

# Silvetus

AT - 251

*A permissionless registry of long-lived trees, with token mechanics that align financial self-interest with ecological survival.*

VERSION 0.4 · BASE MAINNET MAY 2026 · silvetus.com

## Summary

Silvetus is a permissionless registry of long-lived trees. Each tree is represented by a non-fungible token whose continued transferability depends on the continued life of the tree. When a tree dies, its token freezes, permanently non-transferable, permanently non-recoverable. Eleven genesis trees are deployed on Base mainnet under contract designation AT-251. The contract, the operational architecture, the phase roadmap, and the rules under which the protocol commits to operate are described below.

## Abstract

Conservation has been pursued, for a century, as a problem of political will. Political will tires. The trees do not become less alive when public attention turns away. They become more vulnerable, because the financial returns to cutting them down accumulate continuously while the resources marshalled to protect them oscillate.

Silvetus proposes an additional mechanism, not a replacement: a permissionless registry of long-lived trees, in which each tree is represented by a non-fungible token whose continued transferability depends on the continued life of the tree. When a tree dies, its token freezes, permanently non-transferable, permanently non-recoverable. The token does not lose value gradually; it leaves the market entirely.

The intended consequence is that participants who hold tokens hold an asset whose continued transferability depends on the survival of a specific tree, and that this

dependence, distributed across many holders and many trees, constitutes a continuous countervailing pressure against the discrete, episodic, exhausting work of advocacy. The mechanism is described in this document. The protocol is currently deployed on Base mainnet under the contract designation AT-251, with eleven genesis trees minted to a single oracle wallet operated by the founder. The roadmap to phase two and phase three describes the conditions under which oracle authority migrates first to a multisignature arrangement and then to a permissionless verification scheme. The conditions are stated as triggers, not dates, because the architecture is engineered such that each phase's required capabilities are provisioned by the same conditions that necessitate the transition.

This paper describes the mechanism, the contract, the operational architecture of phase one, the roadmap, and the rules under which the protocol commits to operate.

## Contents

Summary

Abstract

1. The problem this protocol addresses
2. Why a token mechanic
3. The contract: AT-251 v0.3
4. Operational architecture: phase one
5. The phase roadmap
6. The three rules
7. Verification: the agent sense-checker
8. Funding: the pool and its endogenous flows
9. What this protocol does not claim
10. Open problems

Related work

Appendix A: Contract addresses

Appendix B: Glossary

# 1. The problem this protocol addresses

Old-growth trees, defined for the purposes of this protocol as individual trees of two hundred and fifty years of age or greater, with the threshold encoded as the contract designation AT-251, are a class of biological entity for which the financial incentives to cut down a particular instance are concentrated, immediate, and assignable to specific actors, while the incentives to keep that instance standing are diffuse, deferred, and assignable mostly to no one in particular.

This asymmetry is the entire problem. It is not a problem of awareness; old-growth loss is widely known. It is not a problem of moral consensus; almost everyone, when asked, would prefer that the world's oldest trees continue to exist. It is a problem of incentive structure. The person standing next to a five-hundred-year-old tree with a chainsaw can, in the next ten minutes, convert that tree into a sum of money that will be deposited into their account by the end of the month. The person standing next to the same tree without a chainsaw, who would prefer the tree continue to live, has no comparable transaction available to them.

Conservation has historically tried to fill this gap with three classes of intervention. The first is regulation, laws prohibiting the cutting. The second is purchase of land, taking the tree out of the cutter's reach by removing the property rights that authorize cutting. The third is moral suasion, campaigns, pressure, advocacy. Each of these works, sometimes, in some places. None of them works continuously, everywhere, on the time horizons over which old-growth trees actually live. Regulations are repealed. Land trusts run out of funding. Public attention shifts to the next emergency. The trees outlive every protective regime that has ever been built around them.

The proposition of this protocol is that a different class of intervention is possible. Not a replacement for the existing three. An addition. Specifically: a financial instrument whose value depends on the continued life of a specific tree,

held by a participant whose financial interest in that tree's continued life is therefore continuous, asynchronous, and not contingent on the participant's attention or willingness to advocate.

If such an instrument exists, and is widely held, then the financial pressure to keep any given tree alive is no longer carried by an underfunded conservation organization or an overstretched regulator. It is carried by the holder of the token, whose self-interest does the work that political will has been doing badly.

