



JITTER PROTOCOL / FINAL MANUAL SECURITY AUDIT

Jitter Protocol Manual Security Audit

Core market security, arithmetic, rewards, adapters, registry, lifecycle, and deployment posture

65	38	25	0
CANONICAL FINDINGS	RESOLVED OR REMOVED	ACKNOWLEDGED / CONDITIONAL	OPEN

Report date	2026-08-17
Assessment baseline	64a477bf9ae945ff585f9dd70231945a9a741ea9 plus reviewed working-tree changes
Primary scope	Jitter core, math, oracle, registry, view, adapters, Jitter rewards, and experimental Jitter orderbook
Classification	Internal Final Security Assessment

Zzyzx Security Review

Contents

01	Executive assessment and status matrix
02	Scope, architecture, and methodology
03	Current release findings
04	Complex remediation evidence
05	Finding register
06	Release gates, limitations, and source record

1. Executive Assessment

Assessment outcome

No confirmed Critical or High attack chain was identified that allows an unprivileged user to steal Jitter V1 core principal. The High experimental-orderbook issue is deployment-excluded. The project explicitly acknowledges the residual PY-index domain, terminal reward dust, and registry-default risks; adapter validation and release reproducibility remain conditional work.

Status matrix

This matrix includes the full Jitter canonical population assigned from every historical audit, including resolved, removed, acknowledged, conditional, open, and deployment-excluded findings.

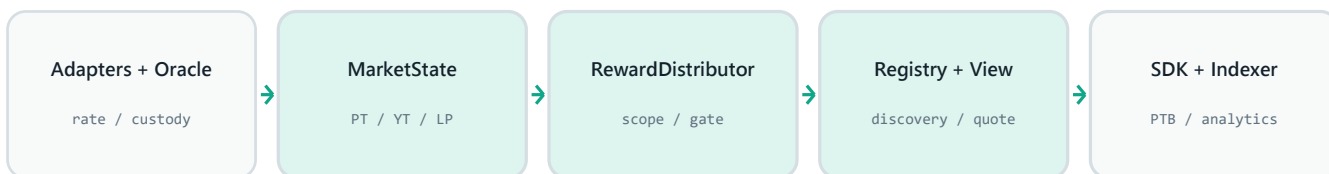
32 RESOLVED	6 ARCHITECTURE ELIMINATED	25 ACKNOWLEDGED
0 CONDITIONAL / PENDING	0 OPEN	2 DEPLOYMENT EXCLUDED

Release posture

The reviewed tree is not a release artifact until a clean commit is frozen, the acknowledged arithmetic, reward-dust, and registry controls are recorded in the release runbook, production adapters are signed off, excluded packages are removed from the deployment graph, and the full Move, formal, mutation, deployment, and operational suites pass from a clean checkout.

2. Scope, Architecture, and Methodology

The Jitter review covers market state, SY registration, PT and YT issuance, AMM math, user positions, liquidity, rewards, oracle aggregation, adapter binding, registry discovery, global version and pause controls, and deployment boundaries. External incentive and policy packages are outside this report and assessed independently.



Primary invariants

1. MarketState, Position, SY registration, adapter vault, and PriceAggregator identities must match exactly.
2. Persistent mutations must obey package version, pause, lifecycle, and reward-gate rules.
3. PT, YT, LP, SY reserves, fees, and reward liabilities must conserve value under the specified rounding direction.
4. Quote and execution must use the same discrete candidate state.
5. Rewards start only after exposure is established and stop at transfer, maturity, configuration cutoff, or campaign close.

Method

The review traces public entrypoints through owned and shared objects, capabilities, hot-potatoes, dynamic fields, custody, transfer, and event emission. Numerical work covers Q64, u128 and u256 intermediates, exact-in and exact-out solvers, maturity, remainder accounting, multi-user exits, and split-versus-batched equivalence.

