ZOMBIE DELETE

Cryptographically Verifiable Deletion Receipts

White Paper | March 2026

Glen Cameron

Dr. Stef Savanah

Delete It. Prove It. Anywhere.

Independent cryptographic proof that data has actually been deleted for regulators, auditors, and data subjects.

Featuring zero-knowledge proof of absence:
mathematical certainty a record does not exist, without revealing anything about the data, the dataset, or the systems it touched.

TogetherAlone

Table of Content

1. The Problem: Data That Refuses to Die

Every organisation that stores personal data must be able to delete it on request. Every regulator is asking the same question: prove it. We are not aware of any existing tool that produces independent cryptographic proof that data has been deleted. The compliance industry relies on workflow logs, screenshots, and self-attestation. That gap is the problem Zombie Delete addresses. ICP Delete is production-live in beta on Internet Computer mainnet today. ZKP-Delete, the zero-knowledge variant for the highest- assurance use cases, follows in 2026.

All components described in this paper are either currently deployed on ICP mainnet or within a six-month engineering horizon using existing, proven cryptographic primitives. No component requires novel cryptographic research.

> We are not aware of any existing tool that produces independent cryptographic proof of data deletion, verifiable by any party without trusting the issuer. The compliance industry relies on workflow logs, screenshots, and self-attestation. Zombie Delete produces mathematical evidence.

1.1 The Compliance Gap

When an enterprise receives a Data Subject Access Request under GDPR Article 17, it deletes the record and logs the action. The log says deletion happened. It does not prove deletion happened. An auditor, a regulator, or a court has no way to independently verify the data is gone. They must trust the organisation's own reporting.

Cumulative GDPR-type fines have surpassed €7 billion. Penalties reach 4% of global annual turnover in the EU, with comparable exposure under South Korean, Indian, and US state frameworks. The average cost of a data breach involving shadow data runs 16% above the already-elevated average. Regulators are moving from trust to accountability. The question is not whether organisations delete data - it is whether they can demonstrate it.

This problem is not specific to blockchain. The vast majority of regulated personal data sits in standard enterprise databases: SQL, NoSQL, cloud-hosted, on-premise. Every organisation that processes personal data under GDPR, CCPA, HIPAA, or equivalent frameworks faces the same gap between deleting a record and proving it was deleted. Zombie Delete addresses this gap across all data environments, from enterprise databases to mutable-state blockchains to cross-chain architectures.

1.2 The Blockchain Tension

Blockchain adoption in regulated industries runs into a practical problem: data governance frameworks require the ability to delete or suppress personal data, while

many blockchain designs use append-only or tamper-resistant ledger structures. The EDPB Guidelines 02/2025[1] are explicit that technical design choices do not create GDPR exemptions.

The standard responses each carry limitations:

- **Crypto-shredding** (encrypt on-chain, destroy the key) makes data inaccessible but does not prove absence. Zombie Delete's position is distinct: it produces evidence that data is absent from the current active state - a materially different and stronger claim than inaccessibility.
- Where crypto-shredding is the only option available Zombie Delete provides proof of the shredding event.
- **Off-chain storage** with on-chain hash references avoids the deletion problem by keeping personal data off the immutable ledger. This is architecturally sound, but it means any project that has already committed personal data to an append-only chain - identity systems, KYC/AML, healthcare records - requires a different remedy.
- **Mutable-state chains** such as the Internet Computer Protocol offer a genuine alternative: canister state can be tombstoned or deleted, while the consensus layer issues immutable cryptographic attestations of what the state is at each moment. This architectural distinction is why ICP was selected as the platform for the first Zombie Delete implementation.

1.3 A Note on AI Training Data

The EU AI Act (enforcement August 2026) introduces training data governance requirements for high-risk AI systems. This is likely to increase demand for auditable remediation and, over time, evidence that problematic training data has been removed or neutralised. This raises a distinct technical question: to what extent do learned statistical patterns in model weights retain identifiable information about specific individuals after source files are deleted? This is genuinely contested, both legally and technically.

The current Zombie Delete suite addresses provable deletion of structured data records. Verification of AI model unlearning requires separate cryptographic techniques[4] and is the subject of a forthcoming white paper. We flag the regulatory context here because it defines a significant part of the forward market.

2. What Zombie Delete Is

Zombie Delete is a family of software configurations that produce Cryptographically Verifiable Deletion Receipts (CVDRs). A CVDR is the common output across the entire

suite. The Zombie Delete product family uses internal configuration codes (MKTd, ZKPd, ICPDL, ICPDT) throughout this paper. The corresponding product names are: ICP Delete for all ICP-native configurations, Chain Delete for standalone non-ICP chain deployments, Zombie Delete Enterprise for standard database environments, and ZKP-Delete for the zero-knowledge proof variant - the thing that makes it possible for a data subject, auditor, or regulator to verify independently that a specific deletion event occurred.

ICP is the reference implementation platform, not a dependency. ICP's architecture - mutable canister state with BLS subnet certification - provides the cleanest environment for building and demonstrating the CVDR mechanism. However, the Zombie Delete architecture is chain-agnostic. Zombie Delete Enterprise operates entirely in standard database environments with no blockchain dependency. Chain Fusion extends verification to any chain with a public RPC endpoint. An enterprise using Ethereum, Solana, Hedera, or any supported chain never touches ICP directly. The enterprise interacts with the Zombie Delete API; the ICP canister operates as an invisible verification engine behind that API.

No blockchain knowledge is required to integrate Zombie Delete Enterprise. A typical REST API integration takes one developer one afternoon. Library-level integrations for ICP-native applications require 50–100 lines of Rust.

A deletion event is a state transition: before deletion, a record exists; after deletion, it does not. Zombie Delete works by generating cryptographic commitments to both states through attested code, binding them into a receipt that can be verified by any party with access to the published procedures, in whichever form required (e.g. Github) - without accessing the underlying database, without trusting the enterprise's reporting, and without any ongoing relationship with Together Alone Ventures.

The term 'Cryptographically Verifiable Deletion Receipt' (CVDR) is used throughout this paper in preference to 'proof,' which is sometimes used loosely in this space. A CVDR is a verifiable receipt: it provides strong, independently checkable evidence of a state transition. The depth of that evidence varies by configuration, from Merkle-commitment receipts (MKTd) to full zero-knowledge proofs (ZKPd). Both are CVDRs; the distinction is in proof strength and verification portability.

The core mechanism applies wherever data can be deleted and that deletion can be observed by attested code. The specific instantiation varies by database platform, proof type, deployment model, and attestation method. Section 3 maps this configuration space. Section 4 walks through the reference implementation, MKTd on ICP in 'Leaf' mode: ICP-Delete(Leaf) (ICPDL).

2.1 What a CVDR Contains

A CVDR always binds together three classes of information:

- **State transition evidence.** Cryptographic commitments to the data state immediately before and after the deletion event. No plaintext personal data appears in these commitments. Any party with knowledge of the record identifier can verify their record was the subject of the transition.
- **Code attestation.** A hash of the Zombie Delete module that executed the deletion, together with attestation from a trusted environment (hardware TEE or ICP subnet signature) that the identified code ran without external modification. This prevents post-hoc receipt fabrication: any code change produces a detectably different hash.
- **Sequencing and anchoring.** A monotonically increasing deletion sequence number preventing replay, a timestamp, and - where applicable - an immutable external anchor (such as an ICP canister certified variable or storage on Arweave) providing an independent reference point for the event.

The exact field structure of a CVDR is configuration-specific and is documented in the relevant integration guide for each product variant. A concrete example appears in Section 4.4. The classes above are stable across all planned variants.

