PUBLIC SAMPLE — REDACTED. This is a genuine Vari review with the client's identity and identifying scope details removed (shown as [redacted]). Methodology, findings, severities, and recommendations are unaltered — shared as a representative example of Vari's full-stack review work.

VARI

FULL-STACK SECURITY REVIEW REPORT

SECURITY REVIEW

[redacted] — [redacted] & [redacted] + Omnichain Stablecoin

CLIENT [redacted]	REPORT DATE 2026-08-11
REVIEW WINDOW 2026-08-11 (point-in-time)	TIER Full — full-stack
CODE REVIEWED branch [redacted] (pre-deploy candidate)	AUDITED BASELINE [redacted], tag [redacted] ([redacted])
LEAD REVIEWER Vari	CLASSIFICATION Public sample — client identity redacted

00 SCOPE & VERSION CONTEXT

This engagement is a full-stack review of the [redacted] omnichain stablecoin system across five layers: **smart contracts, deployment & configuration, cross-chain (LayerZero V2), frontend/backend/API, and DNS/domain hygiene**. It also includes a SEAL-framework alignment pass and dedicated protocol-specific economic attack research.

Contracts reviewed: the core system — `[redacted]`, `[redacted]` (`[redacted]`), `[redacted]+`, `StakingRewardsDistributor`, the redeem/plus silos, the LayerZero OFTs/adapters, and both composers — plus their interfaces, deploy scripts, and deployment config.

> **Version note (critical to how findings are read).** Per the client, the codebase reviewed here — the [redacted] branch — is the set of improvements made **since** [redacted]'s audit and is the **next deployment candidate**; it is **not** the bytecode currently live on mainnet. The client states the **live mainnet contracts are the [redacted]-audited version**. Accordingly, code-level findings below are assessed against the reviewed pre-deployment code and are best treated as **must-resolve-before-deploy** items, while governance/configuration findings (upgrade authority, collateral egress, KYC config, LayerZero wiring) apply to the live system as well.
>
> KEY ADDRESSES (Ethereum): [redacted] · [redacted]+ [redacted] · Admin Safe (3-of-4) [redacted] · Fee treasury [redacted]

01 EXECUTIVE SUMMARY

[redacted] is a genuinely well-engineered protocol. It has already been through a formal [redacted] audit plus several internal remediation rounds, and that discipline shows: the core mint/redeem/backing math is value-conservative, reentrancy is handled consistently, the unbacked-mint cap (timelock + guardian) is sound, share-inflation defenses are in place, the LayerZero [redacted] adapter is correctly locked down, and the domain even runs DNSSEC and a strict DMARC policy — all things most live protocols get wrong. **We found no critical vulnerability and no permissionless path to mint unbacked tokens, drain collateral, or double-spend across chains.**

What we did find is concentrated in the two layers a contract-only audit never covers, and which the protocol's own history proves matter most: **governance/configuration and deployment provenance**. The stablecoin's solvency rests on a small set of powerful, uncapped, un-timelocked admin levers (collateral withdrawal, instant upgradeability); its depeg protection is opt-in and, in the deploy scripts as written, left un-armed; a routine re-run of the deploy tooling could flip a system-wide KYC switch. None of these is a bug an attacker triggers alone, but together they are exactly the surface that has drained "audited" protocols throughout 2025–2026.

0	3	7	13	12
CRITICAL	HIGH	MEDIUM	LOW	INFO

The three High-severity items, all governance/configuration rather than code exploits: **(H-01)** `withdrawCollateral` is an uncapped, un-timelocked lever that can undercollateralize the stablecoin in one transaction, with its on-chain guardrails inert by default; **(H-02)** UUPS upgrades have no timelock, so implementation logic for the contracts that hold all collateral can be replaced instantly; **(H-03)** per-collateral oracle pricing is opt-in and the deploy scripts arm no feeds, leaving collateral priced 1:1 and exposed to a cross-asset depeg drain. Each is actionable before the next deployment.

