TRELYAN

A Post-Quantum Inscription Protocol for the Algorand Era

Version 2.0 — 16 June 2026 Revision: v2.0 — adds §2.4 (Reproducibility and implementation integrity): the deterministic-Falcon build pin, the byte-identity known-answer test, the determinism run, and a resolved upstream memory-safety finding; elevates the side-channel disclosure (§7.1) with the published power-analysis result and the sign-once scoping; pins the on-chain verifier provenance (AVM v12 / consensus v41 / go-algorand v4.3.0); and tightens write-once wording. Supersedes v1.3.1, v1.3, v1.2. Status: Working draft, subject to counsel and audit review prior to any value-bearing deployment

---

Abstract

TRELYAN is a fixed-supply digital-asset protocol issued on the Algorand blockchain and designed from the ground up around post-quantum cryptography. Where existing major cryptocurrencies authenticate value transfer with elliptic-curve signature schemes whose security collapses in polynomial time under Shor's algorithm, TRELYAN is built atop one of the only major Layer-1 networks to expose Falcon-1024 — a NIST-selected lattice signature scheme, designated FN-DSA in the forthcoming FIPS 206 (in development as of mid-2026) — as a native on-chain verification primitive. The protocol, as specified for Genesis (§6), comprises three layers. **(i)** A fungible Algorand Standard Asset of 21,000,000 units, fair-launched at genesis under a zero-address minting configuration, with no pre-sale, no inflation, and no allocation preceding the public unlock; the token functions as the on-chain medium of inscription-fee settlement and is burned — transferred irrecoverably to a one-way burn-vault contract account with no withdrawal path, its address and bytecode committed at genesis — at each invocation of the inscription contract. **(ii)** A cohort of 1,024 non-fungible Algorand Standard Assets — the Vault Cells — each carrying the technical ability to inscribe one artifact of payload ≤ 4 KB onto Algorand under a Falcon-1024 signature, with verification performed by the AVM's native `falcon_verify` opcode, executed by every node during block evaluation, rather than by a hand-rolled in-contract implementation. **(iii)** A Swiss Stiftung (foundation) to be established under articles 80–89a of the Swiss Civil Code, with registration in Zug and federal supervision targeted for 2027 (in formation as of this writing; §6), designed to outlive any single contributor and to hold the protocol as endowment rather than equity. This paper specifies the cryptographic substrate, the inscription mechanics, the economic mechanism, the governance architecture, and the project's known limitations — and does so in the register of a falsifiable engineering claim rather than a marketing claim.

---

1. Context: The Cryptographic Transition

1.1 The threat model

Public-key cryptography secures the on-chain integrity of approximately three trillion United States dollars in digital value as of mid-2026. Bitcoin, Ethereum, Solana, and the majority of production Layer-

1 networks authenticate transactions with elliptic-curve digital signature schemes — ECDSA over secp256k1, Ed25519 over Curve25519, BLS over BLS12-381 — whose security depends on the hardness of the elliptic-curve discrete-logarithm problem (ECDLP) and Diffie–Hellman-type variants — all broken outright by a discrete-log-capable quantum computer. Shor's algorithm (Shor, 1994), executed on a fault-tolerant quantum computer of sufficient logical-qubit capacity, solves ECDLP in time polynomial in the bit-length of the curve order. The structural consequence is that every account whose public key is exposed on-chain becomes subject to key recovery — and hence to arbitrary signature forgery — once such a machine exists.

The resource estimate for that machine has compressed materially over the prior decade. Published resource estimates have fallen by roughly an order of magnitude over the decade (Gidney–Ekerå 2019 → Gidney 2025), and reporting on Google Quantum AI's 2026 estimates places the requirement materially below earlier planning assumptions. The direction of travel is the salient fact: not any single estimate, but the bound on a distribution of estimates, has moved.

1.2 Why most chains are structurally exposed

The vulnerability is not abstract. A quantum adversary capable of executing Shor's algorithm against an ECC-secured Layer-1 can, in principle:

- recover the private key associated with any address whose public key has been revealed on-chain — on some chains the public key is revealed the first time an address signs a spend; on others — including Algorand and Solana — it is exposed from the moment the address first appears;
- forge signatures on previously unsigned transactions, including transactions that move historical balances;
- compromise validators or block-proposers whose consensus signing keys are classical, with second-order effects on liveness and safety.

The canonical mitigation — coordinated migration to a post-quantum signature scheme — imposes severe operational cost on a chain with non-trivial historical state. It typically requires a coordinated hard fork, a synchronized key-rotation event across the user base, and a migration window during which the old and new schemes are simultaneously honored. The institutional, regulatory, and engineering surface of such a migration on Bitcoin or Ethereum is, on any honest accounting, several years of focused work — and the migration must be planned years before it is needed, since the adversary's timeline is not under the issuer's control.

1.3 Algorand's structural position

Algorand has been deploying post-quantum primitives since the 2022 State Proofs upgrade, in which compact attestations of chain state — published every 256 rounds — have been produced under Falcon signatures for nearly four years. In November 2025, the Algorand Foundation executed its first MainNet transaction authorized with Falcon-1024 signatures, verified by the native AVM `falcon_verify` opcode through a logic-signature account abstraction — to our knowledge among the first user-level post-quantum transaction authorizations on a top-tier Layer-1 (distinct from the protocol-level Falcon State Proofs running since 2022). Independent industry assessments in 2026 have repeatedly ranked Algorand among the Layer-1s best prepared for the post-quantum transition. These are not endorsements of TRELYAN; they describe the substrate conditions under which TRELYAN is possible.

The post-quantum property of the protocol is therefore inherited, not asserted: TRELYAN is quantum-resistant at the inscription-authorization layer because Algorand is quantum-resistant at that layer, and the cryptographic claim reduces to the supply-chain integrity of the Falcon-1024 specification (selected by NIST in the third round of the PQC competition; standard form FN-DSA, draft FIPS 206) and the implementation correctness of the Algorand opcode — the specification standing on nearly a decade of academic review, the opcode implementation on open-source auditability.