2.2 What a CVDR Does Not Prove

Zombie Delete is explicit about the limits of what a CVDR demonstrates. The receipt proves that the Zombie Delete module, running in an attested environment, observed and committed to a state transition in the data it was given access to. It does not prove:

- That the Merkle or state commitments correspond exactly to the full enterprise database (this depends on the fidelity of the data feed, whether CDC, direct DB access, or canister-native access).
- That the enterprise cannot restore deleted data from backups or through other out-of-band means. Tombstone enforcement is procedural: a database trigger or canister guard. A malicious administrator with sufficient access could bypass it.
- That the enterprise’s database is the only place the data exists. Zombie Delete operates on what it is given.

These limits are not unique to Zombie Delete - no current tool can guarantee physical deletion from every copy of data across a complex enterprise. What Zombie Delete adds is verifiability: if the enterprise acts honestly - which describes the vast majority of

compliance use cases - any party can independently confirm it. If they act dishonestly, any reinsertion of a tombstoned record is inconsistent with prior CVDRs and detectable by any verifier. This transforms the compliance posture from unverifiable self-reporting to accountable, evidence-based attestation.

3. Configuration Families

Zombie Delete is not a single product. It is a family of configurations, each tailored to a different combination of database platform, proof type, and deployment context. This section describes the main groupings.

3.1 Proof Type: MKTd and ZKPd

The most fundamental configuration dimension is proof type: the cryptographic mechanism underlying the CVDR.

	MKTd - Merkle-Tree Delete	ZKPd - Zero-Knowledge Proof Delete
Mechanism	Cryptographic commitments to deletion-relevant state, using Merkle-set commitments where the configuration uses tree mode.	Sparse Merkle Tree non-membership proofs wrapped in zk-SNARKs (Groth16). Proves the membership-to-non-membership transition without revealing the Merkle path.
Attestation	Hardware TEE attestation (standard DB) or ICP subnet BLS threshold signature (ICP). Provides the independent trust anchor.	Same attestation options, with the addition that the zk-SNARK proof is itself independently verifiable by any party without a trust anchor query.
Verification	Requires published procedures and access to the attestation anchor (ICP certified variables or TEE root cert).	Self-contained. The proof is verifiable offline, on any platform, without querying live systems.
Appropriate for	Routine DSAR compliance, operational audit trails, enterprise pilot entry point.	Litigation-grade proof, long-term archival, offline verification, scenarios where the verifier has no ongoing relationship with the issuer. Required for environments where the proof must reveal nothing about the deleted data, the dataset structure, or the enterprise's data architecture: healthcare, financial services, government, and any enterprise with trade secret sensitivity.
Status	ICP-Delete(Leaf) (ICPDL): beta, live on mainnet. Zombie Delete Enterprise (standard DB): in development	In development. Dependent on circuit compilation and ZK verification infrastructure.

An enterprise that begins with MKTd can migrate to ZKPd as assurance requirements evolve. The CVDR concept and field classes are compatible across both; the proof blob and verification procedure change.

3.2 Database Platform

Configurations also differ by where the enterprise's PII data lives and how it is managed:

Platform	Description	Deletion Options	Notes
Standard (Web2)	Any SQL or NoSQL database (cloud-hosted, on-premise, distributed, etc.)	Hard-delete or tombstone (primary).	Zombie Delete Enterprise: integrates via CDC stream or queue table trigger; runs in TEE if available. Tombstone enforcement via DB trigger.
ICP (mutable chain)	Enterprise data in ICP canister stable memory, with native deletion capability	Tombstone (current implementations). Hard-delete optional.	ICPDL (beta). ICP subnet BLS certification replaces TEE hardware attestation. Mutable canister state enables genuine data modification.
Immutable chain	Append-only or near-immutable blockchain ledgers	Crypto-shredding only (where encryption is in use). Zombie Delete can evidence key destruction; it cannot prove hard-delete from an immutable ledger.	Bespoke solutions required per chain. Highest residual trust assumptions.
Chain Fusion	Off-chain PII with live on-chain references or contract-state entries on external chains	MKTd / ZKPd for off-chain deletion, plus on-chain witness / proof-anchoring for relevant contract-state updates	Extends Zombie Delete into cross-chain deployments where off-chain PII is linked to live on-chain references or contract-state entries. An ICP canister observes target-chain state changes and can anchor proof hashes back to the watched chain, creating a publicly auditable compliance record addressable across 20+ chain environments supported by ICP's production threshold signing infrastructure.

Web2 standard databases are a primary addressable market. They represent the largest volume of regulated personal data and the broadest enterprise buyer base. ICP is the initial deployment platform because its architecture - mutable state with native cryptographic certification - provides the cleanest technical environment for building and demonstrating the core CVDR mechanism. Subsequent configurations will extend Zombie Delete to standard database environments as development progresses.

3.3 Other Configuration Dimensions

Within each platform and proof type, configurations vary further across several axes:

Dimension	Options	Notes
Deployment	Library (compile-time dependency), PlugIn (self-contained TEE module), SaaS (TAV-hosted API)	Current implementations use Library mode. PlugIn and SaaS variants are planned.
Commit mode	Leaf (commitment covers a single record or data subject), Tree (commitment covers the full record-ID set)	ICPDL uses Leaf mode for single canister per user models. ICP-Delete(Tree) (planned) adds Tree mode for multi-subject-per-canister architectures.
Deletion type	Tombstone (permanent null-field marker), Hard-delete (record physically removed), Crypto-shredding (key destroyed)	Current implementations use Tombstone. See note on tombstones below.
Trigger mechanism	ICP native call, CDC (Change Data Capture), Queue table polled via DB triggers	ICPDL is triggered by an authorised canister call. Zombie Delete Enterprise subscribes to a CDC stream.
Code attestation	ICP subnet BLS, TEE hardware, Operator attestation	Attestation strength varies. ICP subnet BLS and TEE hardware provide stronger assurance than operator self-attestation.

Tombstones in Zombie Delete are permanent, not temporary. In standard database replication practice, a tombstone is a short-lived deletion marker used to propagate deletes across distributed nodes before eventual cleanup. In Zombie Delete, the tombstone is a perpetual replacement record: same record-ID, all PII fields set to a published cryptographic constant. It cannot be overwritten (enforced by a trigger or canister guard), and its presence - or any inconsistency between the tombstone and a prior CVDR - is detectable by any verifier holding the receipt.

The tombstone is what transforms a CVDR from a historical record into a living enforcement mechanism. A deletion receipt without a tombstone proves that data was deleted at a point in time. A deletion receipt with a tombstone proves that the data was deleted and cannot be reinserted without producing a detectable inconsistency. For enterprises, this is the difference between a compliance snapshot and a compliance guarantee. For regulators, it means any future reinsertion of personal data is not merely a violation but a violation with pre-existing cryptographic evidence against it.

4. Reference Implementation: MKTd on ICP (ICP-Delete(Leaf))

ICP-Delete(Leaf) (ICPDL) is the first working implementation of Zombie Delete. It is a rust library that integrates directly into an ICP canister's execution context, producing CVDRs for GDPR right-to-erasure requests against data held in ICP canister stable memory.

Status: ICPDL is live on ICP mainnet and is successfully generating CVDRs for GDPR-style deletion events in an operational canister application. The current product status is beta: the core mechanism is available on mainnet.

4.1 Why ICP

ICP was chosen for the initial implementation for specific architectural reasons, not because blockchain is the primary market. ICP canisters have genuinely mutable state: personal data in canister stable memory can be tombstoned or deleted. ICP's consensus layer - subnet BLS threshold signatures - certifies what the canister state is at each round. You can change state; you cannot forge a certificate about what state was.

This is the architectural foundation that makes ICP-based CVDRs independently verifiable: the certified commitment in a CVDR is attested by a supermajority of independent subnet nodes, against a public NNS root key.

