers l'exécute, essai croisé quand plusieurs fournisseurs l'exécutent les uns vers les autres. Son usage dans un contrat demande une clause rédigée à part ; une proposition d'annexe contractuelle est prévue pour décembre 2026 (§7).

1.1 Verdicts

Chaque critère reçoit l'un des quatre verdicts du profil S4. Le rapport les code en anglais.

Verdict	Code du rapport	Sens
Conforme	`CONFORMANT`	Le critère est satisfait sur le périmètre testé
Non conforme	`NON_CONFORMANT`	Le critère n'est pas satisfait. Une fonction revendiquée mais non prise en charge par le service produit aussi ce verdict
Non applicable	`NOT_APPLICABLE`	Le critère sort du périmètre revendiqué (fonction optionnelle non déclarée) ou n'a pas été demandé
Non concluant	`INCONCLUSIVE`	L'environnement de test a fait défaut (réseau, identifiants ou horloge de l'outil, configuration de l'essai) ; le test doit être rejoué

Une règle s'applique à tous les scénarios : le service n'est pas conforme au profil testé si un objet manque, si un contenu est corrompu ou si un accès interdit devient autorisé. Une erreur renvoyée par le service, une fois les reprises épuisées, compte comme un défaut : l'objet refusé ou illisible manque, le listage en échec relève de A6, le contrôle de l'application témoin échoue. Seule une défaillance de l'environnement donne un verdict non concluant ; le rapport conserve alors ce qui a été établi. Les critères obligatoires ne se compensent pas et ne font l'objet d'aucune moyenne pondérée. Les performances et les coûts qualifient ensuite les conditions de la réversibilité sans racheter ces échecs. Le verdict global d'une exécution est non conforme dès qu'un critère l'est. Il est non concluant dès qu'un critère l'est ou qu'un critère obligatoire de la première étape (A1, A2, A3, A4, A10, A11) n'a pas été jugé. Il est conforme sinon. Le rapport liste les critères obligatoires non jugés.

1.2 Contenu de chaque fiche de test

Outre la structure commune, chaque fiche précise les versions, paramètres et commandes, les preuves conservées, les conditions d'arrêt ainsi que les fonctions obligatoires et optionnelles du périmètre.

2. Jeux de données de référence

Les jeux sont produits par le générateur du CTK à partir d'une graine publiée : même graine, mêmes clés, mêmes tailles, mêmes octets, mêmes métadonnées. Leur contenu et leurs sommes de contrôle sont donc connus à l'avance et identiques pour tous les fournisseurs. Ils sont synthétiques et ne contiennent aucune donnée personnelle. Chaque jeu est accompagné de son **manifeste de référence** (annexe A), qui sert d'oracle pour les comparaisons ; le rapport publie l'empreinte SHA-256 de ce manifeste.

2.1 Volumes

L'atelier a retenu deux volumes ; le CTK S4 en ajoute un troisième pour les exécutions rapides.

Volume	Objets	Taille totale	Objets multipartie	Usage
Essai rapide	60	20 Mio	5 Mio + 1 octet, 12 Mio	Vérification de l'installation ; hors profil
Fonctionnel	10 000	10 Go	5 Mio + 1 octet, 12 Mio, 64 Mio, 256 Mio	CTK-01 à CTK-04
Charge	100 000	1 To	5 Mio + 1 octet, 12 Mio, 4 × 1 Gio	CTK-05 et CTK-06

Le volume de charge reste à confirmer selon les moyens des essais croisés.

2.2 Familles de contenu

Chaque volume forme un seul jeu, qui mêle cinq familles de contenu. Le tableau décrit chaque famille et la part que couvre le générateur actuel.

