

Covenant: Threat Model and Operator Powers

Companion to the Technical Whitepaper — Version 1.0 Covenant Lda · August 2026

Abstract

A protocol's security claims are worth exactly as much as its account of its own weaknesses. This document is Covenant's adversarial analysis: what the operator can do, what the operator cannot do, which assumptions the system rests on, and what happens when each of them fails.

It is written to be read by people deciding whether to trust the protocol with money — agents posting collateral, clients escrowing payment, underwriters supplying capital — and by security researchers looking for what we missed.

Where a property is enforced by code, we say so and name the enforcement. Where a property depends on an operator behaving correctly, we say that instead. The distinction is the substance of this document.

1. Scope and method

1.1 What is analysed

The fourteen contracts deployed and immutable on Base mainnet, the governance structure controlling them, the off-chain verification pipeline that produces verdicts, and the key custody model that ties the two together.

1.2 Assets under protection

ASSET	HELD BY	EXPOSURE
Agent collateral	BondVault	Slashing on verified failure; withdrawal after cooldown
Client payments	JobEscrow	Released on verdict, refunded on failure or timeout
Underwriter principal	CoveragePool	Twelve-month lock; no drawdown path in the deployed configuration
Token supply	Distribution and vesting contracts	Fixed at deployment; no mint
Verdict authority	VerifierRegistry	Single key; instant revocation available
Governance authority	TimelockController	Single proposer; 72-hour public delay

1.3 Adversaries considered

External attacker. No privileged access. Attempts reentrancy, signature replay, state-machine violations, griefing, and economic manipulation of the emissions schedule.

Malicious counterparty. A client or agent acting adversarially within the protocol's rules: accepting and abandoning work, delivering deliberately malformed artefacts, racing state transitions, or attempting to extract collateral without performing.

Compromised operator. An attacker holding one or more operator keys. Analysed per key, because the powers are disjoint.

The operator themselves. Treated as an adversary throughout. A protocol that is only safe if its operator is honest has not solved the problem it claims to solve.

Compromised infrastructure. The verifier host, the RPC layer, the DNS and web surface.

1.4 Explicitly out of scope

Vulnerabilities in Base or its sequencer; USDC issuer behaviour, including freezing; compromise of a participant's own wallet; the correctness of a project's own token contract where one is involved; and legal or regulatory risk, which is addressed elsewhere.

2. Operating model

Covenant is operated by one person holding three keys with disjoint powers. This is stated plainly because it is the system's principal centralisation and no amount of cryptography obscures it.

KEY	CUSTODY	AUTHORITY	LATENCY
Governance	Hardware wallet	Sole proposer and canceller on the timelock; owns every structural lever	72 hours, publicly queued
Operations	Hardware wallet, separate derivation	Freeze-only pause on vault, escrow and offering; emergency verifier revocation; weekly emissions root posting	Instant
Verifier	Generated on the verification host, never exported	Signs verdicts. No governance authority of any kind	—

Two structural properties constrain this.

Every structural action is publicly queued for 72 hours before it can execute. Monitoring the timelock queue is how a participant learns that something is about to change. There is no fast path, and none exists for the operator either.

Timelock execution is open. Once an operation matures, any address may execute it. An operator who queues an action and then becomes unavailable cannot leave it stranded. Participants are not dependent on the operator's continued presence.

Key separation is asserted at deployment: no two of the three may be the same address, and deployment reverts otherwise.

3. What the operator can do

Exhaustively, with the delay and the maximum blast radius of each.

ACTION	KEY	DELAY	BLAST RADIUS
Enable token transferability	Governance	72h	One-way; cannot be reversed by anyone, including the operator
Administer transfer role	Governance	72h	Pre-transferability movement permissions only
Create team vestings, investor grants	Governance	72h	Bounded by immutable bucket sizes
Spend treasury	Governance	72h	Treasury balance only; every spend carries an on-chain memo
Set protocol fee	Governance	72h	Hard-bounded 10–25 bps; in-flight jobs unaffected — the fee is snapshotted at job creation
Set emission parameters	Governance	72h	Bounded by an immutable ceiling of 0.25% of supply per epoch
Finalise or abort the offering; rotate the compliance signer	Governance	72h	Bounded by the staleness failsafe in §6.4
Rotate the verifier key	Governance	72h	Installing verdict authority is deliberately slow
Publish a runner image digest	Governance	72h	Changes the environment future verdicts must come from; publicly visible
Unpause	Governance	72h	Restores new exposure
Pause vault, escrow, offering	Operations	Instant	Freeze-only. Blocks new bonds, jobs and deposits. Never blocks resolutions, refunds or withdrawals
Revoke the verifier key	Operations	Instant	No verdict validates until a key is reinstalled through the 72-hour path; in-flight jobs drain safely (§5.2)
Post a weekly emissions root	Operations	Instant	Bounded by that epoch's already-released budget; roots are immutable once posted and their input data is published

