VEAULT

Security Whitepaper

Architecture of a Zero-Knowledge Digital Legacy Vault

August 2026 · Version 1.0

Abstract

Veault is a zero-knowledge vault for digital estate information. Customers record accounts, financial details, documents, instructions and personal messages; all such content is encrypted in the customer's browser under a key derived from a password held only by the customer, before transmission to Veault's servers. Veault stores ciphertext for which it holds no key.

The system additionally implements a controlled release mechanism, Veault Access, through which a vault becomes accessible to a person designated in advance by its owner, under conditions the owner defines and can revoke. The mechanism operates without introducing any capability by which Veault could decrypt vault contents at its own initiative.

This document describes the cryptographic design, storage model, authentication model, infrastructure, operational controls and release mechanism; sets out the principal design alternatives considered and the reasons for their rejection; and states the assumptions on which the security of the system rests.

contact@veault.com · veault.com

Contents

1. Terminology

Vault The complete set of a customer's encrypted content, comprising structured sections, uploaded documents and generated archives.

Vault password A password chosen by the customer, from which the encryption key protecting vault contents is derived. Not transmitted to or stored by Veault.

Master key The key under which vault contents are encrypted, itself protected by the key derived from the vault password.

Trusted contact A person designated by the customer as a permitted recipient of the vault under defined conditions.

Access document A printable document generated in the customer's browser, containing a 256-bit secret associated with a specific trusted contact, transferred to that contact by physical means.

Release The process by which a verified trusted contact obtains a decrypted copy of the vault following satisfaction of all conditions set by the customer.

2. Threat model

The architecture is designed to hold under the following conditions.

2.1 Full compromise of Veault's infrastructure. An adversary in possession of Veault's databases and file storage obtains ciphertext and account records. Vault contents remain encrypted under keys that were never present on Veault's systems and cannot be derived from anything the adversary has obtained. Recovery of vault contents would require an offline attack against each individual customer's password, subject to the derivation cost described in section 5.2.

2.2 Malicious or compromised personnel. No decryption capability exists within the administrative surface of the application. Privileged access confers no ability to read vault contents, as the necessary key material is not held.

2.3 Legal or regulatory compulsion. Veault can disclose only what it holds. Vault contents held by Veault are not readable by Veault and cannot be rendered readable by it in response to a lawful order. Compulsion directed at Veault therefore yields ciphertext.

2.4 Compromise of a customer's application credentials. Authentication to the application and decryption of a vault are governed by separate secrets, of which only the former is transmitted to Veault. An adversary authenticated as the customer obtains access to encrypted data.

2.5 Compromise of the transport layer or of an intermediary. Vault content is encrypted prior to transmission. Interception of traffic between client and server yields ciphertext irrespective of the state of transport security.

2.6 Discontinuation of the service. Vault contents can be exported in a form that decrypts offline, independently of Veault's infrastructure and of the continued existence of the company.

3. Problem statement

A substantial and growing proportion of estate value and estate administration is now digital: banking and investment platforms, cryptocurrency holdings, insurance portals, business systems, subscription services, and the second-factor devices and applications that protect them. This inventory is typically undocumented and known only to the account holder.

On death, the executor or surviving family faces a discovery problem rather than an access problem alone. Assets go unclaimed because their existence is unknown; services continue to charge; account recovery procedures cannot be initiated because nobody knows which accounts exist. Where the estate includes self-custodied cryptocurrency, the loss is absolute and immediate: no institution exists from which recovery can be sought.

Existing practice addresses this poorly. Written records are legible to anyone who obtains them and become inaccurate within months. Password managers secure credentials effectively but assume a living user and treat succession as a peripheral feature, typically requiring the recipient to hold an account with the same provider. Estate professionals hold legal authority without technical access; a testamentary instrument can direct who inherits an account but cannot supply the credential, and providers increasingly decline to do so. Documents stored in general-purpose cloud services are readable by the provider and by any party that compromises it.

Veault addresses the specific interval in which these approaches fail: the transfer of access from an owner who is no longer present to a recipient who must be verified.

4. The two-secret model

Veault separates authentication to the application from decryption of vault contents. The distinction is fundamental to the remainder of this document.

	Application authentication	**Vault encryption**
Protects	Access to the Veault application	Vault contents
Mechanism	Single-use code issued to the registered email address	Password chosen by the customer
Retained by Veault	No — single-use and short-lived	No — never transmitted
Recovery path	Customer's email account	None

Successful authentication grants access to encrypted data only. Decryption requires the vault password, which exists in the customer's memory and in the browser's working memory for the duration of an unlocked session, and is discarded when that session ends.

