Abilian
CS LAB

OPEN INTEROP · S4

# Le CTK S4 : implémentation et premiers résultats

Stefane Fermigier (Abilian SAS) · 9 novembre 2026 · DRAFT

Version 0.4
Licence CC BY 4.0

Le CTK S4 (*Compliance Test Kit*) met en œuvre les scénarios de test du profil S4 de réversibilité du stockage objet. Il migre un jeu de données de référence d'un service source vers un service cible, vérifie le résultat critère par critère et produit un rapport qu'un tiers peut rejouer. Cette note décrit son fonctionnement, sa couverture et les enseignements de ses premiers essais. Le code est publié sous licence Apache 2.0 : https://git.sr.ht/~sfermigier/s4-ctk. La spécification des scénarios se trouve dans la note scénarios de test, les critères dans la scorecard.

## Synthèse


- **Cinq scénarios sur six sont exécutables** : export complet (CTK-01), intégrité avec corruption volontaire (CTK-02), métadonnées et droits d'accès (CTK-03), incident et reprise (CTK-04), exploitation sur la cible après révocation des accès à la source (CTK-05). CTK-06, sur la performance de la migration, attend que le groupe de travail le définisse.
- **Chaque essai produit une preuve reproductible** : jeu de données déterministe, manifestes d'export validés par un schéma JSON, l'un des quatre verdicts par critère, rapport qui donne la commande pour rejouer l'essai.
- **Le kit est lui-même testé** contre un simulateur et contre sept implémentations libres de l'API S3 (Garage, Ceph RGW, MinIO, SeaweedFS, Versity Gateway, RustFS, leBLOB), lancées dans des conteneurs, seules ou en migration de l'une vers l'autre.
- **Les premiers essais montrent l'écart entre déclaration et comportement.** Garage est conforme au niveau de base, sauf sur les lectures conditionnelles (`If-Match`) qu'il ignore ; il ne prend pas en charge les étiquettes d'objet. Ceph RGW, sur son moteur de stockage autonome, ne termine jamais la lecture de certains objets multiparties. Le client AWS de référence rejette par défaut, depuis janvier 2025, des réponses de services compatibles. Aucun de ces écarts n'apparaît dans une déclaration « compatible S3 ».
- **Prochaine étape** : les essais croisés entre fournisseurs de fin novembre 2026, au volume fonctionnel.

# 1. Rôle du CTK

Le profil S4 fixe les exigences ; le CTK vérifie qu'une migration réelle les respecte. Il joue le rôle de l'outil de transfert indépendant que l'atelier du 9 octobre 2026 a demandé : il ne dépend d'aucun fournisseur et s'adresse aux deux services par l'API S3, avec des identifiants ordinaires.

Le rapport du CTK apporte une preuve technique au sens de la proposition de doctrine (§5.2 et §5.3). Il sert aux trois niveaux de preuve : un fournisseur peut l'exécuter et publier son rapport (autoévaluation), un acheteur ou un tiers peut rejouer l'essai à partir du rapport (vérification) ; plusieurs fournisseurs peuvent l'exécuter ensemble lors d'une session organisée (essai croisé). Son usage dans un contrat demande une rédaction propre, que prépare la proposition d'annexe contractuelle attendue en décembre 2026.

# 2. Déroulement d'un essai

Un essai suit toujours les mêmes étapes :