---

2. Cryptographic Substrate

2.1 Falcon-1024

Falcon (**F**ast-Fourier **L**attice-based **C**ompact **S**ignatures over **N**TRU) is a digital-signature scheme due to Fouque, Hoffstein, Kirchner, Lyubashevsky, Pornin, Prest, Ricosset, Seiler, Whyte, and Zhang. It was selected by the United States National Institute of Standards and Technology in the third round of the PQC competition (2022) **for standardization** as one of the post-quantum signature schemes. NIST has designated the **planned** standardized form as **FN-DSA** (FFT over NTRU-Lattice-Based Digital Signature Algorithm), to be published as **FIPS 206**; NIST advanced the draft toward approval in 2025, and the final standard remains **not yet published** as of mid-2026 (finalization expected late 2026 / early 2027). Falcon's three sibling standards — FIPS 203 (ML-KEM / Kyber, key encapsulation), FIPS 204 (ML-DSA / Dilithium, digital signature), and FIPS 205 (SLH-DSA / SPHINCS+, hash-based digital signature) — were published 13 August 2024. The scheme instantiates the Gentry–Peikert–Vaikuntanathan (GPV) hash-and-sign framework over NTRU lattices, following the Ducas–Lyubashevsky–Prest construction (2014), with the secret NTRU basis as its trapdoor and fast-Fourier sampling over that basis (Ducas–Prest, 2016) as its signing engine.

Signature generation operates over the ring $R_q = \mathbb{Z}_q[x] / (x^n+1)$ at $n = 1024$, $q = 12289$ (a small prime chosen for fast NTT). Existential unforgeability under chosen-message attack reduces, in the random-oracle model, to the NTRU and NTRU-SIS problems over the cyclotomic ring R_q — a structured-lattice family whose security stands on roughly three decades of academic cryptanalysis but which does not inherit the clean worst-case-to-average-case guarantee that the Ajtai/Regev chain provides for unstructured SIS/LWE (Ajtai 1996; Regev 2005); Falcon rests on the NTRU and NTRU-SIS assumptions directly. This reduction is non-tight and is the standard argument for *probabilistic* Falcon as submitted; the primitive deployed here is the *deterministic* `det1024` variant (§2.4), whose signing coins are SHAKE256-derived over a fixed versioned salt, and whether the EUF-CMA argument carries over unchanged to that derandomization is **not separately established in this document** — it is flagged as an open question for external cryptographic review (see §2.4 and the reviewer-facing cryptographic dossier). To the limit of public knowledge in 2026, no polynomial-time quantum algorithm is known against NTRU at the Falcon-1024 parameter set, and the cryptanalytic margin against the best lattice-reduction algorithms (BKZ-2.0 with progressive blocksize, sieve subroutines including G6K and BDGL) leaves Falcon-1024 at NIST Category V — the highest of the five NIST PQC security categories, corresponding to security at least as hard to break as an exhaustive key search on AES-256 (Category V is a comparative floor against AES-256 key search, not a literal 256-bit-security figure for Falcon-1024; the quantum security level under standard cost models is governed by the post-quantum security of the underlying lattice problem, not by Grover speedup on a key search).

Falcon-1024 signatures admit two standard encodings: a variable-length compressed encoding averaging ≈1,262 bytes, and a fixed-length padded encoding of 1,280 bytes (the public key is 1,793 bytes). The native Algorand `falcon_verify` opcode consumes the deterministic-Falcon (det1024) *compressed* encoding — a variable-length signature carrying the `0xBA` header byte, at most 1,423 bytes and typically ≈1,222 bytes — together with the 1,793-byte public key, per the AVM v12 opcode specification. The TRELYAN reference deployment (TestNet application 763809096) passes the det1024 signature through unmodified, bounding its length on-chain; verification costs 1,700 opcode-budget units — the published opcode cost (`falcon_verify` is `costly(1700)` in go-algorand `opcodes.go`, opcode `0x85`), below the 1,900 of an Ed25519 verification on the same metric. The compactness is significant: it is materially larger than Ed25519's 64 bytes, but it is the smallest of the three NIST PQC-selected signature schemes at the comparable security category. At NIST Category V (Falcon's category), the comparison is: Falcon-1024 ≈ 1,262 bytes (compressed, average) vs ML-DSA-87 (FIPS 204, the Category V Dilithium parameter set) ≈ 4,627 bytes vs SLH-DSA-SHA2-256s (FIPS 205, the Category V SPHINCS+ small parameter set) ≈ 29,792 bytes (or ≈ 49,856 bytes for the fast-signing 256f variant). Compactness is the dominant selection criterion for on-chain post-quantum verification, where signature bytes consume block bandwidth in perpetuity.

2.2 Algorand's Falcon integration

Falcon-1024 verification is exposed within the Algorand Virtual Machine (AVM) as a TEAL opcode, `falcon_verify`, callable from any smart contract or logic signature. The opcode performs the polynomial-arithmetic verification on-chain, with cost predictable in opcode-budget units and no off-chain dependency. A user can today create an Algorand account whose spending authorization is gated by a Falcon signature, providing post-quantum security at the account level without any protocol-level migration of the user's existing addresses — a property, to our knowledge, not offered natively by any other top-tier Layer-1 as of this writing, via Algorand's rekey-to-logic-signature mechanism.

This primitive is the cryptographic engine of the TRELYAN inscription contract. The contract calls `falcon_verify` against the Cell-holder's public key (bound to the Cell at issuance), the artifact-hash, and the signature; on positive verification it writes the artifact-hash into Cell state and transitions the Cell from `Sealed` to `Inscribed`. There is no off-chain attestation in the inscription path. The trust surface is exactly the protocol.