Several practical properties are also relevant: the Swiss Subnet (launched January 2026, 13 independent node providers across Switzerland and Liechtenstein) provides data sovereignty for EU-regulated data; ICP's reverse gas model means enterprise clients interact through standard canister APIs without holding tokens or managing wallets; and orthogonal persistence means canister state survives canister upgrades without complex migration.

4.2 What ICP-Delete(Leaf) Does

ICPDL operates in Leaf mode: one data subject per canister. This is the architecture used by applications such as OpenChat, where each user's canister holds their own data. The enterprise canister imports the ICPDL Rust library as a compile-time dependency, implements a data adapter describing its PII fields, and exposes a deletion endpoint.

The deletion state transition and certified-commitment update occur atomically within a single ICP message; the finalized receipt then carries the certificate material needed for offline verification:

- Captures the current state hash of the canister's PII fields (pre-deletion commitment).
- Applies the tombstone: each PII field is replaced by a published cryptographic constant; the record-ID is retained.
- Verifies the tombstone was applied correctly; if not, rolls back the entire message execution.
- Captures the post-tombstone state hash (post-deletion commitment).
- Assembles the CVDR and stores it in canister stable memory.
- Updates the canister's certified variable with a compound commitment binding the pre- and post-state hashes to the deletion event and module hash.

ICP's single-threaded message execution model guarantees atomicity throughout. No concurrent modification can occur during the deletion. The CVDR is produced and the certified variable is updated in the same operation.

4.3 Integration Model

The enterprise canister adds four integration points: an import of the ICPDL crate, an adapter implementing the DataSource trait (declaring which fields are PII), lifecycle hooks in init() and post_upgrade(), and a refresh call after each PII-modifying write. The adapter is typically 30–80 lines of Rust. No changes to existing business logic are required.

The library runs inside the enterprise canister's execution context. There is no external network call to a Zombie Delete service at deletion time. This is intentional: for the state commitments to be cryptographically meaningful, the library must have atomic, direct access to the data being deleted. An external service receiving only after-the-fact notification cannot provide the same guarantees.

The enterprise receives CVDRs as structured data returned from the deletion call. Published verification procedures, the ICP subnet public key, and the canister module hash are the only inputs a verifier needs.

4.4 ICP-Delete(Leaf) CVDR Structure

The table below shows the fields of an ICPDL CVDR. The receipt is an unsigned artefact stored in canister stable memory. Its authority derives entirely from its consistency with the certified commitment - a canister certified variable signed by the subnet via BLS threshold signature. If any receipt field is altered after issuance, the recomputed hashes will not match the subnet-certified value, and V2 verification will fail.

The salt used in all state hash computations is derived as SHA-256("MKTd02_SALT_V1" || canister_id_bytes). It is not stored in the receipt but is deterministically recomputable from the canister_id. This prevents cross-canister hash linkage: two canisters with identical PII produce different state hashes.

Table follows

Field	Description	Type
protocol_version	Protocol identifier for this receipt format.	String — "MKTd02-v3"
receipt_id	Deterministic receipt identifier derived from canister_id, record_id, and deletion_seq using a published domain-separated hash formula. Unique per deletion event; not reusable.	[u8; 32]
canister_id	Principal of the canister in which the deletion occurred.	Vec<u8> — raw principal bytes
record_id	Identifies the data subject whose record was deleted. In Leaf mode, derived from the calling principal at deletion time (ic_cdk::caller()). For mediated call paths, derivation must be reviewed against the published integration assumptions.	Vec<u8> — raw principal bytes
pre_state_hash	SHA-256 of the canister's PII fields immediately before tombstoning, computed under the published State Encoding Spec and PII boundary definition. No plaintext PII appears in this value.	[u8; 32]
post_state_hash	SHA-256 of the canister's PII fields after tombstoning. Independently reproducible by any verifier using the tombstone constant, PII boundary definition, and serialisation spec.	[u8; 32]
tombstone_hash	SHA-256("MKTd02_TOMBSTONE_HASH_V1" || canister_id || tombstone_constant || timestamp || deletion_seq). Domain-separated and deterministically reproducible by any verifier.	[u8; 32]
deletion_event_hash	SHA-256("MKTd02_EVENT_V1" || pre_state_hash || post_state_hash || timestamp || module_hash || deletion_seq). Binds the state transition, timestamp, code identity, and deletion_seq into a single digest.	[u8; 32]
certified_commitment	SHA-256("MKTd02_CERTIFIED_V1" || post_state_hash || deletion_event_hash). Set as the canister's ICP certified variable at deletion time. Subnet BLS signature over this value is the trust anchor for V2 verification.	[u8; 32]
module_hash	SHA-256 of the exact deployed WASM binary for this canister, captured at init/upgrade time. Anchors code provenance. Verifiable against the published GitHub release record and reproducible build inputs.	[u8; 32]
timestamp	ICP consensus time at execution. Used as an ordering identifier; combined with deletion_seq to uniquely identify each deletion event.	u64 — nanoseconds
deletion_seq	Monotonically increasing counter, persisted in stable memory across canister upgrades. Strictly non-reusable. Prevents replay and establishes ordering across deletion events on the same canister.	u64
bls_certificate	Raw BLS certificate bytes captured via ic0.data_certificate() during the deletion flow. Embedded in the finalized receipt so that V2 verification can be performed offline from the receipt alone, without querying the live canister or subnet.	Option<Vec<u8>>
trust_root_key_id	Identifier for the NNS root public key used to verify the BLS certificate chain (e.g. "mainnet"). Embedded so that offline V2 verification uses the key that was active at issuance time, supporting verification across root-key rotation.	String
timestamp_iso (JSON export only)	Human-readable ISO 8601 rendering of the timestamp field. Informational only; not used in any verification procedure.	String