The vault password cannot be reset, recovered or reissued by any party, including Veault. This is a necessary consequence of the model rather than an omission: any facility permitting a provider to restore access to a locked-out customer constitutes a facility permitting that provider to grant access to a third party, whether in error, under deception, or under compulsion. Veault does not implement one, and the guarantees in section 2 depend on its absence.

5. Cryptographic design

Vault cryptography is implemented in a single isolated module without framework dependencies, so that its parameters are resistant to inadvertent modification and the implementation can be reused unchanged across client platforms.

Parameter	Value
Symmetric cipher	AES-256-GCM
Key derivation function	PBKDF2-HMAC-SHA-256
Iteration count	1,000,000
Salt	Unique per encryption operation, cryptographically random
Nonce	Unique per encryption operation, cryptographically random
Implementation	Web Crypto API (browser-native)

5.1 Cipher selection. AES-256-GCM provides authenticated encryption with associated data. Integrity is verified as an intrinsic part of decryption, so modified ciphertext fails to decrypt rather than yielding altered plaintext. Unauthenticated modes such as CBC provide confidentiality without this property and permit classes of attack in which an adversary manipulates ciphertext to produce controlled changes in the decrypted output; they are not used.

5.2 Key derivation. Keys are derived from the customer's vault password using PBKDF2-HMAC-SHA-256 at one million iterations. The OWASP Password Storage Cheat Sheet currently recommends 600,000 iterations for this construction.

The parameter governs the cost of an offline guessing attack against a customer's password in the event that ciphertext is obtained. The cost is incurred once per unlock by the customer, where it is imperceptible, and once per candidate password by an adversary, where it is multiplied across the entire search space.

Memory-hard derivation functions such as Argon2id and scrypt offer superior resistance to attack by specialised hardware and are preferred where available. They are not exposed by the Web Crypto API, and adopting them in a browser context requires either a WebAssembly implementation or a JavaScript implementation, each of which introduces a dependency into the most security-critical path in the system and forfeits the assurance of a browser-maintained, natively audited implementation. Veault has elected to remain on PBKDF2 with a substantially elevated iteration count. This position will be revisited if and when memory-hard derivation is exposed natively by the platform.

5.3 Randomness. Salts and nonces are generated for every encryption operation using the browser's cryptographically secure pseudorandom number generator. No general-purpose random function is used in any cryptographic path. Nonce reuse under a given key, which would compromise the security of GCM, is precluded by generating a fresh nonce for each operation rather than by derivation or counter.

5.4 Implementation. All primitives are provided by the browser's native Web Crypto API — the same implementations relied upon for TLS. Veault implements no cryptographic primitives of its own.

5.5 Format versioning. Encrypted payloads carry a format version identifier, permitting the scheme to be strengthened without rendering previously encrypted vaults unreadable. Backward compatibility is maintained against fixed test vectors and against archived vault fixtures predating format changes, so that a vault written under an earlier format remains decryptable following modification of the current one.

6. Data model and storage

Category	Form in which it is stored
Vault content, uploaded documents, generated archive	Ciphertext only; encrypted client-side prior to transmission
Vault password	Not stored in any form
Account record: email address, name, subscription status	Plaintext

6.1 Client-side encryption boundary. Vault content is encrypted before transmission. No plaintext vault content is received, processed or logged by Veault's servers in the ordinary operation of the service.

6.2 Granularity. Vault content is encrypted and stored per section rather than as a single monolithic payload, so that a customer editing one part of their vault does not require decryption and re-encryption of the whole. Each section retains one prior encrypted version, permitting a single step of recovery from an erroneous edit. Veault does not retain a full revision history of vault content; a customer who has overwritten information twice cannot recover the original, and this is intentional, as an accumulating history of drafts would enlarge the body of ciphertext held without corresponding benefit to the customer.

6.3 Non-content data. A count of completed sections is stored unencrypted, so that progress can be displayed without decryption. It conveys the number of sections a customer has completed and nothing of their content.

6.4 File metadata. Uploaded files are stored as opaque encrypted objects. The upload path records all objects identically irrespective of their contents, so filenames, media types and extensions are not present in the storage layer; they are contained within the encrypted payload. An adversary in possession of Veault's file storage cannot determine the nature of any stored object, nor distinguish a testamentary document from an identity document or a photograph.

6.5 Consequence. Compromise of Veault's storage yields ciphertext and a set of account records. It does not yield vault contents, and it does not yield metadata from which the character of vault contents could be inferred.