SEVERITY	COUNT	THEME
CRITICAL	0	—
HIGH	3	Admin collateral lever; no upgrade timelock; depeg/oracle config
MEDIUM	7	Cross-chain compliance gap; KYC config drift; handover/wiring verification; LZ reorg config; API proxy abuse; admin session hardening; account/infra security
LOW	13	Pause asymmetries; cross-chain accounting; DNS CAA + DNSSEC denial anomaly; frontend CSP; disclosure channel; dust/rounding edges
INFORMATIONAL	12	Hardening, hygiene, and monitoring recommendations

02 METHODOLOGY

The review combined AI-assisted broad analysis with targeted human review across all five layers. Automated analysis covered the full contract set for known vulnerability classes and economic surfaces; manual review then examined business logic, the collateral/peg/backing invariants, cross-layer interactions (frontend ↔ contracts ↔ config ↔ deployment), and protocol-specific economic attacks. Every finding was validated by review; where confirmation requires live on-chain state we say so explicitly rather than assume.

```
PROCESS :: SCOPE LOCK → AI-ASSISTED SWEEP → PER-MODULE HUMAN REVIEW → CROSS-LAYER + ECONOMIC ANALYSIS → SEAL ALIGNMENT → DNS/BACKEND → PRIORITIZATION → REPORT
```

This review aligns to the **SEAL (Security Alliance) frameworks** for external smart-contract reviews and protocol operations (see §10). It is a point-in-time assessment and, consistent with both SEAL guidance and our own disclaimer, is not a guarantee of bug-free code and does not replace continuous security practice, formal verification, or exhaustive novel economic modeling.

03 SEVERITY DEFINITIONS

LEVEL	DEFINITION
CRITICAL	Directly exploitable path to loss of user or protocol funds, or complete system compromise. Fix before anything else.
HIGH	Exploitable under realistic conditions with material impact on funds, permissions, or data — or a flaw that becomes critical when combined with common circumstances (incl. privileged-key or single-config compromise of the entire fund base).
MEDIUM	Meaningful weakness with limited direct impact or requiring unusual preconditions; fix promptly.
LOW	Minor issue, defense-in-depth improvement, or best-practice deviation with low direct risk.
INFO	Observation, hardening suggestion, or hygiene note. No direct security impact.

Severity reflects exploitability × impact, not effort to find. Centralization findings (H-01, H-02) require malicious or compromised privileged keys rather than a permissionless exploit; we rate them High because their blast radius is the entire collateral base and they have no on-chain mitigation as configured.

04 FINDINGS

> Findings are ordered by severity. IDs use `PROTO-{SEV}-{NN}`. "Layer" maps the finding to one of Vari's five review layers. "Status" reflects the reviewed pre-deployment code and, where relevant, the action required.

HIGH PROTO-H-01 **withdrawCollateral is an uncapped, un-timelocked lever that can undercollateralize the stablecoin in one transaction**

`CONTRACT / SOLVENCY`

LAYER	Contracts / Governance
LOCATION	[redacted].sol:406–444
DESCRIPTION	`withdrawCollateral` decrements `collateralBalance[asset]` and transfers collateral to an arbitrary address **without reducing [redacted]/[redacted] supply** — so outstanding stablecoin is unchanged while its backing drops. The two on-chain guardrails (`enforceWithdrawalAllowlist` and the per-epoch `withdrawalEpochCap`) are **inert by default**. On an un-armed deployment a single `COLLATERAL_MANAGER` key can move up to 100% of every collateral in one transaction, with no cap, no timelock, and only an after-the-fact event. This is intended ("collateral must leave to fund PayFi"), but the design places the entire solvency of the stablecoin behind one key with no on-chain brake.
IMPACT	Immediate, total undercollateralization; redemptions become first-come-first-served and then revert. Single largest solvency exposure in the system.
SCENARIO	Compromised or malicious `COLLATERAL_MANAGER` calls `withdrawCollateral(USDC, bal, attacker)` and the same for USDT — all [redacted] is now unbacked.
RECOMMENDATION	Egress can't be removed, but bound it on-chain at launch, not opt-in: (1) arm `enforceWithdrawalAllowlist` with known custody addresses; (2) arm a conservative per-epoch `withdrawalRateLimit` so no single key can outrun monitoring; (3) hold `COLLATERAL_MANAGER` in a timelocked multisig distinct from `DEFAULT_ADMIN` ; (4) consider an on-chain collateralization floor that rejects withdrawals below a governance-set ratio; (5) publish proof-of-reserves and monitor `CollateralWithdrawn` .
CONFIDENCE	High.