4.5 ICP-Delete(Leaf) CVDR Verification

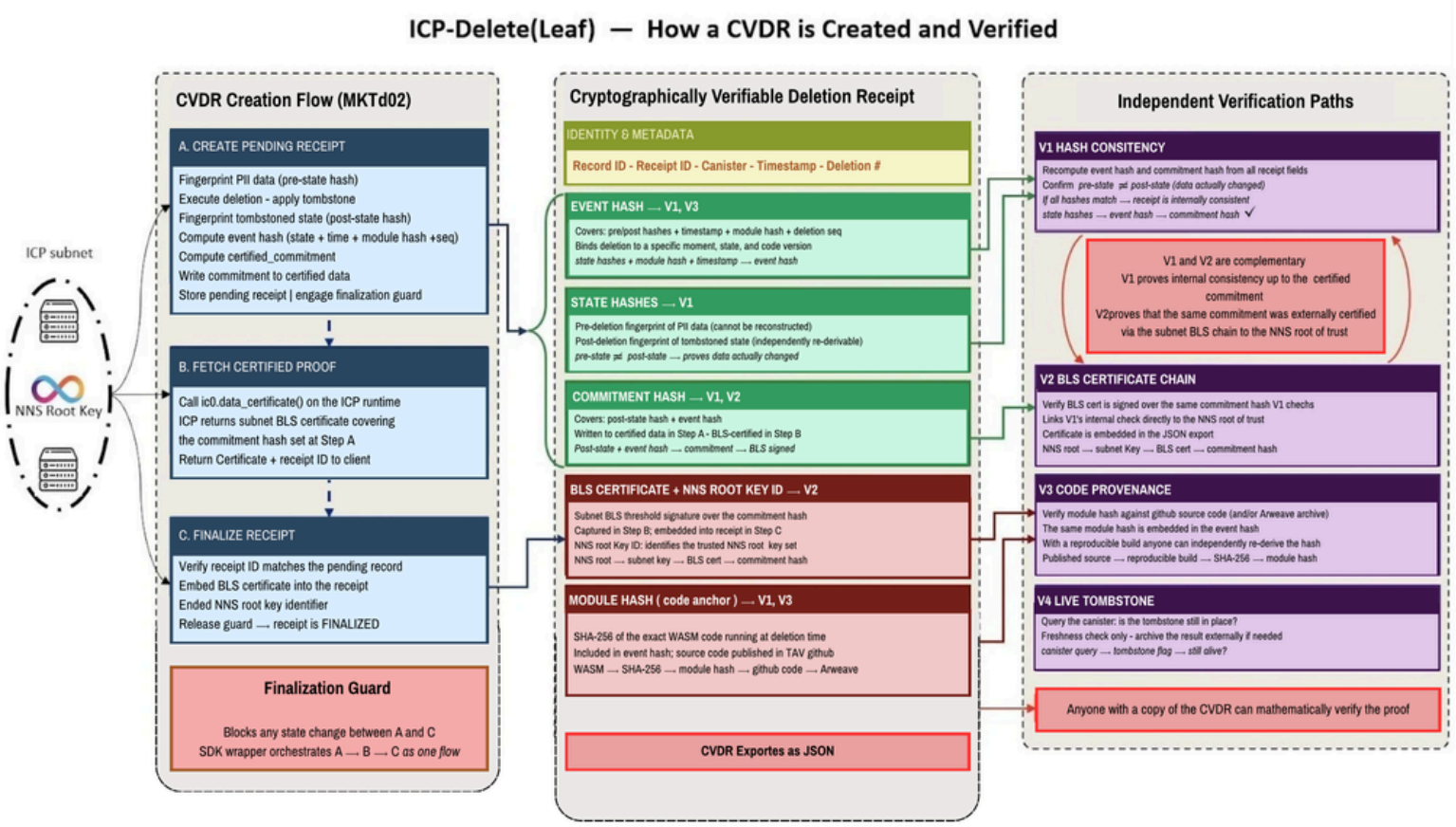
The published verification procedure for ICPDL receipts comprises four independently executable checks. V1 and V2 are the core cryptographic verification checks and can be performed from the receipt alone. V3 and V4 are supplementary checks that may require access to external resources.

Check	What It Confirms	How
V1 State transition consistency	The receipt's state hashes are internally consistent with a genuine tombstone having been applied under the declared PII boundary definition.	Using the published State Encoding Spec, PII boundary definition, and tombstone constant: (1) confirm pre_state_hash ≠ post_state_hash; (2) independently compute the expected post_state_hash — i.e. the hash of a state where every declared PII field equals the published tombstone constant — and confirm it matches the receipt; (3) recompute tombstone_hash from canister_id, tombstone constant, timestamp, and deletion_seq using the published domain-separated formula and confirm it matches the receipt.
V2 Certified commitment	The subnet BLS signature attests that, at the moment of capture, the canister's certified variable held the exact certified_commitment recorded in the receipt - proving the deletion event was committed in canister state at that time.	For finalized receipts (bls_certificate is Some): parse the embedded certificate CBOR; verify the BLS signature chain against the trust root identified by trust_root_key_id; enforce canister-range authorisation from the delegation chain; look up certified_data in the verified hash tree; recompute the expected certified_commitment from post_state_hash and deletion_event_hash and confirm equality. The certificate timestamp should be within a few seconds of the receipt timestamp - strong evidence the certificate was captured during the deletion flow itself. This path requires no live network call and works permanently from the archived receipt.
V3 Code provenance	The canister was running the expected ICPDL-integrated code at the time of deletion. Post-deletion upgrades are detectable but do not invalidate previously issued receipts.	Primary (archival) path: compare the receipt's module_hash against the published GitHub release record, which lists the expected WASM hash alongside the Rust toolchain version, build flags, and source commit for that release. An independent auditor can reproduce the WASM from source using the published build instructions and confirm byte-for-byte identity. This path remains meaningful even if the canister no longer exists. Fallback (live canister) path: compare the receipt's module_hash against the canister's current on-chain module hash (queryable via the ICP management canister) as a corroborating check that the deployed code has not changed since deletion. If module_hash is all zeros, the deployer did not supply the WASM hash at init/upgrade time.
V4 Tombstone persistence	The tombstone has not been reversed since the receipt was issued.	Query the target canister directly: call mktd_get_tombstone_status() to confirm the canister is currently tombstoned; call mktd_get_state_hash() via certified query to confirm the current certified state hash is consistent with the receipt's post_state_hash. Requires the canister to be live and reachable.

Full verification procedures, reference code, and templates are publicly available in Together Alone Ventures' public GitHub repository.

ICP-Delete(Tree) (ICPDT) extends ICPDL to support multi-subject-per-canister architectures using Merkle path proofs over the full record-ID set, suitable for applications that hold many users' data in a single canister.

Figure 1. ICP-Delete(Leaf) - CVDR Creation, Structure, and Independent Verification Paths



5. What Zombie Delete Assures, and What It Does Not

The table below maps each relevant threat to Zombie Delete's capability, clearly distinguishing prevention from detection from evidence.

Threat	Prevents	Detects	Evidence	Mechanism
CVDR is forged or backdated	Yes	Yes	Yes	Attestation (ICP subnet BLS or TEE hardware) cannot be reproduced by any single party. Deletion sequence number is strictly monotonic and non-reusable.
Data remains in source system post-deletion	Yes*	Yes	Yes	Post-state commitment reflects the tombstoned state. Any reinsertion with non-null fields is inconsistent with prior CVDRs and detectable by any verifier. *Prevention depends on tombstone guard remaining in place.
Deletion module tampered with	No	Yes	Yes	Module hash is embedded in every CVDR and verifiable against the published binary hash. Any code change produces a detectably different value.
Enterprise restores deleted data from backup	No	Yes	Yes	Out-of-band bypass is possible for a malicious administrator. But any reinsertion inconsistent with a prior CVDR provides cryptographic evidence of a knowing violation under GDPR's accountability principle - materially different legal exposure from unknowing omission.
State commitments do not reflect full DB	No	Partial	Partial	Residual trust in the data feed (CDC integrity or canister-native access). Source code audits and TEE attestation reduce but do not eliminate this assumption. Explicitly declared in published residual trust statements per configuration.

A further assurance property applies across all configurations. A CVDR issued against a deletion that did not fully occur is not merely a failed proof; it is cryptographically documented false attestation. The enterprise holds signed, timestamped, independently verifiable evidence that it claimed a deletion happened. If a regulator, litigant, or auditor later demonstrates the data persists, the CVDR becomes evidence of a knowing false compliance claim. Under GDPR's accountability principle, the difference in liability exposure between an unverifiable log entry and a cryptographic receipt is material: without a CVDR, an incomplete deletion is an oversight; with a CVDR, it is a documented inconsistency checkable by any adversary.

Scope Boundary

A CVDR proves that a specific record was deleted from a defined system boundary. It does not currently prove deletion from external replicas, third-party processors, or systems outside the attestation scope. It enables provable deletion from all systems within the CVDR's attestation perimeter, not provable erasure from every copy that may exist anywhere. This is a deliberate design choice: the CVDR's value as evidence depends on the precision of what it attests to.

> Without a CVDR, an incomplete deletion is an oversight. With a CVDR, it is cryptographically documented false attestation.

Each configuration publishes a Residual Trust Statement listing the assumptions that remain after the Zombie Delete mechanism is applied.

