SkyVault Security Whitepaper

Version 1.0.0 · Juillet 2026

SkyVault

7 juillet 2026

Résumé exécutif

SkyVault est une application mobile de stockage cloud **zero-knowledge** pour Android (V1.0.0). Les fichiers sont chiffrés **sur l'appareil** avec **AES-256-GCM** avant tout envoi vers Backblaze B2. SkyVault ne détient jamais la clé maître de déchiffrement en clair.

Ce document décrit l'implémentation réelle V1.0.0, pas une feuille de route marketing. Les éléments non livrés (iOS App Store, sync type Drive, déchiffrement web, pentest externe) sont explicitement exclus.

Pilier	V1.0.0
Chiffrement fichiers	AES-256-GCM côté client
Dérivation clé	PBKDF2 + sel unique
Transport	TLS 1.3 (HTTPS)
Stockage blobs	Backblaze B2 (EU / US)
Identité & métadonnées	Firebase Auth + Firestore
Auth renforcée	2FA TOTP, passkeys Android
RGPD	Export manifeste + suppression compte
Audit externe	Prévu 2027 (non réalisé)

Table des matières

1. Introduction et périmètre V1

1.1 Objectif du service

SkyVault permet à un utilisateur de :

- Sauvegarder photos, vidéos et documents dans le cloud ;
- Libérer de l’espace sur son téléphone tout en conservant des copies chiffrées ;
- Partager des fichiers via des liens protégés ;
- Gérer son compte, son quota et ses abonnements.

1.2 Périmètre V1.0.0 (Android)

Inclus :

- Application Flutter flavor `production` (`com.smartstorageliberator.app`) ;
- Chiffrement zero-knowledge bout-en-bout pour le contenu fichier ;

- Upload / download résilients (pause, reprise, multipart ≥ 5 Mo) ;
- Auth e-mail, Google, Facebook, GitHub ;
- Passkeys sur Android (WebAuthn) ;
- 2FA TOTP + codes de récupération ;
- Vérification compte → quota gratuit 5 Go ;
- Abonnements Google Play via RevenueCat ;
- Export RGPD (manifeste JSON) et suppression compte ;
- Trust Center public (`skyvaultcloud.xyz/trust`).

Exclus ou reportés (annoncés honnêtement sur le site) :

- Application iOS / Apple Sign-In / App Store ;
- Synchronisation automatique type Google Drive ;
- Déchiffrement des partages dans le navigateur web ;
- Pentest indépendant (objectif 2027) ;
- Certifications ISO 27001 / SOC 2 ;
- Bug bounty rémunéré (en préparation).

1.3 Principes de conception

1. **Zero-knowledge** : le serveur ne peut pas déchiffrer les fichiers utilisateur.
2. **Défense en profondeur** : chiffrement client + TLS + règles Firestore strictes + URLs signées à durée limitée.
3. **Transparence** : documentation publique, statut automatisé, changelog versionné.
4. **Minimisation des données** : seules les métadonnées nécessaires au service sont stockées.

2. Modèle de menace

2.1 Adversaires considérés

Adversaire	Capacité	Mitigation V1
Opérateur SkyVault malveillant	Accès Firestore, B2, logs	Zero-knowledge : blobs illisibles
Attaquant réseau (MITM)	Interception HTTPS	TLS 1.3, certificate pinning implicite via stacks OS
Attaquant avec compte Firebase valide	Lecture règles Firestore	Règles par UID, pas d’accès cross-user
Attaquant avec lien partage	Téléchargement blob chiffré	Mot de passe partage optionnel, blob reste chiffré
Phishing identifiants	Vol e-mail / MDP	Passkeys (origin binding), 2FA TOTP

Adversaire	Capacité	Mitigation V1
Malware sur appareil	Lecture clés en mémoire	Hors périmètre ZK classique ; biométrie + verrou vault
Compromission B2	Accès blobs	Contenu chiffré, clés non stockées sur B2

2.2 Hors périmètre explicite

- Récupération si l'utilisateur perd **à la fois** sa clé de récupération et l'accès au coffre ;
- Protection contre un appareil déjà rooté / compromis au niveau OS ;
- Résistance quantique (algorithmes post-quantiques non déployés).