HIGH PROTO-H-02 **No timelock on UUPS upgrades — implementation of the contracts holding all collateral can be replaced instantly**

CONFIG / GOVERNANCE

LAYER	Deployment / Governance
LOCATION	All six core proxies are UUPS; `_authorizeUpgrade` gated by `DEFAULT_ADMIN`/owner = the multisig. No `TimelockController` in the deploy config.
DESCRIPTION	There is zero delay between an upgrade decision and its execution. The multisig can replace the implementation of [redacted] or [redacted] — which custody and authorize movement of all collateral — in a single transaction, with no user exit window. For a live stablecoin this is the largest standing trust assumption after H-01.
IMPACT	A compromised/phished multisig quorum, or a malicious insider quorum, can push a draining implementation with no on-chain warning and no time for holders to exit.
SCENARIO	Signers are phished into a single `upgradeToAndCall` on [redacted]/[redacted]; collateral is swept before anyone reacts.
RECOMMENDATION	Route upgrade authority through an OZ `TimelockController` (24–72h) with the multisig as proposer, giving holders an exit window; publish the timelock address and monitor `Upgraded`/queued operations. If a timelock is deferred, document the accepted risk explicitly and compensate with real-time upgrade monitoring + alerting.
CONFIDENCE	High.

HIGH PROTO-H-03 **Depeg exposure: per-collateral oracle pricing is opt-in and the deploy scripts arm no feeds — collateral priced 1:1**

`CONTRACT / CONFIG / ECONOMIC`

LAYER	Contracts / Deployment / Economic
LOCATION	[redacted]`.sol:615–627, 639–646` (no-feed branch prices 1:1); `deploy/01-FullSystem.ts` and `deployConfig.mainnet.ts` arm no feeds.
DESCRIPTION	The reviewed branch adds Chainlink per-collateral pricing (asymmetric: mint credits at `min(price,$1)`, redeem pays at real price) — a genuinely correct design that, **with feeds armed**, is provably value-conservative across any asset pair (verified). But pricing is **opt-in per asset**: a collateral with no feed is priced at a hard 1:1. The deploy scripts set no feeds, and the audited-live version predates this mechanism entirely (all collateral effectively 1:1, mitigated only by pause + the unbacked cap). With two collaterals in 1:1 mode, a depeg of one lets an arber mint with the cheap asset at par and redeem the healthy asset at par, draining the healthy reserve.
IMPACT	Reserve drain of the still-pegged collateral during any real depeg (e.g. the Mar-2023 USDC → $0.88 event), bounded per block by `maxRedeemPerBlock` (1,000,000e18) but repeatable. A single un-fed depeggable asset compromises the backing of every other asset.
SCENARIO	USDT trades $0.92, feeds unset. Deposit 1,000,000 USDT → mint 1,000,000 [redacted] (valued 1:1) → redeem into USDC at par → ~$80k extracted per rotation until USDC reserve is exhausted.
RECOMMENDATION	Treat "a validated feed is configured for every funded collateral" as a launch-blocking invariant. Either require `collateralFeed[asset] != 0` before an asset can accrue balance, or make no-feed assets non-mintable (fail-closed) rather than 1:1. Add the feed addresses + `feedMaxAge` to deploy config with a deploy-time assertion, and confirm on-chain `isOraclePriced(USDC)` and `isOraclePriced(USDT)` are true before relying on the clamp.
CONFIDENCE	High (mechanism + math). **Confirmed on-chain (2026-08-11):** the live [redacted] has no oracle (isOraclePriced reverts) → live collateral is priced 1:1 today, with no unbacked cap either; depeg is mitigated only by pause. Arm feeds as the branch deploys.