ID	Assumption	Mitigation
RT1	ICP subnet consensus is honest (2/3 nodes not colluding)	Foundation of ICP security model. Subnet membership governed by NNS. Sovereign subnets/UTOPIA change this - operator controls all nodes.
RT2	Canister controller does not upgrade canister to reverse tombstones	Module hash verification (V3) detects code changes. deletion_event_hash in certified commitment binds the module hash at deletion time. Governance/DAO control of upgrades mitigates. Receipts with prior module hash remain valid evidence of what happened under that code version.
RT3	MKTdDataSource adapter's pii_field_manifest faithfully declares all PII fields	Open-source adapter code is auditable. The State Encoding Spec / PII boundary definition should be published and version-controlled. Adapter runs inside canister under consensus. If the published boundary omits a PII field, that field is excluded from the hash and survives tombstoning - the receipt is valid but deletion is incomplete. This is an integration correctness concern, not a protocol failure.
RT4	Tombstone execution renders prior PII values irrecoverable	MKTd02 performs logical overwrite of PII fields in stable memory. ICP does not guarantee that prior byte values are physically irrecoverable from individual replica storage. MKTd operates at the logical protocol level; physical data remanence is a platform-level concern outside the protocol's scope. This is consistent with GDPR's standard of "reasonable measures" - logical deletion with cryptographic proof is the appropriate benchmark, not physical media scrubbing.
RT5	No canister snapshots or backups exist that could resurrect deleted data	ICP canister snapshots (backup/migration feature) could contain pre-deletion state. Snapshot restoration would change the certified commitment, making it inconsistent with existing receipts - detectable by any verifier performing V2. However, detection requires active verification; MKTd02 Leaf mode does not include continuous monitoring. Receipt explicitly documents point-in-time deletion; snapshot management is an operational concern outside MKTd scope.

6. Roadmap

The following represents the planned development sequence. Timelines are indicative, pending exact timing of circuit development for ZKPd, and ongoing technical validation.

Configuration	Description	Stage
ICPDL	ICP / Leaf mode / Tombstone. Current reference implementation.	Beta on mainnet. Hardening underway.
ICPDT	ICP / Tree mode / Tombstone. Multi-subject-per-canister. Merkle path proofs over full record-ID set.	In finalisation. Multi-subject-per-canister architecture with Sparse Merkle Tree in StableBTreeMap, BLS subnet certification, freeze guard, and tombstone anti-resurrection on all write paths.
Zombie Delete Enterprise	Standard DB / TEE / CDC trigger / Tombstone. Zombie Delete Enterprise.	In development.
ZKPd01	Standard DB / ZK / TEE. Adds Groth16 zk-SNARK layer to Zombie Delete Enterprise for self-contained proof.	Design phase. Dependent on circuit compilation.
ZKPd02 (ICP)	ICP / ZK / Leaf mode. Adds full ZKP to ICP-native variant.	Planned. Follows ZKPd01.
PQR-Delete	NIST FIPS 204/205 ML-DSA / SLH-DSA post-quantum signatures on CVDR. Any platform.	Planned. Targeted at government and financial institutions with quantum-threat horizons.
Chain Fusion	Cross-chain witness and proof-anchoring capability for blockchain-native enterprises with off-chain PII and live on-chain references. Complements MKTd / ZKPd by adding auditable evidence around target-chain state updates.	Active commercial discussions; observer-form architecture is feasible now, with stronger cross-chain variants on the roadmap.

6.1 ZKPd: The Full Proof

ZKPd adds a zk-SNARK layer to the Merkle-based CVDR. The proof establishes the transition from Sparse Merkle Tree record membership to non-membership without revealing details of the Merkle path. The resulting CVDR is self-contained: verifiable by any party, offline, without querying live systems.

The economic case for ZKPd has strengthened significantly: Groth16 proofs now verify for a fraction of a cent on specialised verification layers such as zkVerify, removing the historical cost barrier to on-chain proof anchoring.

The privacy property is central. A ZKPd CVDR proves the membership-to-non-membership transition without revealing the Merkle path, the record content, or the structure of the surrounding dataset. The enterprise proves deletion occurred; the verifier learns nothing about what was deleted, where it sat in the tree, or what other records exist. For healthcare (HIPAA), financial services (PCI DSS), government, and any enterprise with trade secret concerns in their data architecture, this is the only acceptable proof model[2][6]. An MKTd receipt reveals the state hashes; a ZKPd proof reveals only the fact of deletion.

> Zero-knowledge proof of absence: the enterprise proves deletion occurred. The verifier learns nothing about what was deleted, where it sat in the dataset, or what other records exist.

ZKPd is also the configuration that makes Zombie Delete categorically distinct from any alternative approach. A hash-chain audit log can be forged by anyone with write access. An attestation-based receipt depends on the continued trustworthiness of the attestation infrastructure. A ZKPd CVDR depends on nothing except the mathematics: it is verifiable on any platform, by any party, offline, permanently, without trusting any infrastructure, any issuer, or any attestation service. The patent portfolio's teaching-away claims explicitly declare that hash-chain audit logs and attestation-based certificates are insufficient substitutes for this level of cryptographic proof.

The same ZKPd upgrade path applies across the entire Zombie Delete suite, including Chain Fusion cross-chain configurations. MKTd is the entry point: fast to deploy, sufficient for routine DSAR compliance, and available now. ZKPd is the destination: litigation-grade, privacy-preserving, and self-verifying. Enterprises that begin with MKTd can migrate to ZKPd as their assurance requirements evolve, without changing their integration or their data architecture.

6.2 The Great Clean-Up (Planned)

A number of organisations have historical deletion obligations for which they hold no cryptographic evidence: personal data was deleted on request, but the only record is a workflow log. Zombie Delete has a planned procedure, the Great Clean-Up, to address this. Given the deleted record-id, it derives deletion evidence from existing business

artefacts - DSAR correspondence, CRM logs, retention expiry records - and produces CVDRs attesting to current non-existence of the relevant record identifiers.

The Great Clean-Up does not retroactively prove that deletion occurred at a specific historical moment. It is a remediation tool, not a time machine. The procedure is designed; implementation is planned as a follow-on to the MKTd product family.

6.3 Chain Fusion

Chain Fusion extends Zombie Delete into blockchain-native architectures where personal data remains off-chain while related references or contract-state entries exist on other chains. Enterprises using Ethereum, Solana, Cardano, Polkadot, or any other supported blockchain do not need to migrate to ICP. Their data stays where it is. The ICP canister operates as an independent observer and proof engine: it reads the target chain's state, confirms pre/post transition events, generates the CVDR, and optionally anchors the proof hash back to the enterprise's own chain. The enterprise never touches ICP directly.

> Enterprises do not need to migrate to ICP. Their data stays where it is. The ICP canister operates as an independent observer and proof engine. 20+ chains reachable from a single canister today.

In this model, an ICP canister can observe relevant target-chain state transitions, generate a CVDR or supporting witness record on the ICP side, and anchor a proof hash back to the watched chain. This adds a valuable cross-chain evidentiary layer for enterprises, auditors, and ecosystem partners that want auditable proof surfaces spanning both off-chain systems and live on-chain state. Regulatory interest is already established: Spain's data protection authority (AEPD) has run a formal proof of concept on blockchain erasure[3], confirming the on-chain reference question as an active compliance concern rather than a theoretical one. As the architecture matures, the same ZKPd upgrade path available across the rest of the Zombie Delete suite applies here: observer-form CVDRs in the near term, progressing to self-contained, offline-verifiable cross-chain proofs.

ICP's threshold signing infrastructure currently supports three signature schemes in production on mainnet: threshold ECDSA on secp256k1 (covering all EVM chains, Bitcoin, Dogecoin, Cosmos, Filecoin, Hedera, Stacks, and XRP), threshold Schnorr

BIP-340 on secp256k1 (covering Bitcoin Taproot), and threshold Schnorr Ed25519 (covering Solana, Cardano, Polkadot, Ripple, Stellar, and Toncoin). Combined with HTTPS outcalls to any chain with a public RPC endpoint, a single MKTd03 Tree-mode canister can observe and attest to deletion events across 20+ chain ecosystems today.