7. Authentication

Veault does not issue or store application passwords. Authentication is performed by means of a single-use code transmitted to the customer's registered email address, generated from a cryptographically secure source, valid for a single use and for a limited period.

This eliminates several classes of risk rather than mitigating them. No standing credential exists between sessions; there is consequently no credential subject to reuse across services, no credential to obtain by phishing, and no password database whose compromise would be material. Authenticated sessions carry signed tokens validated at the network edge prior to the execution of application logic.

The model transfers a degree of reliance to the customer's email account, and the extent of that reliance should be stated precisely. An adversary in control of a customer's email account can authenticate to the application. Such an adversary obtains access to encrypted data only; the vault password is not transmitted by email, is not recoverable through the email account, and is not derivable from anything the adversary thereby obtains. Customers are nonetheless advised to protect the registered email account with strong second-factor authentication, as it governs access to the application and to release-related correspondence.

8. Infrastructure and data protection

8.1 Jurisdiction. Veault operates on cloud infrastructure located within the European Union. Vault data is not transferred outside the EU, and no processing of vault content occurs in a third country.

8.2 Architecture. The platform is serverless throughout. No persistent application servers, remote administration interfaces or shared application hosts are maintained, eliminating the corresponding categories of intrusion rather than defending against them. Compute is instantiated per request and does not persist state between invocations.

8.3 Request handling. Requests to protected endpoints are authenticated at the network edge prior to reaching application logic. Vault storage is private, with no public access path to any stored object. File access is granted exclusively through short-lived signed references scoped to the requesting customer's own data.

8.4 Supporting controls. Application secrets are held in a managed secrets service and are not present in source code. Payment provider callbacks are cryptographically verified before being acted upon. The platform is monitored continuously, with automated alerting on error conditions and anomalous activity.

8.5 Durability. Databases operate with continuous point-in-time recovery and scheduled automated snapshots. Backups contain the same ciphertext held in production and are subject to the same absence of key material; a compromise of backup storage is equivalent in consequence to a compromise of production storage.

8.6 Deletion. Deletion of a vault initiates a thirty-day recovery period, protecting against both accidental deletion by the customer and malicious deletion by a party who has obtained access to the customer's session. On expiry the data is destroyed irreversibly. Veault does not retain vault content following deletion, and the absence of key material means that no residual copy would be readable in any event.

8.7 Data minimisation. Personal data held in readable form is limited to that required to operate a subscription service: identity of the account holder, contact address, subscription status, and correspondence. The substantive content a customer entrusts to Veault — the material that constitutes the actual sensitivity of a digital estate record — is held only as ciphertext and is outside the set of data Veault is capable of disclosing, whether to a third party, to an adversary, or to a supervisory authority.

9. Operational access

9.1 Separation. Administrative accounts are maintained in an account system separate from that of customers, with independent authentication, and cannot be self-registered. A customer session cannot reach an administrative endpoint; the separation is structural rather than a permission evaluated within the application.

9.2 Limits of administrative capability. Administrative access does not confer access to vault contents. No decryption capability exists within the administrative surface in any form, whether restricted, audited or reserved for emergency use. Administrative functions extend to billing, subscription and account state. No member of Veault personnel can decrypt a vault, as the required key material is not held by Veault and cannot be obtained by it.

The distinction matters in evaluating any provider. A commitment that personnel will not access customer data is a statement of policy, contingent on enforcement, on the conduct of individuals, and on the provider's resistance to external pressure. The property described here is a statement of capability: the operation cannot be performed, and no assurance regarding personnel conduct is required to support it.

9.3 Accountability. Administrative actions are recorded in an audit log against the identity of the acting administrator and the time of the action.

10. Veault Access: controlled release

Veault Access is the mechanism by which a vault becomes accessible to a designated recipient following the death of its owner. Its design objective is to make release possible without introducing any capability by which Veault could initiate release on its own authority.

10.1 Establishment. The customer designates one or more trusted contacts. For each contact, the customer's browser generates a 256-bit random secret and renders it into a printable access document containing the secret in human-readable form and as a QR code. The document is transferred to the contact, or retained by the customer for later delivery, by physical means.

Veault receives a cryptographic hash of the secret. The secret itself is not transmitted to or held by Veault. The hash permits subsequent verification that a party presenting itself in a release request holds the genuine document.

The vault's master key is then encrypted a second time, in the browser, under a key derived from the contact's secret. Veault stores this wrapped key. It cannot be unwrapped without the secret, which Veault has never possessed. Where multiple contacts are designated, each receives a distinct secret and a distinct wrapped copy; possession of one confers nothing in respect of another.