1. **Génération du jeu de données.** Le CTK produit un jeu de données déterministe à partir d'une graine : mêmes paramètres, mêmes octets. Les données sont générées en flux et ne sont jamais stockées en entier sur la machine qui exécute le kit : le kit atteint ainsi le volume du profil de charge.
2. **Chargement de la source**, qui doit être vide au départ et rester figée pendant l'essai.
3. **Manifeste de référence et manifeste de la source.** Chaque manifeste décrit un objet par ligne (JSON Lines) : clé, taille, empreinte SHA-256 de l'objet complet, métadonnées système et utilisateur, étiquettes. Le manifeste de la source est construit en relisant chaque objet.
4. **Copie vers la cible.** La cible ne peut pas reproduire certaines valeurs (date de dernière modification, ETag d'origine) : elles sont transportées comme métadonnées utilisateur, conformément au contrat MET-04.
5. **Manifeste de la cible**, construit de la même façon que celui de la source.
6. **Évaluation des scénarios demandés**, critère par critère, avec l'un des quatre verdicts : conforme, non conforme, non applicable (fonction hors du périmètre revendiqué ou scénario non demandé), non concluant (environnement de test défaillant). Une erreur renvoyée par le service est un défaut : elle est consignée objet par objet ou contrôle par contrôle, puis l'essai se poursuit jusqu'à son rapport.
7. **Rapport.** Le rapport JSON contient l'exécutant, les fonctions revendiquées, la source et la cible (nom, version, région), le chemin de transfert et la capacité déclarée du lien, les versions des outils, le jeu de données et l'empreinte du manifeste de référence, les résultats avec leurs preuves, les limites de l'essai et la commande qui permet de le rejouer. Le code de sortie vaut 0 si l'essai est conforme, 1 s'il ne l'est pas, 2 s'il n'est pas concluant, 3 si le kit n'a pas pu s'exécuter (option erronée, matrice d'accès invalide, bucket non vide, défaut du kit) : ce dernier cas ne dit rien du service et ne produit pas de rapport. Un essai est aussi non concluant s'il laisse sans verdict un critère obligatoire de la première étape, par exemple faute de matrice d'accès : le rapport liste alors ces critères.

Le CTK valide aussi, contre le schéma JSON du manifeste, des manifestes produits par d'autres outils, par exemple un manifeste fourni par le fournisseur source (contrat EXP-03).

# 3. Couverture

| Scénario | Critères | Mise en œuvre |
|---|---|---|
| CTK-01 Export complet | A1 complétude, A6 listage, A7 ré-import | Inventaire sur plusieurs pages de listage, migration, comparaison clé par clé, treize clés difficiles (Uni- |

| Scénario | Critères | Mise en œuvre |
|---|---|---|
| | | code NFC et NFD, espaces, segments . et .., barres obliques, casse, clé de 1 024 octets) |
| CTK-02 Intégrité | A2 | Empreinte SHA-256 de chaque objet à chaque étape ; trois corruptions injectées dans la cible (octet modifié, troncature, remplacement) doivent être détectées et localisées, puis les objets sont réparés |
| CTK-03 Métadonnées et droits | A3, A4, A10 | Métadonnées système et utilisateur, étiquettes si elles sont revendiquées, valeurs d'origine transportées ; droits d'accès comparés par leur effet, à partir d'une matrice d'accès |
| CTK-04 Incident et reprise | A1, A2 | Coupures de connexion injectées au-delà des nouvelles tentatives du client ; la migration doit se terminer complète et intacte ; incidents, reprises, octets renvoyés et durée de reprise publiés (B9) |
| CTK-05 Exploitation et charge | A8 indépendance, A11 portabilité fonctionnelle | Une application témoin travaille sur la cible seule : lectures vérifiées contre le manifeste, métadonnées, listage paginé, lectures partielles vérifiées contre les octets de référence, lecture conditionnelle (`If-Match`), écriture ; les anciens identifiants de l'application doivent être refusés par la source ; durée totale et débit utile publiés (B10, B11) |
| CTK-06 Performance de la migration | À définir | Non implémenté |

**Matrice d'accès.** Pour CTK-03, l'acheteur décrit les identités (un profil d'identifiants côté source et côté cible, ou l'accès anonyme) et les cas à vérifier (identité, opération, décision attendue). Le CTK exécute chaque cas des deux côtés. Si la source elle-même ne donne pas la décision attendue, le critère A10 est « non concluant » : la matrice ne décrit pas la source. Si la source la donne et pas la cible, il est « non conforme ». Sauf indication contraire, la lecture et le listage anonymes doivent être refusés.

**Couverture S3 datée.** L'API S3 évolue au gré des décisions d'AWS : un résultat n'a de sens que pour une couverture identifiée et datée. Le rapport indique donc la version datée du profil vérifiée (`S4 draft 2026-11-04`) et les opérations S3 que le kit utilise. Au niveau de base : `AbortMultipartUpload`, `CompleteMultipartUpload`, `CreateMultipartUpload`, `DeleteObject`, `GetObject`, `HeadObject`, `ListObjectsV2`, `PutObject`, `UploadPart` ; avec les étiquettes, `GetObjectTagging` en plus. Un test du kit vérifie que cette liste

correspond aux opérations réellement appelées. Elle décrit les opérations exercées par le kit ; les fonctions prises en charge relèvent de la déclaration du fournisseur.