Encoding and binding. The message presented to `falcon_verify` is not the bare artifact digest but a domain-separated preimage: the ASCII tag `TRELYAN-INSCRIPTION-v1`, followed by the application ID, the Cell ID, the 32-byte artifact digest, and the network genesis hash. A valid signature therefore authorizes exactly one act — this artifact, into this Cell, under this application, on this network — which closes cross-Cell replay, cross-application replay, and TestNet→MainNet replay at the verification step itself. The sender's address is deliberately excluded from the preimage: holdership is enforced separately by the contract's ASA-ownership check, so a Cell transferred before inscription does not invalidate a holder's prepared signature. The contract bounds the operand lengths (det1024 signature ≤1,423 bytes; 1,793-byte public key) before invoking verification.

2.3 Acknowledged limitations — the honest envelope

The protocol is post-quantum at the *inscription-authorization* layer and at the State Proofs layer. It is **not yet** post-quantum at the consensus-internal layer: block-proposal signatures, the Verifiable Random Function (VRF) used in committee-sortition, and committee-vote signatures remain classical

(Ed25519, with VRF over Curve25519) as of mid-2026. The Algorand Foundation has publicly committed these primitives to the next major upgrade window; independent 2026 assessments identify the gap in the same terms. Until the upgrade lands, a sufficiently capable quantum adversary could in principle target the consensus layer even while individual user transactions remain secure under Falcon.

TRELYAN inherits the limitation and discloses it without qualification. The protocol does not claim total quantum resistance. It claims post-quantum authorization at the inscription layer — a defensible and meaningful advance over peer chains, supported by the documents of record and verifiable in the contract bytecode — and it commits in writing not to permit any communication of the project to elide the distinction.

2.4 Reproducibility and implementation integrity

A protocol whose thesis is the trustworthiness of a signature must be able to show that the signer it ships is the signer it describes — byte for byte, on someone else's machine. TRELYAN treats this as a falsifiable engineering claim, not an assurance.

Deterministic signing, and why it is delicate. The reference signer uses the *deterministic* Falcon-1024 variant (`falcon_det1024`): the per-signature randomness (the Gaussian sampler's coins) is derived as `SHAKE256(logn ‖ private-key ‖ message)` and the salt is a fixed versioned constant, so a given key and message yield exactly one signature. This is a deliberate **deviation** from reference Falcon and from the forthcoming standard: draft **FIPS 206 (FN-DSA) mandates *randomized* signing and flags deterministic signing as dangerous** — exactly because a floating-point implementation mistake could produce two different signatures for one hash. TRELYAN adopts deterministic `det1024` because Algorand's `falcon_verify` opcode verifies precisely that variant, and the pinned fixed-point backend, byte-identity KAT, and sign-once lifecycle below are the controls that prevent the specific failure NIST names — two different signatures for one hash — in this pinned build (the byte-identity KAT demonstrates it). We make no FIPS-206-compliance claim; the broader security of deterministic signing stands as the open question flagged below. Determinism is load-bearing rather than cosmetic — Falcon's signing draws discrete-Gaussian samples in floating point, and floating-point non-determinism across compilers and platforms is the well-known portability hazard the field has documented since Falcon's submission. The pinned build neutralises it by selecting Falcon's **emulated fixed-point backend** (`FALCON_FPEMU=1` , `FALCON_FPNATIVE=0`): integer-only arithmetic that is bit-exact across toolchains. A clarifying point of scope: Falcon's Gaussian sampling is precisely what makes an *honest* signature reveal nothing about the secret basis — the historical defence against the transcript cryptanalysis that broke earlier deterministic-rounding lattice signatures — so determinism here is about reproducibility of an already-hiding signature, not about hardening a leaky one. (Whether this SHAKE256 derandomization preserves Falcon's EUF-CMA argument — which is proved for *probabilistic* signing — is a distinct, open question, flagged for external review in §2.1 and the cryptographic dossier; determinism's effect on the *transcript-leakage* posture and its effect on the *security reduction* are separate matters and we keep them separate.)

What is pinned, and how anyone can check it. The signer is built from a frozen upstream commit (`algorand/falcon` `ce15e75b…`), recorded by a source-tree digest (`sha512_256 = c6adf487…` , 27 files) and a `deterministic.c` digest (`601390dc…`); a CI step recomputes both and fails the build on any drift. A committed **known-answer test (KAT)** fixes `(key, message) → signature` golden vectors and asserts that the rebuilt signer reproduces them **byte-for-byte** (with a one-byte-flip negative control), and a 100,000-message run comparing the `-O0` and `-O3` builds produced **zero** mismatches. These are

regression controls for *this pinned build* — evidence of reproducibility, not a general Falcon portability theorem — and cross-operating-system byte-identity is **confirmed** by a three-OS continuous-integration job that rebuilds the pinned signer and reproduces the golden signatures byte-for-byte on Linux (gcc), macOS (clang), and Windows (MSVC). (Cross-*endianness* remains argued by construction — the CI runners are all little-endian x86-64.)

Memory-safety, found and closed. A reviewer who runs a sanitizer over the signer's C will encounter a misaligned 64-bit load in the upstream pseudo-random generator — an *optional* fast path for which the upstream code already ships a portable equivalent. It is a portability / memory-safety matter, **not** a cryptographic weakness: it is closed by a build-flag selection (`-DFALCON_UNALIGNED=0 -fno-strict-aliasing`) that needs no source edit, leaves the pinned digest unchanged, and is proven byte-identical to the golden signatures and clean under the alignment and undefined-behaviour sanitizers across key generation, signing, and verification. A continuous-integration gate enforces it.

The point is not that these controls are exotic; it is that they are **executable**. A reviewer need not trust the description — the digests, the known-answer test, and the sanitizer gate are public and re-runnable, and the full per-claim cryptographic analysis, with its honest "what this does not prove" envelope, is set out in the project's reviewer-facing cryptographic dossier.

---

3. The TRELYAN Token

3.1 Fixed parameters

Parameter	Value
Token name	TRELYAN
Symbol	TRELYAN
Decimals	6
Total supply	21,000,000 (fixed at genesis)
Issuance	Single ASA-create transaction at genesis; manager / clawback / freeze / reserve fields set to the zero-address
Inflation	None — no minting authority survives the genesis transaction
Standard	Algorand Standard Asset (ASA)
Primary function	Burned as the inscription-fee unit at each call to `inscribe()` on the inscription contract

The 21,000,000 figure is not arbitrary. It is a deliberate register choice: the cardinality popularized by Bitcoin as the convention for fixed supply — and the choice fixes TRELYAN as a fixed-supply settlement unit for inscription fees, rather than as a programmatic-emission utility token whose value rests on a yield curve. The selection is a register choice, not a derivation.

