Liberty Shield — Whitepaper sécurité

Document de référence technique. Décrit l'architecture cryptographique réellement déployée, le modèle de menace, et les mesures de protection. À jour au 9 juillet 2026.

---

1. Principe

Liberty Shield est un coffre-fort numérique **zero-knowledge**. Tout le chiffrement et le déchiffrement ont lieu **sur l'appareil de l'utilisateur**, dans le navigateur. Le serveur ne reçoit, ne stocke et ne transmet jamais que des données déjà chiffrées, opaques. Il ne détient aucune clé, et ne connaît jamais le mot de passe maître. Conséquence assumée : si l'utilisateur perd ses accès, personne — pas même nous, pas même sur injonction — ne peut les récupérer à sa place.

Cette garantie repose sur des primitives cryptographiques standard, fournies par **libsodium** (aucune cryptographie « maison »), et sur une hiérarchie de clés vérifiable décrite ci-dessous.

---

2. Architecture cryptographique

2.1 Primitives

Fonction	Algorithme	Paramètres
Chiffrement des données	XChaCha20-Poly1305 (AEAD)	nonce aléatoire de 24 octets, généré à chaque opération
Dérivation de clé (mot de passe → clé)	Argon2id	64 Mio de mémoire, 3 itérations, sortie 256 bits, sel aléatoire de 16 octets
Empreinte d'authentification	BLAKE2b (avec séparation de domaine)	domaine `shield:auth:v1`
Partage / messages scellés	X25519 (`crypto_box_seal`)	scellé anonyme vers la clé publique du destinataire
Transmission (héritage)	SLIP-39 (partage de secret de Shamir, à seuil)	seuil M-sur-N configurable
Multisig Sanctuaire	SLIP-39 (découpage) + X25519 (scellement par détenteur)	seuil M-sur-N configurable, à la demande
Stockage serveur de l'empreinte d'auth	scrypt + sel par compte	N=16384, r=8, p=1
Scellé du déverrouillage biométrique	SHA-256 d'une sortie WebAuthn PRF	clé dérivée du matériel d'authentification, jamais transmise

Le chiffrement est **authentifié** (AEAD) : toute altération d'un message chiffré est détectée et provoque un échec de déchiffrement. Le nonce de 24 octets de XChaCha20-Poly1305 est tiré aléatoirement à chaque chiffrement ; sa taille étendue rend toute collision négligeable et évite la classe de failles liée à la réutilisation de nonce.

2.2 Hiérarchie de clés

- Le **mot de passe maître** est transformé en **MasterKey** par Argon2id (jamais transmis au serveur).
- La MasterKey enveloppe une **AccountKey** aléatoire (256 bits) — la racine du compte.
- L'AccountKey enveloppe la **clé privée X25519** de l'utilisateur (pour recevoir des éléments partagés ou un legs).
- Chaque coffre possède une **VaultKey** aléatoire, enveloppée par l'AccountKey.

- Chaque fiche possède sa **propre clé** (ItemKey), enveloppée par la VaultKey.

Ce modèle « une clé par fiche » permet de partager un élément précis sans exposer le reste du coffre, et de révoquer un accès sans tout rechiffrer. Le changement de mot de passe maître ne ré-enveloppe que l'AccountKey : aucune fiche n'est rechiffrée.

2.3 Ce que le serveur voit — et ne voit jamais

Reçu et stocké par le serveur : des enveloppes chiffrées opaques (base64), une empreinte d'authentification (elle-même re-hachée par scrypt avec un sel propre au compte, puis comparée en **temps constant**), les paramètres publics de dérivation (sel, mémoire, itérations), et la clé **publique** de l'utilisateur.

Jamais reçu : le mot de passe maître, la MasterKey, l'AccountKey, les VaultKeys, les ItemKeys, la clé privée, le contenu des fiches en clair. Une fuite intégrale de la base de données ne révèle donc ni les mots de passe, ni le contenu des coffres.