**Fonction revendiquée mais absente.** Si un fournisseur revendique une fonction, comme les étiquettes d'objet, sans que le service la mette en œuvre, le critère concerné est « non conforme ». Une erreur renvoyée par le service l'est aussi, une fois les reprises épuisées. Un point d'accès injoignable, ou un défaut propre à l'outil (identifiants refusés, horloge décalée), donne un rapport « non concluant », qui conserve les manifestes déjà construits.

## 4. Jeux de données

| Profil | Objets | Volume | Usage |
|---|---|---|---|
| Essai rapide | 60 | 20 Mio | Développement du kit, tests automatisés |
| Fonctionnel | 10 000 | 10 Go | Profil fonctionnel de l'atelier ; essais croisés |
| Charge | 100 000 | 1 To | Profil de charge de l'atelier ; CTK-05 en conditions réelles |

Chaque jeu contient des cas limites choisis pour révéler les écarts de comportement : objet vide, objet d'un octet, objets juste au-dessus du seuil du transfert multipartie, objets multiparties, clés difficiles, métadonnées système (dont `Expires`) et utilisateur (dont des valeurs non ASCII encodées selon la RFC 2047). Ces cas ont déjà prouvé leur utilité (§6).

## 5. Tests du kit

Un kit de conformité doit être lui-même vérifié. Le CTK S4 compte 141 tests automatisés, à trois niveaux :

- **tests unitaires** : jeux de données, manifestes et schéma, matrice d'accès, verdicts ;
- **tests contre un simulateur S3** (moto), en mémoire et par HTTP : chaque scénario, l'injection de corruptions et leur détection, la reprise avec et sans incident, un service qui répond mal, la ligne de commande ;
- **tests de bout en bout** du kit sur de vrais serveurs (Garage v1.0.1, MinIO `RELEASE.2025-09-07T16-13-09Z`), lancés dans des conteneurs jetables et supprimés à la fin de l'essai, y compris une migration d'une implémentation vers une autre.

Ces tests vérifient le kit. Ils se distinguent des essais, qui utilisent le kit pour établir si un service se comporte correctement. Les essais contre sept implémentations libres de l'API S3 (Garage v1.0.1, Ceph RGW v19.2.7, MinIO, SeaweedFS 4.52, Versity Gateway 1.10.0, RustFS 1.0.4, leBLOB 0.4.0.dev1) ont leur propre suite, hors des tests du kit ; le §6 en donne les résultats.

## 6. Premiers résultats

Le tableau suivant donne les résultats au 5 novembre 2026 (au 8 novembre pour leBLOB), tous obtenus sur un même serveur. Les essais ont été menés au volume d'essai rapide (60 objets, 20 Mio), sauf deux au volume fonctionnel, signalés dans le tableau ; aucun essai n'a encore été mené au volume de charge.

| Source et cible | Scénarios | Résultat |
|---|---|---|
| Simulateur → simulateur | CTK-01 à CTK-05, avec et sans étiquettes | Conforme |

