Etornie: Verifiable On-Chain Custody and Zero-Knowledge Compliance for Tokenized Intellectual Property on Solana

Ilknur Timur-Tuncer and the Etornie Research Team
Etornie
Ruessenstrasse 5, Baar, Canton of Zug, 6340 Switzerland
https://etornieagent.xyz

***Abstract*: Etornie is an intellectual-property (IP) protocol that binds legally meaningful IP records to verifiable, privacy-preserving on-chain state on Solana. Rather than tokenizing IP as tradeable financial instruments, Etornie issues non-transferable, soul-bound records whose authenticity, ownership, and lifecycle are cryptographically attestable while the underlying documents and identities remain off-chain and protected. The protocol combines four constructions: (i) a canonical-hash attestation layer that anchors case metadata and its full lifecycle to append-only on-chain events; (ii) a soul-bound Token-2022 issuance scheme in which transferability is disabled at the protocol level through frozen-by-default token accounts under program-controlled authority; (iii) a Groth16 zero-knowledge layer over the BN254 curve that lets a holder prove file ownership and query compliance without revealing the file, the query, or the witness; and (iv) a sponsored-transaction custody model in which a backend operator funds and co-signs user-authorized transactions under a strict instruction allowlist that provably prevents operator-key drainage. We formalize the attestation, tokenization, proof, and custody constructions, analyze their security under a defined threat model, and describe how the design reconciles blockchain immutability with the EU General Data Protection Regulation and the Swiss Federal Act on Data Protection by keeping only non-identifying commitments on-chain. We report compute-cost measurements for single and batched proof verification and describe an AI-assisted filing agent that integrates with European intellectual-property offices.**

***Index Terms*: intellectual property, real-world assets, Solana, zero-knowledge proofs, Groth16, BN254, Token-2022, soul-bound tokens, verifiable custody, sponsored transactions, data protection.**

I. Introduction

Intellectual property is among the most valuable and least liquid asset classes in the modern economy. Registering, proving, licensing, and transacting rights across trademarks, patents, industrial designs, and copyrights remains slow, jurisdiction-fragmented, document-centric, and opaque. Ownership history is scattered across national registries and private files; provenance is difficult to establish; and the chain of custody over a rights record is rarely independently verifiable.

Public blockchains offer append-only, independently auditable state and are a natural substrate for provenance and custody. Yet naive tokenization of IP as freely tradeable non-fungible tokens is both legally hazardous and technically inappropriate. A transferable token that purports to represent a registered right invites securities-law ambiguity, decouples the on-chain artifact from the off-chain legal instrument, and exposes confidential filing data on a permanent public ledger.

Etornie takes a deliberately different position. Its thesis is captured by a single design goal: the intellectual-property operating system for the on-chain era. IP records are anchored on-chain as *non-transferable, verifiable attestations*, not as tradeable instruments. The on-chain layer stores only cryptographic commitments and lifecycle events; all personally identifying and commercially sensitive material stays off-chain under conventional data-protection controls. Holders can prove properties of their records, such as ownership of a specific file or the compliant provenance of an AI-assisted action, in zero knowledge, revealing nothing beyond the truth of the statement.

A. Contributions

- A canonical-hash *attestation protocol* (Section III) that binds IP case metadata and its complete lifecycle to on-chain program-derived accounts and append-only events, with init-once replay resistance.
- A *soul-bound tokenization* scheme (Section IV) built on the SPL Token-2022 standard, in which non-transferability is enforced by the token runtime itself through frozen-by-default accounts whose freeze authority is a program-derived address.
- A *zero-knowledge layer* (Section V) with three Groth16 circuits over BN254, verified on-chain through Solana pairing syscalls, supporting private proofs of file ownership and query compliance, plus a batch verifier that amortizes pairing cost across up to four proofs.
- A *verifiable sponsored-custody model* (Section VI) in which a backend operator pays fees and co-signs, and a formal instruction allowlist guarantees that co-signing cannot be abused to move operator funds, closing a hot-wallet drainage vector inherent to naive sponsorship.
- A *data-protection analysis* (Section IX) showing how on-chain immutability coexists with the right to erasure by confining on-chain state to non-identifying commitments.