This is the core thesis. The remainder of this document describes the mechanism by which the proposition is made operational.

## 2. Why a token mechanic

The mechanism could in principle be implemented through several different technical substrates. Conservation easements, syndicated land ownership, performance bonds, and various contractual derivatives have all been proposed at various points by various authors, and several have been implemented in narrow contexts. The reasons this protocol uses a public-blockchain non-fungible token are specific and worth stating directly.

### Permissionlessness

A token registered on a public blockchain is accessible to anyone with an internet connection. There is no application process, no jurisdiction filter, no minimum financial threshold, no requirement that participants identify themselves or commit to any ongoing relationship with the protocol. The barrier to acquiring a financial interest in a tree's survival is reduced to the lowest level the underlying network supports.

This matters because the alternative, a curated investor pool, a registered fund, a legal vehicle of any kind, concentrates the protective pressure in the hands of a small number of participants whose continued participation depends on factors orthogonal to the trees themselves. A fund can be wound up. An organization can be defunded. A treasury can be raided. A permissionless token registry persists for as long as the underlying chain persists, and is held by whoever holds it.

## Verifiability without trust

A public blockchain provides a substrate on which the state of the registry, which trees exist, which are alive, which have died, who holds which token, is independently verifiable by any participant or observer. The protocol does not need to be trusted to report its own state honestly, because its state is not produced by reporting. It is produced by computation that anyone can replicate.

This property is essential because the protocol's claim to do useful work depends on the persistence of specific trees, and the persistence of specific trees depends on a record-keeping system that cannot be quietly altered by any party with an incentive to alter it. The blockchain provides that property as a substrate-level guarantee.

## The freeze property

A non-fungible token contract can implement an irrevocable state transition, in this protocol, the freezing of a token's transferability when the underlying tree is reported and confirmed dead. Once a token is frozen, no transaction, no governance decision, no coordinated action of any kind can transfer it again. The token persists in the wallet of its last holder as a permanent record of the tree it referenced.

This is the load-bearing innovation. The token does not become worthless when the tree dies; it becomes inert. There is no liquidation event, no recovery, no salvage value. The financial interest in the tree's survival is total, because no portion of the holder's investment is recoverable upon failure. The incentive to protect the tree is not diluted by hedging strategies, because there are no hedging strategies, the only way the token continues to be a transferable asset is for the tree to continue to be a living tree.

## Non-claim of land rights

Critically: the token does not represent ownership of the tree, the land the tree stands on, the right to cut the tree, the right to prevent the tree being cut, or any other interest in the physical world recognized by any legal system. The token references the tree. It does not own the tree. Referencing is not claiming. Observing is not owning.

This is not a technical limitation to be solved. It is a deliberate feature. A protocol that purported to grant land rights would immediately collide with the

property law of every jurisdiction containing trees, and would necessarily either fail to operate in those jurisdictions or operate as a contested presence within them. A protocol that references trees without claiming any rights over them is operationally compatible with every property regime that exists, because it does not interact with property regimes at all. The reference is the protocol's claim. The tree's persistence in the physical world is a separate matter, governed by whoever governs the land.

This separation is what makes the financial incentive of the token holder into a useful pressure rather than a property dispute. The holder of a token cannot prevent the landowner from cutting the tree. The holder of a token holds an asset whose value is contingent on the landowner not cutting the tree. The pressure flows through markets, advocacy, purchase offers, conservation easements that the holder might fund, public attention that the holder might generate, through every legitimate channel, none of them adversarial to the property regime.

### 3. The contract: AT-251 v0.3

The AT-251 contract is an ERC-721 implementation deployed on Base mainnet, with the following load-bearing properties. The contract is a token; its companion contract, the verification pool, is a separate smart contract that holds the protocol's operational treasury and processes recertification fees and dead-tree claim fees. Both contracts are source-verified on BaseScan.

#### The freeze mechanism

The contract implements ERC-721 with an override on the internal transfer function. Before any transfer is permitted, the contract checks the alive-status flag of the token. If the flag is set to dead, the transfer reverts. The flag, once set to dead, cannot be set back to alive. The token becomes permanently non-transferable. There is no governance hook, no admin override, no recovery path.

This is the entire mechanism. It is small. It is small deliberately. The contract's correctness is the protocol's foundational guarantee, and the foundation is more durable when there is less to verify.