| Source et cible | Scénarios | Résultat |
|---|---|---|
| Garage → seconde instance Garage, 10 000 objets, 10 Go | CTK-01 à CTK-05, niveau de base, matrice d'accès | Tous les critères obligatoires sont jugés. Conforme sur tous sauf A11, pour le défaut décrit à la ligne suivante. Migration en 479 s (inventaire 93 s, transfert 290 s à 32,8 Mio/s, vérification 95 s) sur une seule machine |
| Garage → Garage | CTK-01 à CTK-05, niveau de base | Conforme sur tous les critères sauf A11, avec reprises après les coupures injectées ; A8 conforme, l'ancienne clé étant refusée. Non conforme sur A11 : Garage ignore `If-Match` en lecture, un ETag périmé renvoie l'objet au lieu d'une erreur 412 |
| Garage, étiquettes revendiquées | CTK-03 | Non conforme sur A3 : la lecture des étiquettes renvoie `NotImplemented` |
| Garage, matrice d'accès (clé en lecture seule, accès anonyme) | CTK-03 | Conforme sur A10 : lectures autorisées, écritures et suppressions refusées, accès anonyme refusé. Verdict global non concluant : CTK-03 seul laisse d'autres critères obligatoires sans verdict |
| Garage, client boto3 par défaut | Aucun | Échec du client : depuis la version 1.36 (janvier 2025), boto3 valide par défaut des sommes de contrôle CRC et rejette la réponse de Garage pour un objet multipartie |
| Ceph RGW (moteur `dbstore`) | Aucun | Défaut du serveur : la lecture d'un objet multipartie dont la dernière partie fait un octet (5 Mio + 1 octet) ne se termine jamais, le serveur restant à 100 % de processeur ; même comportement en v20.2.4 |
| Ceph RGW (moteur `dbstore`) | CTK-01 à CTK-05 avec étiquettes ; matrice d'accès par politiques de bucket | Non exécutés tant que ce défaut n'est pas corrigé |
| MinIO, une instance | CTK-01 à CTK-05 avec étiquettes, matrice d'accès, ancienne clé | Tous les critères obligatoires sont jugés. Non conforme sur A1 seul : refuse les segments `.` et `..`, `//` et la clé de 1 024 octets. Conforme sur `If-Match` |
| SeaweedFS, une instance | Idem | Non conforme sur A1 et A3 : refuse les segments `.` et `..` et la clé de 1 024 octets ; réécrit sans le signaler `//` en `/` et `keys/trailing-` |

| Source et cible | Scénarios | Résultat |
|---|---|---|
| | | `slash/` en `keys/trailing-slash/trailing-slash` ; remplace un type `application/octet-stream` déclaré par un type déduit de l'extension `.txt` |
| Versity Gateway (moteur POSIX), une instance | Idem | Non conforme sur A1 : refuse les segments `.` et `..`, la clé de 1 024 octets et une clé finissant par `/` qui contient des données ; réécrit sans le signaler `//` en `/` |
| RustFS, une instance | Idem | Non conforme sur A1 et A3 : refuse les segments `.` et `..`, `//` et la clé de 1 024 octets ; conserve `Expires` et le renvoie en réponse à HEAD, mais pas à GET |
| Garage → MinIO, SeaweedFS, Versity Gateway, RustFS | CTK-01 à CTK-05, niveau de base, matrice d'accès, ancienne clé | Non conforme sur A1 et A7, ainsi que sur A3 pour SeaweedFS et RustFS : les clés refusées ou réécrites par chaque cible se perdent dans la migration. Contenu intact (A2) dans tous les cas |
| MinIO → Garage | Idem | Non conforme sur A1, pour les clés que MinIO a refusées au chargement ; sur A11, pour `If-Match` côté Garage. Rien n'est perdu entre les deux |
| MinIO → RustFS | CTK-01 à CTK-05 avec étiquettes, matrice d'accès, ancienne clé | Non conforme sur A1, pour les clés que MinIO a refusées au chargement ; sur A3 et A7, pour `Expires`. Tous les objets arrivent avec leurs étiquettes |
| leBLOB 0.4.0.dev1, une instance | CTK-01 à CTK-05 avec étiquettes, matrice d'accès (clé en lecture seule), ancienne clé | Conforme sur tous les critères. Conserve les treize clés difficiles et les étiquettes ; conforme sur `If-Match` ; la clé en lecture seule est refusée en écriture et en suppression |
| leBLOB 0.4.0.dev1 → seconde instance leBLOB, 10 000 objets, 10 Go | Idem | Conforme sur tous les critères. Migration en 1 110 s (inventaire 215 s, transfert 680 s à 14,0 Mio/s, vérification 215 s) sur une seule machine, chargée par d'autres travaux ; 494 s le même jour sous l'ancien nom, avec le même code, sur la machine moins chargée |
| MinIO → leBLOB 0.4.0.dev1 | CTK-01 à CTK-05 avec étiquettes, matrice d'accès, ancienne clé | Non conforme sur A1, pour les clés que MinIO a refusées au charge- |