---

3. Les niveaux de sensibilité

Shield organise les secrets en niveaux, chacun avec son propre verrou.

- **Quotidien** — déverrouillé par le mot de passe maître seul. Usage quotidien (identifiants, cartes, codes wifi, abonnements).
- **Documents** — exige le mot de passe maître **et** un second code. La clé de ce niveau combine l'AccountKey et le code : le code seul ne suffit pas, et le mot de passe maître seul non plus. Les codes trivialement faibles (suites, répétitions) sont refusés. Recommandation produit : ce code doit être substantiel pour résister à un attaquant qui aurait déjà le mot de passe maître.
- **Sanctuaire** — exige le mot de passe maître **et** une phrase secrète. Destiné aux secrets les plus critiques (seeds de wallets). La phrase, à haute entropie, place ce niveau hors d'atteinte d'un brute-force même si le mot de passe maître est compromis.

3.1 La feuille de secours

À l'inscription, une feuille de secours unique est générée : 160 bits d'aléa, formatés en groupes lisibles, à imprimer et conserver hors-ligne. Elle enveloppe une seconde fois l'AccountKey, permettant de **récupérer un mot de passe maître oublié**. Son entropie la rend mathématiquement imbruteforçable ; son seul risque est physique (perte ou vol du papier). Elle est régénérable à tout moment (la nouvelle invalide l'ancienne) et n'est jamais stockée en clair côté serveur.

---

4. Transmission / héritage

Le legs ne repose sur aucune confiance dans l'éditeur : il repose sur les mathématiques.

- Une **clé de legs** aléatoire (256 bits) chiffre le coffre transmis (le *bundle*).
- Cette clé est découpée en **fragments SLIP-39** selon un seuil M-sur-N choisi par le titulaire.
- Les fragments sont remis **en main propre** aux porteurs désignés (bénéficiaires, exécuteurs).
- Le coffre chiffré est conservé ; **la clé ne l'est jamais** — elle n'existe que recomposée à partir du seuil de fragments.
- Les héritiers recomposent la clé et ouvrent le coffre via un **décrypteur hors-ligne** (fichier statique autonome), **sans aucune dépendance à nos serveurs** : la récupération fonctionne même si Shield, l'entreprise, n'existe plus.

4.1 Machine à états du déclenchement

Le déclenchement est gouverné par une machine à états automatisée (vérification horaire) :

`ARMED` → `PING_OVERDUE` → `PENDING_RELEASE` → `RELEASED`

- **ARMED** : signal de vie actif. Le titulaire confirme sa présence à intervalle régulier (`livenessIntervalDays`).
- **PING_OVERDUE** : un signal de vie a été manqué ; un email de rappel est envoyé. Un délai de grâce (`graceDays`) s'applique.
- **PENDING_RELEASE** : faute de réponse après le délai de grâce — ou sur demande des porteurs atteignant le seuil — une fenêtre de refus (`refusalWindowDays`) s'ouvre. Le titulaire est notifié et peut **annuler à tout moment**.
- **RELEASED** : à la fin de la fenêtre, le legs devient disponible aux porteurs. L'ouverture exige toujours la réunion des fragments physiques.

Le titulaire garde le contrôle à chaque étape : répondre au signal de vie réarme le système et annule tout déclenchement.

4bis. Multisig Sanctuaire (optionnel)

À la différence de la Transmission (§4), qui prépare une remise post-mortem, le multisig Sanctuaire protège le déverrouillage **du vivant du titulaire**, à la demande. Désactivé par défaut, il s'active uniquement sur le niveau Sanctuaire.