3.2 Distribution

The TRELYAN token is distributed via fair-launch liquidity bootstrapping on Algorand-native automated market makers at genesis. There is no pre-sale of the fungible token. There is no venture allocation. There is no advisor allocation that precedes the public unlock. There is no founder allocation that unlocks before any community-allocation unlock.

Allocation	Share	Vesting / discipline
Fair-launch liquidity	60 %	Open at genesis; bootstrapped via LBP price-discovery
Community treasury (Stiftung)	20 %	4-year linear, governed by Cell-holder treasury vote
Builder grants	15 %	4-year linear, allocated by quorum-weighted treasury proposal
Founder	5 %	5-year linear with a contractually-binding precondition that no founder-allocation unlock precedes any community-allocation unlock

The founder allocation is deliberately small, back-weighted, and conditional. The structure encodes an economic incentive identical to a long-term Cell-holder's: the founder's outcome on the protocol is identical to the cohort's outcome on the protocol, by design.

3.3 Supply mechanics after genesis

The token is non-inflationary by construction — the minting authority is destroyed at genesis, and no governance pathway exists by which it can be restored. Two mechanisms tighten effective circulating supply monotonically over time:

1. **Inscription burn.** Each invocation of `inscribe()` on the inscription contract consumes a fixed quantity of TRELYAN (parameterized at genesis). The consumed tokens are transferred to a one-way burn-vault: an application account that opts in to the asset at genesis and whose approval program contains no transfer, close-out, or update path. The vault's address and program bytecode are committed at genesis, so the burn is verifiable by any indexer and irrecoverable by construction (Algorand Standard Assets cannot be sent to the zero address, which cannot execute the required opt-in). As inscription demand accumulates, circulating supply contracts in a manner observable by any indexer in real time.
2. **Treasury sequestration.** The Stiftung's 20 % community-treasury allocation is held in a multi-signature configuration with the 4-year vest schedule binding the rate at which any of it can re-enter circulation. During the formative-years window, the treasury is, in effect, removed from circulating supply.

No mechanism exists by which supply can ever increase. This is a property of the ASA configuration, not of an undertaking — and is verifiable by reading the asset's on-chain parameters directly.

---

4. The Vault Cells

4.1 Concept

The Vault Cells are 1,024 non-fungible Algorand Standard Assets, each carrying exactly one substantive function: the technical ability to inscribe one artifact of payload ≤ 4 KB onto Algorand under a Falcon-1024 signature whose verification is performed on-chain by the native TEAL `falcon_verify` opcode. The cardinality 1,024 references the ring degree of the signature scheme (Falcon-n at n = 1024, the dimension of $\mathbb{Z}_q[x]/(x^n+1)$) — and bounds the Vault above at the population where it remains a curated archive rather than a marketplace.

Each Cell is sovereign with respect to its inscription: only its owner can perform the inscription, and once the inscription is committed on-chain, no party — not the holder, not the foundation, not the Stiftungsrat, not the founder, not any subsequent transferee — can modify, revoke, or overwrite it. The inscription contract has no pathway to revision. The foundation has no pathway to revision. This write-once property is enforced by the substrate, not represented by the issuer.

4.2 The inscription payload and what lives on-chain

The inscription mechanism writes the artifact itself directly to on-chain box storage, indexed by Cell ID. The on-chain record per inscribed Cell is the tuple `{artifact_hash, artifact_blob, falcon_pubkey, inscribed_at_round, inscriber_address}` — written into the box keyed `cell_{cell_id}` — together with the Falcon-1024 signature accepted by the contract's `falcon_verify` call. The artifact payload is bounded above at 4,096 bytes (4 KB) per Cell to keep the per-inscription block-bandwidth cost predictable, to keep the verification call within the contract's opcode budget, and to keep the long-tail storage cost bounded by the chain's box-storage economic model.

Artifacts in scope at the 4 KB ceiling include: short-form prose, structured records, key-attestation payloads, public-key bundles, hash-attestations of larger off-chain artifacts (where the off-chain content is too large to fit on-chain), poems, will-fragments, scientific dataset hashes, blueprints expressed as terse formal specifications, deed references, and the Cell-holder's own public-key material.

For artifacts that exceed the 4 KB ceiling, holders may inscribe a `multihash` (the IPLD multi-codec content-addressed identifier) referencing content held on a content-addressed storage layer (IPFS, with contracted pinning) or a durability-oriented layer (Arweave); the on-chain commitment attaches to the hash, not to off-chain availability. In that configuration, the on-chain artifact is the multihash (encoding a SHA-256 digest) and the Falcon attestation; the larger content lives off-chain but is bound to the Cell by a cryptographic commitment that no future actor — including the foundation — can alter without finding a SHA-256 second preimage (and a colluding inscriber, who controls the artifact before commitment, would need a prepared collision pair — the reason collision resistance, not merely second-preimage resistance, is required); neither attack is known to be feasible, classically or quantumly, at SHA-256's security margins.