ID	Contenu visé	Sert à tester	Générateur actuel
DS-S (*small*)	Objets nombreux, de tailles tirées autour de la moyenne du volume (environ 1 Mo au volume fonctionnel), dont un objet vide et un objet d'un octet, sur une arborescence de préfixes	Pagination, listage, complétude	Couvert, sauf l'arborescence : deux préfixes seulement (`objects/`, `keys/`) ; une arborescence de profondeur variable est à ajouter pour le listage avec délimiteur
DS-L (*large*)	Objets déposés en multipartie, parties de 5 Mio : de 5 Mio + 1 octet (dernière partie d'un octet) à 256 Mio au volume fonctionnel, jusqu'à 1 Gio au volume de charge	Multipartie, intégrité des gros objets, reprise, débit	Couvert. Des tailles de parties variées (pour l'ETag) et des objets de plusieurs Gio sont prévus dans une version ultérieure
DS-M (*metadata*)	Toutes les métadonnées système fixables, métadonnées utilisateur (dont valeurs non ASCII et taille proche de la limite), étiquettes	Préservation des métadonnées	Couvert, sauf les tailles limites : `Content-Type` sur tous les objets ; `Content-Encoding`, `Content-Language`, `Content-Disposition`, `Cache-Control` et `Expires` sur un objet sur quatre ; deux métadonnées utilisateur par objet, dont une valeur non ASCII encodée selon la RFC 2047 sur un objet sur cinq ; étiquettes sur un objet sur trois quand la fonction est revendiquée
DS-K (*keys*)	Clés difficiles : Unicode NFC et NFD de même apparence, emoji, espaces de tête et de fin, `//`, segments `.` et `..`, clés finissant par `/`, clés de longueur maximale, clés ne différant que par la casse	Restitution exacte des clés	Couvert : treize clés (NFC et NFD, emoji, espaces intérieur, de tête et de fin, `//`, segments `.` et `..`, `/` final, casse, clé de 1 024 octets)
DS-V (*versioned*)	Clés à plusieurs versions, marqueurs de suppression, sous-ensemble avec Object Lock et *legal hold*	Périmètre avancé (CTK-X3)	Absent

L'objet de 5 Mio + 1 octet, dont la dernière partie fait un octet, a mis au jour un défaut de Ceph RGW (§3, CTK-02).

3. Scénarios retenus

Vue d'ensemble

ID	Scénario	Volume	Critères de la scorecard	CTK S4
CTK-01	Export complet	Fonctionnel	A1, A5, A6, A7, A9	Implémenté (A1, A6, A7)
CTK-02	Intégrité	Fonctionnel	A2	Implémenté
CTK-03	Métadonnées et droits	Fonctionnel	A3, A4, A7, A10	Implémenté (A3, A4, A10)
CTK-04	Incident et reprise	Fonctionnel	A1, A2, A6 ; B9	Implémenté pour les coupures de connexion (A1, A2 ; B9)
CTK-05	Exploitation et charge	Charge	A8, A11 ; B10, B11	Implémenté
CTK-06	Performance de la migration	Charge	B1 à B3, B7	Non implémenté (scénario à définir)

CTK-01 : Export complet

- **Préconditions** : source figée, manifeste de référence connu ; un contrat avec B et une connexion entre A et B sont établis ; identifiants programmatiques avec droits de lecture complets chez A et d'écriture chez B ; outil de transfert de référence installé à une version fixée.
- **Action**
  1. Charger le jeu fonctionnel dans un bucket de A.
  2. Produire l'inventaire de A sur plusieurs pages de listage et le comparer au manifeste de référence.
  3. Migrer vers B avec l'outil de référence, sans intervention manuelle.
  4. Produire l'inventaire de B et le comparer au manifeste de A.
- **Données** : jeu fonctionnel, dont DS-K.
- **Résultat attendu** : tous les objets attendus sont présents chez B avec les mêmes clés ; aucun objet inattendu n'apparaît dans le périmètre ; aucune clé n'est listée deux fois.
- **Mesure** : comptage et correspondance des clés ; doublons et omissions de pagination ; étapes manuelles.
- **Seuil** : 100 % de correspondance ; zéro objet en plus ; zéro intervention pendant l'exécution nominale.
- **Verdict** : A1, A5, A6, A7 et A9.
- **Contrats couverts** : EXP-01, EXP-03, EXP-04, API-01, API-04, API-05.
- **État** : implémenté pour A1, A6 et A7. L'export de la configuration du bucket (A5) n'est pas évalué ; A9 est constaté (aucune intervention) sans recevoir de verdict.