- À l'activation, la **VaultKey du Sanctuaire** est découpée **une seule fois** en fragments SLIP-39 selon un seuil M-sur-N choisi par le titulaire.
- Chaque fragment est **scellé individuellement** (X25519 `crypto_box_seal`) pour la clé publique de son détenteur. Un détenteur sans compte Shield reçoit son fragment scellé symétriquement (sous la VaultKey) en attente ; il est re-scellé pour sa clé publique dès la création de son compte — mécanisme identique à l'invitation du coffre familial (aucune obligation de souscrire à Shield).
- Pour déverrouiller, le titulaire crée une **demande de déverrouillage** bornée dans le temps (préréglages 1h/3h/6h/12h/24h, ou durée personnalisée).
- Chaque détenteur, indépendamment, ouvre son propre fragment avec sa clé privée puis le **re-scelle spécifiquement pour la clé publique du demandeur** — jamais pour le serveur.
- Dès que le seuil de fragments re-scellés est réuni, seul le demandeur, détenteur de la clé privée correspondante, peut les ouvrir et **reconstituer la VaultKey localement**, sur son appareil.

À chaque étape, le serveur ne stocke et ne relaie que des enveloppes chiffrées : il ne voit jamais un fragment en clair, ni la VaultKey reconstituée. Une demande expirée invalide les approbations déjà reçues. Le multisig s'ajoute à la phrase secrète du Sanctuaire — il ne la remplace jamais.

---

5. Coffre familial

Compartiment partagé pour les secrets communs (wifi, abonnements, codes). La VaultKey du coffre familial est enveloppée par l'AccountKey du gérant, puis partagée aux membres via scellé X25519 (chaque membre déchiffre avec sa propre clé privée). Les droits d'édition sont gérés par membre. L'éditeur ne peut jamais lire le contenu.

---

6. Extension de navigateur & déverrouillage rapide

L'extension de navigateur apporte le coffre dans le navigateur (remplissage automatique, générateur, codes 2FA) **sans modifier le modèle zero-knowledge** : tout le chiffrement reste local, et le serveur ne reçoit

toujours que des enveloppes opaques. Le service worker de l'extension conserve la session déverrouillée dans un stockage **éphémère** (`chrome.storage.session`), assorti d'un délai de verrouillage automatique ; les clés ne sont jamais persistées en clair sur le disque.

6.1 Déverrouillage rapide — principe

Pour éviter de retaper le mot de passe maître à chaque ouverture, l'utilisateur peut activer un **déverrouillage rapide** : code PIN et/ou biométrie. Deux propriétés sont garanties par construction :

- Le déverrouillage rapide est **strictement local et par navigateur** : les éléments de scellé vivent dans `chrome.storage.local` de l'appareil et ne transitent jamais par le serveur.
- Il **ne remplace jamais** le mot de passe maître : il ne fait que sceller, sur l'appareil, la clé déjà dérivée. La protection serveur (empreinte d'auth, Argon2id) reste inchangée.

Le déverrouillage rapide est **multi-comptes** : chaque compte dispose de son propre PIN et/ou de sa propre clé biométrique, indexés par adresse e-mail. Choisir un compte n'expose jamais les autres.

6.2 Code PIN

Le PIN scelle, sur l'appareil, un paquet contenant la clé maître du compte, sous une clé dérivée du PIN et d'un sel aléatoire propre au compte. Le paquet scellé et son sel sont stockés localement, par e-mail. Une **limite de tentatives** (cinq essais) protège contre une saisie répétée : au-delà, le déverrouillage rapide de ce compte est effacé et le mot de passe maître redevient nécessaire.

Le PIN est une **commodité sur un appareil que l'utilisateur contrôle**, pas un mur cryptographique : son entropie est faible par nature. Sa sécurité repose sur (a) le fait qu'il ne quitte jamais l'appareil, et (b) la limite de tentatives. Il n'affaiblit en rien la protection serveur, qui demeure gouvernée par le mot de passe maître.