MEDIUM PROTO-M-01 **[redacted] Composer does not compliance-check the cross-chain share recipient, unlike its [redacted]+ sibling**

CONTRACT / CROSS-CHAIN

LAYER	Contracts / Cross-chain
LOCATION	`[redacted] Composer.sol:332–361` (no beneficiary check) vs `[redacted]+ Composer.sol:98–111` (has one)
DESCRIPTION	On the cross-chain mint path the composer validates the originator ([redacted] + hub blacklist) but never validates the destination beneficiary `_sendParam.to`, to which minted [redacted] is delivered on the spoke. The [redacted]+ composer deliberately added exactly this beneficiary check to stop a restricted user routing shares to a fresh address; the [redacted] composer has the identical exposure and no equivalent guard. The only backstop is spoke-side quarantine keyed off the spoke blacklist, and cross-chain restriction sets are explicitly non-atomic.
IMPACT	A credentialed depositor can mint [redacted] directly to an address blacklisted on the hub but not yet synced on the spoke — weakening the same compliance control the team judged worth enforcing on [redacted]+.
RECOMMENDATION	Mirror the [redacted]+ beneficiary check: decode `_sendParam.to` and reject if `isBlacklisted(beneficiary)`, using the same malformed-message → refund pattern.
CONFIDENCE	High (the asymmetry is definite); real-world exploitability depends on blacklist sync lag.

MEDIUM PROTO-M-02 **[redacted] config is non-zero while the live system is KYC-off — a routine deploy re-run can flip a system-wide switch** CONFIG

LAYER	Deployment / Config
LOCATION	`deployConfig.mainnet.ts` ([redacted]`Address = 0xb0B5…E666`, `policyId` [redacted]); `01-FullSystem.ts` Phase-2 re-applies `set`[redacted]`Config` on drift.
DESCRIPTION	A non-zero [redacted]`Address` enforces KYC on every mint/redeem, including the cross-chain composer path. The team states live is KYC-off ([redacted]`Address == 0`). But Phase-2 of the deploy script is idempotent and applies deltas: run against the live KYC-off system to change any parameter, and it detects the mismatch and calls `set`[redacted]`Config`, silently turning KYC on.
IMPACT	A routine re-run can lock every non-credentialed holder out of redemption (funds frozen) and block mints; if the address/policy is wrong, all mint/redeem revert — protocol-wide DoS — from one config action with no guardrail.
RECOMMENDATION	Reconcile config to live on-chain state now: if KYC is intended off, set config [redacted]`Address = address(0)`. Gate the [redacted] write behind an explicit flag rather than automatic drift-application, and e2e-test a real mint/redeem + composer path before any re-run if KYC is intended on.
CONFIDENCE	Config value is definite. **Confirmed on-chain (2026-08-11):** live [redacted]`Address == 0x0` (KYC off) — so a re-run with the non-zero config would flip KYC ON. Set config to `address(0)` unless KYC is intended.

MEDIUM PROTO-M-03

Silo wiring-lock and two-step ownership handover are not verified — deployer/owner keys may retain control of escrowed user funds

CONFIG / DEPLOY

LAYER	Deployment / Contracts
LOCATION	[redacted]RedeemSilo.sol / [redacted]+Silo.sol (setVault / setStakingVault repointable until lockWiring()); 04-Handover.ts:224–283 (Ownable2Step transferOwnership only, no acceptOwnership verification; failures caught and logged, script still exits 0).
DESCRIPTION	The silos hold users' escrowed redemption/cooldown funds and trust a settable vault address until lockWiring() is called. Separately, the silos and distributor use two-step ownership: transferOwnership only sets a pending owner — authority does not move until the multisig calls acceptOwnership(), which the handover script prints as a reminder but never verifies. All handover failures are swallowed by try/catch.
IMPACT	If lockWiring() is skipped and the silo owner key is compromised, escrowed funds can be drained by repointing the vault. If acceptOwnership is forgotten, the deployer EOA remains full owner of the silos/distributor indefinitely — a hot-key single point over user funds.
RECOMMENDATION	Add a post-handover verification pass that asserts, on-chain, owner()/DEFAULT_ADMIN == multisig on every contract, the deployer holds no roles anywhere, and wiringLocked() == true on both silos. Make handover fail loudly. (Note: OPEN_ITEMS.md reports several of these verified on mainnet — confirm on the live deployment.)
CONFIDENCE	High on the script behavior; live state must be confirmed on-chain.