10.2 Release conditions. The customer specifies a release policy independently for each trusted contact:

- **Verified review** — release is conditional on submission of a death certificate and identity document, subject to verification.
- **Waiting period** — release occurs after a period specified by the customer, during which the customer is notified of the request and may cancel it.

The choice is a deliberate allocation of the customer's own risk. Verified review introduces human verification at the cost of delay; a waiting period grants release without documentary proof but preserves the customer's ability to intervene throughout. Both terminate in the same requirement: possession of the access document.

10.3 Release sequence.

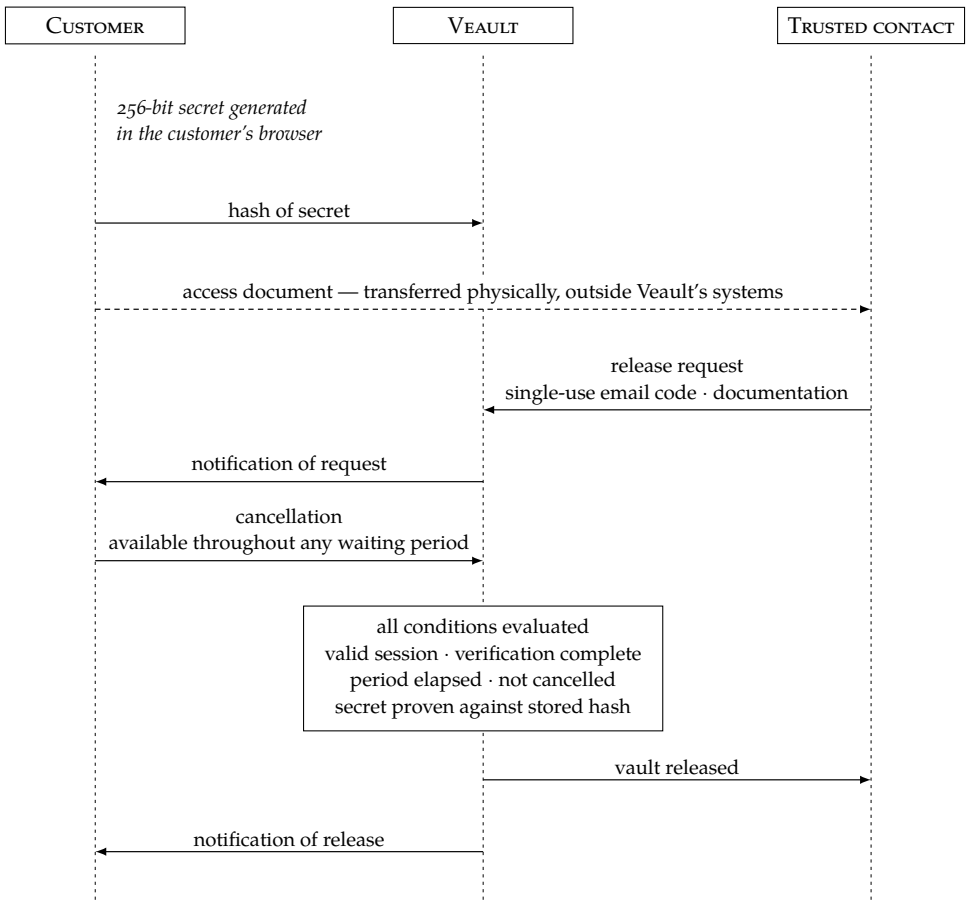

Figure 1. Release sequence. The access document is transferred outside Veault's systems; Veault holds only the hash of the secret and a copy of the master key wrapped under it. Release requires the secret to be presented by the contact.

A contact initiating a request must demonstrate control of their registered email address by means of a single-use code and operate within a time-limited session. Under the verified review policy, the contact must additionally submit a death certificate and identity document for review.

The customer is notified by email upon initiation of a request. Where a waiting period applies, the customer may cancel the request for its duration. A request initiated in error, or in bad faith, against a living customer is therefore visible to that customer and reversible by them.

Release occurs only where all conditions hold simultaneously: a valid session, completion of the applicable verification, expiry of any waiting period, absence of cancellation, and verification of the presented secret against the stored hash by constant-time comparison. Constant-time comparison is used to prevent an adversary from inferring the correct value incrementally from variations in response timing.

On satisfaction of these conditions, the wrapped master key is unwrapped

in memory using the secret supplied by the contact, used to assemble the vault archive, and discarded. No key material or plaintext is persisted. The scope of this operation is the single release for which all conditions were satisfied; it is not a capability available to Veault absent the secret, which the contact supplies and Veault does not hold.