6.3 Biométrie (WebAuthn PRF)

La biométrie s'appuie sur **WebAuthn** et l'extension **PRF**. À l'activation, un *passkey* de plateforme (Touch ID, Windows Hello) est créé avec PRF activée ; la sortie PRF (haute entropie, dérivée du matériel d'authentification) est condensée en une clé de scellé (SHA-256) qui scelle localement la clé maître du compte. Au déverrouillage, une assertion biométrique régénère la même sortie PRF, donc la même clé de scellé, et ouvre le paquet.

Depuis Chrome 122, une extension peut asserter un `rpId` couvert par ses `host_permissions` : la cérémonie biométrique tourne **directement dans le panneau de l'extension**, sans iframe ni fenêtre intermédiaire. La clé biométrique et la clé maître ne quittent jamais l'appareil ; le serveur n'est jamais sollicité. Le zero-knowledge est intégralement préservé.

6.4 Déverrouillage inline sur une page

Sur une page de connexion, lorsque le coffre est verrouillé, l'extension peut afficher une **carte de saisie de PIN isolée**, injectée dans la page via une iframe d'origine extension. Le PIN est traité **à l'abri de la page hôte** : il est transmis au service worker, qui déchiffre ; **aucun secret ne transite par le site visité**, et seul un signal « déverrouillé » ressort. Le déverrouillage inline cible le compte de la dernière session ; le choix d'un autre compte, ou la biométrie, s'effectue depuis le panneau de l'extension.

6.5 Déconnexion totale

Deux gestes distincts coexistent. **Verrouiller** ferme la session ouverte ; le déverrouillage rapide la rouvre. **Se déconnecter** est une **déconnexion totale** : le déverrouillage rapide du compte courant (PIN **et** biométrie) est effacé de l'appareil, de sorte que la prochaine ouverture exige de nouveau le mot de passe maître. La déconnexion ne touche que le compte courant ; les autres comptes conservent leur déverrouillage rapide.

6.6 Remplissage et capture

Le remplissage automatique propose, dans le champ, les seules fiches dont le domaine correspond au site courant. La capture à la soumission d'un formulaire permet de proposer l'enregistrement ou la mise à jour d'un identifiant. Les codes 2FA sont calculés localement par le service worker : le secret TOTP **ne quitte jamais**

l'extension, seul le code à usage unique est copié. Toutes ces opérations se déroulent dans le contexte de l'extension ; la page hôte ne reçoit jamais de secret en clair.

---

7. Modèle de menace

Le point essentiel, et contre-intuitif : **le chiffrement n'est pas le maillon faible.** Les primitives employées sont à l'état de l'art et alignées sur les meilleurs acteurs audités du marché. Les risques réels se situent ailleurs.

- **La force du secret choisi par l'utilisateur.** C'est le seul mur réellement franchissable par force brute (voir §8). Mesure : politique de robustesse à l'inscription (longueur minimale, jauge d'entropie, encouragement d'une phrase), et rejet des secrets triviaux.
- **L'appareil de l'utilisateur.** Un logiciel malveillant (keylogger, vol de presse-papier) peut capter le mot de passe **avant** que le chiffrement ne s'applique. Aucun chiffrement ne protège de ce cas, par construction : la protection porte sur les données stockées et transmises, pas sur la frappe en clair. Mesure : hygiène de l'appareil, déverrouillage rapide pour limiter les saisies, déconnexion totale sur un poste partagé, sensibilisation.
- **Le déverrouillage rapide.** Le PIN et la clé biométrique vivent sur l'appareil. Le PIN, à faible entropie, est protégé par une limite de tentatives et par le fait qu'il ne quitte jamais l'appareil ; la biométrie s'appuie sur une clé haute entropie issue du matériel. Sur un poste que l'on ne contrôle plus, la déconnexion totale efface ces scellés.
- **La livraison du code.** Comme tout coffre opérant dans un navigateur, le code qui chiffre est servi par le serveur ; un code remplacé pourrait capter le secret. Mesure **en place** : Content-Security-Policy stricte (`default-src 'self'`, `connect-src 'self'`) qui empêche tout script d'exfiltrer des données vers un domaine tiers, et publication en open source du module de chiffrement client et du décrypteur (vérifiabilité). L'extension de navigateur, distribuée et versionnée, offre une surface de livraison contrôlée pour l'usage quotidien.
- **Les bugs d'implémentation.** Vérifiés par revue de code, par l'ouverture du cœur cryptographique, et, à terme, par audit tiers (voir §10).
- **Le 2FA rangé dans le coffre.** Regrouper mot de passe et code 2FA dans Shield protège pleinement contre les fuites et le phishing côté sites — mais fait du coffre l'unique gardien des deux facteurs. Choix assumé, documenté et signalé dans le produit (avertissement à la première fiche 2FA) : pour les comptes les plus critiques (email principal, banque), Shield recommande un second facteur séparé, application dédiée ou clé matérielle.