3. Architecture système

3.1 Vue d'ensemble

```
[Appareil Android]
    │ Chiffrement AES-256-GCM (clé locale)
    │ Dérivation PBKDF2 (mot de passe vault)
    ▼
[Cloud Functions europe-west1]
    │ Auth Firebase · callables métier
    │ URLs signées upload/download
    ▼
[Firestore]  métadonnées (UID-scoped)
[Backblaze B2]  blobs chiffrés (EU / US)
[Firebase Hosting]  site + Trust Center
[Cloudflare]  DNS + CDN download
[RevenueCat]  abonnements Play
[Resend]  e-mails transactionnels
```

3.2 Composants et responsabilités

Composant	Rôle	Voit le contenu fichier ?
App Flutter	Chiffrement, UI, transferts	Oui (après déverrouillage vault)
Firebase Auth	Identité, OAuth, passkeys	Non

Composant	Rôle	Voit le contenu fichier ?
Firestore	Métadonnées fichiers, profil, plans	Non (noms, tailles, hashes)
Cloud Functions	Orchestration, URLs signées, RGPD	Non
Backblaze B2	Stockage objet	Non (blobs chiffrés)
RevenueCat	Statut abonnement	Non
CDN Cloudflare	Cache download	Non (blobs chiffrés)

3.3 Régions

- Cloud Functions : `europe-west1` ;
- Firestore : région Europe lorsque applicable ;
- B2 : bucket EU (`eu-central-003`) ou US (`us-east-005`) selon `complianceRegion` utilisateur.

4. Cryptographie des fichiers

4.1 Algorithme

AES-256-GCM (Galois/Counter Mode) :

- Taille de clé : 256 bits ;
- Nonce unique par fichier (généré localement, CSPRNG) ;
- Tag d’authentification GCM : intégrité + confidentialité.

4.2 Flux upload

1. L’utilisateur sélectionne un fichier et déverrouille le vault (mot de passe / biométrie).
2. L’app lit les octets du fichier en local.
3. Génération d’une clé de fichier (ou utilisation de l’envelope encryption · voir §5).
4. Chiffrement AES-256-GCM → blob `ciphertext || tag`.
5. Calcul SHA-256 du contenu **avant** chiffrement (déduplication locale, intégrité debug).

6. Demande d’URL signée upload via callable Cloud Function.
7. PUT direct vers B2 (sans transit par les Functions pour le corps du fichier).
8. Écriture métadonnées Firestore (nom, taille, `storageKey`, hash, etc.) via Function.

4.3 Flux download

1. Callable `getDownloadUrl` (ou batch) avec auth Firebase.
2. URL signée B2 ou CDN (durée limitée).
3. Téléchargement du blob chiffré.
4. Déchiffrement AES-256-GCM sur l’appareil avec la clé maître déverrouillée.

4.4 Ce que le serveur ne reçoit jamais

- Mot de passe vault en clair ;
- Clé maître en clair ;
- Contenu fichier en clair ;
- Nonces ou clés de fichier en clair.

5. Gestion des clés et envelope encryption

5.1 Clé maître

- Générée localement (CSPRNG) à la création du coffre ;
- Jamais envoyée au serveur en clair ;
- Une **clé de récupération** est montrée une fois à l’utilisateur (hachée côté stockage local).

5.2 Enveloppe chiffrée (serveur)

Une copie de la clé maître est stockée côté serveur **chiffrée** :

- Dérivation via **PBKDF2** à partir du mot de passe vault ;
- Sel unique par utilisateur (stocké Firestore, pas secret en soi) ;
- Le serveur stocke l’enveloppe, pas la clé dérivée.

Sans le mot de passe vault, l’enveloppe est inutilisable.