CTK-02 : Intégrité

- **Préconditions** : celles de CTK-01 ; les corruptions sont injectées dans une copie de test ou dans la cible, puis réparées avant la suite des vérifications.
- **Action**
  1. Migrer un mélange de tailles comprenant des objets multipartie.
  2. Calculer, pour chaque objet, la taille et l'empreinte SHA-256 du contenu complet, chez A puis chez B, indépendamment des ETags.
  3. Corrompre quelques objets (octet modifié, troncature, objet remplacé) et rejouer la vérification.
- **Données** : jeu fonctionnel, dont DS-L.
- **Résultat attendu** : les contenus valides sont reconnus ; chaque corruption injectée est détectée et localisée (clé, nature de l'écart).

- **Mesure** : objets dont taille et empreinte correspondent ; corruptions détectées sur corruptions injectées.
- **Seuil** : 100 % d'empreintes conformes sur les objets valides ; 100 % des corruptions détectées ; aucun faux verdict positif.
- **Verdict** : A2.
- **Contrats couverts** : INT-01, INT-02, INT-03.
- **État** : implémenté. Les trois corruptions sont injectées dans la cible, détectées par clé, puis réparées. Les essais ont révélé un défaut de Ceph RGW (v19.2.7 puis v20.2.4, sur le backend `dbstore`) : la lecture d'un objet multipartie dont la dernière partie fait un octet ne se termine pas. Ils ont aussi montré que boto3 (à partir de 1.36), avec sa validation des sommes de contrôle activée par défaut, rejette la réponse de Garage pour un objet multipartie. Le client de référence désactive donc cette validation et le CTK vérifie l'intégrité lui-même par SHA-256.

CTK-03 : Métadonnées et droits

- **Préconditions** : liste des champs obligatoires du profil arrêtée ; **matrice d'accès** définie, c'est-à-dire un ensemble d'identités (lecteur, rédacteur, administrateur, public anonyme) et, pour chacune, les opérations autorisées et interdites sur des objets désignés.
- **Action**
  1. Migrer le jeu fonctionnel, dont DS-M.
  2. Comparer, objet par objet, les champs obligatoires chez A et chez B.
  3. Reconstruire chez B les droits convenus, par les moyens propres au modèle IAM de B.
  4. Rejouer la matrice d'accès chez A puis chez B et comparer les décisions.
- **Données** : jeu fonctionnel, dont DS-M ; matrice d'accès.
- **Résultat attendu** : propriétés préservées ; valeurs d'origine non reproductibles (date de dernière modification, ETag) transportées ; mêmes décisions d'accès chez A et chez B ; aucune exposition publique involontaire.
- **Mesure** : champs obligatoires conformes sur champs attendus ; cas d'accès correctement appliqués sur cas prévus.
- **Seuil** : 100 % des champs et 100 % des cas d'accès obligatoires.
- **Verdict** : A3, A4, A7 et A10.
- **Contrats couverts** : MET-01 à MET-04, MET-06, SEC-01.
- **État** : implémenté pour A3, A4 et A10 (A7 est jugé dans CTK-01). Les valeurs d'origine de `Last-Modified` et de l'ETag sont copiées chez B en métadonnées utilisateur préfixées `oi-orig-`. La matrice d'accès est un fichier TOML (identités, opérations, décisions attendues) ; l'accès anonyme en lecture et en listage y est ajouté avec la décision attendue `deny`. Sans matrice, A10 reçoit le verdict non applicable. Si la source s'écarte déjà de la matrice, A10 est non concluant. Sur Garage, revendiquer les étiquettes donne un verdict non conforme sur A3 (`GetObjectTagging` non implémenté).

Les droits se comparent par leur effet sur les accès, car les modèles IAM diffèrent d'un fournisseur à l'autre. La gestion des identités et les droits nécessaires à l'export et à l'import relèvent aussi du socle transverse.

CTK-04 : Incident et reprise

- **Préconditions** : celles de CTK-01 ; liste d'incidents définie (coupure réseau d'une durée donnée, erreurs 5xx et `SlowDown` injectées, identifiants expirés en cours de transfert).
- **Action**
  1. Lancer la migration du jeu fonctionnel.
  2. Provoquer chaque incident à un moment défini, puis laisser l'outil reprendre.
  3. Comparer le résultat final au manifeste de référence.