4.3 The inscription contract — end-to-end flow

The inscription is mediated by an Algorand stateful smart contract written in Algorand Python (algopy), targeting the AVM version in which `falcon_verify` was activated (AVM v12 / consensus v41, go-algorand v4.3.0) or later. The flow:

1. The Cell-holder generates a Falcon-1024 keypair locally; the private key never leaves the holder's possession and is never escrowed, including by the foundation. The corresponding public key is registered to the Cell via the contract's `register_pubkey()` call at Cell-acquisition time.
2. The holder composes the artifact bytes (≤ 4 KB on-chain, or any size off-chain with a multihash committed on-chain).
3. The holder signs, under the Falcon-1024 private key, the domain-separated preimage `"TRELYAN-INSCRIPTION-v1" || app_id || cell_id || artifact_hash || genesis_hash`, where `artifact_hash` is the SHA-512/256 digest of the artifact payload (see § 2.2, *Encoding and binding*). The signature, the public key, the artifact hash (together with the artifact blob where it is carried on-chain), and the Cell ID form the inscription transaction. A holder who loses the Falcon private key after registration but before inscription cannot inscribe that Cell; a key-rotation path prior to inscription is a designated audit item (§6).
4. The transaction calls `inscribe(cell_id, artifact_hash, falcon_pubkey, falcon_signature, artifact_blob)` on the inscription contract.
5. The contract asserts: (a) Cell is in `Sealed` state, (b) sender holds the Cell ASA, (c) the registered public key for the Cell matches `falcon_pubkey`, (d) `sha512_256(artifact_blob) == artifact_hash` where the blob is supplied, (e) the inscription-fee burn is present as an inner transaction.
6. The contract calls `falcon_verify(message, falcon_signature, falcon_pubkey)` over the domain-separated preimage — verification is performed by the AVM's native opcode, executed by every node during block evaluation, not by a hand-rolled in-contract implementation.
7. On positive verification the contract writes the box `cell_{cell_id}` and transitions Cell state `Sealed → Inscribed`, emits the `inscription_v1(cell_id, artifact_hash, inscriber_address, round)` event, and the inscription is final at deterministic 2.8-second finality.

After confirmation, the inscription is write-once and tamper-evident for the lifetime of the Algorand network. The trust surface of the inscription is exactly the trust surface of `falcon_verify` plus the contract bytecode — both publicly auditable, to be audited by an independent firm before any value-bearing deployment (a precondition of Genesis, §6).

Specification note. The flow above describes the genesis-target design. The TestNet reference implementation (application ID 763809096) implements a variant pending audit: the full 1,793-byte Falcon public key is registered into the Cell's box at registration time (working around the AVM's 2,048-byte argument ceiling); digests are SHA-512/256, the AVM's native hash; the artifact payload is held off-chain in all cases, with only the hash commitment, the registered key, and metadata stored in the Cell's box; and the inscription-fee transfer is grouped with the application call rather than issued as an inner transaction. A registered key cannot currently be rotated before inscription — key loss before inscription leaves the Cell uninscribable — and a guarded re-registration path is a designated deliverable of the external review (§6), alongside reconciliation of the remaining deltas enumerated in the repository.

4.4 Cell #1 — the genesis inscription

Cell #1 is reserved by the foundation, not sold. It is inscribed at genesis in a public ceremony, with a fresh Falcon-1024 keypair generated live from a hardware-true RNG visible to attendees, and with the inscription text (the project's canonical self-description; see the *Cell #1 Inscription Manifesto*) committed to an IPFS multihash whose CID is itself recorded on-chain. The Cell-#1 inscription is the protocol's genesis artifact and its canonical on-chain origin record.

The Cell-#1 transaction is the genesis end-to-end exercise of the inscription contract on Algorand, performed in public, on the network users will be asked to use — a deliberate falsifiability test of the protocol's operational claim.

4.5 Secondary market and transferability

Cells are freely transferable as standard Algorand NFTs subject to two cohort-specific rules. **Pre-sale cohort (Cells #2 → #301):** Cells are non-transferable until *both* of two cumulative conditions are satisfied — (a) the holder has inscribed the Cell, *and* (b) 24 months have elapsed from the genesis round. The conjunction is enforced in the inscription contract, not by undertaking. Cells in this cohort that satisfy the conjunction transition into the `Released` state, in which they become freely transferable while retaining their inscription as a write-once record. **Genesis-day cohort (Cells #302 → #1024):** Cells carry no transferability lock and are tradeable from issuance, subject only to the inscription mechanism. **Cell #1:** non-transferable by foundation reserve.

The Cell state machine therefore admits four states: `Sealed` (pre-inscription), `Inscribed` (post-inscription), `Released` (post-inscription and post-lock for the pre-sale cohort), and `Reserved` (Cell #1 only; non-transferable, terminal). All state transitions are deterministic functions of on-chain conditions; the foundation has no authority to advance or rewind a Cell's state.

An inscribed Cell carries its inscription with it durably; subsequent custodians inherit the Cell and inherit the inscription but cannot modify it. The inscription is bound to the Cell, not to its owner.

---

5. Governance

5.1 The Foundation

TRELYAN will be steward-operated by the **TRELYAN Foundation**, a Stiftung in formation, to be incorporated under the Swiss Civil Code, articles 80–89a, with registration in Zug under the supervision of the Eidgenössische Stiftungsaufsicht (see §6). The Stiftung's *Stiftungsurkunde* (foundation deed) binds the entity to its purpose — the perpetual stewardship of the TRELYAN protocol and the integrity of the inscriptions written to it — and that purpose is irrevocable as a matter of Swiss civil law: the Stiftung cannot pivot to a different purpose, cannot be dissolved by its founder, and cannot redirect its assets to beneficiaries outside its purpose-defined class. The Stiftungsrat (foundation board) is composed of three to five members under quorum rules specified in the foundation deed.

The Foundation holds the community treasury, deploys upgrades to *ancillary* infrastructure (gateway tooling, indexer code, the Council operational stack, the public site), and represents the project to regulators, partners, and counsel. The Foundation does **not** hold authority over the asset itself: the TRELYAN ASA is fixed at genesis and the inscription contract is non-upgradeable with respect to any state-bearing function that touches existing inscriptions.

The foundation deed, by binding charter, prohibits the Foundation from:

- Issuing new TRELYAN tokens (technically impossible after genesis; reaffirmed here as charter discipline);
- Modifying or upgrading the deployed inscription contract in any manner that affects, mutates, or revokes existing inscriptions or existing Cell-state transitions;

- Selling, lending, or otherwise encumbering treasury holdings outside the disclosed vesting schedule;
- Acting as a counterparty or market-maker for the TRELYAN token on any venue;
- Holding keys outside a multi-signature configuration whose composition is disclosed in the Treasury Custody Architecture document.