5.3 PBKDF2 (V1)

- Fonction de dérivation standard (PKCS#5) ;

- Itérations configurées pour ralentir les attaques par force brute ;
- **Migration Argon2id** planifiée (roadmap sécurité, non livrée V1).

5.4 Coffre-fort PIN (local)

- PIN 6 chiffres : haché **PBKDF2-SHA256** + sel aléatoire ;
- Stockage : `flutter_secure_storage` (Keychain/Keystore) ;
- Jamais synchronisé cloud ;
- Auto-verrouillage : inactivité 2 min + passage arrière-plan.

5.5 Dossiers secrets

- Verrou par dossier : `FolderLockService` (PBKDF2 local) ;
- Dossiers `isSecret` filtrés des vues normales.

6. Authentification et session

6.1 Méthodes V1

Méthode	Statut V1
E-mail / mot de passe	✅
Google Sign-In	✅
Facebook	✅
GitHub	✅
Apple Sign-In	❌ (iOS / V2)
Téléphone / SMS	✅ (vérification profil)
Passkeys	✅ Android
2FA TOTP	✅

6.2 Session Firebase

- Token JWT Firebase géré par SDK ;
- Callables exigent `requireAuth` (UID) ;
- Déconnexion : purge session + verrou vault local.

6.3 Vérification compte

- Nouveau compte : `accountStatus: pending`, `storageLimit: 0` ;
- Profil complet + e-mail vérifié → demande vérification ;
- Après validation : `verified`, quota **5 Go** gratuit ;
- Abonnements Play bloqués tant que compte non vérifié.

7. Passkeys (WebAuthn / FIDO2)

7.1 Implémentation V1 (Android)

- Enrôlement et authentification via WebAuthn ;
- Clé privée dans Secure Enclave / TPM ;
- Seule la **clé publique** est enregistrée côté serveur ;
- **Origin binding** : protection anti-phishing.

7.2 App Links / Digital Asset Links

- `public/.well-known/assetlinks.json` déployé sur Hosting prod ;
- Associe le domaine `skyvaultcloud.xyz` au package Android production.

7.3 Limites

- iOS / passkeys Safari : non disponible V1 ;
- Nécessite appareil compatible FIDO2.

8. Authentification à deux facteurs (TOTP)

8.1 Standard

- **RFC 6238** (TOTP) ;
- Codes 6 chiffres, fenêtre 30 secondes ;
- Compatible Google Authenticator, Authy, etc.

8.2 Codes de récupération

- Générés à l’activation 2FA ;
- Usage unique ;
- Stockés **hachés SHA-256** (jamais en clair) ;
- Collection `twoFactorSecrets` : **inaccessible client** (Firestore rules).

8.3 Flux login

1. Auth primaire (e-mail, Google, passkey…) ;
2. Si 2FA activé : demande code TOTP ou récupération ;
3. Vérification côté Function avant émission tokens complets.

9. Coffre-fort local (PIN / biométrie)

9.1 VaultLockService

- Création PIN : double saisie + confirmation ;
- Déverrouillage : PIN ou biométrie (`local_auth`) ;
- Récupération : clé de récupération (affichée une fois).

9.2 VaultGateScreen

- Écran unique : création, déverrouillage, récupération ;
- `VaultGuard` protège routes sensibles (Settings coffre, dossiers secrets).

9.3 Menace couverte

- Accès physique court à l'appareil déverrouillé : auto-lock ;
- **Non couvert** : appareil rooté avec extraction Keystore avancée.

10. Transferts et résilience réseau

10.1 File d'attente

- Uploads et downloads gérés par moteur de transferts dédié ;
- Écran **Transferts** : progression, pause, reprise, annulation.

10.2 Multipart upload

- Fichiers ≥ 5 Mo : session multipart B2 ;
- Parts signées individuellement ;
- Reprise après coupure réseau.

10.3 Foreground Service (Android)

- Transferts ≥ 32 Mo : service premier plan pour éviter kill OS ;
- Notification persistante conforme Android 14+.