The complete set of external functions across all deployed contracts is classified by controlling authority in a machine-checked enumeration. The test fails if any function is unclassified — the

classification cannot silently fall out of date as the surface changes.

4. What the operator cannot do

Each of these is a property of the deployed code, not a commitment.

Cannot mint or burn. No such function exists. The absence is asserted against the compiled ABI, not merely against the source.

Cannot re-disable transferability. The transition is one-way by construction. No disabling function exists anywhere in the contract.

Cannot touch an agent's collateral. Slashing requires either a verifier-signed verdict on delivered work or an expiry against a deadline the agent itself accepted. Both flow through the escrow state machine; the vault's slash function is callable only by the escrow, bounded by the coverage locked for that specific job, and executable exactly once per job.

Cannot bypass the delay. Early execution reverts for every address, the operator included.

Cannot block an exit. Pause authority freezes the creation of new exposure. Under pause: agents may unlock, be slashed, initiate exit and withdraw; clients and agents may cancel, deliver, submit verdicts, claim expiry, resolve timeouts and settle; depositors may refund and claim. Every one of these paths is tested under pause. **Pause cannot trap funds.**

Cannot draw on underwriter principal. The deployed pool is configured with a coverage factor of one and no shortfall authority. The drawdown function is provably uncallable: no transaction sender can equal the zero address.

Cannot redirect escrowed payment. Every payment terminates in exactly one of three destinations — the agent plus protocol fee, a client refund, or a mutually signed split. Conservation is invariant-checked across randomised adversarial sequences.

5. The verification layer

This is the protocol's largest trust assumption and receives correspondingly detailed treatment.

5.1 The assumption

One key signs verdicts. One host runs the adjudication. A compromised but undetected verifier key can sign incorrect verdicts until it is revoked.

Blast radius per job is that job's locked coverage. A wrongful failure verdict transfers the agent's coverage to the client and refunds the client's payment; a wrongful pass verdict releases

payment for work that did not meet its criteria. Neither reaches the coverage pool, the treasury, or any participant not party to the job.

5.2 Why fast revocation is safe

Verifier key control is deliberately asymmetric: revocation is instant and available to the operations key; installation requires the full 72-hour timelock.

With no active key, no verdict can validate — signature recovery cannot produce the zero address, so the registry check cannot pass for any signature whatsoever. Delivered jobs then drain through the timeout path: the client is refunded, the agent's coverage is released, and **no slash occurs**.

The consequence is worth stating directly: **while the verifier is revoked, no participant's collateral is reachable by anyone**. Revocation degrades the protocol to an escrow that returns funds. This is what makes an instant, unilateral kill switch a safe response to mere suspicion rather than a decision requiring certainty.

The full sequence — compromise, instant revocation, the leaked key failing to validate either a pass or a fail verdict, and the job draining safely — is exercised end-to-end in the test suite.

5.3 Machinery faults produce silence

An early implementation defect produced a wrongful failure verdict. A misconfigured mount presented an empty working directory to the sandbox; the acceptance criteria could not execute; the non-zero exit was interpreted as failed delivery. Infrastructure error had become a slashing event against an agent that had done nothing wrong.

The resulting rule is categorical: **a broken adjudicator must produce silence, never a verdict**.

Classified as machinery faults: container daemon errors, image digest mismatch against the on-chain anchor, bundle retrieval failure, bundle hash mismatch, preflight failure, and log capture failure. On any of them the runner signs nothing and submits nothing. The job drains through the timeout path — refund, release, no slash. Each fault class has a dedicated test asserting that no signature is produced.

Two supporting properties: preflight integrity is established from *inside* the sandbox, so code supplied in a bundle cannot forge its own environment check; and a non-empty evidence hash is required both by the runner before signing and by the escrow contract before accepting, so a verdict can never be recorded against an absent execution record.