Proof anchoring, writing the CVDR hash back to the target chain via threshold signing, is a distinct value proposition that applies regardless of whether the on-chain reference itself constitutes personal data under GDPR. It provides the enterprise with a publicly auditable deletion record on the chain their regulators and auditors are already monitoring.

A further capability on the roadmap is canister-executed deletion, where the ICP canister signs and submits deletion transactions directly to target chain smart contracts via threshold signing, then verifies the result and generates the CVDR within a single trust boundary. This provides a stronger Article 28 compliance position: the processor executes and proves, rather than relying on the controller to execute correctly while the processor merely witnesses.

What is Proven Today

Live on ICP mainnet (beta): ICP Delete (ICPDL, Leaf mode) generates CVDRs for single-canister deletion events. The DaffyDefs test dApp demonstrates the full lifecycle: pre-state commitment, tombstone application, post-state commitment, CVDR assembly, and BLS subnet certification. The CVDR is verifiable against the NNS root public key without requiring a live subnet or any relationship with Together Alone Ventures.

What the verifier can independently check: That the pre-deletion state hash and post-deletion state hash are bound to a specific canister, module hash, and deletion sequence number; that the tombstone constant is present in the post-state; and that the CVDR carries a valid BLS threshold signature from the hosting subnet.

Shipping (within 60 days): ICP Delete Tree mode (ICPDT) extends coverage to multi-subject canisters with per-record Sparse Merkle Tree non-membership proofs.

Within four months: Zombie Delete Enterprise (standard databases via CDC/library integration) and ZKP-Delete (zero-knowledge proof of absence). Both build on the same CVDR architecture using proven primitives (Sparse Merkle Trees, zk-SNARKs, BLS threshold signatures). No further cryptographic research is required.

7. Delivery Models

Zombie Delete reaches enterprises through several integration patterns, determined by their technical architecture and integration capacity:

Model	Description	Suited For
Library / compile-time dependency	The Zombie Delete module is a code library compiled into the enterprise's existing application or canister. Current primary model.	ICP-native applications, enterprises with development resource, maximum cryptographic assurance.
PlugIn / TEE module	A self-contained module deployed within the enterprise's TEE-capable infrastructure (enclave selection is configuration-dependent; the architecture supports Intel SGX, ARM TrustZone, and AWS Nitro Enclaves), connecting to their database via CDC or queue table.	Enterprises wanting minimal application code changes; Web2 standard DB deployments.
SaaS / hosted API	Enterprise deploys tombstone locally; calls the Zombie Delete REST API for remote proof computation and CVDR return. Planned as a self-serve platform with API key provisioning, usage dashboard, and CVDR archive accessible to the enterprise's compliance team.	Enterprises preferring an external service over local library deployment. Companies onboarding via self-serve without development resource. Compliance teams requiring audit-ready CVDR archives.
OEM / embedded	White-label integration within existing privacy compliance platforms (e.g. OneTrust, BigID, TrustArc).	Privacy vendors seeking to add cryptographic proof capability to existing customer bases.

The following additional integration patterns support developer-led and self-serve adoption across blockchain and Web2 environments:

Model	Description	Suited For
REST API / webhook	Application calls the Zombie Delete API endpoint after executing a deletion. The API reads target chain or database state, generates the CVDR, and returns it to the caller. Webhook option for event-driven architectures.	dApps on any chain (EVM, Solana, Cardano, Polkadot) via Chain Fusion. Web2 applications with existing deletion workflows. Minimal integration effort.
SDK (npm, pip, cargo)	Open-source client libraries for major languages and frameworks. Wraps the REST API with typed interfaces, handles authentication, and provides helper functions for CVDR verification.	Developer-led adoption. Product-led growth. Companies integrating Zombie Delete into existing applications without architectural changes.
Chain Fusion verification API	A dedicated endpoint for cross-chain deletion verification. The caller specifies chain, contract address, and record identifier. The ICP canister reads target chain state via HTTPS outcalls, confirms deletion, generates CVDR, and optionally anchors the proof hash back to the target chain via threshold signing.	Any dApp or enterprise on any of the 20+ chains reachable via Chain Fusion. No ICP knowledge required by the integrator. EVM, Solana, Cardano, Polkadot, Bitcoin (off-chain content only).

The OEM path is strategically significant. Existing privacy compliance platforms serve tens of thousands of enterprise customers and handle the workflow of deletion request management. Zombie Delete as a verification layer within their platforms is a faster route to scale than direct enterprise sales.

A forward-looking extension of the CVDR system will generate Universal Compliance Credits (UCCs) - optional, transferable instruments representing verified deletion events, mintable on ICP and presentable to regulators, auditors, or insurers. UCCs are not required to use Zombie Delete; the CVDR system operates independently. Details are reserved for a subsequent paper.

8. Regulatory Context

Several independent regulatory deadlines converge in 2026 and 2027, each requiring demonstrable deletion capability rather than process attestation:

Regulation	Deadline	Key Requirement
EDPB Guidelines 02/2025	In force (adopted April 2025)	Blockchain immutability is not a GDPR defence[1]. All on-chain projects handling personal data are subject to Article 17.
California Delete Act (SB 362) / DROP	Operational since Jan 2026; audits from Jan 2028	500+ registered data brokers must comply. $200 per request per day overdue.
EU AI Act Article 10	Aug 2, 2026	Training data governance for high-risk AI systems. Increased pressure for documented training-data governance and auditable remediation pathways.
India DPDP Act	May 13, 2027	Comprehensive data protection for 1.4 billion population. Deletion rights enforceable.
20+ US state privacy laws	Various, 2025–2027	Fragmented but cumulative. All include deletion rights. All require demonstrable compliance.

9. Competitive Landscape

We are not aware of any mainstream deletion-compliance platform that currently issues independently verifiable cryptographic deletion receipts as a standard product output. Current enterprise deletion tools generate workflow logs and self-attestations[5]. Recent academic work on verifiable deletion in cloud storage [8][11] addresses integrity verification for outsourced data but does not produce a self-contained compliance receipt, does not implement tombstone anti-resurrection, and does not support the zero-knowledge privacy property required for regulated environments; the structural difference is that a CVDR can be checked by any party without relying on the issuer's own reporting. The following comparison reflects publicly available product information:

Existing tools: "We deleted it. Trust us."
Zombie Delete: "We deleted it. Check for yourself."

Zombie Delete is designed to operate as an embedded verification layer within existing compliance platforms, not as a replacement for workflow management. The OEM delivery model enables platforms such as OneTrust, BigID, and TrustArc to add cryptographic deletion proof to their existing deletion workflows without architectural changes.

Zombie Delete will seek SOC 2 Type II and ISO 27001 certifications. We recognise the irony: these are attestation-based trust frameworks, precisely the model our product is designed to surpass. We pursue them because enterprise procurement requires them, not because they represent the standard of proof we believe the industry should accept.

Capability	Zombie Delete	OneTrust	BigID	Transcend	Blancco
Cryptographic deletion receipt (CVDR)	**Yes (CVDR)**	No	No	No	No
Independent third-party verification	**Yes**	No	No	No	Partial
ICP-native deletion (mutable chain)	**Yes (ICPDL)**	No	No	No	No
Zero-knowledge proof variant	**ZKP-Delete (planned)**	No	No	No	No
Post-quantum readiness	**PQR-Delete (planned)**	No	No	No	No
Cost per deletion event	**~$0.001 (ICPDL on ICP)**	Bundled	Bundled	Bundled	$1,400+ (manual)

An additional differentiator not captured in the table above: a Zombie Delete CVDR, if issued against a deletion that subsequently proves incomplete, constitutes cryptographically documented false attestation, independently verifiable by any party. No existing competitor produces an output with this enforcement property. Furthermore, the ZKPd configuration will produce self-verifying proofs requiring no trust in any attestation infrastructure, issuer, or live system. This is a categorically different class of evidence from any workflow log, erasure certificate, or attestation-based receipt currently available in the market.