10.4 Compression vidéo

- Vidéos > 50 Mo : phase compression locale avant chiffrement ;
- Réduit bande passante et quota.

10.5 Anti-doublon

- Comparaison nom + taille + hash local avant upload ;
- Évite re-envoi de fichiers déjà présents.

11. Partage sécurisé

11.1 Modèle

- Callable crée document `shares` avec token unique ;
- Paramètres : expiration, nombre max de téléchargements, mot de passe optionnel ;
- Mot de passe partage : sel + hash (jamais stocké en clair).

11.2 App Links `/s/`

- Lien `https://skyvaultcloud.xyz/s/#k=...` ;
- Ouverture dans l'app Android (intent-filters + assetlinks) ;
- Page web : message honnête si déchiffrement navigateur non disponible (V1.2).

11.3 Zero-knowledge préservé

- Le blob téléchargé via lien reste **chiffré** ;
- Déchiffrement dans l'app avec clés utilisateur.

12. Infrastructure cloud

12.1 Firebase (Google Cloud)

- **Authentication** : identité uniquement ;
- **Firestore** : métadonnées, profils, partages, plans ;
- **Hosting** : site vitrine, Trust Center, `.well-known` ;
- **Cloud Functions v2** : Node.js 22, région `europe-west1`.

12.2 Backblaze B2

- API S3-compatible ;
- Buckets séparés EU / US ;
- Credentials par région (secrets Firebase) ;
- Lifecycle : suppression à la purge compte / corbeille.

12.3 Cloudflare

- DNS `skyvaultcloud.xyz` ;
- CDN download : `cdn-eu.skyvaultcloud.xyz`, `cdn-us.skyvaultcloud.xyz` ;
- **Ne déchiffre pas** les fichiers (cache blobs chiffrés).

12.4 RevenueCat

- Webhook HTTP vers Function `revenueCatWebhook` ;
- Synchronise `plan` et `storageLimit` Firestore ;
- Paiements : exclusivement Google Play (V1 Android).

12.5 Resend

- E-mails transactionnels : reset MDP, alertes internes, notifications marque ;
- Pas de contenu fichier dans les e-mails.

13. Règles d’accès Firestore

13.1 Principes

- Lecture / écriture client limitée au `uid` authentifié ;
- Collections sensibles (`files`, `shares`, `twoFactorSecrets`) : **écriture Functions uniquement** ;
- Pas d’énumération cross-tenant.

13.2 Exemples de restrictions

Collection	Client read	Client write
`users/{uid}`	Propre UID	Champs limités
`files`	Propre UID	❌
`twoFactorSecrets`	❌	❌
`shares`	Token public contrôlé	❌

13.3 Vérification

- Tests manuels : tentative écriture directe `files` → refus ;
- `PERMISSION_DENIED` attendu sans auth valide.

14. URLs signées Backblaze B2

14.1 Upload

- Callable génère URL PUT signée (durée courte) ;
- Client envoie directement le blob chiffré ;
- Function enregistre métadonnées après succès.

14.2 Download

- Callable génère URL GET signée ou route CDN ;
- Expiration minutes ;
- Pas de listing bucket côté client.

14.3 Suppression RGPD

- `deleteObject` B2 lors suppression compte ou purge corbeille ;
- Clés `storageKey` par région utilisateur.

15. CDN et transport

15.1 TLS

- HTTPS obligatoire (Hosting, Functions, B2, CDN) ;
- TLS 1.3 négocié par les stacks modernes.

15.2 Intégrité

- GCM tag par fichier ;
- SHA-256 pré-chiffrement (debug / déduplication).

15.3 Performance

- Batch `getDownloadUrls` pour éviter N+1 callables ;
- CDN réduit latence download EU/US.

16. Notifications (FCM)

16.1 Usage V1

- Push transferts terminés, sauvegarde auto WiFi nuit ;
- Inbox serveur (collection dédiée) ;
- **Pas de contenu fichier** dans les notifications.

16.2 Tokens

- FCM token lié à `uid` ;
- Révoqué à la déconnexion / suppression compte.