II. System Model and Trust Assumptions

A. Actors

- **Client** U: the rights holder. Controls a Solana wallet w_c and signs the transactions that authorize records about their own IP.
- **Operator** $\mathcal{O}$: an Etornie backend key that funds transactions (pays network fees and rent) and co-signs sponsored transactions. The operator provides liveness and gas abstraction; it is not a custodian of the client's rights.
- **Programs**: three on-chain Solana programs, for attestation, tokenization, and zero-knowledge verification, whose bytecode is public and whose upgrade authority is intended to be held by a multi-signature.
- **IP offices**: external registries (for example the European Union Intellectual Property Office) with which the platform interoperates through authenticated adapters.

B. Trust and threat model

We assume the Solana ledger [5] provides safety and liveness under its consensus assumptions, and that BN254 pairings and the SHA-256 and Poseidon hash functions are secure. The client's wallet key is trusted by the client. The operator is trusted for *liveness* (it may refuse service) but is *not* trusted for *integrity*: the protocol is designed so that a compromised or malicious operator cannot forge a record the client did not authorize, cannot make the client's soul-bound token transferable, and cannot, through the sponsorship mechanism, be coerced into signing a transaction that spends operator or client funds outside the defined fee. Adversaries may submit arbitrary transactions to the programs and arbitrary payloads to the backend. We do not defend against a fully compromised client wallet, nor against off-chain legal disputes over the underlying rights, which remain the province of the competent registries and courts.

III. On-Chain Attestation Protocol

The attestation program anchors an IP case and its evolution to on-chain state. A case is identified by a 128-bit identifier cid.

A. Metadata commitment

Let m denote the structured metadata of a case, including its identifier, type, jurisdiction, classification, filing and deadline dates, and the client wallet. Etornie serializes m into a canonical form $\mathrm{canon}(m)$ using lexicographically sorted keys and compact separators, then commits to it with SHA-256:

$$h_m \;=\; \mathrm{SHA256}\big(\mathrm{canon}(m)\big) \in \{0,1\}^{256}. \tag{1}$$

Canonicalization ensures the commitment is independent of serialization order, so a verifier who reconstructs m can recompute and check h_m deterministically. Only h_m, never m, reaches the chain.

B. Record and derivation

The attestation record is a program-derived account (PDA) keyed by the case identifier:

$$\mathrm{PDA}_{\mathrm{att}} \;=\; \mathcal{D}\big(\texttt{"case"} \,\|\, \mathrm{cid};\; \mathsf{prog}_{\mathrm{att}}\big), \tag{2}$$

where $\mathcal{D}$ is Solana's PDA derivation. The stored record is the tuple

$$R_{\mathrm{att}} = \big(\mathrm{cid},\, h_m,\, w_{\mathrm{cr}},\, w_c,\, w_{\mathcal{O}},\, t\big), \tag{3}$$

binding the case to its metadata commitment h_m, the creator signer w_{cr}, the client wallet w_c, the operator $w_{\mathcal{O}}$, and the ledger timestamp t. The account is created with an init-once constraint, so a given cid can be attested exactly once; a replayed creation fails at the runtime level. This yields the uniqueness invariant

$$\forall\, \mathrm{cid}: \;\; \big|\{R_{\mathrm{att}} : R_{\mathrm{att}}.\mathrm{cid} = \mathrm{cid}\}\big| \le 1. \tag{4}$$

C. Lifecycle events

State transitions (for example filing, examination, grant, or closure) are recorded by an update instruction that overwrites h_m with the fresh commitment and emits a typed event

$$\mathsf{Ev} = \big(\mathrm{cid},\, h_m^{\mathrm{old}},\, h_m^{\mathrm{new}},\, \eta,\, w_{\mathrm{actor}},\, t\big), \tag{5}$$

where η is a lifecycle discriminator. Because every transition is an on-chain event, the complete, tamper-evident history of a case is reconstructable by replaying the ordered event log

$$\mathcal{T} = \big\langle \mathsf{Ev}_0, \mathsf{Ev}_1, \ldots, \mathsf{Ev}_k \big\rangle, \tag{6}$$

without exposing any case content beyond successive commitments.