10. Intellectual Property

28 provisional patent families filed between October 2025 and March 2026 by Together Alone Ventures OÜ (Estonia), sole assignee. The portfolio is designed to provide broad coverage across the CVDR lifecycle, with claims that span core proof mechanisms, delivery configurations, and application contexts. The intent is comprehensive protection of the architecture rather than narrow claims around individual components.

The cryptographic primitives used in Zombie Delete (Sparse Merkle Trees, zk-SNARKs, BLS threshold signatures) are well-established in the literature. The novelty protected by the portfolio is the specific system that assembles them into a deletion verification lifecycle with properties no existing academic or commercial system provides. Beyond the core CVDR mechanism, the portfolio includes patent families covering capabilities with no known prior art in the deletion compliance space: Viral Tombstone (cross-system deletion propagation with governance controls), Universal Compliance Credits (tokenised deletion receipts as transferable compliance instruments), Inverted Zombie (proving data that should exist is absent, the inverse verification problem), and MLUVC (verifiable AI model unlearning via zero-knowledge circuits). Each extends the CVDR architecture into adjacent compliance markets that current academic work on verifiable deletion[8][11] does not address.

Patent Family	Coverage
ZKP-Delete Lifecycle	60+ claims. Core deletion proof lifecycle. Teaching-away claims explicitly declare that hash-chain audit logs and access revocation are insufficient substitutes for cryptographic deletion proof.
MKT-Delete	Merkle-commitment deletion receipts. The foundation product patent family.
PQR-Delete	Post-quantum resistant deletion receipts using NIST FIPS 204/205 ML-DSA and SLH-DSA.
VDF Timing	Temporal ordering enforcement. Anti-backdating via Verifiable Delay Functions.
Liability Airlock	Cryptographic liability transfer boundary. Timestamped commitment establishing a legally meaningful handoff point.
UCC	Universal Compliance Credits: tokenised compliance instruments with full derivatives infrastructure.
Hardware	Hardware-anchored deletion verification for air-gapped and classified environments (HSM / TEE attestation).
MLUVC + Extensions	AI model unlearning verification and architecture-aware ZK circuits. Subject of a forthcoming technical white paper.
Viral Tombstone	Cross-system deletion propagation and resurrection prevention. Subject of a forthcoming technical white paper.
Omnibus Extension	Coverage for combinations and applications across the portfolio.

11. Executive Team

Glen Cameron, CEO and Inventor. 20+ years C-suite experience in global communications and strategy. Second-time blockchain founder.

Dr Stef Savanah, Chief Scientist and Inventor. 26 patents granted, 470+ pending. Former Director of Scientific Research at nChain. PhD from the ARC Centre of Excellence in Cognition at Macquarie University.

Robin Davies, COO. FinTech/blockchain specialist.

Advisory Board: Matt Dangerfield (30+ years in regulated distributed systems, ICP architecture), Richard Slater (former partner, Simmons and Simmons), Vikram Seth (former Head of Web3 at Shell), Kanan Dhru (LegalTech VC pioneer).

12. Conclusion

Zombie Delete is a family of software configurations that produce Cryptographically Verifiable Deletion Receipts - independently auditable evidence that a specific data deletion event occurred. The CVDR is the common output across all variants, from Merkle-commitment receipts (MKTd) to full zero-knowledge proofs (ZKPd), across standard databases and mutable-state blockchains. Chain Fusion extends this reach into blockchain-native architectures, with active commercial engagement across multiple chain ecosystems.

The first live implementation, ICPDL, is running on ICP mainnet today as beta and generating CVDRs for deletion events in an operational canister application.
For ICP-native applications: Zombie Delete is a library integration that generates a verifiable receipt each time a user exercises the right to erasure. For enterprise databases: the same capability is in development for standard database environments.

For regulators and auditors: the receipt is independently verifiable without any relationship with Together Alone Ventures.

Zombie Delete is, to our knowledge, the only platform that produces cryptographic proof of data deletion, across on-chain and off-chain systems, verifiable by any party without trusting the issuer.

13. References

[1] European Data Protection Board, Guidelines 02/2025 on processing of personal data through blockchain technologies, Version 1.1, Adopted 8 April 2025.

[2] INATBA Privacy Working Group, Leveraging Zero-Knowledge Proofs for GDPR Compliance in Blockchain Projects (Updated Edition), 26 August 2025.

[3] Agencia Española de Protección de Datos (AEPD), Technical note: Proof of concept — Blockchain and the right to erasure, 13 November 2024.

[4] T. Eisenhofer, D. Riepel, V. Chandrasekaran, E. Ghosh, O. Ohrimenko, and N. Papernot, "Verifiable and Provably Secure Machine Unlearning," in Proc. IEEE Conference on Secure and Trustworthy Machine Learning (SaTML), Copenhagen, 2025, pp. 479–496.

[5] A. Zafar, "Reconciling blockchain technology and data protection laws: regulatory challenges, technical solutions, and practical pathways," Journal of Cybersecurity, vol. 11, no. 1, 2025.

[6] A. Barthoulot, O. Blazy, and S. Canard, "Cryptographic Accumulators: New Definitions, Enhanced Security, and Delegatable Proofs," in AFRICACRYPT 2024, LNCS vol. 14861, Springer, 2024, pp. 94–119.

[7] F. Baldimtsi, I. Karantaidou, and S. Raghuraman, "Oblivious Accumulators," in PKC 2024, LNCS vol. 14602, Springer, 2024, pp. 99–131.

[8] X. Li, J. Wu, and J. Ni, "Accelerating Secure and Verifiable Data Deletion in Cloud Storage via SGX and Blockchain," in Proc. IEEE GLOBECOM, 2024, pp. 4138–4143.

[9] G. Xu et al., "Empowering Data Owners: An Efficient and Verifiable Scheme for Secure Data Deletion," Computers & Security, vol. 144, 2024.

[10] E. Podda, P. Hölzmer, A. Amard, J. Sedlmeir, and G. Fridgen, "The impact of zero-knowledge proofs on data minimisation compliance of digital identity wallets," Internet Policy Review, vol. 14, no. 3, 2025.