The Stiftungsrat is accountable to the Swiss supervisory authority and not to a discretionary equity-holder class, because no such class exists.

5.2 Cell-holder governance

Substantive treasury allocations, grant disbursements, parameter adjustments to ancillary infrastructure, and any amendment to non-charter operating policy are conducted by on-chain vote of the Cell-holder cohort. The voting architecture:

- **Eligibility.** One Cell, one vote. Cells in the `Released` state and `Inscribed` state vote at full weight; `Sealed` Cells vote at half weight (the inscription is the act that converts the Cell from instrument to standing).
- **Quorum.** One-third of outstanding Cells (342 of 1,024) must vote for a proposal to reach quorum.
- **Approval threshold.** Simple majority of votes cast for ordinary proposals; supermajority (two-thirds of votes cast, with the same one-third quorum) for any proposal touching the foundation's charter-restricted set.
- **Cycle.** Proposals are posted to the on-chain governance contract with a seven-day discussion window, a seven-day voting window, and a seven-day implementation buffer; the buffer exists to allow the Stiftungsrat to escalate to counsel if the proposal carries regulatory implications it must surface before execution.
- **Implementation.** On-chain execution is performed via the governance contract directly where possible; off-chain execution (treasury wire, partner engagement) is performed by the Foundation under quarterly public reporting.

Routine operational decisions — those that fall below the materiality threshold codified in the deed — are made by the Foundation directly under the Stiftungsrat's standing authority, with public quarterly disclosure of every such decision.

5.3 The Council

Day-to-day operations — research, drafting, on-chain monitoring, narrative cohesion, threat detection, archival memory — are augmented by the **AI Intelligence Council**, a three-seat configuration of frontier large-language-model agents drawn from structurally independent training lineages so that cross-seat triangulation is not a shared-training artifact. The seats are: **Seat I — Strategy**, Claude Opus (Anthropic), responsible for protocol exposition, communication, and brand register; **Seat II — Diligence**, Llama 3.3 70B hosted on IBM watsonx (us-south region), responsible for internal due diligence on the Foundation itself, on prospective counterparties, and on any plan or claim Seat I produces; **Seat III — Independent**, Mistral Large (Mistral AI, EU data residency), responsible for lineage-independent triangulation, multilingual review across EN/FR/IT/DE/ES, and European regulatory framing under FINMA, ESMA, GDPR, and the EU AI Act. The Council drafts; humans publish. The Council reviews; humans decide. The Council does not hold private keys, does not sign transactions, does not make commitments on behalf of the Foundation, and does not have any pathway to bind the foundation to any obligation. The Council's operating charter is subject to Stiftungsrat

ratification once the Stiftung is constituted, with public quarterly disclosure of every substantive decision the Council has informed.

The Council is a means, not a principal. Its architecture, its model-tier assignments, its escalation rules, its failure modes, and its cost envelope are documented in the Council brief and are subject to Stiftungsrat oversight on the same cadence as any other operational tool the Foundation employs.

---

6. Roadmap

The roadmap is expressed as a sequence of falsifiable conditions, not a calendar. Each phase has an entry condition and an exit condition; the project does not pass to the next phase until the exit condition is satisfied. Calendar windows are nominal anchors derived from current best estimates and are subject to revision against the entry/exit conditions, which are the binding artifacts.

Phase	Anchor window	Entry condition	Exit condition
Pre-Foundation	Q2–Q4 2026	Founder commitment letter signed in full (Sections 1, 2, 3)	Phase 1 legal opinion received from Swiss counsel (engagement in selection); Stiftung formation funded
Foundation	Q1–Q3 2027	Stiftung formation capital landed (Accelerator + F&F bridge)	Stiftung registered with the Handelsregister Kanton Zug; Stiftungsrat seated; counsel of record retained
Contracts	Q3 2027	Stiftung registered; inscription-contract draft v1.0 complete	Inscription contract audited by an independent firm; audit report published; public bug-bounty programme opened
Pre-sale	Q4 2027	Audit clean; legal-viability stop condition (month-18) cleared	300 Cells (Cells #2–#301) sold to KYC-verified buyers under Reg D 506(c) + Reg S
Genesis	Q2 2028	Pre-sale capital settled; final audited contracts deployed to Algorand	Cell #1 inscription transaction confirmed on-chain; fair-launch liquidity opens on Algorand AMMs
Establishment	Q3–Q4 2028	Genesis transaction confirmed	First Cell-holder governance vote completes quorum; first quarterly Foundation report published
Maturation	2029 onward	Establishment phase exit	Pre-sale cohort enters `Released` state at T+24 months; Algorand consensus-layer PQ upgrade integrated as it lands

The project does not commit to specific calendar dates. Premature deployment of a post-quantum asset whose audit envelope is incomplete would undermine the integrity claim the protocol exists to make. The entry/exit conditions are the contract; the calendar is the projection.

Reproducibility. The reference inscription contract compiles and verifies against the live `falcon_verify` opcode on Algorand TestNet (application ID 763809096), with a public test suite

covering the full `Sealed` → `Inscribed` → `Released` lifecycle. The deployment and its test vectors are reproducible from the project repository.

---

7. Risks and Limitations

This section is unusually long for a token whitepaper. That is intentional. A project whose central thesis is the trustworthiness of cryptography has an affirmative obligation to enumerate every condition under which that trustworthiness could fail — and to do so in the register of an academic disclosure, not the register of a marketing disclaimer. The risks below are not a fence against liability; they are the complete envelope of what the project does not control.