IV. Soul-Bound Tokenization

Once attested, a case may be tokenized. Etornie issues a token that is *soul-bound*: bound to the client wallet at issuance and non-transferable thereafter. Transferability is not merely discouraged by convention; it is disabled by the token runtime.

A. Construction

The token is an SPL Token-2022 [6] mint with zero decimals and unit supply. Its mint and freeze authorities are a program-derived authority π, and the mint is configured with the *default account state* extension set to Frozen. Consequently, the client's associated token account a is created in the frozen state. Minting proceeds as an atomic thaw-mint-refreeze under the program authority:

$$\pi: \quad \mathsf{Thaw}(a) \;\to\; \mathsf{MintTo}(a,1) \;\to\; \mathsf{Freeze}(a). \tag{7}$$

B. Non-transferability invariant

After issuance, a is frozen and only π can thaw it. The Token-2022 runtime rejects any transfer, delegation, or burn on a frozen account. Therefore, for any transaction τ authored by the client,

$$\begin{aligned} \mathrm{state}(a) = \mathsf{Frozen} \;\wedge\; \mathsf{FreezeAuth}(a) = \pi \\ \Rightarrow\; \mathsf{Transfer}(a) \notin \tau_{\mathrm{valid}}. \end{aligned} \tag{8}$$

Burning is likewise gated: only an operator-authorized program instruction can thaw, burn one unit, and mark the record burned, with an already-burned guard preventing replay. The token thus functions as a verifiable, revocable certificate of record rather than a tradeable asset, which is the property that keeps it outside the ambit of transferable financial instruments.

V. Zero-Knowledge Proof Layer

Etornie proves properties of off-chain artifacts without revealing them, using Groth16 arguments [1] over the BN254 pairing-friendly curve [2], verified on-chain through Solana's `alt_bn128` syscalls by the Etornie Mosaic verifier.

A. Groth16 verification

For a circuit with public input vector $x = (x_1, \ldots, x_\ell)$ and proof $\pi = (A, B, C)$ with $A, C \in \mathbb{G}_1$ and $B \in \mathbb{G}_2$, verification checks the pairing equation

$$e(A, B) \;=\; e(\alpha, \beta) \cdot e\left(\sum_{i=0}^{\ell} x_i \,\mathrm{IC}_i,\ \gamma\right) \cdot e(C, \delta), \qquad (9)$$

with $x_0 = 1$ and verifying-key elements $\alpha, \beta, \gamma, \delta, \{\mathrm{IC}_i\}$. Writing the public-input linear combination as

$$L \;=\; \mathrm{IC}_0 + \sum_{i=1}^{\ell} x_i \,\mathrm{IC}_i \in \mathbb{G}_1, \qquad (10)$$

(9) is equivalent to the single multi-pairing identity

$$e(-A, B) \cdot e(\alpha, \beta) \cdot e(L, \gamma) \cdot e(C, \delta) \;=\; 1_{\mathbb{G}_T}, \qquad (11)$$

which the verifier evaluates in one batched `alt_bn128` pairing syscall. A proof is 256 bytes ($64 \,\|\, 128 \,\|\, 64$), and the verifying key is serialized canonically as $\alpha \,\|\, \beta \,\|\, \gamma \,\|\, \delta \,\|\, \{\mathrm{IC}_i\}$ with $\ell + 1$ input-commitment points.

