PUBLIC SAMPLE — REDACTED · client identity removed; ▮▮ bars mark removed identifiers · methodology & findings intact

ARI

FULL-STACK SECURITY REVIEW

SECURITY REVIEW — [redacted] PROTOCOL

[redacted] · Track T2 — Membership-gated fixed-term lending pools + factory

PROJECT	TRACK / TIER	DATE
[redacted]	T2 · Full	19 April 2026
CHAINS	**CODE VERSION**	**CLASSIFICATION**
Ethereum + EVM	locked at kickoff (pending)	PUBLIC SAMPLE

Disclaimer. This report describes a time-boxed technical security review of the materials identified in the Scope of Review as they existed at the referenced commit/version during the review window. It is a point-in-time assessment: subsequent changes to code, configuration, infrastructure, or dependencies are not covered. No security review can identify all vulnerabilities. This report does not constitute a guarantee, warranty, or insurance of any kind, and it is not an endorsement of the project, its business model, or its tokens. It is not investment, legal, or financial advice. The review does not include formal verification, continuous monitoring, or exhaustive novel economic attack modeling unless expressly stated in scope. [redacted] may publish or share this report in full, unmodified form, including this disclaimer. Partial reproduction that omits findings, statuses, or this disclaimer is not authorized. Vari accepts no liability for decisions made by third parties on the basis of this report.

00 EXECUTIVE SUMMARY

[redacted] is a membership-gated, uncollateralized fixed-term lending protocol. Pools are pure accounting layers — funds route peer-to-peer (lender→borrower on lend, borrower→lender/treasury on repay), so a pool never custodies assets, removing the usual drain surface. Access control, checks-effects-interactions ordering, reentrancy guards, SafeERC20 and parameter caps are all sound. The novel risk lives in the interest/penalty accounting: the monthly-repayment path pays interest up to the current time but only reduces the recorded liability up to the scheduled payment time, so a late sub-interval is charged more than once. No attacker-driven theft path was found (the borrower controls repayment; lender actions are membership-gated).

CRITICAL	HIGH	MEDIUM	LOW	INFO
0	0	1	4	2

No critical or high-severity issues; findings are medium and below.

01 SCOPE OF REVIEW

[redacted] membership/KYC gating, lending Pool + PoolFactory, Asset accounting, utility guards. Beacon-proxy upgradeable. [redacted]'s own code.

`TRACK :: T2 · TIER :: Full · CHAINS :: Ethereum + EVM · CODE :: locked at kickoff (pending)`

Out of scope: third-party dependencies (base libraries, external protocols, bridge/oracle endpoints), off-chain keeper/relayer infrastructure, and key management, assumed correct unless expressly listed. Test, mock and deploy-script files were read for context but are not the subject of the findings.

02 METHODOLOGY

AI-assisted broad analysis over the full source followed by targeted human review of business logic, value-moving paths, roles/permissions and higher-risk areas; cross-layer reconciliation; and live-state verification of the deployed settings (see §05). The review is manual + AI-assisted + static; it does not include formal verification or runtime fuzzing unless stated. Severity reflects exploitability + impact, never effort: **Critical** (direct path to loss of funds / full compromise) · **High** (exploitable, realistic conditions, material impact) · **Medium** (limited impact or unusual preconditions) · **Low** (defense-in-depth / best practice) · **Info** (observation).

03 FINDINGS

ID	SEVERITY	LAYER	FINDING
PROTO-T2-01	MEDIUM	contracts	Monthly interest double-charged for an overdue interval (pay-through vs deduction mismatch)
PROTO-T2-02	LOW	contracts	Default is terminal and unilaterally lender-triggerable after a short grace; no on-chain cure
PROTO-T2-03	LOW	contracts	Origination-fee unused-time refund gated on callback existence, contradicting its own comment
PROTO-T2-04	LOW	contracts	balanceOf overstates accrued interest after a roll / after maturity
PROTO-T2-07	LOW	contracts	KYC-revoked lender cannot cancel callback or accept roll ([redacted])
PROTO-T2-05	INFO	contracts	Accounting assumes standard non-fee, non-rebasing tokens
PROTO-T2-06	INFO	contracts	Upgradeable contracts lack storage gaps

MEDIUM PROTO-T2-01

Monthly interest double-charged for an overdue interval (pay-through vs deduction mismatch)

LOCATION

`Pool.sol: _repayInterestTo / dueInterestOf`

DESCRIPTION

In the monthly branch, interest is computed to endDate = max(block.timestamp, nextPaymentTimestamp), but _repayInterestTo reduces totalInterest and advances accrualTs only to nextPaymentTimestamp, which does not depend on block.timestamp. When repayInterest is called late, the interval [nextPaymentTimestamp, block.timestamp] is paid to the lender yet never deducted from totalInterest, so it is charged again on the next call and/or on final repay(). Traced numerically: a 3-month loan settled in same-block catch-up calls at maturity pays ~2× the intended interest.