---

8. Coût d'une attaque par force brute

Hypothèses : attaquant très bien doté (~1 000 GPU haut de gamme). Argon2id à 64 Mio / 3 itérations limite l'attaque à quelques centaines de milliers d'essais par seconde, chaque essai étant coûteux en mémoire. Le chiffrement rend chaque tentative chère ; c'est l'entropie du secret qui fixe le nombre de tentatives.

Secret choisi	Entropie	Temps de cassage estimé
Mot de passe faible (8 lettres)	~38 bits	quelques jours
Phrase de 4 mots	~52 bits	plusieurs siècles
Phrase de 5 mots	~64 bits	hors échelle humaine
Feuille de secours	160 bits	mathématiquement hors d'atteinte

Par niveau : au **Quotidien**, le mur est le mot de passe maître. Au **Documents**, un second secret s'ajoute — un attaquant sans le mot de passe maître doit casser les deux ; avec le mot de passe maître, la sécurité dépend

de la qualité du code. Au **Sanctuaire**, la phrase additionnelle place le niveau hors d'atteinte quoi qu'il arrive.

Le cas du déverrouillage rapide. Le PIN n'entre pas dans ce tableau : ce n'est pas un mur cryptographique mais une commodité locale, gouvernée par une limite de tentatives sur l'appareil. Il ne change rien au coût d'attaque côté serveur, qui reste fixé par le mot de passe maître.

Économie de l'attaque. Pour un compte individuel, le coût d'un brute-force dépasse de loin la valeur du contenu : l'attaque par force n'est jamais le scénario rationnel. Un attaquant ira vers le moins cher — hameçonnage, logiciel malveillant, code piégé — d'où la priorité donnée aux défenses du §7 plutôt qu'au chiffrement, déjà imprenable.

---

9. Positionnement face au marché

Primitives vérifiées chez les concurrents de référence :

Capacité	Shield	Bitwarden	Proton Pass	1Password	Vault12	Inheriti
Chiffrement authentifié moderne	XChaCha20	✓	AES-GCM	AES-GCM	✓	AES-256
KDF Argon2id	✓	option	✓	— (PBKDF2)	n/a	n/a
Une clé par fiche	✓	—	✓	—	n/a	n/a
Zero-knowledge	✓	✓	✓	✓	✓	✓
Niveaux de sensibilité gradués	✓	—	—	—	—	—
Coffre familial partagé	✓	✓	✓	✓	—	—
Seeds crypto en type natif	✓	—	—	—	✓	✓
Transmission par seuil (M-sur-N)	✓	—	—	—	✓	✓
Récupération héritiers hors-ligne	✓	—	—	—	✓	✓
Déverrouillage multisig à la demande (M-sur-N, du vivant du titulaire)	✓	—	—	—	—	—
Déverrouillage biométrique sans serveur (WebAuthn PRF)	✓	✓	✓	✓	n/a	n/a
Tout-en-un (mdp + docs + seeds + héritage)	✓	—	—	—	—	—
Audit tiers indépendant	à venir	✓	✓	✓	~	✓

Shield emploie les mêmes briques modernes que les meilleurs (XChaCha20-Poly1305 comme NordPass, Argon2id et clé-par-fiche comme Proton Pass), et la fondation à seuil des spécialistes de l'héritage (Vault12, Inheriti) — mais intégrée à un coffre complet plutôt que réservée aux seeds.