- **Données** : jeu fonctionnel, dont DS-L.
- **Résultat attendu** : reprise automatique ; rapport explicite des incidents ; aucune perte ni duplication logique.
- **Mesure** : complétude et intégrité finales ; délai de reprise ; volume retransmis ; interventions nécessaires.
- **Seuil** : résultat final identique au manifeste ; seuils de délai et de retransmission à calibrer après les premiers essais.
- **Verdict** : A1, A2 et A6 ; B9 publié.
- **Contrats couverts** : API-02, API-03, REP-01.
- **État** : implémenté pour les coupures de connexion, injectées côté client au-delà des nouvelles tentatives du SDK ; A1 et A2 sont jugés ; incidents, reprises, octets retransmis et durée de reprise sont publiés (B9). La copie reprend aussi après une erreur 5xx réelle. Restent à injecter les erreurs 5xx ou `SlowDown` ainsi que l'expiration d'identifiants.

CTK-05 : Exploitation et charge

- **Préconditions** : volume de charge ; application témoin (lecture, écriture, listage) et sa suite de tests ; durée d'engagement définie pour le scénario.
- **Action**
  1. Migrer le jeu de charge vers B.
  2. Révoquer tous les accès de l'application à A.
  3. Reconfigurer l'application vers B (adresse du point d'accès et identifiants uniquement), puis exécuter sa suite de tests.
- **Données** : jeu de charge.
- **Résultat attendu** : application opérationnelle sur B ; engagement de durée respecté.
- **Mesure** : tests applicatifs réussis sur tests prévus ; fonctions encore liées à A ; durée totale du lancement à la validation sur B ; débit utile.
- **Seuil** : 100 % des tests applicatifs du profil ; zéro dépendance résiduelle pour l'usage testé ; durée et débit publiés avec leurs conditions.
- **Verdict** : A8 et A11 ; B10 et B11 publiés.
- **Contrats couverts** : REC-01, REC-02, REC-04.
- **État** : implémenté. L'application témoin travaille sur B seul : lectures d'un échantillon vérifiées contre le manifeste, métadonnées, listage paginé, lectures partielles vérifiées contre les octets de référence (milieu de l'objet, frontière de partie, fin), lecture conditionnelle (`If-Match`, qui doit échouer avec 412 sur un ETag périmé), écriture. Quand les anciens identifiants de l'application chez A sont fournis, A doit les refuser (A8) ; sinon A8 est non applicable. Si A les accepte encore, A8 est non concluant. Les essais menés à ce jour sur Garage ont utilisé le volume d'essai rapide et le volume fonctionnel ; l'exécution au volume de charge suivra les essais croisés, menés au volume fonctionnel.

CTK-06 : Performance de la migration

L'atelier a retenu ce scénario sans en fixer le contenu. Il vise à vérifier l'absence de limitation du transfert, à conditions d'infrastructure équivalentes. La méthode « déclaré puis vérifié » sert de point de départ :

- **Préconditions** : fiche de déclaration avec B1, B2, B3 et B7 ; client placé de sorte que le réseau ne limite pas la mesure ; capacité du lien mesurée et consignée ; quotas suffisants confirmés par le fournisseur.
- **Action** : lecture continue de DS-L au parallélisme déclaré pendant au moins 30 minutes (B1), même mesure sur plusieurs buckets (B2), courbe débit et parallélisme (B7), montée en charge des requêtes jusqu'aux premières réponses de limitation (B3).

- **Mesure et seuil** : valeurs mesurées comparées aux valeurs déclarées, avec une tolérance à fixer ; réponses de limitation conformes au profil.
- **Conditions de comparaison** : même jeu de données, même client, même concurrence, mêmes régions, mêmes quotas et même chemin réseau. Le rapport publie le débit réseau disponible et les temps d'inventaire, de transfert, de vérification et de reconstruction. Un seuil universel de débit serait prématuré.
- **Contrats couverts** : PERF-01 à PERF-04.
- **État** : non implémenté, en attente de la définition du scénario par le groupe de travail. Les mesures de débit du CTK S4 restent indicatives.