3. Current Release Findings

These are the active Jitter release findings. Resolved and historically acknowledged findings remain visible in the complete domain register.

MAN-01 PY index arithmetic domain guard is not enforced

MEDIUM

ACKNOWLEDGED

CONDITION	The schema reserves PyIndexGuard fields, but observation, update, interest, and settlement do not enforce them.
IMPACT	Extreme but representable supply and index combinations may overflow a full-width intermediate and cause arithmetic denial of service.
PROJECT DECISION	The project will not activate an uncalibrated guard. Safe growth and window values have not been selected, and premature enforcement could reject legitimate index growth or block settlement. Deployment is restricted to the tested domain with monitoring and an upgrade path.
FUTURE HARDENING	Enable a calibrated step and window guard or prove a safe reduction that cannot overflow in the production domain.

MAN-02 CoinRewarder can retain allocated terminal fractional dust

MEDIUM

ACKNOWLEDGED

CONDITION	Calendar lots allocate budget through Q128 accumulators. After every claimant retires, sub-raw-unit fractions may remain allocated without a claimant able to cross one raw coin.
IMPACT	A small amount of reward budget may remain permanently accounted in the vault; user principal is unaffected.
PROJECT DECISION	The project accepts claimant-less sub-raw terminal dust. Its expected value is lower than the complexity and attack surface of proving claimant exhaustion or assigning a final claimant. Closed-lot residuals will be monitored and the decision reopened if they become material.
FUTURE HARDENING	Define a terminal policy that proves claimant exhaustion before treasury sweep, or deterministically assigns the final residual.

MAN-03 Ember freshness may use local read time

MEDIUM

CONDITIONAL

CONDITION	An old upstream rate can appear fresh if the adapter writes the local Clock time instead of the authoritative source update time.
IMPACT	A stale rate may pass freshness checks and affect conversion, quoting, or maturity settlement.
PROJECT DECISION	The project excludes Ember from the production dependency graph until the authoritative upstream timestamp and its validation rules are confirmed.
FUTURE HARDENING	Use Ember's authoritative update timestamp and reject zero, future, or stale values before production eligibility.

MAN-04 Current and NAVI payout equality is not proven

MEDIUM

CONDITIONAL

CONDITION	The core burns an exact SY obligation, but the adapter integration has not proven that the third-party redemption returns exactly that underlying amount.
IMPACT	Third-party rounding or fees could create an adapter deficit or surplus.
PROJECT DECISION	The project excludes Current and NAVI from production until exact payout equality is demonstrated against the live upstream ABI or the adapter is redesigned around actual output with a user minimum.
FUTURE HARDENING	Assert actual Coin.value equals the obligation, or redesign the request to settle actual output with a user min-output guard.

MAN-05 Experimental orderbook maker fee is fragmentation-sensitive

HIGH

DEPLOYMENT EXCLUDED

CONDITION	Each partial fill applies fee_ceil independently. Splitting one economic fill into many small fills can repeatedly charge at least one raw unit.
IMPACT	Maker escrow can be eroded by rounding; the package is not eligible for the V1 production graph.
PROJECT DECISION	The project excludes the experimental Jitter orderbook from V1 because hybrid orderbook routing is not required for launch and the fee invariant is not acceptable for production custody.
FUTURE HARDENING	Use order-level cumulative fee debt: $\text{ceil}(\text{cumulative_filled} * \text{rate})$ minus fee_already_paid.

MAN-06 Registry default can reference a delisted market

LOW

ACKNOWLEDGED

CONDITION	Default selection and delisting are not yet proven to preserve the invariant that the default is listed and active.
IMPACT	SDK and indexer discovery can route users toward an inactive market.
PROJECT DECISION	The project accepts this operational discovery risk because Registry does not authorize custody or execution and MarketState lifecycle, pause, and version checks remain fail-closed. SDK and indexer revalidation plus dashboard runbooks are the selected controls.
FUTURE HARDENING	Validate listed and active state on set and read; require a replacement or clear the default when delisting it.