---

10. Mesures en place et feuille de route

Déjà en place : chiffrement libsodium état de l'art ; Argon2id 64 Mio / 3 ; zero-knowledge vérifié ; empreinte d'auth re-hachée scrypt + comparaison à temps constant ; Content-Security-Policy stricte (anti-exfiltration) ; limitation des tentatives de connexion ; protection contre l'énumération des comptes ; politique de robustesse du mot de passe et rejet des secrets triviaux à l'inscription ; déverrouillage rapide local (PIN limité en tentatives, biométrie WebAuthn PRF) ; déconnexion totale ; consentement horodaté à l'inscription ; suppression de compte conforme RGPD ; décrypteur hors-ligne autonome ; **module de chiffrement client et décrypteur publiés en open source** (licence MIT) ; sauvegarde indépendante chiffrée, hors de l'hébergeur

principal (§ Continuité et sauvegarde).

Crédibilité (progressive) : programme de divulgation responsable ; audit de sécurité tiers indépendant lorsque la base de membres le justifiera.

Liberty Club LLC — Sheridan, Wyoming, USA. contact@libertyclub.finance

Annexe — mises à jour du 5 juillet 2026

Remplissage des cartes de paiement (extension v0.1.25 → v0.1.28)

Le remplissage d'une carte n'est jamais déclenché par la page : le choix d'une carte dans le menu inline ouvre une **fenêtre de confirmation appartenant à l'extension** (origine `chrome-extension://`, hors du DOM de la page), que le site ne peut ni afficher, ni recouvrir, ni imiter — mitigation du clickjacking et des overlays trompeurs. La demande en attente vit exclusivement dans `chrome.storage.session` (mémoire volatile de l'extension, TTL 120 secondes, purgée à la fermeture du navigateur) ; les données de carte ne transitent jamais par les serveurs Shield. Après confirmation explicite, le service worker diffuse le remplissage à **tous les cadres de l'onglet** : les intégrations de paiement isolant numéro et cryptogramme dans des iframes distinctes (Stripe, Adyen) sont couvertes, chaque cadre ne remplissant que ses propres champs. Les champs mois/année séparés (texte et menus déroulants) sont gérés. Le remplissage est programmatique : **aucune frappe clavier n'est émise**, le rendant invisible aux enregistreurs de frappe. Le menu n'expose que le titre et les quatre derniers chiffres. Une carte saisie manuellement au paiement peut être enregistrée au coffre depuis le même menu (chiffrement standard des items, déduplication par numéro) ; lorsque numéro et cryptogramme résident dans des cadres distincts, la capture est limitée au cadre courant.

Pavés de saisie mélangés (anti-keylogger)

Le déverrouillage rapide de l'extension et le code du niveau Documents (application web) proposent un pavé 0-9 dont l'ordre est **mélangé à chaque affichage**. La saisie s'effectue intégralement à la souris : aucun événement clavier n'est produit, et la permutation aléatoire rend la position des clics inexploitable par un observateur des coordonnées. Le clavier physique reste accepté. Modèle de menace assumé : un enregistreur de frappe au niveau du système d'exploitation ne peut capter que ce qui est tapé — la stratégie de Shield est de réduire la frappe des secrets à sa plus simple expression : le mot de passe maître n'est saisi qu'à la première connexion d'un appareil, les déverrouillages suivants passant par la biométrie (WebAuthn/Touch ID, zéro frappe) ou le pavé mélangé. Aucun logiciel ne peut détecter ni neutraliser un enregistreur de frappe depuis un navigateur ; Shield ne le prétend pas.