#### Liveness and the recertification cycle

Trees are presumed alive by default. To preserve the integrity of the registry over long time horizons, the contract implements a recertification cycle: each token must be recertified within a liveness period of twenty-five years. A holder who fails to recertify within the period is subject to having their token's alive-status flag re-examined by the protocol's verification mechanism.

Recertification is the protocol's primary mechanism for keeping the registry honest at scale. It is also the primary endogenous funding flow for the verification pool. A recertification fee is paid by the holder at the moment of recertification, and routes to the pool, which uses the funds to support whatever verification work is current at that phase of the roadmap. The recertification fee is a protocol fact, encoded in the contract, and is not subject to discretionary adjustment.

## Claim fees

A separate flow funds the pool when a token is claimed. The current claim fee is set as the recertification fee plus a displacement premium of fifty percent, encoded immutably. The displacement premium reflects the additional verification cost of admitting a new tree to the registry relative to maintaining a tree already on the registry. The premium is structural, not punitive.

## What the contract does not do

The contract does not implement a royalty mechanism. There is no ERC-2981 hook, no transfer fee, no protocol-side cut on any transaction in which a token changes hands between holders. The protocol takes nothing on sales between participants. This property is described in greater detail in section six.

The contract does not implement a per-cycle mint cap. The security model that such a cap would provide is provided instead by the phase roadmap and the operational architecture of phase one.

The contract does not depend on any external price oracle. All fees are denominated in the native unit of the underlying chain. There is no Chainlink dependency, no USD-pegged parameter, no external feed that, if compromised, could produce malformed protocol behavior. The pool's solvency is a function of the registry's activity, not of any oracle's prices.

## 4. Operational architecture: phase one

The protocol is currently in phase one of a three-phase roadmap. Phase one is operated by a single oracle wallet, controlled by the founder. The wallet is the only entity authorized to mark a tree dead, to admit a new tree to the registry, or to perform any other action on the contract that requires oracle authority.

This is, by design, the most centralized configuration the protocol will ever have. The reason for this configuration is operational, not philosophical.

### Why a single oracle in phase one

The protocol's correctness in phase one rests on the integrity of the registry, that trees admitted to the registry are real trees, that their reported coordinates correspond to those real trees, that their reported ages are accurate, and that their alive-status flags reflect their actual condition. At the scale of phase one, eleven genesis trees, expanding through controlled submission, this integrity is maintainable by a single careful operator with appropriate tooling. Distributing the oracle authority across multiple parties at this scale would not improve integrity; it would increase coordination cost without commensurate benefit, and would expose the protocol to griefing attacks that a single careful operator can reject by inspection.

The trade-off this configuration accepts is a centralization risk. The operator of a single-oracle configuration has the technical capacity to deviate from the protocol's intent, by marking living trees dead, by admitting fictitious trees to the registry, by failing to recertify legitimate trees. This risk is real. It is not eliminated by the contract; it is constrained by the rules described in section six and made operationally visible through the architecture described in section seven.

### The oracle wallet and its commitments

The oracle wallet, as of phase-one deployment, holds the eleven genesis tokens. It is the only address from which the contract has accepted mints. The wallet's address is published in appendix A; its activity is independently observable on BaseScan.

The wallet operates under an explicit commitment, stated as the first of the three rules in section six: the genesis cohort is sealed at exactly eleven, and the oracle wallet will not be the recipient of any additional mints under any circumstances.

Future tokens, when admitted via the user-submission flow described below, will mint directly to the submitter's wallet. The oracle wallet's role in that flow is to authorize the mint, not to receive it.

This commitment is not enforced by the contract. It is enforced by discipline. The site exists in part to make this commitment publicly visible, so that any deviation from it is immediately legible to any observer.

## The submission flow

In phase one, new trees enter the registry through a submission flow. A would-be holder submits the coordinates of an old-growth tree they wish to see registered. The submission is reviewed by an automated coordinate sense-checker (described in section seven) and then by the founder. Submissions that pass review are admitted to a mintable set. When walletless onboarding lands in phase two, see section five, submitters whose trees have been admitted to the mintable set will be able to claim those trees first.

This flow is intentionally narrower than a fully open mint. The narrowness is a property of phase one; the conditions under which the flow widens are described in the phase-roadmap section.