4. Complex Remediation Evidence

4.1 Exact-out solver bracket recovery

The repaired path recovers the last valid raw PT input after an exponential probe leaves the quote domain, then resumes maximality search. This preserves feasible depth near pool exhaustion without weakening exact-output safety.

Resolved production excerpt: amm_math.move lines 1294-1339

```

1294     while (i < MAX_EXPONENTIAL_BRACKET_EVALUATIONS) {
1295         let (candidate_sy_out, _) = quote_exact_pt_for_sy_prepared(
1296             total_pt,
1297             total_sy_raw,
1298             total_sy_asset,
1299             sy_index_raw,
1300             rate_scalar,
1301             rate_anchor,
1302             fee_factor,
1303             high,
1304         );
1305         quoted_sy_out = candidate_sy_out;
1306         if (candidate_sy_out >= sy_out) {
1307             break
1308         };
1309         if (candidate_sy_out == 0 && previous_sy_out > 0) {
1310             // A power-of-two probe can jump past the curve's valid
1311             // domain. Recover the last quotable raw PT input so a target
1312             // between the previous probe and that boundary is not lost.
1313             let mut valid_low = previous_high;
1314             let mut invalid_high = high;
1315             let mut boundary_iteration: u64 = 0;
1316             while (
1317                 boundary_iteration < MAX_EXACT_IN_SEARCH_ITERATIONS
1318                 && valid_low < invalid_high - 1
1319             ) {
1320                 let mid = valid_low + (invalid_high - valid_low) / 2;
1321                 let (mid_sy_out, _) = quote_exact_pt_for_sy_prepared(
1322                     total_pt,
1323                     total_sy_raw,
1324                     total_sy_asset,

```

```

1325         sy_index_raw,
1326         rate_scalar,
1327         rate_anchor,
1328         fee_factor,
1329         mid,
1330     );
1331     if (mid_sy_out > 0) {
1332         valid_low = mid;
1333     } else {
1334         invalid_high = mid;
1335     };
1336     boundary_iteration = boundary_iteration + 1;
1337 };
1338 high = valid_low;
1339 let (boundary_sy_out, _) = quote_exact_pt_for_sy_prepared(

```

4.2 Calendar-based parallel reward lots

Each top-up has an explicit start, end, rate, and funded budget. New capital cannot backpay empty-vault history, while parallel lots allow deliberate APR changes without rewriting prior exposure.

Resolved production excerpt: `coin_rewarder.move` lines 849-888

```

849 fun add_funding_lot<R: drop>(
850     rewarder: &mut CoinRewarder<R>,
851     coin: Coin<R>,
852     start_delay_ms: u64,
853     duration_ms: u64,
854     now_ms: u64,
855 ) {
856     assert!(!rewarder.closed, E_REWARDER_CLOSED);
857     let budget = coin.value();
858     assert!(budget > 0, E_ZERO_AMOUNT);
859     assert!(duration_ms > 0, E_INVALID_SCHEDULE);
860     assert!(now_ms >= rewarder.last_updated_ms, E_TIME_REGRESSION);
861     assert!(rewarder.funding_lots.length() < MAX_FUNDING_LOTS, E_TOO_MANY_FUNDING_LOTS);
862
863     let start_ms_u256 = (now_ms as u256) + (start_delay_ms as u256);
864     assert!(start_ms_u256 <= MAX_U64, E_INVALID_SCHEDULE);
865     let end_ms_u256 = start_ms_u256 + (duration_ms as u256);
866     assert!(end_ms_u256 <= MAX_U64, E_INVALID_SCHEDULE);
867
868     let lot_id = rewarder.next_lot_id;
869     rewarder.next_lot_id = checked_add(lot_id, 1);
870     let start_ms = start_ms_u256 as u64;
871     let end_ms = end_ms_u256 as u64;
872     rewarder.reward_vault.join(coin.into_balance());
873     rewarder.funding_lots.push_back(FundingLot {
874         lot_id,
875         start_ms,
876         end_ms,
877         budget,
878         accounted_amount: 0,
879     });
880     assert_vault_accounting(rewarder);
881
882     event::emit(CoinRewarderFundedEvent {
883         rewarder_id: object::id(rewarder),
884         lot_id,
885         budget,
886         start_ms,
887         end_ms,
888     });