17. Abonnements et paiements

17.1 Flux

1. Utilisateur vérifié achète sur Google Play ;
2. RevenueCat reçoit l'événement ;
3. Webhook signé → Function prod ;
4. Mise à jour `plan`, `storageLimit` Firestore ;
5. App reflète via listener / refresh.

17.2 Données paiement

- SkyVault **ne stocke pas** de coordonnées bancaires ;
- Google Play gère PCI ;

- RevenueCat : identifiants abonnement uniquement.

17.3 Plans V1

- Gratuit vérifié : 5 Go ;
- Plans payants : Starter, Pro, etc. (catalogue Play).

18. RGPD : export et suppression

18.1 Export (`exportUserDataManifest`)

Callable authentifié retournant un **manifeste JSON** :

- Profil sanitizé (pas de secrets) ;
- Inventaire fichiers (métadonnées, pas contenu déchiffré) ;
- Dossiers, partages actifs ;
- Horodatages, plan, consentements.

L'export ZIP déchiffré complet est produit **côté client** à partir du manifeste + téléchargements.

18.2 Suppression (`deleteUserAccount`)

- Confirmation textuelle obligatoire : `SUPPRIMER` ;
- Séquence : blobs B2 → collections Firestore → index téléphone → Firebase Auth ;
- Irréversible ; perte définitive si pas de backup local.

18.3 Sous-traitants documentés

- Google (Firebase), Backblaze, Cloudflare, RevenueCat, Resend ;
- Politique : `skyvaultcloud.xyz/privacy`.

19. Journalisation et données visibles

19.1 Ce que SkyVault peut voir

Donnée	Visible serveur ?
E-mail, UID, plan	Oui
Nom fichier, taille, dates	Oui

Donnée	Visible serveur ?
Hash SHA-256 pré-chiffrement	Oui
Contenu fichier	**Non**
Mot de passe vault	**Non**
Clé maître	**Non**

19.2 Logs Cloud Functions

- Erreurs techniques, IDs corrélation ;
- Pas de log intentionnel de contenu utilisateur ;
- Webhook RevenueCat : statuts HTTP, pas de PAN.

19.3 Demandes légales

- Métadonnées fournissables si obligation légale ;
- Contenu chiffré : **indéchiffrable** sans clé utilisateur ;
- Rapport transparence : `/trust/transparency`.

20. Trust Center et transparence opérationnelle

20.1 Pages publiques

- Hub `/trust` : cryptographie, architecture, zero-knowledge, status, bug bounty ;
- Schémas interactifs `/security` ;
- Changelog versionné `/changelog`.

20.2 Statut automatisé (V1)

- Endpoint `GET /api/status` : sondes Firestore, B2, Hosting ;
- Cache 5 minutes ;
- Page `/trust/status` mise à jour par JavaScript ;
- Incidents : `public/api/incidents.json` (format date, durée, cause, impact, résolution).

20.3 Alertes internes

- Scheduler `probePublicStatusAndAlert` toutes les 10 minutes ;
- E-mail `support@skyvaultcloud.xyz` si dégradation / rétablissement.

21. Divulgation responsable

21.1 Contact

- `support@skyvaultcloud.xyz` (objet : Security disclosure) ;
- `security.txt` : `/.well-known/security.txt` ;
- Délai réponse cible : 72 h ouvrées.

21.2 Scope

- App Android production, site, Functions, règles Firestore, flux chiffrement.

21.3 Hall of Fame

- Section publique sur `/trust/bug-bounty` ;
- Publication avec accord explicite du chercheur.

22. Limites connues V1

1. **PBKDF2** au lieu d'Argon2id (roadmap) ;
2. **Pas de pentest externe** (prévu 2027) ;
3. **iOS absent** : surface d'attaque réduite mais pas de parité plateforme ;
4. **Pas de SLA contractuel** publié (historique uptime en construction) ;
5. **security@** : routage e-mail dédié en cours (contact fonctionnel via support@) ;
6. **Malware local** : hors garantie zero-knowledge standard.