IMPACT

Deterministic borrower overpayment (to lenders + spread to treasury) proportional to lateness. Not attacker-triggerable (repayInterest is onlyBorrower) — an accounting-correctness / fund-misallocation bug, not a theft vector.

RECOMMENDATION

Pay interest over the SAME interval that is deducted from totalInterest (bound endDate to nextPaymentTimestamp, or advance accrualTs/totalInterest to endDate); charge lateness solely through the penalty path.

Status: OPEN · Confidence: inference — traced numerically from code; no on-chain repro

LOW PROTO-T2-02

Default is terminal and unilaterally lender-triggerable after a short grace; no on-chain cure

LOCATION

`Pool.sol: repay/repayAll/repayInterest (nonDefaulted) & markPoolDefaulted/canBeDefaulted`

DESCRIPTION

All repayment paths carry nonDefaulted. Any single lender with principal can markPoolDefaulted once canBeDefaulted() is true (bullet: maturity + 3-day grace). defaultedAt is never cleared, so the borrower can no longer repay through the contract.

IMPACT

A lender can permanently block on-chain settlement of a borrower only marginally past grace; a borrower wishing to make lenders whole after an inadvertent default has no on-chain path. Appears intended (workout model) but concentrates timing power in any one lender.

RECOMMENDATION

Consider allowing repay() on a defaulted pool (on-chain make-whole) and/or a governor confirmation or longer/tunable grace before lender-initiated default becomes terminal.

Status: OPEN · Confidence: code-cited (design behavior)

LOW PROTO-T2-03 **Origination-fee unused-time refund gated on callback existence, contradicting its own comment**

LOCATION

`Pool.sol: _getOriginationFee`

DESCRIPTION

The unused-tenor origination-fee refund is applied only when the lender has an active callback, yet the inline comment describes the opposite (“if lender hasn’t created callback and borrower repays early, not all origination fee is used”). As written, a borrower who voluntarily repays early with no callback is charged the full origination fee for the whole tenor.

IMPACT

Borrower over-charged origination fee (to treasury) on early voluntary repayment without a callback — or, if the code is intended, a latent trap for maintainers. Either way a spec/implementation mismatch on a fee path.

RECOMMENDATION

Decide the intended policy and align code + comment.

Status: OPEN · Confidence: code-cited (intent ambiguous)

LOW PROTO-T2-04 **balanceOf overstates accrued interest after a roll / after maturity**

LOCATION

`Pool.sol: balanceOf vs acceptRoll`

DESCRIPTION

balanceOf prorates each position’s interest by (now-startAt)/(endAt-startAt). acceptRoll bumps totalInterest and extends maturity but does not update _lenderPositions, so a position’s endAt stays at the pre-roll maturity; once it passes, the ratio exceeds 1 and grows unbounded. Payment paths do not use positions, so payments are unaffected.

IMPACT

View-only inconsistency — off-chain integrators or contracts relying on balanceOf() read inflated balances post-roll/post-maturity.

RECOMMENDATION

Update positions on acceptRoll, or clamp the ratio at 1 (clamp now to endAt), or document balanceOf as unreliable after rolls.

Status: OPEN · Confidence: code-cited

LOW PROTO-T2-07 **KYC-revoked lender cannot cancel callback or accept roll ([redacted])**

LOCATION

`Pool.sol: cancelCallBack / acceptRoll / requestCallBack`

DESCRIPTION

cancelCallBack and acceptRoll are [redacted]. A lender holding an active callback who is blacklisted at [redacted] can no longer cancel it, so the active-callback counter stays inflated and requestRoll (needs zero callbacks) is blocked for the pool.

IMPACT

A revoked lender’s stale callback can lock the pool out of rolls. Resolvable (borrower can still repay, which clears the callback), so a griefing/liveness nuisance, not a fund risk.

RECOMMENDATION

Allow the borrower/governor to clear a callback for a non-member lender, or decrement the counter when a lender loses membership.