# 5. The phase roadmap

The protocol is structured around a three-phase migration of oracle authority. Each phase is defined by a configuration of who holds the authority to admit trees and to mark them dead, and by the conditions under which authority transitions to the next phase. The conditions are stated as triggers, not dates, because the protocol's correct behavior in any given phase depends on whether the operational preconditions for that phase have been satisfied, not on whether any particular calendar interval has elapsed.

The phases are summarized below. Each is described in the corresponding subsection.

## Phase one: founder-operated single oracle

The current phase. Single wallet, single operator. Submissions reviewed by the founder, with assistance from the coordinate sense-checker. Eleven genesis trees minted; user submissions admitted by review.

Trigger to phase two: the registry has grown to a size at which single-operator review is no longer sufficient for integrity, the operational preconditions for multisignature oversight have been assembled, and the verification process has been documented in sufficient detail that additional reviewers can be onboarded against a written specification rather than against the founder's tacit judgment.

[The brief specifies a registered-tree threshold around one hundred thousand as a working anchor for the first condition. Mitch, confirm or revise the figure before this section ships to the live site or paper.]

## Phase two: multisignature oversight

Oracle authority moves to a multisignature arrangement. The threshold is set such that the founder's signature is never structurally required for any decision, three of five with the founder as one signer, or two of three with the founder recused, both satisfy this property; three of three with mandatory founder participation does not. The reason for this construction is that the multisig must be capable of acting in the founder's absence or against the founder's preference; otherwise the migration from phase one to phase two is nominal rather than structural.

In phase two, walletless onboarding lands. Submitters without crypto wallets can claim trees through a custodial flow that mints to a server-generated wallet on the submitter's behalf. The submitted-trees-first ordering becomes operationally meaningful: the trees that participants have submitted during phase one are claimable by those participants without their needing to navigate the wallet-creation step.

Trigger to phase three: twelve or more months of pool self-funding through recertification and claim fees alone, without external subsidy; a verifier population of sufficient size and diversity to bootstrap the Schelling game's economic dynamics; ten or more active verifiers who have demonstrated independent capability to perform tree-status verifications under the documented protocol.

## Phase three: permissionless verification

Oracle authority is removed from the multisig and replaced with a permissionless verification mechanism, the working design is a Schelling-game architecture in which any participant can submit a verification claim and any participant can challenge it, with stakes resolved by the protocol's economic logic. The interface for this mechanism is already defined in the AT-251 contract; the full design of the Schelling game is deferred to a subsequent research note (Note 008, in preparation).

In phase three, the founder becomes a regular user of the protocol. The founder retains whatever genesis tokens the founder still holds, and may participate in verification on the same terms as any other participant. The protocol no longer has any role for which the founder is irreplaceable.

## On the structure of the triggers

The triggers above are constructed so that each phase's required capabilities are provisioned by the same conditions that make the phase necessary. Multisignature oversight is unnecessary while the registry is small enough for a single careful operator to maintain integrity by inspection; it becomes necessary as the registry grows; and the registry grows by attracting the participants who are themselves the candidate signers of the multisig. The growth of the protocol furnishes the operational substrate of its next phase.

The same logic applies to phase three. A permissionless verification mechanism is unnecessary while the multisig is small enough to handle the verification load directly; it becomes necessary as the load grows; and the load grows by attracting the verifiers who themselves operate the permissionless mechanism. Each transition is engineered to be a function of the protocol's success, not a separate effort that the protocol must execute on top of its core operation.

This is the answer to the question of how the protocol scales without a centralized scaling effort: the conditions of expansion enable the capability that expansion requires, and the conditions of non-expansion make additional capability irrelevant.

## 6. The three rules

The protocol's behavior is constrained by three commitments. They are not enforced by the contract. They are enforced by discipline, made visible here so that any deviation

from them is immediately legible to any observer.

The reason these commitments are not encoded in the contract is that the contract is, by design, small. The commitments are operational; the contract is structural. Encoding the commitments in code would either constrain the protocol's ability to perform its core function or would require contract complexity that increases the surface area of the foundational guarantee. The trade is preferable in this direction: keep the contract small, state the rules publicly, accept that the rules' enforcement depends on the visibility of any breach.