7.1 Cryptographic risks

- **Falcon-1024 lattice-hardness assumption.** Falcon's existential-unforgeability reduces, in the random-oracle model, to the hardness of the NTRU and NTRU-SIS problems over the cyclotomic ring. This structured-lattice setting does **not** inherit the clean worst-case-to-average-case guarantee that the Ajtai/Regev chain provides for *unstructured* SIS/LWE; Falcon rests on the NTRU and NTRU-SIS assumptions directly. No signature scheme in cryptographic history has been proven unconditionally secure; lattice cryptography in particular has had its security margins narrowed by improvements in BKZ-2.0, sieving (G6K, BDGL), and structured-lattice cryptanalysis. A novel algorithmic advance against any of these — or a quantum algorithm specific to NTRU structure — would compromise TRELYAN's central premise.
- **Implementation bugs in `falcon_verify`.** The Algorand Falcon implementation is open-source and has been independently reviewed, but consensus-layer software bugs in the verification primitive could in principle accept forged signatures. The mitigation is the open-source verifiability of the opcode and the existence of independent re-implementations against which the Algorand version can be tested.
- **Side-channel exposure at signing time.** The published key-recovery attacks on Falcon target the *signing* operation and are **multi-signature**: electromagnetic recovery from the key-dependent FFT multiplications (Karabulut–Aysu, DAC 2021) and power-analysis of the discrete-Gaussian sampler (Guerreau–Martinelli–Ricosset–Rossi, TCHES 2022, with its 2023 and 2025 follow-ups including *Thorough Power Analysis on Falcon Gaussian Samplers*, PKC 2025), building on the constant-time-sampling line (Howe–Prest–Ricosset–Rossi, 2020), all reconstruct the secret by *learning the parallelepiped* from leakage across **many** signatures of one key. TRELYAN's reference signer narrows this surface structurally — it signs **once per key, offline, in an isolated short-lived process, then destroys the key** (there is no online signing oracle, and the on-chain verification path exposes no secret) — so the multi-signature attacks do not apply. What remains — which we **flag rather than defend** — is single-trace leakage of the unmasked software sampler *during that one signing* on a compromised endpoint; and because the deterministic variant derives the sampler's coins from the key, that one execution is itself key-sensitive. Holders securing significant value should sign on constant-time, hardware-isolated, ideally masked implementations. This surface is treated per-claim in the project's cryptographic dossier.
- **Signer reproducibility and implementation integrity.** Deterministic Falcon signing is byte-reproducible only under a frozen build (the emulated fixed-point backend; § 2.4). A determinism regression — native floating point, unsafe optimization, or upstream drift — would be a correctness

failure and, in the limit, a forgery risk, which is why the pinned source digests, the byte-identity known-answer test, the 100,000-message determinism run, and the alignment / undefined-behaviour sanitizer gate are continuous-integration controls rather than prose. A reviewer can re-run every one of them (§ 2.4).

7.2 Substrate risks

- **Algorand consensus is not yet fully post-quantum.** As stated in § 2.3, block-proposal signatures, the consensus-sortition VRF, and committee-vote signatures remain classical (Ed25519, Curve25519-VRF) as of mid-2026. TRELYAN inherits this exposure: a quantum adversary with sufficient logical-qubit capacity could in principle attack the consensus layer even while user-level Falcon signatures remain secure. The Algorand Foundation has publicly committed these primitives to the next major upgrade window; TRELYAN's risk profile improves materially when that upgrade ships.
- **Substrate continuity risk.** TRELYAN cannot survive the catastrophic failure of the underlying chain. A material technical, governance, or community failure on Algorand propagates directly to TRELYAN. Diversification across chains is incompatible with the protocol's substrate-dependency claim and is therefore not a mitigation we will adopt.

7.3 Operational risks

- **Treasury key management.** The Stiftung's TRELYAN-treasury keys and operational signing material are a critical dependency. Compromise would not affect the integrity of already-issued tokens or already-inscribed Cells (both are write-once under the inscription contract), but would expose the treasury holdings and the founder-allocation vesting schedule. The mitigation is a multi-signature configuration with hardware-isolated key material distributed across Stiftungsrat members and jurisdictions, and is specified in the Treasury Custody Architecture document.
- **Single-jurisdiction concentration.** The Swiss Stiftung structure is standard for token foundations and has been stress-tested through five years of Swiss DLT-Act practice and nearly a decade of FINMA token-classification practice; nonetheless, the project's legal-personhood is concentrated in one jurisdiction and is exposed to Swiss regulatory and legislative regime change. The mitigation is the legal-viability stop condition in Section 2 of the founder's commitment letter, which terminates the project at month 18 if Swiss formation has not closed under the conditions originally specified.
- **Key-person risk.** Small teams carry key-person risk. The Foundation charter, the autonomous Council architecture, and the irrevocable nature of the inscription contract are all designed to bound the blast radius of any single contributor's exit, but cannot eliminate the risk that the founder's departure materially affects execution quality.

7.4 Regulatory risks

- **Securities classification — Swiss.** The TRELYAN token is structured to satisfy FINMA's Zahlungstoken criteria (no issuer claim, no fundraising event, endogenous network use, decentralized issuance); the Vault Cells are structured to satisfy Nutzungstoken criteria with three concurrent features designed to suppress Anlagetoken re-characterization (utility-at-issuance, conjunctive transferability lock, per-buyer cap). FINMA's classification is, however, supervisory and reserved to the regulator; we cannot determine it ourselves. The Token Classification Memo prepared for counsel review is the project's working position, not a legal conclusion.

- **Securities classification — United States.** Pre-sale Cells offered to US-resident accredited investors are offered under Regulation D 506(c) with verified accreditation; non-US persons are offered under Regulation S safe harbor from US registration. The classification of the fungible TRELYAN token under US securities law is a separate question; the project relies on US securities counsel for the cross-jurisdictional analysis.
- **MiCA classification — European Union.** Under the Markets in Crypto-Assets Regulation (2023/1114), TRELYAN's posture is that the token is "neither an asset-referenced token (ART) nor an e-money token (EMT)" — that is, a crypto-asset falling into the residual category whose distribution to EU-resident purchasers may require a Crypto-Asset White Paper under MiCA art. 6, notified under art. 8. The project will publish the MiCA White Paper to the relevant home-Member-State competent authority prior to any post-genesis distribution that meets the MiCA offering thresholds.
- **Jurisdictional restrictions.** TRELYAN will not be offered for sale in jurisdictions where doing so would require registrations the Foundation has not obtained. The current restricted-jurisdictions list is published and maintained on the project website and is incorporated by reference into the pre-sale terms.