23. Roadmap sécurité (post-V1)

Item	Cible annoncée
Migration Argon2id	V1.x
Déchiffrement web partages	V1.2
Bug bounty rémunéré	Budget disponible
Pentest indépendant	2027
SLA uptime public	Après 90 j historique
ISO 27001 / SOC 2	Long terme

24. Glossaire

- **AES-GCM** : chiffrement symétrique authentifié.
- **CSPRNG** : générateur aléatoire cryptographiquement sûr.
- **Envelope encryption** : clé de données chiffrée par clé dérivée du mot de passe.
- **Zero-knowledge** : le fournisseur ne possède pas les clés de déchiffrement.
- **TOTP** : mot de passe à usage unique basé sur le temps.
- **WebAuthn** : standard passkeys FIDO2.
- **Blob** : objet binaire stocké tel quel (ici, chiffré).

Annexe A · Inventaire des Cloud Functions V1

Callable / HTTP	Rôle sécurité
`getUploadUrl`	URL signée PUT, auth requise
`registerUploadedFile`	Métadonnées post-upload
`initMultipartUpload` / `completeMultipartUpload`	Gros fichiers
`getDownloadUrl` / `getDownloadUrls`	URLs signées GET
`softDeleteStoredFile` / `restoreStoredFile`	Corbeille
`deleteStoredFile`	Suppression définitive
`createShareLink` / `revokeShareLink`	Partage
`getSharedFile` / `peekSharedLink`	Accès partage contrôlé
`exportUserDataManifest`	RGPD export
`deleteUserAccount`	RGPD suppression
`revenueCatWebhook`	Webhook signé Play
`getPublicStatusHttp`	Statut public
`probePublicStatusAndAlert`	Alerte interne scheduler
`sendBrandedPasswordResetEmailCallable`	Reset MDP Resend

Toutes les callables métier fichiers exigent un UID Firebase valide. Les webhooks exigent une autorisation dédiée (secret RevenueCat).

Annexe B · Matrice de tests RC1 (juillet 2026)

Domaine	Tests validés device	Preuve
Auth e-mail + Google ✅		RC1 smoke §1

Domaine	Tests validés device	Preuve
2FA TOTP	✅	RC1 smoke §1
Upload ZK + reprise	✅	RC1 smoke §2
Download déchiffrement	✅	RC1 smoke §2
Partage + App Links	✅	RC1 smoke §6
Vérif compte 5 Go	✅	RC1 smoke §10.5
Abonnements Play	✅	RC1 smoke §11
Corbeille	✅	RC1 smoke §7
Analyse stockage	✅	RC1 smoke §11bis
Passkeys Android	✅	RC1 P1
Webhook RevenueCat HTTP 200	✅	`setup_revenuecat_webhook.sh --test`
Export RGPD	🔜	Callable livré, test device à formaliser
Suppression compte	🔜	Callable livré, test device à formaliser
Facebook / GitHub OAuth	🔜	Code livré, test device à formaliser

Annexe C · Scénarios de menace détaillés

C.1 Compromission compte Firebase d’un autre utilisateur

Scénario : attaquant obtient un token valide pour UID victime.

Impact : lecture métadonnées, demande URLs download (blobs chiffrés), pas de déchiffrement sans MDP vault.

Mitigations : 2FA, passkeys, pas de stockage MDP vault serveur, alertes connexion (roadmap).

C.2 Fuite bucket B2

Scénario : accès read-only massif aux objets.

Impact : téléchargement blobs chiffrés uniquement.

Mitigations : clés IAM minimales, URLs signées courtes, chiffrement client.

C.3 Ingénierie sociale support

Scénario : demande « réinitialiser mon coffre ».

Procédure : refus de récupération clé maître ; pas de backdoor ; suppression compte si demandé (RGPD).

C.4 Lien partage deviné

Scénario : brute-force token partage.

Mitigations : tokens longs aléatoires, expiration, max downloads, mot de passe optionnel.