MEDIUM PROTO-M-04

LayerZero block-confirmation config is ambiguous ([1,1] vs [15,20]) and enforced compose-gas is unverified — BSC reorg / stuck-message risk

CONFIG / CROSS-CHAIN

LAYER	Deployment / Cross-chain
DESCRIPTION	Internal docs contradict each other on whether ETH⇄BSC confirmations were raised from [1,1] to [15,20] (OPEN_ITEMS.md says raised; SECURITY_AUDIT_R5_POSTFIX.md says the repo still holds [1,1]). BSC is reorg-prone; 1-block confirmation is unsafe. Enforced options/gas for the heavier lzCompose deposit path (vs the documented ~200k lzReceive reserve) are likewise unverified.
IMPACT	If a re-wire pushes [1,1] on-chain, cross-chain messages become exposed to reorg-based re-processing. If compose gas is under-provisioned, composes revert and user funds sit on the composer awaiting rescue (degraded availability, not loss).
RECOMMENDATION	Confirm on-chain ULNConfig.confirmations is the intended [15,20] on both directions and that repo source matches; gas-profile the worst-case deposit/redeem compose and set enforced options with headroom; verify OFT peers on both hub adapters and spoke OFTs.
CONFIDENCE	Plausible (the internal-doc contradiction is real). Verify on-chain.

MEDIUM PROTO-M-05 **API proxy access control is a spoofable Referer check with no rate/complexity limiting — open GraphQL proxy** BACKEND

LAYER	Backend / API
LOCATION	`app/api/subsquid/route.ts:26–36` ; `app/api/points-subsquid/route.ts:26–36`
DESCRIPTION	The only gate for non-server callers is a `Referer` -host match, which any non-browser client can spoof. There is no rate limiting, no per-session quota, and no GraphQL depth/complexity limiting or persisted-operation allow-list. (The upstream master token is correctly not exposed to these callers, which caps severity.)
IMPACT	Anyone can use the app server as a free proxy to the Subsquid indexers with arbitrary GraphQL and no throttling — abuse amplification and DoS of both the app and the indexer.
RECOMMENDATION	Do not treat `Referer` / `Origin` as authentication. Add per-IP/per-session rate limiting and GraphQL cost/depth limiting or a persisted-operation allow-list; keep the token attachment but still throttle.
CONFIDENCE	High.

MEDIUM PROTO-M-06 **[redacted] admin SIWE session lacks domain binding + expiry and is returned to the browser as a long-lived replayable bearer credential**

BACKEND / AUTH

LAYER	Backend / Frontend auth
LOCATION	`lib/[redacted]/admin-auth.ts:20–55` ; `features/[redacted]/hooks/use-[redacted]-admin-login.ts:27–85`
DESCRIPTION	The admin flow is genuinely server-verified and address-gated (good — not client-side theater), but `siweMessage.verify()` is called without the `domain` parameter and the SIWE message carries no `expirationTime` , so the signed credential is valid forever and not domain-bound. The nonce is not rotated/invalidated on verify, and on success the server returns `{signature, message}` to the client, which stores it in browser JS — a long-lived (up to 30-day cookie) replayable bearer credential.
IMPACT	Any exposure of the signature+message+nonce triple (XSS — see L-04 CSP — a leaked log, a shared machine) yields admin access for the cookie lifetime, with no expiry to bound the window.
RECOMMENDATION	Pass `domain` into `verify()` ; include and enforce a short `expirationTime` ; make the nonce server-issued, single-use, and rotated on verify; hold the session as an httpOnly server-side record (opaque id) rather than returning the raw signature to client JS. Confirm the admin verify/nonce routes set cookies `httpOnly; secure; sameSite=strict` .
CONFIDENCE	High on domain/expiry/client-storage; nonce single-use is Plausible pending review of the nonce route.

MEDIUM PROTO-M-07 **Account & infrastructure security — enforce hardware-key 2FA and lockdown across registrar, DNS, hosting, source, and infra**

OPERATIONAL / FRONTEND

LAYER	Operational / Frontend-DNS
DESCRIPTION	The contracts can be flawless and users still lose funds through the "keys beat code" path: takeover of the registrar ([redacted]), DNS, hosting ([redacted]), source (GitHub), or CI/cloud accounts lets an attacker repoint the domain or ship a poisoned frontend. Several of 2026's largest losses and the CoW-Swap-class frontend attacks came this way, not through contract bugs. The team's own SEAL self-assessment already marks account security as **ACTION REQUIRED**.
IMPACT	Frontend hijack / domain repoint / malicious deploy → users sign drainer approvals with the real contracts untouched. The single highest-leverage non-contract risk for a live protocol.
SCENARIO	A phished [redacted] or [redacted] credential (no hardware 2FA) lets an attacker repoint [redacted] or deploy a wallet-drainer build; users approve malicious spenders.
RECOMMENDATION	Enforce hardware-key (FIDO2) 2FA and account lockdown across registrar, DNS, hosting, source, and cloud/CI; enable registrar lock (DNSSEC already on — good); restrict and review deploy permissions; add DNS-change and deploy monitoring with alerting. Follow the SEAL Operational Security and Account Management practices.
CONFIDENCE	High that the surface exists; current 2FA/account posture is an operational item to verify and enforce.