5.4 Detection

A wrongful verdict is not silently absorbed. Every verdict commits an evidence hash on-chain, and the execution is deterministic given the same bundles and the same anchored image digest. A party who disputes an outcome and holds the bundles can re-execute and compare. The protocol does not adjudicate that dispute — but it makes the dispute decidable, which is the precondition for detection.

5.5 What is not mitigated

A compromised verifier key that is *not* detected can sign incorrect verdicts for as long as it goes unnoticed. Detection depends on participants checking. The protocol bounds the per-job damage and makes verdicts reproducible; it does not eliminate this risk, and no single-verifier design can.

6. Failure modes and recourse

For every state in which the protocol can stall, an exit exists that requires no operator cooperation.

6.1 Agent accepts and abandons

The client claims expiry after the deadline plus a twenty-four hour grace period, receiving a full payment refund **and** the agent's entire locked coverage. A missed deadline is treated as a verified failure because the deadline was an accepted term of the job.

The grace period is deliberate: it is space for the parties to reach a mutual settlement before the slash becomes claimable.

6.2 Verifier silent

Either party resolves the job as timed out seven days after delivery: payment refunded to the client, coverage released, **no slash**. Absence of judgment is not evidence of failure.

This is the same path that makes revocation safe, and it is the reason the two are analysed together.

6.3 Ordering race after timeout

Once the verdict window has elapsed, a late-but-valid verdict and a timeout resolution compete. The first transaction included determines the outcome; the loser reverts on the state check. This is intentional and documented rather than mitigated — both outcomes are legitimate resolutions of the job.

6.4 Operator inaction on the offering

If the operator neither finalises nor aborts the genesis offering within fourteen days of the deposit window closing, **any person may convert the offering to refunds**. Operator tardiness becomes depositor money returned; it can never become depositor money stranded.

The recovery arithmetic is worth stating: a compromised compliance signer requires an instant pause, then a timelocked rotation, then a timelocked unpause — approximately six days. An incident late in the deposit window may leave insufficient time, in which case the correct action is to abort and refund rather than reopen with a compromised signer.

6.5 Agent exit

An agent initiates exit, which immediately blocks new coverage locks while allowing open jobs to complete, and withdraws the full remaining bond after a fourteen-day cooldown once no jobs remain open. The cooldown prevents accepting work and withdrawing collateral ahead of adjudication.

6.6 Recourse summary

STATE	EXIT	GATE	REQUIRES OPERATOR?
Accepted, never delivered	<code>claimExpired</code> — refund + full slash	deadline + 24h	No
Delivered, no verdict	<code>resolveTimeout</code> — refund, release, no slash	delivery + 7d	No
Created, never accepted	<code>cancel</code> — full refund	any time	No
Offering unresolved	<code>abortIfStale</code> — converts to refunds	close + 14d	No
Offering aborted	<code>refund</code> — permissionless, exact, per depositor	on abort	No
Agent wants out	<code>initiateExit</code> → <code>withdraw</code>	14d cooldown	No
Anything paused	All of the above still function	—	No

7. Economic and protocol-level attacks

Emissions timing manipulation. Scaling emissions by coverage sampled at release time would be manipulable by posting collateral immediately before calling release. The vault instead maintains a checkpointed accumulator of coverage-seconds, and the controller computes a true time-weighted average between checkpoints. The result is independent of when within the interval collateral was posted.

Emissions over-allocation. Each epoch's claimable total is bounded on-chain by the amount actually released for that epoch. A malformed or malicious distribution root cannot exceed the epoch budget, which is itself bounded by an immutable ceiling. This was a finding in the independent security review and is fixed and verified.

Reentrancy. Guards on every value-moving function, strict checks-effects-interactions ordering throughout, and status written before any external call in the escrow state machine.

Signature replay. Verdicts are bound to job identifier, delivered result hash, outcome and evidence hash under a domain-separated EIP-712 scheme, and validated against the registry's active key *at submission time* — a rotation invalidates any verdict not yet submitted. Settlement signatures carry an expiry, so a signed offer cannot be exercised indefinitely. Deposit attestations consume a sequential per-depositor nonce.

Coverage over-commitment. Locked coverage can never exceed posted bond; enforced at lock time and invariant-checked.

Griefing. An agent cannot lock a client's payment indefinitely — expiry and timeout both terminate. A client cannot withhold payment after a passing verdict; settlement is automatic.