| Source et cible | Scénarios | Résultat |
|---|---|---|
| | | ment. Tous les objets acceptés par MinIO arrivent, avec leurs étiquettes |
| Garage → leBLOB 0.4.0.dev1 | CTK-01 à CTK-05, niveau de base, matrice d'accès, ancienne clé | Conforme sur tous les critères : tous les objets arrivent intacts avec leurs métadonnées ; la clé en lecture seule le reste |
| leBLOB 0.4.0.dev1 → Garage | Idem | Non conforme sur A11, pour `If-Match` côté Garage. Rien n'est perdu |

Les essais à 10 Go vérifient que le kit et le service tiennent ce volume. Ils ne remplacent pas les essais croisés : les deux instances de chaque essai utilisent le même logiciel, sur la même machine, sans réseau réel entre elles. Les durées ne préjugent donc pas de celles d'une migration entre fournisseurs. CTK-05 y a tourné au volume fonctionnel, alors que le scénario prévoit le volume de charge.

Ces résultats illustrent trois arguments de la note de constat et propositions :

- **« Compatible S3 » ne précise pas les fonctions prises en charge.** Garage est un bon service S3 pour l'usage courant ; il ne met pas en œuvre les étiquettes et ignore `If-Match` en lecture. Ce second écart compte pour une migration : un outil qui reprend un transfert par plages ne peut pas détecter que l'objet a changé entre deux lectures. Une déclaration de conformité doit donc préciser les fonctions revendiquées, que le CTK vérifie.
- **Les cas limites révèlent des défauts invisibles à l'usage courant.** Le défaut de Ceph RGW ne touche qu'un moteur de stockage destiné aux tests, pour une seule taille d'objet. Un utilisateur ordinaire ne le rencontrera probablement jamais ; une migration complète, si. C'est la raison d'être d'un jeu de données conçu pour les cas limites.
- **Le client compte autant que le serveur.** L'incident de boto3 a rendu inutilisables, du jour au lendemain, de nombreux services compatibles avec la version précédente du client. Le CTK fixe donc le comportement de son propre client et vérifie l'intégrité par une empreinte SHA-256 qu'il calcule lui-même, sans se fier à l'ETag.

Seuls Garage et leBLOB conservent les treize clés difficiles. Le kit juge non conforme sur A1 aussi bien un service qui refuse une clé qu'un service qui la réécrit sans le signaler. Le rapport les distingue : une clé rejetée figure dans `service_errors` avec le code renvoyé, une réécriture apparaît comme une clé manquante et une clé inattendue. C'est la règle du contrat API-04 : toute clé de DS-K doit être acceptée et restituée à l'identique, car un client qui a stocké des objets sous une telle clé doit pouvoir partir avec eux. Ces essais concernent des configurations de test à un seul nœud, à des versions figées ; ils ne valent que pour elles.

leBLOB (`leblob` sur PyPI), anciennement theBLOB, est écrit par l'auteur du CTK S4 ; ses droits par clé ont été ajoutés pour ces essais. Le kit, le jeu de données, les critères et la matrice d'accès n'ont pas été modifiés pour lui. Un tiers peut rejouer ces essais à partir des rapports. La version 0.4.0.dev1, une version de développement, a été installée depuis PyPI. Depuis la version 0.3.0, leBLOB met en œuvre les étiquettes : les essais où il est seul ainsi que la migration depuis MinIO les revendiquent ; avec Garage, qui n'en a pas, les essais restent au niveau de base. La migration depuis MinIO est le seul de ces essais où les étiquettes sont revendiquées entre deux implémentations indépendantes ; elle offre une sortie au projet archivé de MinIO. leBLOB reste un logiciel expérimental, testé sur un seul nœud. Ses verdicts conformes

ne démontrent pas la réversibilité entre fournisseurs. La version 0.1.0, sans clé en lecture seule, obtenait un verdict non conforme sur A10 lors d'une migration depuis Garage : ce contrôle échoue donc bien quand la limite manque.