## Rule one: the genesis cohort is sealed at eleven

The oracle wallet holds eleven tokens. It will hold no others. Future tokens admitted to the registry, whether through user submission in phase one or any other flow in subsequent phases, will mint to addresses other than the oracle wallet. The founder will not be the recipient of any tree admitted to the registry after the genesis cohort.

The reason for this rule is that the founder's incentives, in phase one, must be visible. A founder operating as oracle has the technical capacity to admit trees to a wallet they themselves control, accumulating a private inventory under cover of protocol operation. A sealed genesis cohort eliminates this possibility by commitment: any growth in the founder's holdings is immediately visible as a deviation from the rule.

## Rule two: the protocol takes nothing on sales

When a participant transfers a token to another participant in exchange for any consideration, the protocol receives no portion of that consideration. There is no royalty. There is no transfer fee. There is no protocol-side hook on transfers. The pool funds itself through recertification fees and claim fees only, both of which are paid by participants who are actively engaging the protocol's operational mechanisms, not by participants who are merely trading.

The reason for this rule is that the protocol's claim to align the holder's interest with the tree's survival is undermined to the extent that the protocol takes a fee on transfers. A protocol that profits from transaction volume has an incentive to encourage transaction volume, which is at best orthogonal to the protocol's claimed

purpose and at worst contrary to it. A protocol that takes nothing on transfers cannot have its incentives questioned on this axis.

### Rule three: ownership of the contract will move toward distributed control, not be renounced

The contract's ownership will migrate, on the schedule described in section five, from the founder to a multisignature arrangement, and then from the multisignature arrangement to a permissionless verification mechanism. The contract's ownership will not, at any point, be renounced. An unowned contract cannot transition to its next phase, and the protocol's continued correctness over the time horizons relevant to old-growth trees depends on the ability to perform such transitions.

The reason for this rule is that contract renunciation is sometimes proposed as a credibility-signal in token projects, a way to say to participants, your tokens are now safe from the founder's discretion. In the context of this protocol, contract renunciation would be the opposite of credibility. It would mean the protocol can no longer migrate to its next phase, and would therefore be permanently locked at whatever level of decentralization existed at the moment of renunciation. The right move toward distributed control is not to abandon the wheel; it is to add hands to it on a schedule, with the founder's hands moving away last.

## 7. Verification: the agent sense-checker

The integrity of the registry depends on the trees admitted to it being real, located where they are reported to be located, of the age claimed, and alive when admitted. In phase one, this verification is performed by the founder, with assistance from an automated coordinate sense-checker.

The sense-checker is a software agent that, given a submitted set of coordinates and a claimed species and age, performs a series of consistency checks against publicly available data sources: satellite imagery, species-distribution databases, treeline elevation models, and where applicable, registries of named significant trees maintained by botanical organizations and government agencies. The agent does not make admissibility decisions; it produces a structured report that the founder reviews.

The data sources currently consulted include high-resolution satellite imagery from publicly available sources, IUCN species-distribution polygons for cross-checking that the claimed species is plausible at the submitted coordinates, treeline and elevation models for upper-bound checks, and registries of named significant trees maintained by national parks services, forestry agencies, and botanical organizations. The list is not exhaustive and is expected to expand as the registry grows. No single data source is treated as authoritative; a submission's plausibility is assessed across multiple checks, and disagreements between sources surface as flags rather than as automatic rejections.

The purpose of the sense-checker is to make the founder's review tractable at scale. A single careful operator cannot personally inspect a hundred thousand candidate locations against satellite imagery; an operator assisted by an agent that does the first-pass inspection and surfaces the candidates that warrant human attention can. The agent does not replace human judgment; it allocates human judgment to the cases where it matters.

The sense-checker is the operational substrate of phase one's claim to maintain registry integrity at a scale beyond what manual review alone could support. It is also the proof-of-existence for the broader claim, made in section five, that each phase's required capabilities are provisioned by the same conditions that make the phase necessary. The agent exists because the registry has grown to a size at which it is needed; the agent's existence is what allows the registry to continue growing.

The agent's outputs are auditable. The protocol does not currently publish per-submission audit trails, phase one operates on the basis of the founder's review, and the agent is a tool of that review rather than a replacement for it. In phase two, when the multisignature arrangement makes the audit trail consequential to multiple parties, agent outputs will be made available to the signers as part of the standard review packet. The detailed specification of this audit interface is part of the phase-two engineering work.

