Abilian
CS LAB

OPENINTEROP · S4

Le CTK S4 : implémentation et premiers résultats

Stefane Fermigier (Abilian SAS) · 31 octobre 2026 · DRAFT

Version 0.1
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 ce que ses premiers essais ont montré. 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 deux implémentations libres de l'API S3, Garage et Ceph RGW, lancées dans des conteneurs.
- **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 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, ce qui permet d'atteindre 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).
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.

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 (Unicode 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

Scénario	Critères	Mise en œuvre
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.

Fonction revendiquée mais absente. Si un fournisseur revendique une fonction, comme les étiquettes d'objet, et que le service ne la met pas en œuvre, le critère concerné est « non conforme ». Un point d'accès injoignable ou des identifiants refusés donnent un rapport « non concluant ».

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 39 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, la ligne de commande ;
- **tests de bout en bout** contre deux implémentations libres de l'API S3, Garage v1.0.1 et Ceph RGW v19.2.7, lancées dans des conteneurs jetables et supprimées à la fin de l'essai.

Les tests de bout en bout servent aussi de cibles de référence : ils montrent comment le kit se comporte face à des services réels, avec leurs écarts.

6. Premiers résultats

Le tableau suivant donne les résultats au 31 octobre 2026. Tous les essais ont été menés au volume d'essai rapide (60 objets, 20 Mio) ; aucun essai n'a encore été mené aux volumes fonctionnel ou de charge.

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

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.

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.
- 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, ce qui double 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.
2. **CTK-04** : ajouter les erreurs serveur (5xx) et l'expiration des identifiants aux coupures de connexion.
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, et de l'écart sur `If-Match` au projet Garage.
5. **Publication du code**, sous licence Apache 2.0, conformément au principe 4 de la doctrine, avec la documentation nécessaire pour qu'un fournisseur ou un acheteur l'exécute lui-même.

Références

- Série S4 : synthèse, constat et propositions, contrats comportementaux, scorecard, scénarios de test.
- Doctrine OpenInterop ; fiche cloud.
- Garage, service de stockage objet compatible S3 : https://garagehq.deuxfleurs.fr/
- Ceph RGW (RADOS Gateway) : https://docs.ceph.com/en/latest/radosgw/
- moto, simulateur des services AWS : https://github.com/getmoto/moto