Le projet MinIO est archivé sur GitHub ; au 4 novembre 2026, ses images sur Docker Hub et son site de téléchargement ne sont plus disponibles. Le kit construit donc son image à partir de la dernière version publiée avec ses fichiers sur GitHub, vérifiés par les empreintes qui y sont publiées. Un logiciel libre de référence peut ainsi cesser d'être distribué sans que l'interface qu'il implémente change : d'où l'intérêt d'une sortie vérifiable.

Le défaut de Ceph RGW reste à signaler au projet Ceph. Un moteur RGW adossé à une grappe RADOS, plus proche des déploiements de production, permettrait d'exécuter les scénarios complets contre une seconde implémentation.

## 7. Limites

- Le CTK couvre le niveau de base du profil et les étiquettes. Les versions d'objets, le verrouillage (Object Lock), le stockage froid et l'export de la configuration des buckets relèvent des périmètres avancés et ne sont pas couverts.
- Les données transitent par la machine qui exécute le kit. Les mesures de durée et de débit dépendent donc de son réseau ; elles sont publiées avec leurs conditions et restent informatives. Elles couvrent la migration seule : les tests eux-mêmes et les coupures injectées en sont exclus. Le kit compte aussi, de chaque côté, les requêtes par opération et les octets lus, y compris ceux des reprises après une coupure injectée ; il ne calcule pas de coût, qui dépend des tarifs de chaque fournisseur.
- La source doit rester figée pendant l'essai : la migration pendant les écritures n'est pas traitée.
- Le manifeste de la source est construit en relisant tous les objets et double donc la lecture des gros buckets. Un manifeste fourni par le fournisseur (contrat EXP-03) l'éviterait.

## 8. Suite

1. **Essais croisés de fin novembre 2026** entre au moins deux fournisseurs indépendants, au volume fonctionnel, puis CTK-05 au volume de charge. Les demandes faites aux fournisseurs et le modèle de déclaration de conformité figurent dans la note Participer aux essais croisés.
2. **CTK-04** : injecter les erreurs serveur (5xx) et l'expiration des identifiants en plus des coupures de connexion ; la copie reprend déjà après une erreur 5xx réelle.
3. **CTK-06**, une fois défini par le groupe de travail.
4. **Seconde cible de référence** : Ceph RGW adossé à RADOS, ou une version corrigée du moteur autonome ; signalement du défaut au projet Ceph, de l'écart sur `If-Match` au projet Garage, des réécritures de clés et de l'absence d'`Expires` à leurs projets.
5. **Documentation d'usage**, pour qu'un fournisseur ou un acheteur exécute lui-même le kit à partir du dépôt public.
6. **Périmètres avancés : versions et Object Lock** (scénario CTK-X3, jeu DS-V des scénarios de test), comme un niveau revendiqué, à la manière des étiquettes aujourd'hui. Toutes les versions et tous les marqueurs de suppression doivent arriver dans leur ordre, avec leur contenu, leurs métadonnées et leur date d'origine (contrats EXP-02 et REC-03). Le mode de rétention, la date de fin et le *legal hold* doivent arriver aussi. La cible doit appliquer la rétention, ce que vérifie une suppression refusée (MET-05). La source doit permettre de lire et d'exporter les objets verrouillés. Les objets verrouillés d'essai ne peuvent être supprimés avant leur échéance, en mode *compliance* pas même par leur

propriétaire : il faut des rétentions courtes, le mode *governance* quand c'est possible et des buckets réservés à ces essais.

## Références

- Série S4 : synthèse, constat et propositions, contrats comportementaux, scorecard, scénarios de test.
- Proposition de doctrine pour Open Interop ; fiche cloud.
- CTK S4, code source (licence Apache 2.0) : https://git.sr.ht/~sfermigier/s4-ctk
- Garage, service de stockage objet compatible S3 : https://garagehq.deuxfleurs.fr/
- Ceph RGW (RADOS Gateway) : https://docs.ceph.com/en/latest/radosgw/
- MinIO : https://github.com/minio/minio ; SeaweedFS : https://github.com/seaweedfs/seaweedfs ; Versity Gateway : https://github.com/versity/versitygw ; RustFS : https://github.com/rustfs/rustfs
- moto, simulateur des services AWS : https://github.com/getmoto/moto