## 8. Funding: the pool and its endogenous flows

The verification pool is a separate smart contract that holds the protocol's operational treasury. The pool's purpose is to fund whatever verification work is current at the protocol's present phase: in phase one, the operational costs of

running the sense-checker and the founder's review apparatus; in phase two, the costs of multisignature operation and walletless onboarding infrastructure; in phase three, the staking and reward flows of the permissionless verification mechanism.

The pool funds itself through two flows, both endogenous to the protocol's operation.

## Recertification fees

Each token must be recertified within twenty-five years of its previous certification. The recertification is performed by the holder, who pays a recertification fee at the moment of recertification. The fee routes to the pool. The recertification fee is encoded in the contract as a protocol fact and is not subject to discretionary adjustment.

The structural argument for this flow is that the holders of tokens are the participants whose continued participation depends on the registry being maintained. They are therefore the appropriate funders of the maintenance work. The fee is paid at the moment the holder is using the protocol's verification capacity, recertification is the formal expression of that use, which makes the fee an expression of the holder's actual relationship to the protocol's work, not a tax on incidental activity.

## Claim fees

When a participant claims a token, either by admitting a new tree to the registry under the user-submission flow or by claiming a dead-tree token under the protocol's death-claim mechanism, the participant pays a claim fee. The claim fee is set as the recertification fee plus a displacement premium of fifty percent, encoded immutably.

The displacement premium reflects the additional verification cost of admitting a new tree to the registry relative to maintaining a tree already on the registry. New admissions require sense-checker review, founder review, and acceptance into the mintable set, work that maintenance does not require. The premium is so named because it reflects the displacement of verification load from the maintenance regime to the admission regime. The premium is structural, not punitive, and it is encoded as a protocol fact rather than as a discretionary parameter to remove the temptation, in any future configuration, to adjust admissions costs in response to incentives orthogonal to the protocol's purpose.

## Why no transfer fee

As stated in rule two, the protocol does not take a fee on transfers between holders. The pool's solvency is therefore a function of the registry's activity at the operational interfaces, recertifications and claims, rather than at the trading interface. This has the consequence that the pool's funding depends on the registry being used as a registry, not on the tokens being used as tradeable assets. The pool is healthy when the protocol's core function is in active operation.

This is, deliberately, a more conservative funding model than the alternatives. A transfer-fee model would yield more pool revenue at any given level of trading activity. The trade is accepted because the alternative pollutes the protocol's credibility on rule two, and because the pool's required funding is bounded, verification at any given phase has identifiable operational costs that scale with the registry, not with the number of times tokens change hands.

## 9. What this protocol does not claim

This section exists because the failure modes of token projects, in the field, are predominantly failures of overclaim. Stating what the protocol does not promise is at least as important as stating what it does. The following is not exhaustive; it is a list of the most important non-claims.

### It does not promise a return

Holders of AT-251 tokens hold tokens. The tokens are not investment contracts. The protocol does not promise that the tokens will appreciate in value, that they will be tradeable on any particular market, that the registry will grow, that the value of one tree's token will bear any particular relationship to the value of another tree's token, or that any holder will be able to transact their token at any particular price. The tokens are what the contract makes them: non-fungible, transferable until frozen, frozen when the underlying tree is reported and confirmed dead.

Whatever value the tokens have in any market is a market phenomenon, produced by the participants in that market. The protocol observes this phenomenon. The protocol does not produce it and does not commit to producing it.

## It does not grant rights to land or trees

As stated in section two: the token references the tree. It does not own the tree. A holder of a token cannot prevent the landowner from cutting the tree, cannot enter the land where the tree stands, cannot file a legal claim against the cutting of the tree on the basis of the token, and cannot, in any jurisdiction known to the authors of this protocol, assert any property right of any kind by virtue of holding the token. The token is a digital reference to a physical entity. The reference confers no rights in the physical entity.

A holder who wishes to acquire rights in the physical entity must do so through whatever legal mechanism applies in the jurisdiction where the tree stands, purchase of land, purchase of conservation easement, contractual agreement with the landowner. The token does not provide these mechanisms. It also does not displace them; participation in the protocol does not preclude participation in any of them, and indeed the protocol is operationally compatible with all of them.