B. Statements

Two application circuits share a commitment structure. A 256-bit content hash h (a SHA-256 digest of a file or of an AI query) is split into two 128-bit halves $h_{\mathrm{hi}}, h_{\mathrm{lo}}$, each embedded in a BN254 field element, and bound to a Poseidon [3] commitment of a secret witness s:

$$\mathcal{R}_{\mathrm{own}} = \big\{(h_{\mathrm{hi}}, h_{\mathrm{lo}}, \mathrm{cm});\ s :\ \mathrm{cm} = \mathsf{Poseidon}(s, h_{\mathrm{hi}}, h_{\mathrm{lo}})\big\}, \qquad (12)$$

$$\mathcal{R}_{\mathrm{cmp}} = \big\{(q_{\mathrm{hi}}, q_{\mathrm{lo}}, \mathrm{cm});\ s :\ \mathrm{cm} = \mathsf{Poseidon}(s, q_{\mathrm{hi}}, q_{\mathrm{lo}})\big\}. \qquad (13)$$

Relation $\mathcal{R}_{\mathrm{own}}$ is a proof of *file ownership*: the prover demonstrates knowledge of a secret s that commits to a specific file hash, without revealing s or the file. Relation $\mathcal{R}_{\mathrm{cmp}}$ is a proof of *query compliance* for AI-assisted actions, where the secret is derived from a wallet signature off-chain. Zero knowledge keeps the file, the query, and the witness private; soundness prevents a party from producing a valid commitment for content whose witness it does not know. On success the verifier initializes a record PDA keyed by $(\mathrm{domain}, w_c, h)$, giving per-holder, per-artifact replay resistance.

TABLE I
On-chain Groth16 verification cost (BN254)

Operation	Compute units
BN254 pairing check (single proof)	$\approx 83{,}500$
Single verification, end to end	$\approx 120{,}000$
Requested compute budget (single)	180,000
Batched verification, per proof (n=4)	$\approx 52{,}000$

C. Payment binding

For compliance proofs that gate a paid action, the on-chain payment is bound to the specific proof. The paying transfer carries a memo

$$\mu \;=\; \mathsf{Base58}\big(\mathrm{SHA256}(q \,\|\, \mathrm{cm})\big), \qquad (14)$$

where q is the query-hash encoding and cm the commitment. A payment is accepted only if its memo matches the proof under verification, preventing a valid payment from being reused for a different query or a valid proof from being settled by an unrelated payment.

D. Batch verification

For throughput, up to $n \in [2, 4]$ proofs are verified together by a batch verifier that aggregates their pairing checks into a single evaluation, at the cost of one field element of randomness per proof. Each batch entry is 288 bytes (256-byte proof and a 32-byte journal digest), so four entries fit within Solana's transaction size limit. The batch is keyed by

$$d = \mathrm{SHA256}\Big(\big\|_{j=1}^{n} \mathrm{SHA256}\big(\big\|_i\, x_i^{(j)}\big)\Big), \qquad (15)$$

recomputed on-chain to prevent seed spoofing. Batching reduces the marginal verification cost from approximately $83{,}500$ to approximately $52{,}000$ compute units per proof, a reduction of about 38% (Table I).

VI. Verifiable Sponsored Custody and Fee Enforcement

A central usability requirement is that clients need not hold or spend SOL to create records. Etornie therefore sponsors transactions: the operator is the fee payer, and the client contributes only the signature that authorizes an action about their own IP.

A. Co-signing

A Solana transaction has an ordered list of required signers; the first is the fee payer. In Etornie's flow the client's application assembles a versioned transaction, the client wallet signs its slot, and the backend splices the operator signature into the fee-payer slot before submission. This gives the client gas abstraction while preserving the property that the client's authorization is a first-class signature over the exact message.

B. The drainage hazard and the allowlist

Naive sponsorship is dangerous: an operator that blindly co-signs whatever a client submits can be induced to authorize instructions that spend operator funds. Concretely, an adversarial client could submit a transaction containing both the expected instruction and a system transfer $\mathcal{O} \to \mathrm{adv}$; because the operator is a signer, the transfer would execute and drain the operator's hot wallet.