```

4.3 Acknowledged PY index guard boundary

Compatible PyIndexGuard fields exist, but field presence is not enforcement. The project keeps the guard disabled until safe values are calibrated, restricts deployment to the tested domain, monitors index growth, and retains an upgrade path. This is an acknowledged proof boundary, not a technical resolution.

Current production evidence: market_state.move lines 92-104

```
92  /// Reserved configuration and rolling state for a future PY-index growth guard.
93  ///
94  /// New markets store this as disabled. The current package does not read these
95  /// values while observing, updating, or settling the PY index.
96  public struct PyIndexGuard has copy, drop, store {
97      mode: u8,
98      max_step_growth_bps: u64,
99      max_window_growth_bps: u64,
100     window_ms: u64,
101     anchor_index_raw: u128,
102     anchor_timestamp_ms: u64,
103 }
104
```

5. Finding Register

The register below contains all 65 Jitter canonical findings assigned from the historical audit corpus. Original severity and current disposition remain separate.

Oracle and Adapter Findings

ID / ORIGINAL SEVERITY	FINDING / CURRENT DISPOSITION / REMEDIATION
ORC-01 High / For Testing	Demo adapter accepts a caller-supplied SY index Deployment Excluded - Deployment Excluded Keep the demo package testnet-only and reject it in production deployment manifests.
ORC-02 High	Adapter vault lacked canonical market and underlying binding Resolved - Resolved Bind request, vault, market, SY registration, and underlying type at both core and adapter layers.
ORC-03 High	Oracle source identity could be forged Resolved - Resolved Require an admin-registered source witness and exact aggregator rule binding.
ORC-04 Medium	Market did not bind one canonical price aggregator Resolved - Resolved Persist and validate the exact aggregator ID, market ID, freshness, and single-use PriceInfo.
ORC-05 Medium	Outlier filtering provides no independent single-source corroboration Acknowledged - Accepted Retain source-count, freshness, required-source, deviation monitoring, and emergency pause controls.
ORC-06 Medium	Adapter freshness could use read time instead of source update time Acknowledged - Acknowledged / Not Deployed Project decision: acknowledge this because Ember is not deployed. Before activation, use an authoritative upstream update timestamp and repeat freshness validation against the production ABI.
ORC-07 Medium	Upstream package upgrades can block adapter liveness Acknowledged - Accepted Operational Monitor upstream versions and retain adapter upgrade and recovery runbooks.
ORC-08 Medium	Scallop withdrawal queue lifecycle and storage exposure Acknowledged - Partial / Accepted Use owner cancel, admin approve or reject, linked storage, and explicit queued-reserve accounting.
ORC-09 Medium	Current and NAVI payout equality is not proven Acknowledged - Acknowledged / Not Deployed Project decision: acknowledge this because Current and NAVI are not deployed. Before activation, require actual payout equality before consuming the burn obligation and repeat lifecycle tests.
ORC-10 Low	Current rate narrows into a u64 domain Acknowledged - Acknowledged / Not Deployed Project decision: acknowledge this because Current is not deployed. Validate the production rate domain and overflow vectors before activation, and abort cleanly outside u64 representation.
ORC-11 Low	Scallop first withdrawal window starts with zero quota Acknowledged - Accepted Keeper Model Require a keeper refresh immediately after vault creation and monitor the baseline.
ORC-12 Low	Scallop accrual and version checks expand liveness dependencies Acknowledged - Accepted / Monitored Treat accrual and upstream version availability as monitored integration dependencies.
ORC-13 Low	NAVI AccountCap and residual balance lifecycle is incomplete Acknowledged - Acknowledged / Not Deployed Project decision: acknowledge this because NAVI is not deployed. Complete AccountCap, residual recovery, and migration semantics before activation.
ORC-14 Low	Oracle package originally lacked version and pause enforcement Resolved - Resolved Gate persistent oracle paths with package registration, version, and pause checks.
ORC-15 Info	Suilend conversion may read a pre-compound ratio Acknowledged - Acknowledged / Not Deployed Project decision: acknowledge this because Suilend is not deployed. Validate compound and refresh semantics against the production ABI before activation.