## It does not promise that any particular tree will survive

Trees die. They die of natural causes. They die of fire and disease. They die of cutting, despite all of the protective pressure their token holders bring to bear. The protocol's mechanism is to align the holder's interest with the tree's survival; it is not to guarantee that survival. A holder whose tree dies receives the structural consequence of that death, the token freezes, and not a refund or compensation of any kind. The freeze is the mechanism, and the mechanism's correctness is not contingent on the outcome.

## It does not claim to be a complete solution to old-growth loss

The protocol is one mechanism. The literature on old-growth loss describes many causes; this protocol addresses one, the asymmetry of incentives, and even within that one, the protocol provides one form of countervailing pressure rather than a complete solution. Regulation, land trusts, advocacy, and direct conservation work all remain necessary. The protocol's claim is that an additional source of continuous protective pressure is useful, not that the existing sources can be replaced.

# 10. Open problems

Several aspects of the protocol's operation are not yet fully resolved. They are listed here, with reference to the research notes that develop them in greater detail.

## Verification at scale

The phase-one architecture handles verification through founder review supported by the sense-checker. This scales to a registry of some size; it does not scale indefinitely. The phase-two architecture extends review to a multisignature group; the phase-three architecture replaces the multisignature group with a permissionless mechanism. The detailed design of the phase-three mechanism, the Schelling game, is the subject of Research Note 005 and a forthcoming Research Note 008.

## The hundred-year problem

The protocol's intended time horizon, old-growth trees commonly live for centuries, and the oldest are millennial, exceeds by a wide margin the time horizons over which any blockchain has so far been demonstrated to operate continuously. Research Note 007 develops this problem at length. The current architecture's response is to minimize dependencies on any particular external substrate (no external oracles, no off-chain governance, no parameters denominated in non-native units) and to design transitions between phases as protocol-internal rather than externally-driven events. The problem is not closed. It is not, on present analysis, soluble in advance; the protocol commits to engineering its longevity as an ongoing operational problem rather than as a one-time design problem.

## Funding sufficiency

The pool's endogenous funding model, recertification fees plus claim fees, is, in principle, sufficient to support the protocol's verification work at any phase. Whether it is sufficient in practice depends on the relationship between fee rates and verification costs, which is the subject of Research Note 006. The note's conclusion, in summary, is that the model is sufficient under reasonable assumptions about the registry's growth and the per-tree verification cost in phases two and three; the assumptions are stated and the sensitivity analysis is published.

## Permissionless grieving

The phase-one configuration accepts that a single oracle is exposed to griefing, submitters who flood the review queue with junk submissions, false reports of death, fictitious tree submissions designed to embarrass the protocol or its operator. The sense-checker and the founder's discretion handle this in phase one. The migration to phase two introduces multisignature review, which distributes the griefing load. The migration to phase three introduces the Schelling game's economic disincentive against bad-faith claims. Each phase's response to griefing is appropriate to the scale at which griefing becomes operationally consequential. The earlier-phase mechanisms are not vulnerable to scales of griefing they cannot encounter.

## Legal and jurisdictional questions

The protocol is operated from a specific jurisdiction by an identifiable founder; participants are located in many jurisdictions; trees are located in many jurisdictions. The protocol's operations are designed to be compatible with the property law of every jurisdiction containing trees, by virtue of not making property claims. The protocol's operations are designed to be compatible with the securities law of jurisdictions where applicable, by virtue of not promising returns. These are descriptions of the protocol's design choices, not legal conclusions about how any particular jurisdiction will treat the protocol or its participants. Participants in any specific jurisdiction may need to seek legal advice specific to their circumstances. The protocol does not provide such advice.

## Related work

This document is the protocol's primary description; it does not stand alone. The mechanisms it describes have been developed over a series of research notes, each treating one component in greater depth than is appropriate for a general-purpose whitepaper. Readers seeking the underlying analysis are referred to the following.

Research Note 001, The Permissionless Instrument. Develops the argument for a public-blockchain registry as opposed to alternative substrates, treating in detail the trade-offs section two of this document summarizes.

Research Note 002, The Death-Profit Paradox. Develops the freeze mechanism's incentive logic, including the reasons for one-way irreversibility and the analysis of attempted hedging strategies.

Research Note 003, The Gold Rush. Treats the question of registry growth dynamics, including the conditions under which a registry of long-lived trees can scale without overwhelming verification capacity.