Etornie closes this vector by verifying the full instruction set before co-signing. Let τ be the submitted transaction, $\mathcal{P}$ the set of program identifiers expected for the flow, Sys the system program, Φ the treasury, and ϕ the required fee. Define the safety predicate

$$\mathsf{Safe}(\tau) = \underbrace{\neg\,\mathsf{LUT}(\tau)}_{\text{no lookup tables}} \;\wedge\; \underbrace{\bigwedge_{I\in\tau} \mathrm{prog}(I) \in \mathcal{P} \cup \{\mathsf{Sys}\}}_{\text{program allowlist}} \;\wedge\; \mathsf{Fee}(\tau), \tag{16}$$

where the fee clause requires that every system instruction be exactly a transfer to the treasury sourced from a non-operator signer, and that the total so transferred meets the fee:

$$\mathsf{Fee}(\tau) = \Big(\sum_{\substack{I\in\tau,\ I=\mathsf{tr}(u\to\Phi) \\ u\in\mathsf{Signers}(\tau)\setminus\{\mathcal{O}\}}} \mathrm{lamports}(I) \;\geq\; \phi \Big). \tag{17}$$

The operator signs τ if and only if $\mathsf{Safe}(\tau)$ holds. Rejecting address lookup tables prevents accounts, including the treasury or the operator, from being resolved dynamically to evade the static check; restricting programs to $\mathcal{P} \cup \{\mathsf{Sys}\}$ blocks invocation of unexpected programs under the operator signature; and constraining every system instruction to be the user-to-treasury fee blocks the drainage transfer. Thus a compromised or adversarial client cannot spend operator funds, and the fee is non-bypassable: a client that omits the transfer fails Fee, and a client that adds a hostile instruction fails the allowlist.

C. Custody guarantee

Because the client's own signature authorizes each record, and the operator's signature is constrained by (16) to fee payment plus the expected program instruction, the operator can neither forge a record nor divert funds. The operator is reduced to a liveness and gas provider, which is the intended minimal-trust custody posture.

VII. Economic Model

Etornie's on-chain economics are denominated in SOL and collected into a treasury Φ. Table II lists the protocol fees. A registration fee is charged at attestation, a mint fee at tokenization, and a micro-fee per AI query settled through the compliance-proof and payment-binding path of Section V. All fees are enforced by the custody predicate of Section VI and are inert until a treasury is configured, so the mechanism is safe by default in development.

At launch the protocol has no native token; value accrues to the treasury in SOL per registered and tokenized asset. A native protocol token is planned to support decentralized governance and incentive alignment among filers, integrators, and the treasury. Its design, distribution, and any utility are outside the scope of this paper and will be specified separately. Nothing in this document constitutes an offer of, or solicitation for, any security or financial instrument.

TABLE II
Protocol fees

Action	Fee (SOL)	Enforcement
Registration (attestation)	0.01	bundled, allowlisted
Tokenization (mint)	0.05	bundled, allowlisted
AI query (compliance)	0.0001	payment-bound proof

VIII. AI-Assisted Filing

Etornie exposes an agentic interface, EtornieGPT, that assists rights holders through search, drafting, fee estimation, and submission. The agent operates over tool interfaces to external IP offices through authenticated adapters. At present the European Union Intellectual Property Office is integrated as an OAuth2 REST client for trademark search and electronic filing, and the United Kingdom Intellectual Property Office is integrated through a supervised browser-automation adapter that drives the official electronic form up to the payment step. Additional offices, including the World Intellectual Property Organization Global Brand Database, are planned through the same adapter abstraction. Every AI query that gates a paid action is settled through the compliance-proof mechanism of Section V, so the agent's paid actions are both metered and privately attestable.

IX. Compliance and Data Protection

Etornie processes personal and commercially sensitive data and operates from Switzerland, and is therefore designed against both the EU General Data Protection Regulation [7] and the revised Swiss Federal Act on Data Protection [8]. The central tension between these regimes and a public ledger is the right to erasure versus on-chain immutability.

Etornie resolves the tension by construction. The on-chain layer stores only *non-identifying commitments*: SHA-256 metadata hashes, Poseidon commitments, content-hash halves, and wallet public keys, none of which reveals the underlying documents, identities, or filing content. All personal and confidential material remains off-chain in conventional storage subject to access control, retention limits, and deletion. Because the on-chain commitments are one-way and content-hiding, erasing the off-chain data renders the on-chain hashes non-identifying and inert, so a data subject's erasure request can be honored off-chain without any conflict with ledger immutability. The soul-bound design further limits exposure: records are not tradeable, so IP data is never propagated through a secondary market.