C.5 Mise à jour app malveillante

Scénario : APK sideload modifié.

Mitigations : distribution Play Store officielle, signature release, pas de sideload documenté prod.

Annexe D · Données Play Console (Data safety)

Résumé aligné zero-knowledge (voir `V1/PLAY_DATA_SAFETY.md` pour formulaire complet) :

Type	Collecté ?	Chiffré transit	Chiffré repos	Partage tiers
E-mail	Oui	Oui	Oui (Firebase)	Non vendu
Nom / profil	Oui	Oui	Oui	Non vendu
Photos / fichiers	Oui (blob)	Oui TLS	Oui AES-GCM client	Backblaze sous-traitant
Identifiants achat	Via Play	Oui	RevenueCat	Non
Contenu fichier lisible SkyVault	**Non**	.	.	.

Message clé pour reviewers : *« Le contenu des fichiers est chiffré sur l’appareil avant upload. SkyVault ne peut pas lire les fichiers utilisateur. »*

Annexe E · Procédure incident public

1. Détection (sonde auto, support, monitoring).
2. Triage fondateur (< 2 h pour indisponibilité).
3. Mise à jour `scripts/data/public-incidents.json`.
4. Rebuild site + deploy Hosting.
5. Vérification `/trust/status` + e-mail testeurs si beta active.

Format obligatoire : date, durée, cause, impact, résolution (document `V1/TRUST_INCIDENTS_PROCESS.md`).

Annexe F · FAQ technique sécurité

Q : SkyVault peut-il lire mes photos ?
R : Non. Seuls des blobs chiffrés sont stockés. La clé reste sur votre appareil (ou dérivée de votre mot de passe vault).

Q : Que se passe-t-il si je perds mon mot de passe vault ?
R : Utilisez la clé de récupération montrée à la création. Sans elle ni accès au coffre déverrouillé, les fichiers sont irrécupérables (by design).

Q : Les employés Google / Backblaze peuvent-ils lire mes fichiers ?
R : Ils peuvent voir des blobs binaires opaques, pas le contenu déchiffré.

Q : Le site /trust/status est-il fiable ?
R : Oui · sondes automatisées toutes les 5–10 min sur Firestore, B2, Hosting. Pas encore de SLA contractuel.

Q : Un pentest a-t-il validé l'app ?
R : Non en V1. Premier pentest indépendant annoncé pour 2027.

Q : iOS est-il sécurisé pareil ?
R : L'app iOS n'est pas publiée en V1. Le modèle sera le même à la sortie.

Annexe G · Parcours utilisateur détaillés (sécurité)

G.1 Première installation

1. Téléchargement depuis Google Play (test interne ou production).

2. Écran autorisations bootstrap (stockage, notifications si demandées).
3. Création compte e-mail ou OAuth.
4. Création coffre vault : génération clé maître CSPRNG.
5. Affichage **unique** de la clé de récupération · l'utilisateur doit la sauvegarder.
6. Choix mot de passe vault (jamais envoyé serveur en clair).
7. Enveloppe chiffrée uploadée (PBKDF2 + sel).
8. État compte `pending`, quota 0 jusqu'à vérification profil.

G.2 Upload quotidien

1. Utilisateur sélectionne médias depuis galerie.
2. Si vault verrouillé : demande PIN / biométrie / MDP vault.
3. Pour chaque fichier :
    - Lecture octets locaux ;
    - Hash SHA-256 ;
    - Génération nonce unique ;
    - Chiffrement AES-256-GCM ;
    - Demande URL signée ;
    - PUT B2 ;
    - Callable enregistre métadonnées.
4. Barre de progression + notification fin (FCM).
5. En cas de coupure : reprise multipart ou simple selon taille.

G.3 Consultation cloud

1. Liste fichiers depuis Firestore (métadonnées uniquement).
2. Tap download → URL signée → blob chiffré.
3. Déchiffrement mémoire → affichage / export galerie.
4. Miniatures : sidecar chiffré ou génération locale.