4. Scénarios complémentaires et périmètres avancés

Ces scénarios ne font pas partie du socle CTK-01 à CTK-06. Ils restent au backlog.

ID	Scénario	Statut
CTK-X1	Grande cardinalité (10^6 objets) et listage	Complément de CTK-01 pour les gros inventaires (B4)
CTK-X2	Conformité au schéma d'un manifeste produit par le fournisseur	Schéma JSON des entrées publié (annexe A) ; le CTK S4 sait valider un manifeste fourni. Reste à obtenir des manifestes de fournisseurs (EXP-03, B5)
CTK-X3	Versions, marqueurs de suppression, Object Lock	Périmètre avancé, à approfondir
CTK-X4	Classes froides	Périmètre avancé (B6)

Sont aussi classés à part : versions historiques, marqueurs de suppression, stockage archivé, Object Lock, règles de cycle de vie, migration pendant les écritures, dépendances aux KMS, cohérence forte.

5. Architecture

Le CTK assemble les composants suivants. Leur conception et leur usage sont décrits dans la note sur le CTK S4.

Composant	Rôle	État
Orchestrateur de scénarios	Exécute les scénarios choisis, fixe paramètres, versions et graine, produit le rapport	Disponible
Générateur de jeux de données	Produit les objets et le manifeste de référence à partir d'une graine	Disponible (§2.2 pour la couverture des familles)
Outil de transfert	Copie de A vers B	Copie intégrée au CTK S4, à travers la machine qui l'exécute ; rclone ou s5cmd à brancher, avec version et paramètres fixés
Comparateur de manifestes	Compare manifestes de référence, source et cible ; écarts par critère	Disponible
Matrice d'accès	Compare les décisions d'accès chez A et chez B (CTK-03)	Disponible

Composant	Rôle	État
Injection d'incidents	Provoque les incidents de CTK-04	Coupures de connexion côté client ; un proxy réseau couvrirait les autres incidents
Application témoin	Suite de tests applicatifs de CTK-05	Disponible
Tests unitaires de conformité	Conformité de l'interface au profil	Existant (ceph/s3-tests), sélection par périmètre à faire
Cibles de référence	Réimport sur des implémentations libres	Garage v1.0.1 et Ceph RGW v19.2.7 en conteneurs ; RGW sur un cluster RADOS reste à mettre en place

Les composants nouveaux seront publiés sous une licence open source permissive ou faiblement copyleft, compatible avec les briques réutilisées ; le CTK S4 est publié sous licence Apache 2.0, le choix pour les autres composants reste à arrêter.

6. Format du rapport

Le rapport est un fichier JSON lisible par machine. Son empreinte est publiée et l'exécutant (fournisseur ou tiers) peut le signer. Il accompagne trois manifestes au format JSON Lines : référence, source et cible.

Élément	Contenu	CTK S4
Version du CTK et profil	Version de l'outil ; profil et version du profil	Produit
Fonctions revendiquées	`core`, `tags`…	Produit
Couverture S3	Opérations S3 que le kit utilise pour ces fonctions, dans la version datée du profil	Produit
Service	Fournisseur, service, version, région	Produit, pour la source et la cible (nom et version déclarés par l'exécutant)
Exécutant	Fournisseur ou tiers, nom	Produit (déclaré)
Environnement	Chemin de transfert, versions des outils, capacité du lien, début et fin	Produit (capacité du lien déclarée par l'exécutant)
Jeu de données	Volume, graine, nombre d'objets, octets, empreinte du manifeste de référence	Produit
Mesures	Durées de la migration seule (inventaire, transfert, vérification) et débit utile (B10, B11) ; requêtes par opération et octets lus de chaque côté (B12) ; reprise (B9 : incidents, reprises, octets renvoyés, durée de reprise) dans les résultats de CTK-04	Produit, à titre indicatif
Verdict global	Selon la règle du §1.1	Produit
Résultats	Par scénario et critère : verdict, nombre d'écarts, preuves (écarts détaillés, auto-test de corruption, décisions d'accès)	Produit