X. Security Analysis

We summarize the guarantees against the threat model of Section II.

Record integrity. A record requires the client's signature; the operator cannot forge one. Attestation is init-once, so records cannot be overwritten or replayed for a given case identifier.

Non-transferability. Soul-binding is enforced by the token runtime through a frozen account under a program freeze authority, not by off-chain policy; a malicious operator cannot make a token transferable without the program authority.

Proof soundness and privacy. Groth16 soundness prevents forging proofs for statements without a valid witness; zero knowledge hides files, queries, and witnesses. Per-holder, per-artifact record PDAs and on-chain recomputation of batch digests prevent proof replay and seed spoofing.

Custody safety. The allowlist predicate (16) guarantees the operator co-signs only fee payment plus the expected program instruction, eliminating the hot-wallet drainage vector and making fees non-bypassable.

Residual assumptions. We rely on the security of BN254 pairings, SHA-256, and Poseidon; on the correctness of the deployed program bytecode, for which independent audit is planned before mainnet; and on the client protecting its own wallet key.

XI. Related Work

On-chain attestation frameworks provide general schemas for making signed claims about arbitrary subjects; Etornie specializes attestation to IP lifecycles with canonical-hash commitments and program-enforced uniqueness. Non-transferable tokens, or soul-bound tokens [4], motivate identity- and credential-bound assets; Etornie realizes the property at the token-runtime level rather than through social convention, using Token-2022 account freezing. Succinct non-interactive arguments, and Groth16 [1] in particular, underpin our proof layer, with Poseidon [3] providing an arithmetization-friendly commitment and BN254 [2] enabling native on-chain pairing verification. Relative to real-world-asset tokenization that emphasizes tradeable representations, Etornie deliberately forgoes transferability in favor of verifiable custody and privacy, which we argue is the appropriate posture for regulated intellectual-property records.

XII. Conclusion and Future Work

Etornie shows that intellectual property can be given verifiable, private, and legally coherent on-chain representation without turning rights into tradeable tokens. Canonical-hash attestation anchors provenance and lifecycle; soul-bound Token-2022 issuance binds records to their holders; a Groth16 layer proves ownership and compliance in zero knowledge; and a formally constrained sponsorship model provides gas abstraction without custodial risk, while keeping only non-identifying commitments on-chain for data-protection compliance. Future work includes independent security audit, a multi-signature upgrade authority, expansion of the IP-office adapter set, validated priority-fee support for mainnet transaction landing, and a separately specified governance token.

Disclaimer

This document is a technical description for informational purposes only. It does not constitute legal, financial, or investment advice, nor an offer or solicitation of any security or financial instrument. Nothing herein creates a representation as to the legal status of any right in any jurisdiction, which remains subject to the competent registries and courts.

References

[1] J. Groth, "On the size of pairing-based non-interactive arguments," in *Advances in Cryptology (EUROCRYPT)*, 2016, pp. 305–326.

[2] P. S. L. M. Barreto and M. Naehrig, "Pairing-friendly elliptic curves of prime order," in *Selected Areas in Cryptography (SAC)*, 2005, pp. 319–331.

[3] L. Grassi, D. Khovratovich, C. Rechberger, A. Roy, and M. Schofnegger, "Poseidon: A new hash function for zero-knowledge proof systems," in *USENIX Security Symposium*, 2021, pp. 519–535.

[4] E. G. Weyl, P. Ohlhaver, and V. Buterin, "Decentralized society: Finding Web3's soul," SSRN Working Paper 4105763, 2022.

[5] A. Yakovenko, "Solana: A new architecture for a high performance blockchain," Whitepaper, 2018.

[6] Solana Program Library, "Token-2022 program and extensions," https://spl.solana.com/token-2022, accessed 2026.

[7] European Parliament and Council, "Regulation (EU) 2016/679 (General Data Protection Regulation)," *Official Journal of the European Union*, 2016.

[8] Swiss Confederation, "Federal Act on Data Protection (revised FADP)," in force 1 September 2023.