LOW PROTO-L-01 **No CAA record on [redacted] — any CA may issue certificates for the domain** DNS

LAYER	DNS / Domain
DESCRIPTION	[redacted] publishes no CAA record. DNSSEC (good), SPF, and DMARC `p=reject` (both strong) are present, but without CAA there is no DNS-level restriction on which certificate authorities may issue for the domain, widening the mis-issuance surface for a phishing/interception cert.
IMPACT	A mis-issued certificate (via a compromised or tricked CA) would face no CAA barrier — a precondition many frontend/interception attacks rely on.
RECOMMENDATION	Publish a CAA record restricting issuance to the CA(s) actually used ([redacted]'s issuer, e.g. Let's Encrypt, plus any Google Trust Services for Workspace), with an `iodef` mailto for violation reports. Since DNSSEC is already enabled, CAA is cheap defense-in-depth.
CONFIDENCE	High (confirmed no CAA via authenticated DoH).

LOW PROTO-L-02 **Frontend CSP is Report-Only and permits 'unsafe-inline' scripts — no active XSS defense-in-depth** FRONTEND

LAYER	Frontend
LOCATION	`next.config.ts:10–41`
DESCRIPTION	The strong CSP is delivered only as `Content-Security-Policy-Report-Only` (never blocks) and allows `script-src 'unsafe-inline'`. Only `frame-ancestors 'none'` is enforced. HSTS, X-Frame-Options DENY, nosniff, Referrer-Policy, Permissions-Policy are correctly enforced. This compounds M-06 (admin credential in client JS).
IMPACT	No CSP-based mitigation is active; an injected script runs unhindered and could exfiltrate the [redacted] admin credential.
RECOMMENDATION	Promote to enforcing CSP after the observation window; replace `'unsafe-inline'` with nonce/hash-based script allow-listing (Next supports nonces); add real RPC + `chainlist.onerpc.com` hosts to `connect-src` first.
CONFIDENCE	High.

LOW PROTO-L-03 **unstake MIN_SHARES bypass enables a donation deposit-DoS at low TVL** CONTRACT

LAYER	Contracts
LOCATION	[redacted]`+.sol:447–450, 699–701` (the `_unstaking` skip of `_checkMinShares`)
DESCRIPTION	`unstake` is the one exit that skips the MIN_SHARES floor. A holder who is ~100% of supply can cooldown all-but-1-wei and unstake, leaving `totalSupply` at 1 wei; a subsequent large [redacted] donation inflates per-share NAV so that every normal `deposit` (which must mint ≥ 1e18 shares) reverts with `MinSharesViolation`.
IMPACT	Liveness/DoS of new deposits at low TVL. No theft (the floor still bounds any rounding loss; the donor sacrifices capital). Admin recovery is awkward because `burnAssets` cannot reduce supply below 1e18.
RECOMMENDATION	Enforce the floor on `unstake` too, permitting only `totalSupply → 0` or `≥ MIN_SHARES`; or permanently seed/lock ≥ MIN_SHARES of dead shares at genesis so supply can never reach dust.
CONFIDENCE	High mechanism; real impact limited to bootstrap/low-liquidity windows.

LOW PROTO-L-04 **Pause asymmetry traps user escrow: cancelRedeem is pause-gated while solver completions can bypass pause** CONTRACT

LAYER	Contracts
LOCATION	[redacted].sol:666 (`cancelRedeem` is `whenNotPaused`) vs the completion path's `completionsAllowedWhilePaused` hatch.
DESCRIPTION	During a pause the solver can still settle queued redemptions (if completions-while-paused is enabled), but users cannot `cancelRedeem` to reclaim their own escrowed [redacted]. The asymmetry lets the protocol push settlements while users cannot pull their escrow back. ([redacted]+.`cancelCooldown` is correctly pause-exempt — apply the same reasoning here.)
IMPACT	User funds locked in the silo for the (unbounded, admin-controlled) pause duration. No theft.
RECOMMENDATION	Allow `cancelRedeem` (and the decrease branch of `updateRedemptionRequest`) while paused — cancellation only returns the user's own escrow and cannot worsen solvency.
CONFIDENCE	High.