Jitter Core Findings

ID / ORIGINAL SEVERITY	FINDING / CURRENT DISPOSITION / REMEDIATION
JIT-01 Medium	PY index arithmetic domain guard is not enforced Acknowledged - Accepted / MAN-01 Project decision: acknowledge the unbounded extreme domain and keep PyIndexGuard disabled until safe growth and window values are calibrated. Deploy only in the tested domain, monitor index growth, and retain an upgrade path.
JIT-02 Medium	Global and profile rewarder composition could create topology drift Architecture Eliminated - Architecture Eliminated Bind one canonical distributor per market and validate scope topology atomically.
JIT-03 Medium	Maturity settlement depends on an authorized live keeper Acknowledged - Accepted Operational Use a canonical fresh PriceInfo plus keeper SLA, alerting, and maturity-settlement drills.
JIT-04 Medium	Disabling a RewardDistributor could freeze exits Resolved - Resolved Separate emergency pause from campaign close so liability-reducing exits remain available.
JIT-05 Medium	Small liquidity removal could round PT output to zero Resolved - Resolved Use candidate-state execution and reject zero-output removals; cover multi-user final-drain cases.
JIT-06 Medium	Market creation could accept pre-issued PT or YT supply Resolved - Resolved Require zero PT and YT TreasuryCap supply when a MarketState is created.
JIT-07 Medium	SY deregistration could strand an active market Resolved - Resolved Reject deregistration while any live market or liability still references the SY registration.
JIT-08 Medium	Underlying type was not fully enforced across mint and burn Resolved - Resolved Bind the exact underlying type through mint request, burn request, vault, and market state.
JIT-09 Medium	Fee recipient and treasury accounting could diverge Resolved - Resolved Use one on-chain treasury source of truth and exact recipient accounting.
JIT-10 Medium	Global pause lookup is linear in configured domains Acknowledged - Partially Resolved / Accepted Retain bounded operational use and explicit unpaused cleanup; accepted due to small pause sets.
JIT-11 Low	External YT coin bridge could bypass canonical reward mutation gates Architecture Eliminated - Architecture Eliminated Remove the external YT bridge and keep YT exposure changes on canonical gated position routes.
JIT-12 Low	A position could be destroyed while accrued interest remained Resolved - Resolved Require PT, YT, LP, and accrued interest to be zero before position destruction.
JIT-13 Low	A zero deadline allows an intentional PTB-level opt-out Acknowledged - Accepted Integration Risk Treat zero deadline as an explicit integration choice and enforce nonzero deadlines in user SDK routes.
JIT-14 Low	Post-swap event values could be recomputed from a different state Resolved - Resolved Emit values captured from the executed candidate state rather than recomputing after mutation.
JIT-15 Info	RouterMintEvent was dead or inconsistently emitted Resolved - Resolved Remove the unused event and its dead construction path.
JIT-16 Medium / Design	Ownerless minimum LP complicated final pool withdrawal Resolved - Resolved by C2 Adopt the C2 lifecycle so circulating LP can reach zero and reserves can be drained safely.