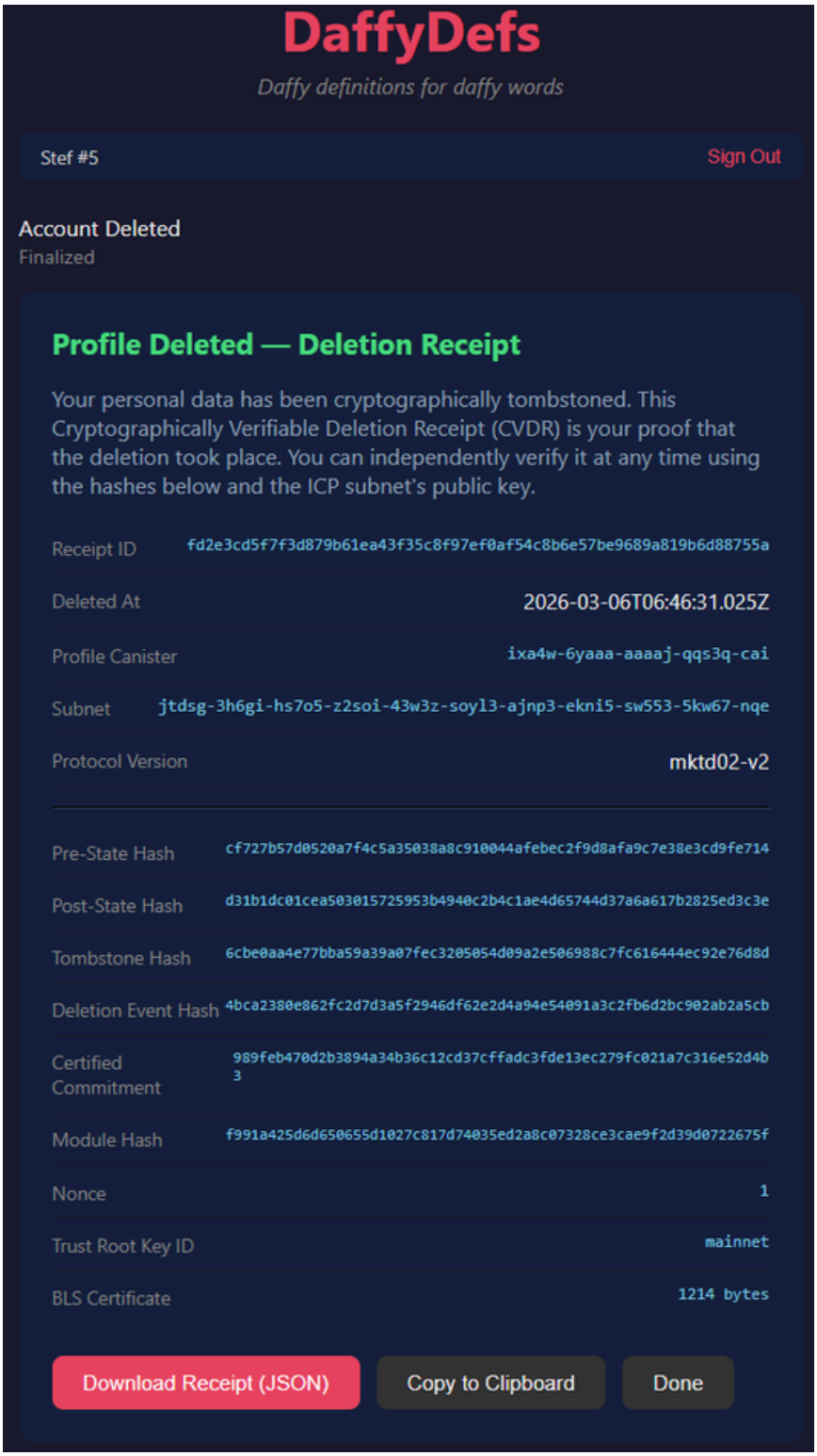
[11] M. A. Darwish, E. A. Markatou, and G. Smaragdakis, "Provable Co-Owned Data Deletion with Zero-Residuals and Verifiability in Multi-Cloud Environment," in Proc. 18th European Workshop on Systems Security (EuroSec'25), Rotterdam, 2025, pp. 77–83.

14. Appendix

14.1 Appendix 1: Glossary

Term	Definition
CVDR	Cryptographically Verifiable Deletion Receipt. The core output of every Zombie Delete configuration: an independently auditable artefact providing cryptographic evidence that a specific data deletion event occurred.
Web2 Delete	Also called Zombie Delete Enterprise.
Zombie Data	Data that has been nominally deleted from production but persists in backups, replicas, warehouses, third-party processors, or other shadow stores. A continuing GDPR compliance liability.
Tombstone (Zombie Delete)	A permanent replacement record retaining only the original record-ID, with all PII fields set to a published cryptographic constant (the Tombstone Constant). Permanent by design. Distinguished from replication tombstones, which are temporary.
MKTd	Merkle-Tree Delete. The Zombie Delete product family producing CVDRs using Merkle tree commitments and hardware TEE or ICP subnet attestation. Does not require zero-knowledge circuit compilation.
ZKPd	Zero-Knowledge Proof Delete. The Zombie Delete product family producing self-contained CVDRs using Sparse Merkle Trees and Groth16 zk-SNARKs. Proofs are verifiable offline by any party without querying live systems.
Leaf mode	CVDR commit mode where the cryptographic commitment covers a single data subject's record. Appropriate for one-subject-per-canister architectures (ICPDL).
Tree mode	CVDR commit mode where the cryptographic commitment covers the full set of record identifiers via a Merkle tree. Used in Web2-Delete and planned ICPDT.
Sparse Merkle Tree (SMT)	A data structure enabling efficient non-membership proofs: cryptographic evidence that a specific key does NOT exist in a dataset. Central to ZKPd configurations.
zk-SNARK	Zero-Knowledge Succinct Non-interactive Argument of Knowledge. A proof that a statement is true without revealing any information beyond the truth of the statement.
BLS Threshold Signature	ICP's subnet certification mechanism. No single node can produce a valid signature; a threshold of independent nodes must agree. Verifiable against the NNS root public key.
Certified State Proof	ICP's mechanism for authenticating canister state. Each consensus round, the subnet signs the state tree root. Enables offline verification by any party with the NNS root public key.
TEE	Trusted Execution Environment. Hardware-enforced isolated computation providing attestation: cryptographic proof that specific code ran on specific hardware without external modification.
CDC	Change Data Capture. Infrastructure that streams database change events in real time, enabling Zombie Delete modules to observe deletion events without direct database access.
DSAR	Data Subject Access Request. The mechanism by which individuals exercise GDPR rights, including the right to erasure under Article 17.
UCC	Universal Compliance Credit. An optional tokenised instrument minted against a verified deletion event and burned when consumed for compliance reporting.
NNS	Network Nervous System. ICP's on-chain governance system and root of trust for subnet signatures.
Residual Trust	The set of assumptions that remain after a Zombie Delete configuration is applied. Explicitly declared per configuration. Substantially weaker than the implicit trust requirements of log-based compliance.

14.2 Appendix 2: Testnet CVDR



14.3 Appendix 3: Chain Fusion Connections

Chain	Authentication	Available Integrations	Assets
Aptos	ECDSA, EdDSA	RPC via HTTPS outcalls	
Arweave	Not yet supported	RPC via HTTPS outcalls	
Avalanche	ECDSA	RPC via HTTPS outcalls	
Bitcoin	ECDSA, Schnorr	Direct	ckBTC
Cardano	ECDSA, EdDSA, Schnorr	RPC via HTTPS outcalls	
Cosmos	ECDSA, EdDSA	RPC via HTTPS outcalls	
Dogecoin	ECDSA	Direct	ckDOGE (upcoming)
Ethereum	ECDSA	EVM RPC, ic-web3-rs	ckETH, ckERC20
Filecoin	ECDSA	RPC via HTTPS outcalls	
Hedera	ECDSA, EdDSA	RPC via HTTPS outcalls	
Kaspa	ECDSA	RPC via HTTPS outcalls	
Monero	EdDSA	RPC via HTTPS outcalls	
NEAR	EdDSA	RPC via HTTPS outcalls	
Polkadot	ECDSA, EdDSA	RPC via HTTPS outcalls	
Solana	EdDSA	SOL RPC	
Stacks	ECDSA	RPC via HTTPS outcalls	
Stellar	EdDSA	RPC via HTTPS outcalls	
Sui	ECDSA, EdDSA	RPC via HTTPS outcalls	
XRP	ECDSA, EdDSA	RPC via HTTPS outcalls	
Toncoin	EdDSA	RPC via HTTPS outcalls	
TRON	ECDSA	RPC via HTTPS outcalls	

Delete It. Prove It. Anywhere.

The regulatory deadlines are fixed.

The cryptographic foundation is proven.

The build is underway.

glen@together-alone.com
together-alone.com | zombiedelete.com