7.5 Market risks

- **Volatility.** All cryptocurrencies are volatile, and a fair-launched asset with no foundation-sponsored market-maker is structurally more so. Holders should expect significant price variance and should not infer a price floor from any historical trade.
- **Liquidity.** A fixed-supply asset with a 60 % fair-launch allocation may exhibit limited secondary-market depth in the bootstrapping window. Listed venues are not a substitute for liquidity; the absence of market-making is a deliberate structural choice and a known cost.
- **Narrative durability.** The post-quantum thesis is robust if Q-Day materializes within the window the field currently estimates. If quantum-computing progress stalls materially — that is, if the resource-estimate distribution moves *back* — the urgency of the thesis weakens, even though the structural correctness of the protocol does not. The project does not commit to a specific Q-Day estimate; we rely on the consensus of the standardization community as it evolves.

7.6 What this paper does not promise

This paper does not promise a price, a return, an exchange listing, a partnership, a Q-Day date, a regulatory outcome, or a calendar-bound milestone whose timing depends on parties outside the foundation. It describes a design, a substrate, a structure, and a set of falsifiable claims. Outcomes depend on execution, on market conditions, on regulatory developments, and on the choices of Cell-holders. Prospective participants should treat TRELYAN as a high-risk asset, should not hold more of it than they can afford to lose entirely, and should consult their own legal and tax counsel before any acquisition.

---

8. References

1. Algorand Foundation, *Technical Brief: Quantum-resistant transactions on Algorand with Falcon signatures*, Developer Blog Series, 3 November 2025. https://algorand.co/blog/technical-brief-quantum-resistant-transactions-on-algorand-with-falcon-signatures

2. Algorand Foundation, *Algorand Post-Quantum Ledger*, 2026. https://algorand.co/blog/algorand-post-quantum-ledger
3. NIST, FN-DSA (Falcon) — selected for standardization, third round of the NIST PQC process; draft FIPS 206 in development (not yet published as of mid-2026). See NIST IR 8413.
4. National Institute of Standards and Technology, *FIPS 205: Stateless Hash-Based Digital Signature Standard (SLH-DSA / SPHINCS+)*, published 13 August 2024.
5. National Institute of Standards and Technology, *FIPS 204: Module-Lattice-Based Digital Signature Standard (ML-DSA / Dilithium)*, published 13 August 2024.
6. National Institute of Standards and Technology, *SP 800-208: Recommendation for Stateful Hash-Based Signature Schemes (XMSS / LMS)*, October 2020.
7. Fouque, P.-A., Hoffstein, J., Kirchner, P., Lyubashevsky, V., Pornin, T., Prest, T., Ricosset, T., Seiler, G., Whyte, W., Zhang, Z., *Falcon: Fast-Fourier Lattice-based Compact Signatures over NTRU*, NIST PQC Submission, Round 3 Final Specification, 2020.
8. Gentry, C., Peikert, C., Vaikuntanathan, V., *Trapdoors for Hard Lattices and New Cryptographic Constructions*, STOC 2008.
9. Ducas, L., Lyubashevsky, V., Prest, T., *Efficient Identity-Based Encryption over NTRU Lattices*, ASIACRYPT 2014.
10. Ducas, L., Prest, T., *Fast Fourier Orthogonalization*, ISSAC 2016.
11. Howe, J., Prest, T., Ricosset, T., Rossi, M., *Isochronous Gaussian Sampling: From Inception to Implementation*, PQCrypto 2020.
12. Regev, O., *On Lattices, Learning with Errors, Random Linear Codes, and Cryptography*, STOC 2005.
13. Ajtai, M., *Generating Hard Instances of Lattice Problems*, STOC 1996.
14. Shor, P. W., *Algorithms for Quantum Computation: Discrete Logarithms and Factoring*, FOCS 1994.
15. Gidney, C., Ekerå, M., *How to factor 2048 bit RSA integers in 8 hours using 20 million noisy qubits*, Quantum 5, 433 (2021; arXiv:1905.09749, 2019). See also Gidney, C., *How to factor 2048 bit RSA integers with less than a million noisy qubits*, arXiv:2505.15917, 2025.
16. Chen, J., Micali, S., *Algorand: A secure and efficient distributed ledger*, Theoretical Computer Science 777, pp. 155-183, 2019 (extended journal version of the earlier *Algorand* preprint, arXiv:1607.01341, 2016). See also Gilad, Y., Hemo, R., Micali, S., Vlachos, G., Zeldovich, N., *Algorand: Scaling Byzantine Agreements for Cryptocurrencies*, SOSP 2017.
17. Voshmgir, S., *Token Economy: How the Web3 Reinvents the Internet*, BlockchainHub Berlin, 2nd ed., 2020.
18. Guerreau, M., Martinelli, A., Ricosset, T., Rossi, M., *The Hidden Parallelepiped Is Back Again: Power Analysis Attacks on Falcon*, IACR Transactions on Cryptographic Hardware and Embedded Systems (TCHES) 2022(3); presented at CHES 2022.
19. Karabulut, E., Aysu, A., *Breaking FALCON Post-Quantum Signature Scheme through Side-Channel Attacks*, 58th ACM/IEEE Design Automation Conference (DAC), 2021, DOI 10.1109/DAC18074.2021.9586131 (the NIST PQC 2022 version is titled *Falcon Down*).
20. Nguyen, P. Q., Regev, O., *Learning a Parallelepiped: Cryptanalysis of GGH and NTRU Signatures*, EUROCRYPT 2006, LNCS 4004 (Journal of Cryptology version following).

21. Algorand / go-algorand, release **v4.3.0** (consensus **v41**) enabling the `falcon_verify` AVM opcode at **AVM v12** (AVM PR #6416, "Prepare for falcon_verify in v12"); go-algorand release notes.

This document is a draft for public review. It is not a solicitation, an offer, or a prospectus. TRELYAN has not been launched at the time of publication. Token availability, pricing, and timing are not specified and are subject to change.