Rewards and Points Findings

ID / ORIGINAL SEVERITY	FINDING / CURRENT DISPOSITION / REMEDIATION
EXT-01 Medium	Point accounting could run without a canonical market reward gate Architecture Eliminated - Architecture Eliminated Attach PointConfig and reward topology to the canonical market and distributor gate.
EXT-02 Medium	Global point rewarders could contaminate unrelated markets Architecture Eliminated - Architecture Eliminated Use market-scoped rewarder selection instead of a mutable global union.
EXT-03 Medium	Rewarder re-registration could backpay historical exposure Resolved - Resolved Bind registration generation and start-time baselines to prevent historical backpay.
EXT-04 Medium	LP referral state lacked exact policy and project binding Resolved - Resolved Bind referral state to the exact policy, project, position, owner, and registration generation.
EXT-05 Medium	LP point claims were not reliably reachable Resolved - Resolved Provide canonical permissionless checkpoint and claim routes with detached pending preservation.
EXT-06 Low	Pool reward scope used ambiguous zero-address semantics Acknowledged - Accepted/Architecture Clarified Define scope topology explicitly and pair LP and pool checkpoint capabilities.
EXT-07 Low	Reward withdrawal authority lacked a narrow governance lane Acknowledged - Accepted Governance Keep withdrawal behind a narrow ACL or multisig governance operation with auditable events.
EXT-08 Low	Reward accounting could strand rounding and terminal dust Acknowledged - Accepted / MAN-02 Project decision: acknowledge claimant-less sub-raw terminal dust because principal is unaffected and a terminal claimant proof would add disproportionate state and attack surface. Monitor closed-lot residuals and reopen if material.
EXT-09 Low	PointCap was not bound to the canonical Scoreboard Architecture Eliminated - Architecture Eliminated Use Project and Stamp witnesses in canonical integrations; isolate manual PointCap issuance.
EXT-10 Low	PointConfig catch-up could require too many versions Acknowledged - Accepted + Mitigated Use permissionless bounded checkpoints and require state to catch up before mutation.
EXT-11 Low	LP multiplier updates could skip an uncheckpointed interval Resolved - Resolved Version parameters by effective time so old intervals settle under old configuration.

Policy and Auxiliary Findings

ID / ORIGINAL SEVERITY	FINDING / CURRENT DISPOSITION / REMEDIATION
AUX-02 Medium	Reward pause could freeze order escrow exits Resolved - Resolved for canonical lifecycle Separate emergency pause from normal campaign close and preserve canonical escrow exits.
AUX-03 Medium	Experimental Jitter orderbook retained unbounded traversal risk Deployment Excluded - Deployment Excluded Exclude the experimental package from V1; retain keeper cleanup and bounded level operations.
AUX-06 Medium	Orderbook reward exposure was not a reliable canonical source Architecture Eliminated - Architecture Excluded Exclude orderbook exposure from canonical rewards until a stable topology exists.
AUX-10 Low	Experimental orderbook could strand terminal fee dust Resolved - Resolved in experimental package Use cumulative fee accounting in the experimental package before deployment.

Math and Solver Findings

ID / ORIGINAL SEVERITY	FINDING / CURRENT DISPOSITION / REMEDIATION
MATH-01 Medium	SY-to-PT quote and execution could disagree at a curve boundary Resolved - Resolved Execute the exact candidate state returned by the quote solver and preserve rounding direction.
MATH-02 Low	Exact-out search could miss feasible depth after an invalid probe Resolved - Resolved Recover the last valid bracket after a zero-output probe, then continue maximality search.
MATH-03 Low / Performance	exp2 implementation had excessive gas consumption Acknowledged - Accepted / Optimized Use the optimized bounded implementation and keep gas and dense-vector regressions.
MATH-04 Low	last_In used a sentinel that overlapped a valid value Resolved - Resolved Replace the overlapping sentinel with an explicit option or validity flag.
MATH-05 Low	Aggregate multiplication can exceed the intended arithmetic domain Acknowledged - Acknowledged / MAN-01 Project decision: acknowledge the remaining extreme index frontier because the reviewed replacement package is not live and normal validated-domain arithmetic is covered. Recalibrate PyIndexGuard or extend the proof before enabling a broader production domain.
MATH-06 Info	Unused public math helpers enlarged the audited ABI Resolved - Resolved Remove unused public helpers and keep only audited production math surfaces.
MATH-07 Info	Signed exponential inputs require an explicit negative domain Acknowledged - Accepted Domain Boundary Define the signed Q64 negative domain and fail closed outside it.