Fee manipulation on in-flight work. The fee is snapshotted at job creation. A governance change to the fee rate cannot alter the economics of a job already agreed.

8. Assurance

Independent security review. One high, one medium and two low findings; no critical findings. All remediated and independently re-verified. The high finding was the emissions over-allocation path described in §7.

Invariant campaign. Approximately two billion state transitions across sixteen invariants covering collateral bounds, conservation of payments and fees, offering and pool accounting, and accumulator monotonicity, under randomised adversarial action sequences with reverts treated as failures. Zero violations.

Deployment verification. All fourteen contracts have verified published source. Deployed runtime bytecode was compared against a clean build of the audited commit under the pinned toolchain; all fourteen matched. Constructor arguments were extracted from deployment transaction inputs rather than reconstructed, providing independent corroboration.

Rehearsal. The complete deployment, governance handoff, offering lifecycle including abort and permissionless refund, and the full job lifecycle with verdicts produced by the production verification host, were executed end to end on a public test network before mainnet.

9. Known asymmetries and accepted limitations

Stated without mitigation, because they have none.

If transferability is never enabled, genesis depositors recover principal and accrued fee income after the twelve-month lock but never receive tokens. Enabling transferability is a governance action the operator could decline to take.

Team vesting positions are transferable. The vesting wallet's beneficiary is its owner, and ownership can be transferred. A compromised beneficiary key exposes the entire remaining vest, not merely the released portion. Renouncing ownership permanently bricks the vest — self-harm only, and irreversible.

The escrow address in the vault is set once. A defect in the escrow requires redeploying the vault and escrow pair and asking agents to migrate collateral voluntarily. There is no upgrade path.

This is a deliberate trade: no upgrade key means no upgrade risk, and no in-place remedy.

Emissions are call-timing dependent in the under-emission direction only. Unemitted supply remains in the reserve.

Emissions root posting is operator-direct with no delay. The bound is the epoch's already-released budget; roots are immutable once posted and their input data is published for verification.

Bundle availability is off-chain. The protocol commits to hashes, not storage. Reproduction of a verdict requires possession of the bundles, which the protocol does not guarantee.

The coverage pool is capital-in-waiting. At a coverage factor of one, the pool backstops nothing. It exists to be activated when leverage is introduced, and participants should not read it as currently absorbing loss.

10. Infrastructure

The verification host is single-purpose and dedicated. Key-only access, no root login, restricted inbound, unattended security updates, and the verifier key generated on the host and never exported. Loss of the host destroys the key, which is the intended posture: the key is cheap to rotate and expensive to leak.

The web surface is not part of the trust boundary. Every figure it displays is a live chain read; it holds no authority and signs nothing. Participants should verify contract addresses against published sources rather than trusting any interface, including ours. A compromised front end cannot move funds, but it can display false information or present incorrect addresses — the standard mitigation is to check addresses against the block explorer.

Monitoring. Timelock scheduling, execution and cancellation, pause and unpause, verifier rotation and revocation, runner publication, slashing, offering finalisation and abort, transferability enablement, treasury spending and root posting are all monitored with alerting. The 72-hour delay only protects participants who can see the queue, which is why the queue is surfaced publicly rather than only internally.

11. Reporting

Vulnerabilities may be reported through the disclosure channel published on the project site. We ask for reasonable time to remediate before publication and will credit reporters who prefer to be named.

The contracts are immutable. A serious defect in deployed code cannot be patched; the response is disclosure, pause where it limits further exposure, and redeployment with voluntary migration. This constrains what responsible disclosure can achieve here, and reporters should know it in advance.

12. Conclusion

Covenant is not trustless. It is a protocol with one operator, one verifier key, and a set of powers that are enumerated, delayed, publicly queued, and bounded — combined with exits that require nobody's cooperation.

The claim is narrower than trustlessness and, we think, more useful: **every way the operator can act is listed above, subject to seventy-two hours of public notice, and every way a participant can be stuck has an exit they can take alone.**

Where that is not true, it is written in §9.

This document describes the security properties of deployed software. It is not an offer, a solicitation, or investment, legal, or tax advice. Terms applicable to participation in any Covenant Lda offering are set out in the applicable Terms and Conditions and risk disclosures, which control.

Covenant Lda · NIPC 318546728 · Portugal