Verrouillage automatique par inactivité (web)

Réglage par appareil (1 à 60 minutes, ou jamais ; défaut 15) : sans interaction, la session est verrouillée localement — les clés quittent la mémoire, sans révocation du jeton, le déverrouillage rapide restant disponible. La valeur est lue à chaque cycle : un changement s'applique immédiatement.

Compléments du 5 juillet 2026 — extension v0.1.31 → v0.1.38

Biométrie sur la carte de déverrouillage inline. La carte PIN affichée dans la page propose aussi la biométrie : l'assertion WebAuthn (PRF) s'exécute dans l'iframe d'origine extension, avec délégation explicite `publickey-credentials-get` posée à la création du cadre. La clé maître descellée rejoint le service worker par le canal interne de l'extension (`chrome.runtime`) — jamais par `postMessage` — et n'est donc jamais exposée à la page hôte.

Sélecteur de carte hors page pour les cadres de paiement. Lorsque le champ carte vit dans une iframe (Stripe, Adyen), tout menu injecté serait rogné par le cadre. L'icône ouvre alors une **fenêtre appartenant à l'extension**, centrée à l'écran, listant les cartes (titre et quatre derniers chiffres seulement) : choisir vaut confirmation, la demande expire après 120 secondes, et le remplissage est diffusé à tous les cadres de l'onglet. Mêmes garanties que la fenêtre de confirmation : aucun site ne peut l'afficher, la recouvrir ni l'imiter.

Remplissage restreint aux champs visibles. Le remplissage de carte ignore tout champ sans dimensions rendues : un formulaire invisible superposé ou dissimulé dans la page ne peut plus capter silencieusement les valeurs destinées au formulaire visible.

Distribution versionnée et mise à jour volontaire. L'extension est publiée en archive versionnée accompagnée d'un manifeste de version public. Le panneau compare sa version au manifeste (au plus une requête par six heures) et affiche un bandeau de mise à jour. Hors magasin officiel, la mise à jour reste un geste volontaire de l'utilisateur — re-télécharger, remplacer, recharger : aucune mise à jour silencieuse n'est possible, ni souhaitée.

Continuité et sauvegarde (7 juillet 2026)

Le zero-knowledge protège la confidentialité des données ; il ne protège pas, en soi, contre leur disparition (panne, erreur opérationnelle, incident chez un hébergeur). Shield traite ces deux risques séparément.

Sauvegarde native de l'hébergeur. L'infrastructure principale bénéficie d'une sauvegarde continue gérée par l'hébergeur (archivage en continu + sauvegardes complètes et incrémentales régulières), permettant une restauration à un instant précis en cas d'incident.

Sauvegarde indépendante. En complément, une copie chiffrée de l'ensemble des données est exportée quotidiennement vers un **second hébergeur, distinct et sans lien opérationnel avec le premier** — de façon à survivre à un incident qui toucherait l'hébergeur principal lui-même (compte, plateforme, région), pas seulement une panne matérielle isolée. Cette copie est chiffrée une seconde fois côté serveur (AES-256-GCM, clé dédiée) avant l'envoi : le second hébergeur ne reçoit, comme le premier, que des données opaques.

Choix de transparence. À l'image des acteurs de référence du secteur (qui documentent publiquement leur infrastructure d'hébergement et leurs pratiques de continuité), Shield documente l'existence et le principe de cette double sauvegarde. Le nom précis du second hébergeur n'est en revanche pas publié : cette information n'apporte aucune garantie supplémentaire à l'utilisateur, et sa divulgation ne ferait qu'élargir la surface exploitable par un attaquant ciblant spécifiquement ce second maillon (ingénierie sociale, usurpation de support). Ce choix est cohérent avec le principe du §7.4 : la protection réelle ne repose jamais sur le secret d'une architecture, mais elle n'a pas non plus à en faire une carte au trésor.