Research Note 004, The Arbitration Problem. Treats the question of how the protocol resolves disputes about a tree's status, including the role of multisignature oversight in phase two.

Research Note 005, The Verification Problem. Develops the multi-phase verification architecture, including the design constraints on the phase-three Schelling game.

Research Note 006, The Funding Problem. Treats pool solvency under realistic registry growth scenarios, including sensitivity analysis on fee parameters and verification costs.

Research Note 007, The Hundred-Year Problem. Treats the problem of operating a protocol on time horizons longer than any blockchain has yet been demonstrated to operate.

Research Note 008 (in preparation). Will treat the detailed design of the phase-three Schelling-game verification mechanism.

Adjacent published work by the same authors, available at Zenodo with persistent DOIs, includes the megapaper Designing for Permanence and the methodological paper on Gomboc Engineering. Both inform the protocol's broader design philosophy without being protocol-specific.

## Appendix A: Contract addresses

Network: Base mainnet (chain ID 8453)

AT-251 v0.3 token contract:

0x316198c9375Ad340DFFb919d98540e77b5ab6Cca

Verification pool contract:

0xe8be800bfB6869C7eF611474Caa87f280569C977

Oracle wallet (genesis cohort holder, phase one):

0xA378Cc944370177916971620bA4abC3B03Aa4518

Both contracts are source-verified on BaseScan. The deployed bytecode of each contract has been independently confirmed against the source published in the protocol's open-source repository.

## Appendix B: Glossary

AT-251. The contract designation. The number encodes the protocol's age threshold for admissible trees: two hundred and fifty years and above. The designation is fixed; it is a property of the protocol, not a version number.

ARES. The deterministic visual fingerprint generator that produces a unique image from any pair of GPS coordinates. The protocol uses ARES to produce a visual identifier for each tree in the registry. ARES is a separate component from the contract; it operates client-side from the coordinates encoded in the token's metadata.

Claim fee. The fee paid by a participant claiming a new tree to the registry, or claiming a dead-tree token under the death-claim mechanism. Set immutably as the recertification fee plus a displacement premium of fifty percent.

Death-claim. The mechanism by which a participant proves that a tree referenced by a registered token has died, triggering the token's freeze. A claim fee is paid; the verification mechanism (phase-appropriate) confirms the death; the contract sets the token's alive-status flag to dead.

Displacement premium. The additional fee charged on claims relative to recertifications, reflecting the higher operational cost of admitting a new tree

relative to maintaining one already on the registry. Set at fifty percent, encoded immutably.

Freeze. The state of a token whose alive-status flag has been set to dead. Frozen tokens are permanently non-transferable. The state is irreversible.

Genesis cohort. The eleven trees held in the oracle wallet from phase-one deployment. Sealed at eleven by rule one. Mollestadeika, the eldest, is token zero.

Liveness period. Twenty-five years. The maximum interval permitted between recertifications of a token. A token whose holder fails to recertify within the liveness period is subject to re-examination of its alive-status flag by the protocol's verification mechanism.

Mintable set. The collection of trees that have been submitted to the protocol, reviewed, and admitted as mintable but not yet minted. In phase two, walletless onboarding will permit submitters to claim from this set.

Mollestadeika. The thousand-year-old pedunculate oak in southern Norway whose coordinates produce the protocol's brand mark via ARES. Token zero.

Oracle wallet. The single wallet holding oracle authority in phase one. Holds the eleven genesis tokens. Will not hold any subsequent tokens (rule one).

Recertification. The act of confirming, within the liveness period, that the tree referenced by a token is still alive. Performed by the holder; a recertification fee is paid; the fee routes to the pool.

Recertification fee. The fee paid by a holder at the moment of recertification. Encoded as a protocol fact in the contract.

Schelling game. The phase-three verification mechanism, in which any participant can submit a verification claim and any participant can challenge it, with stakes resolved by the protocol's economic logic. Detailed design in preparation as Research Note 008.

Sense-checker. The automated coordinate-validation agent that supports the founder's review of submitted trees in phase one. Validates submissions against satellite imagery, species-distribution data, and registries of named significant trees, surfacing candidates that warrant human attention.

Verification pool. The smart contract holding the protocol's operational treasury. Funded by recertification fees and claim fees. Disburses funds to whatever verification work is current at the protocol's present phase.