Status: OPEN · Confidence: code-cited

INFO PROTO-T2-05 **Accounting assumes standard non-fee, non-rebasing tokens**

LOCATION

`Pool.sol: _lend / _repayTo / _repayInterestTo`

DESCRIPTION

All transfers move nominal amounts directly between parties and the recorded principal/interest use those nominal amounts; a fee-on-transfer or rebasing token would drift from recorded values. Mitigated because pools may only use assets on [redacted]'s governor-curated allowlist.

IMPACT

Mis-accounting for non-standard tokens; mitigated by the allowlist.

RECOMMENDATION

Keep the allowlist restricted to standard stablecoins and document the assumption.

Status: OPEN · Confidence: code-cited

INFO PROTO-T2-06 **Upgradeable contracts lack storage gaps**

LOCATION

`Pool.sol / PoolFactory.sol / [redacted].sol`

DESCRIPTION

Pool, PoolFactory and [redacted] are upgradeable but declare no __gap; Pool in particular appends state after mappings. Beacon upgrades stay safe only if every future variable is strictly appended and inherited layouts are never reordered — and all pool proxies share one beacon implementation.

IMPACT

No current bug, but elevated storage-collision risk on future upgrades.

RECOMMENDATION

Add a __gap to each upgradeable contract and enforce storage-layout diff checks in CI.

Status: OPEN · Confidence: code-cited

04 WHAT'S SOLID

- Funds route peer-to-peer; the Pool never holds asset balances, eliminating a pooled-funds drain target.
- CEI respected: _lend and _repayTo update all state before token transfers, and external entrypoints carry nonReentrant.
- Membership gating is consistent; crucially repay/repayAll are onlyBorrower (not [redacted]), so a lender whose KYC is later revoked can still be repaid.
- Interest and penalty accrual are correctly frozen at defaultedAt across every view.
- Parameter space bounded (rateMantissa ≤ 1e18, spread/origination ≤ 1e18, penalty ≤ 1e19); constructor _disableInitializers + initializer guards present.

05 LIVE ON-CHAIN VERIFICATION

> **Pending deployed addresses.** Governance/config findings must reflect the deployed state, not the source. The items below are queued to be read live (via Etherscan V2 proxy `eth_call`) once [redacted] provides the deployed addresses + chains per system; each will be reported PASS / ACTION / VERIFY / INFO. Until then they are marked VERIFY.

- [redacted]: owner/governor, treasury, spreadRate, originationRate, incrementPerRoll, penaltyRatePerYear, and the stablecoin allowlist — confirm only standard non-FoT stablecoins.
- PoolFactory: owner (who can default pools and set beacon/[redacted]), poolBeacon implementation, and that the [redacted] linkage matches the intended [redacted].
- Per deployed Pool: rateMantissa, spread/origination/increment/penalty rates, tenor, depositWindow, isBulletLoan, maxSize vs currentSize, isPublic vs intended access.
- Beacon upgrade authority: who controls the UpgradeableBeacon behind every pool (single point upgrading ALL pools) and its timelock/multisig.
- Deployed timing vars (monthly round 30d, roll range 48h, grace 3d, factory minimums) — confirm mainnet values, not loosened testnet ones.

06 ACTION POINTS

PRIORITY	ID	ACTION	STATUS
P1	PROTO-T2-01	Monthly interest double-charged for an overdue interval (pay-through vs deduction mismatch)	◦ OPEN
P2	PROTO-T2-02	Default is terminal and unilaterally lender-triggerable after a short grace; no on-chain cure	◦ OPEN
P2	PROTO-T2-03	Origination-fee unused-time refund gated on callback existence, contradicting its own comment	◦ OPEN
P2	PROTO-T2-04	balanceOf overstates accrued interest after a roll / after maturity	◦ OPEN
P2	PROTO-T2-07	KYC-revoked lender cannot cancel callback or accept roll ([redacted])	◦ OPEN
P2	PROTO-T2-05	Accounting assumes standard non-fee, non-rebasing tokens	◦ OPEN
P2	PROTO-T2-06	Upgradeable contracts lack storage gaps	◦ OPEN

P0 = before deploy / now · P1 = soon · P2 = hardening. Status column is filled at the re-review round (report goes to v1.1: ◦ OPEN · ✔ FIXED · ~ ACKNOWLEDGED · ✗ NOT FIXED).

07 RESIDUAL RISK

After the recommended fixes, residual risk concentrates in the privileged roles (owner/manager/strategist/governor keys) and in operational posture — key custody, multisig thresholds, timelocks, and monitoring — which code alone cannot guarantee. Because this system holds user funds, a companion Economic Modeling Addendum accompanies this report and models the solvency and adverse-market mechanics summarized above. This is a point-in-time review; re-verify after any change to code, configuration, or deployed settings.