Élément	Contenu	CTK S4
Limites	Périmètre non couvert, conditions particulières	Produit
Reproduction	Commande et paramètres qui rejouent l'exécution	Produit
Signature	Signature de l'exécutant	Absent

Exemple abrégé :


```
{
  "profile": "S4 draft 2026-11-04",
  "features_claimed": ["core", "tags"],
  "s3_operations": ["CompleteMultipartUpload", "CreateMultipartUpload",
"GetObject", "…"],
  "source": {"name": "Fournisseur A", "version": "…", "endpoint": "https://…",
             "region": "…", "bucket": "…"},
  "target": {"name": "Garage", "version": "v1.0.1", "endpoint": "http://…",
             "region": "garage", "bucket": "…"},
  "environment": {"executor": "…", "transfer_path": "direct, through the machine
running the kit",
                  "link_capacity_mbps": 1000, "boto3": "1.x",
                  "started_at": "2026-…", "finished_at": "2026-…"},
  "dataset": {"volume": "functional", "seed": 0, "objects": 10000,
              "bytes": 10000000000, "reference_manifest_sha256": "…"},
  "measures": {"inventory_seconds": 0.0, "transfer_seconds": 0.0,
"verification_seconds": 0.0,
               "migration_seconds": 0.0, "bytes_copied": 0,
"useful_throughput_mib_s": 0.0,
               "source_usage": {"requests": {"GetObject": 0, "…": 0},
"bytes_read": 0},
               "target_usage": {"requests": {"PutObject": 0, "…": 0},
"bytes_read": 0}},
  "verdict": "NON_CONFORMANT",
  "not_judged": [],
  "results": [
    {"scenario": "CTK-03", "criterion": "A3", "verdict": "NON_CONFORMANT",
     "findings": 1, "evidence": {"…": ["…"]}}
  ],
  "limits": ["Source frozen during the run.", "…"],
  "reproduction": {"command": "…"}
}
```


7. Calendrier et TechSprint

L'atelier a proposé le calendrier suivant :

Date proposée	Livrable	État
Début novembre 2026	TechSprint et CTK minimal	En cours ; le CTK S4 couvre déjà CTK-01 à CTK-05
Fin novembre 2026	Essais croisés entre fournisseurs et rapport d'écarts	À venir
Décembre 2026	Profil V0.1 et proposition d'annexe contractuelle	À venir

- **Objectif du TechSprint** : CTK-01 à CTK-03 exécutés contre les deux cibles de référence et au moins deux fournisseurs indépendants ; CTK-04 et CTK-05 en complément.
- **Engagements recherchés** : deux fournisseurs indépendants (environnements, documentation, essais), un utilisateur (critères de réception, scénario applicatif), des développeurs du CTK et un référent sécurité (matrice d'accès).
- **Livrables** : premiers rapports CTK, liste des écarts constatés sur le profil, signalements aux projets amont (dont le défaut de Ceph RGW).

8. Points ouverts

- Liste des champs obligatoires de CTK-03 et transformations admises.
- Format de la matrice d'accès : le format TOML du CTK S4 est une proposition.
- Contenu de CTK-06, tolérances entre valeurs déclarées et mesurées.
- Seuils de délai et de retransmission de CTK-04.
- Volume de charge définitif, selon les moyens des essais croisés.
- Compléments du générateur pour couvrir entièrement DS-S, DS-L, DS-M et DS-K (§2.2), puis DS-V.
- Export de la configuration du bucket (A5) et en-tête d'export (annexes A.2 et A.3), non encore implémentés.
- Licence des composants autres que le CTK S4.

Annexe A. Manifeste d'export

Le manifeste utilise le format JSON Lines, avec une ligne par objet (par version dans le périmètre versionné). La configuration du bucket figure dans un fichier séparé. Un en-tête d'export indique le fournisseur, le bucket, la date, la version du schéma et les algorithmes. Une variante Parquet pourra servir aux très gros inventaires. Lorsque les données et le manifeste sont livrés ensemble, l'empaquetage s'inspire de BagIt (RFC 8493).