LOW PROTO-L-05 **burnQuarantine breaks the hub-locked == spoke-supply invariant with no on-chain reconciliation**

CONTRACT / CROSS-CHAIN

LAYER	Cross-chain
LOCATION	[redacted]OFT.sol:186–193 ; [redacted]+OFT.sol:161–168
DESCRIPTION	A compliance seizure burns quarantined spoke supply, but the matching [redacted](+) stays locked in the hub adapter with no on-chain link forcing a hub-side burn. Over-collateralized (not inflationary — no theft), but a solvency monitor comparing hub-locked to aggregate spoke supply diverges until a manual reconciliation.
IMPACT	Orphaned locked funds; monitoring drift.
RECOMMENDATION	Provide a dedicated hub-side burn tied to seizures and expose an on-chain reconciliation value; add a monitoring alert on hub-locked vs spoke-supply divergence.
CONFIDENCE	High (as documented).

LOW PROTO-L-06

Cross-chain redeem sends the outbound collateral leg with minAmountLD = 0 (no destination slippage protection)

CONTRACT / CROSS-CHAIN

LAYER	Cross-chain
LOCATION	[redacted] `Composer.sol:457–459` (redeem send), `:358–360` (deposit send)
DESCRIPTION	Both send paths overwrite `minAmountLD` to 0; `_assertSlippage` applies only to the hub-side amount. The share leg is a 1:1 lockbox (safe), but the redeem collateral leg goes over a collateral OFT (e.g. Stargate) that can apply pool fees/slippage on delivery with zero protection.
IMPACT	On cross-chain redeem, a user can receive materially less collateral than expected with no revert and no refund; bounded by the bridge's fee/slippage but unprotected.
RECOMMENDATION	Carry the user's destination minimum through the compose message and pass it as `minAmountLD` for collateral OFTs (a slippage revert then routes to refund), or restrict whitelisting to 1:1 no-fee collateral OFTs and document it.
CONFIDENCE	Plausible; severity depends on the fee model of whitelisted collateral OFTs.

LOW PROTO-L-07

redistributeLockedAmount(to = 0) desyncs the [redacted] : [redacted] 1:1 ratio used by cross-chain quotes

CONTRACT

LAYER	Contracts
LOCATION	[redacted]`.sol:918–961` (burn branch) interacting with ERC4626 preview functions consumed by the composer.
DESCRIPTION	Burning a blacklisted balance to `address(0)` burns [redacted] but not the corresponding vault-held [redacted], leaving `totalAssets() > totalSupply()`. The real mint/redeem paths use oracle/normalized math (unaffected), but the residual ERC4626 `previewDeposit/Mint` — consumed by cross-chain `quoteSend` — drift after any burn-to-zero.
IMPACT	No fund loss on core paths; cross-chain share/fee quotes drift and the orphaned [redacted] is non-distributable.
RECOMMENDATION	On the `to == 0` branch, also [redacted]`.burn(totalAmount)` to preserve the 1:1 invariant, mirroring the user `burn()` path.
CONFIDENCE	High (ratio drift); composer impact Plausible.

LOW PROTO-L-08 **rescueTokens blast radius + composer admin must be confirmed as the multisig**

CONTRACT / CONFIG

LAYER	Cross-chain / Config
LOCATION	[redacted] `Composer.sol:203–207` ; [redacted]+ `Composer.sol:148–152` (admin granted to deployer at construction).
DESCRIPTION	`rescueTokens` can sweep any ERC20 the composer holds — correctly limited to the composer's own balance (it cannot reach the adapters' locked backing), but the composer transiently holds user funds and holds them until rescue on a failed refund. The role is granted to the deploying EOA and only "should" be moved to the multisig. Handover of the composers is also the step most at risk of being silently skipped (M-03/F4).
IMPACT	Until the role is the multisig, one key can sweep stuck/in-flight collateral or [redacted] on the composer.
RECOMMENDATION	Confirm on-chain that `DEFAULT_ADMIN` on both composers is the multisig and the deployer holds no role; consider event-monitored/timelocked rescue.
CONFIDENCE	High mechanism; live admin must be verified.