Admin and Registry Findings

ID / ORIGINAL SEVERITY	FINDING / CURRENT DISPOSITION / REMEDIATION
ADM-01 Low	Registry allowed multiple instances and raw object IDs Acknowledged - Partial / Accepted Use canonical registry discovery, exact IDs, and explicit multi-instance administration.
ADM-02 Low	Zero-output checks had ambiguous zero-basis-point semantics Resolved - Resolved Validate nonzero result semantics independently from the configured basis-point value.
ADM-03 Info	AdminCap remains a broad trusted authority Acknowledged - Accepted Trust Assumption Treat AdminCap as a multisig trust boundary and prefer scoped ACL operations.
ADM-04 Info	ACL events and role tombstones were incomplete Acknowledged - Accepted / Event Hardened Emit role lifecycle events and prevent deleted-role generation reuse.
ADM-05 Info	Registry default state can outlive a delisted market Acknowledged - Accepted / MAN-06 Project decision: acknowledge the discovery-layer default risk because transaction paths still fail closed. Require SDK and indexer revalidation plus dashboard and lifecycle runbooks; retain on-chain hardening as a future option.

Version-Control Findings

ID / ORIGINAL SEVERITY	FINDING / CURRENT DISPOSITION / REMEDIATION
VC-01 High	Root config did not fully validate its own package version Resolved - Resolved Assert the root package marker and allowed version before governance and downstream checks.
VC-02 Medium	Deployment discovery could miss a nested package marker Resolved - Resolved Recursively discover package markers and regression-test deployment manifests.
VC-03 Low	Position transfer-lock mutation lacked a root version gate Resolved - Resolved Require GlobalConfig and core package version on persistent position mutators.
VC-04 Medium	A paused adapter could still create new custody state Resolved - Resolved Use strict-active checks for creation and version-only checks only for liability reduction.
VC-05 Low	Package-registration resume was not idempotent Resolved - Resolved Make same-version registration idempotent and require explicit approval for version changes.
VC-06 Info	NAVI exposed an ungated public helper Resolved - Resolved Reduce the helper to private scope and expose only the gated ticket route.
VC-07 Low	A paused campaign could be re-enabled through a recovery path Resolved - Resolved Permit disable under recovery rules but require strict package-active status to enable.

6. Release Gates, Limitations, and Source Record

Mandatory release gates

1. Freeze a clean release commit and dependency graph.
2. Record the accepted PY-index domain, monitoring threshold, and upgrade trigger.
3. Record the accepted terminal CoinRewarder dust tolerance and reopen threshold.
4. Require SDK and indexer revalidation for registry defaults.
5. Enforce adapter eligibility and package exclusions.
6. Re-run first-party Move packages, formal proofs, and mutation checks with pinned tools.
7. Exercise multisig, ACL, keeper, oracle, pause, upgrade, maturity, reward-close, and recovery runbooks.
8. Obtain final independent sign-off on the frozen release commit and published dependency graph.

Limitations

This independent security assessment is not a guarantee of defect absence and does not prove third-party protocol behavior, frontend correctness, RPC availability, wallet behavior, key custody, governance honesty, or economic assumptions.

Source record

The register reconciles the final manual audit source, the July 2026 status and working reviews, the July 27 full audit, the August 4 DEEP audit, subsequent lifecycle and checkpoint reviews, and formal verification work through August 17, 2026.