The customer is notified upon completion of a release. All requests, cancellations and completions are recorded in a dedicated event log.

10.4 Absence of inactivity-based release. Veault does not implement proof-of-life monitoring and does not release vaults in response to account inactivity. Release requires initiation by a human party.

Inactivity is a weak proxy for death. It is equally consistent with hospitalisation, travel, device loss, illness, or infrequent use of a product that is by nature consulted rarely. Any system releasing vaults on the basis of silence will, across a sufficient population and interval, disclose the contents of a living customer's vault to a third party. The error is silent at the point it occurs, is not detectable by the customer in advance, and cannot be reversed once the contents have been disclosed.

The design accepts a modest additional burden on the recipient — a request must be made, identity demonstrated, and a physical object produced — in exchange for the elimination of that failure mode.

11. Design alternatives considered

Several conventional approaches were evaluated and rejected. Each would have simplified the system materially.

11.1 Server-side encryption. Encrypting vault content on receipt, under keys held by the provider, is the prevailing model in cloud storage and permits password recovery, server-side search and administrative support of a kind not available under the present design. It requires that the provider hold the means to decrypt customer content, and so fails conditions 2.1 through 2.3 of the threat model in their entirety.

11.2 Key escrow with recovery. Retaining a recovery key, held separately or under split control, would allow a customer who forgets their vault password to regain access. The recovery capability is not selective: a mechanism sufficient to restore access to a verified customer is sufficient to grant access to a party who has successfully presented as one, and constitutes an asset that can be compelled, stolen or misused. The permanence of password loss is accepted as the cost of its absence.

11.3 Provider-mediated release without a customer-held secret. Releasing a vault upon Veault's own verification of a death certificate would remove the requirement for the customer to distribute a physical document. It would also require Veault to hold the means of decryption, and would place the decision to open a vault within Veault's discretion — a decision susceptible to forged documentation, to social engineering, and to legal compulsion. The customer-held secret exists so that Veault's satisfaction that a customer has died is a necessary condition of release under one policy, and never a sufficient one.

11.4 Inactivity-triggered release. Addressed at section 10.4.

11.5 Recipient accounts holding key material. Distributing key material to a trusted contact's Veault account, rather than to a physical document, would remove the risk of the document being lost. It would introduce a requirement that the recipient maintain an account, and hold credentials over a period likely to be measured in decades, during which the account may be abandoned, compromised or deleted. A printed document held with a will, deed or other estate paperwork is a custody model the intended recipients already understand and already practise.

12. Independence from the service

A finalised vault may be exported as a single self-contained file comprising the encrypted vault data and the code required to decrypt it.

The file opens in any browser, on any machine, without network connectivity, without a Veault account, and without Veault continuing to operate. Entry of the vault password decrypts the contents locally. The decryption performed is the same operation, using the same primitives and the same parameters, as that performed within the application; the exported file is not a reduced or degraded copy.

This addresses a structural risk particular to this product category. The interval between purchase and use may extend to decades, over which the continued operation of any given company cannot reasonably be assumed. A design in which the data is meaningless without the service converts an estate instrument into a dependency on corporate longevity. Under the present design, continuity of the service is a convenience rather than a precondition of access.

13. Assumptions and scope

The guarantees described in this document rest on two assumptions, both inherent to the model rather than deficiencies within it.

13.1 Integrity of the customer's device. Encryption and decryption occur in the customer's browser. A device compromised by malicious software — for example a keylogger or a hostile browser extension — lies within the trust boundary, and no client-side encryption scheme can exclude it. This assumption is common to all end-to-end encrypted systems. Its corollary is that vault contents are exposed only at the endpoint at which the customer is already present, and at no point in transit or at rest within Veault's infrastructure.

13.2 Custody of the access document. A trusted contact's secret is not retained by Veault and cannot be reconstructed by Veault, which is the property that makes the release mechanism trustworthy. It follows that the access document must be preserved by its holder. The customer may reissue a document and invalidate its predecessor at any time during their lifetime. A secret that Veault could regenerate on the holder's behalf would be a secret Veault could regenerate on anyone's behalf.

Veault · Sandnes, Rogaland, Norway · contact@veault.com · veault.com

This document describes Veault's architecture as implemented at the date of publication and is revised as the system develops. Suspected vulnerabilities should be reported directly to Veault prior to public disclosure.

Version	Date	Notes
1.0	August 2026	Initial publication