G.4 Partage à un tiers

1. Utilisateur crée lien depuis fiche fichier.
2. Options : expiration, max downloads, mot de passe.
3. Token aléatoire long stocké Firestore.
4. Lien `skyvaultcloud.xyz/s/#k=...` partagé.
5. Destinataire : app Android ou page web informative.
6. Blob téléchargé reste chiffré jusqu'à ouverture dans app autorisée.

G.5 Activation 2FA

1. Réglages → Sécurité → 2FA.
2. Scan QR (secret TOTP généré).
3. Codes récupération affichés une fois (hachés serveur).
4. Prochain login : challenge TOTP obligatoire.

G.6 Passkey Android

1. Réglages → Passkeys → Enregistrer.
2. WebAuthn crée paire clés dans Secure Enclave.
3. Clé publique stockée Firestore.
4. Login suivant : biométrie / PIN device, anti-phishing par origin.

G.7 Export RGPD

1. Réglages → Confidentialité → Exporter mes données.
2. Callable `exportUserDataManifest` → JSON profil + inventaire.
3. App télécharge blobs et produit export local (ZIP déchiffré côté client).
4. Aucun envoi automatique du ZIP au support.

G.8 Suppression compte

1. Réglages → Supprimer mon compte.
2. Saisie confirmation `SUPPRIMER`.
3. Callable supprime B2, Firestore, Auth.
4. Irréversible · rappel clé de récupération perdue.

Annexe H · Corbeille et rétention

- Soft delete : flag `isDeleted`, fichier masqué UI ;
- Rétention 30 jours (`purgeAfter`) ;
- Scheduler `purgeExpiredTrash` : suppression B2 + doc Firestore ;
- Restauration possible avant purge via `restoreStoredFile`.

Annexe I · Résidence données (EU / US)

- Champ `complianceRegion` sur profil utilisateur ;
- Bucket B2 EU ou US selon région ;
- Migrations planifiées via `runResidencyMigrations` ;

- CDN aligné `cdn-eu` / `cdn-us`.

Annexe J · Comparatif cloud (synthèse)

Critère	SkyVault V1	Drive / Dropbox classique
Chiffrement client avant upload	Oui	Non (chiffrement serveur)
Opérateur peut lire fichiers	Non (ZK)	Oui (techniquement)
Résilience transferts	Pause / reprise	Variable
Open source client	Non (V1)	Non
Pentest public	2027 prévu	Variable

Annexe K · Schéma métadonnées Firestore (extrait)

Document `users/{uid}`

Champs principaux (aucun contenu fichier) :

- `email`, `displayName`, `plan`, `accountStatus`
- `storageUsed`, `storageLimit`, `complianceRegion`
- `personal`, `account`, `security`, `wallet`, `consents`
- `profileCompletion`, `createdAt`, `verifiedAt`

Document `files/{fileId}`

- `userId`, `name`, `size`, `mimeType`
- `storageKey` (chemin B2)
- `sha256` (hash pré-chiffrement)
- `folderId`, `isDeleted`, `purgeAfter`
- Timestamps upload / modification

Document `shares/{shareId}`

- `token`, `fileId`, `ownerId`
- `expiresAt`, `maxDownloads`, `downloadCount`
- `passwordHash`, `passwordSalt` (si mot de passe)

Collections opaques client

- `twoFactorSecrets` : lecture/écriture interdites côté SDK
- Écriture `files` : Functions uniquement

25. Références

- NIST SP 800-38D (GCM)
- RFC 6238 (TOTP)
- WebAuthn Level 2 (W3C)
- RGPD (UE) 2016/679
- Documentation publique : https://skyvaultcloud.xyz/trust
- Architecture transferts : dépôt `ARCHITECTURE_TRANSFERTS.md`
- Checklist conformité : `V1/PRE_PROD_CHECKLIST.md`

Document version : 1.0.0
Date : 7 juillet 2026
Alignement code : branche production RC1
Contact : support@skyvaultcloud.xyz