Le schéma JSON des entrées (version 0.1, JSON Schema 2020-12) couvre le niveau Core et les étiquettes. Le CTK S4 l'emploie pour valider ses propres manifestes et ceux que fournit un service. Un manifeste de référence indique `null` pour les champs attribués par le serveur (`last_modified`, `etag`).

A.1 Entrées objet

Champs du schéma 0.1. Tous sont présents dans chaque entrée, avec la valeur `null` quand l'objet n'a pas ce champ ; aucun autre champ n'est admis. Ce choix facilite la comparaison des manifestes.

Champ	Type	Reproductible par l'API	Remarques
`key`	chaîne, au plus 1 024 octets UTF-8	Oui	Le schéma limite les caractères, le CTK vérifie les octets ; encodage à préciser pour les octets non imprimables
`size`	entier ≥ 0	Oui	
`checksum`	`{algorithm, value}`	Oui (recalculable)	Objet complet ; `algorithm` vaut `sha256`, `value` en hexadécimal

Champ	Type	Reproductible par l'API	Remarques
`last_modified`	horodatage RFC 3339 ou `null`	Non	Date d'origine ; transportée chez B en métadonnée `oi-orig-last-modified`
`etag`	chaîne ou `null`	Non (multipartie, SSE-KMS)	Informatif, jamais utilisé comme preuve d'intégrité ; transporté chez B en métadonnée `oi-orig-etag`
`content_type`, `content_encoding`, `content_language`, `content_disposition`, `cache_control`	chaîne ou `null`	Oui	`null` si absent
`expires`	horodatage RFC 3339 ou `null`	Oui	`null` si absent
`user_metadata`	objet clé → valeur	Oui	Objet vide si aucune
`tags`	objet clé → valeur, ou `null`	Oui	Objet vide si la fonction n'est pas revendiquée ; `null` si elle l'est mais que le serveur ne la prend pas en charge

Les versions suivantes du schéma ajouteront les champs ci-dessous.

Champ	Type	Présence	Reproductible par l'API	Remarques
`version_id`	chaîne	Si versionné	Non	Identifiant d'origine
`is_latest`	booléen	Si versionné	Oui (par ordre de rejeu)	
`is_delete_marker`	booléen	Si versionné	Oui (par rejeu)	
`checksum` (autres algorithmes)	`{algorithm, value}`		Oui	CRC64NVME en plus de SHA-256
`multipart`	`{part_count, part_size}`	Si multipartie	Partiellement	Permet de reproduire l'ETag si souhaité
`storage_class`	chaîne	Toujours	Partiellement	Classe d'origine ; la correspondance avec les classes de la cible relève d'un chantier ultérieur
`retention`	`{mode, retain_until}`	Si Object Lock	Oui (avec la même échéance)	
`legal_hold`	booléen	Si Object Lock	Oui	
`encryption`	`{type, key_ref}`	Toujours	Non applicable	Informatif ; les données exportées sont

Champ	Type	Présence	Reproductible par l'API	Remarques
				en clair sauf chiffrement côté client
`owner`	chaîne	Facultatif	Non	Propriétaire d'origine

A.2 Configuration du bucket

La configuration du bucket n'a pas encore de schéma et le CTK S4 ne la produit pas.

Élément	Niveau	Remarques
Versionnage (état)	Core	
Politique d'accès, ACL, propriété des objets	Core	Exportées telles quelles, sans traduction
Chiffrement par défaut	Core	Type ; référence de clé à titre informatif
Object Lock par défaut	Governance	Mode, durée
Étiquettes de bucket	Governance	
Cycle de vie	Extended	
CORS	Extended	
Notifications, réplication	Extended	Souvent liées à des services propriétaires : à signaler comme dépendance résiduelle (C5)

A.3 En-tête d'export

L'en-tête n'a pas encore de schéma et le CTK S4 ne le produit pas.

Champ	Remarques
`schema_version`	Version du schéma
`profile`, `declared_level`	Profil et niveau
`provider`, `service`, `region`, `bucket`	
`generated_at`	Horodatage de production
`consistency`	Instantané cohérent ou listage au fil de l'eau ; borne temporelle
`object_count`, `version_count`, `total_size`	Totaux de contrôle
`checksum_algorithms`	Algorithmes utilisés