LOW PROTO-L-09 **Wallet sanctions/T&C attestation is enforced client-side only with a static, replayable message** FRONTEND

LAYER	Frontend
LOCATION	`components/layout/wallet-signature-verification-modal.tsx:34–114`
DESCRIPTION	On connect, the user `personal_sign` s a fixed legal/sanctions attestation (no nonce, domain, timestamp, or expiry); verification is browser-only and cached in `sessionStorage` . It is a UX gate, not a control — trivially bypassed by skipping the signature or seeding sessionStorage.
IMPACT	The compliance gate provides no real enforcement; the static message is replayable and not domain-bound (low risk since it authorizes nothing on-chain and is human-readable — no blind-signing concern).
RECOMMENDATION	If this gate carries compliance weight, verify server-side with a server nonce + domain + expiry bound into the message; otherwise document it explicitly as a non-security acknowledgement.
CONFIDENCE	High.

LOW PROTO-L-10 **depositCollateral credits the raw amount rather than the measured balance delta (fee-on-transfer inconsistency)** CONTRACT

LAYER	Contracts
LOCATION	[redacted].sol:451–460 (vs the delta-measuring `mint` at :223–236)
DESCRIPTION	`mint` measures a `balanceOf` delta (fee-on-transfer safe) but `depositCollateral` credits the requested `amount`. A fee-on-transfer collateral would over-count `collateralBalance` vs real holdings, so a later liquidity check can pass while the transfer reverts (redeem/withdraw DoS for that asset). No unbacked [redacted] is created here.
IMPACT	Accounting can overstate holdings for a fee-on-transfer asset; onboarding policy already excludes such assets, so this is defense-in-depth.
RECOMMENDATION	Mirror `mint` — credit the measured delta.
CONFIDENCE	High.

LOW PROTO-L-11 **SPF softfail and unverified DKIM selector on the mail domain** DNS / EMAIL

LAYER	DNS / Domain
DESCRIPTION	[redacted] uses Google Workspace mail with `v=spf1 include:_spf.google.com ~all` (softfail) and a strong DMARC `p=reject`. DMARC-reject makes the SPF softfail largely moot for aligned mail, but a DKIM selector was not confirmed from DNS. Since DMARC alignment can pass on DKIM alone, DKIM must be correctly published for legitimate mail to survive the reject policy.
IMPACT	Minor: mail deliverability/spoofing-resistance hygiene. DMARC `p=reject` already provides strong protection.
RECOMMENDATION	Confirm the Google DKIM selector (`google._domainkey.`[redacted]) is published and signing; optionally tighten SPF to `-all` once all senders are enumerated.
CONFIDENCE	High on SPF/DMARC (confirmed); DKIM selector not checked from DNS.

LOW PROTO-L-12

No published vulnerability-disclosure channel (security.txt) — slower whitehat contact during an incident

FRONTEND / OPS

LAYER	Frontend / Ops
DESCRIPTION	No `security.txt` was observed at [redacted]`/.well-known/security.txt` (404). An RFC 9116 `security.txt` and a clear disclosure policy give a good-faith reporter an unambiguous, fast path to reach the team — the difference between a quiet fix and a public exploit.
IMPACT	Ad-hoc / delayed whitehat contact during a live incident; no safe-harbor signal to responders.
RECOMMENDATION	Publish `/.well-known/security.txt` (Contact, Policy, Expires) on [redacted] and [redacted]; adopt the SEAL Whitehat Safe Harbor to give good-faith responders a legal path.
CONFIDENCE	High (404 at the apex); confirm whether one exists on the app subdomain.

LOW PROTO-L-13

DNSSEC authenticated-denial (NSEC3) anomaly — NSEC3PARAM returns SERVFAIL / EDE-12 on the validating path

DNS / DNSSEC

LAYER	DNS / Domain
DESCRIPTION	The [redacted] DNSSEC chain of trust validates end-to-end (root → `io` → [redacted]; algorithm 13 ECDSA-P256-SHA256; 2 KSK + 2 ZSK; 2 SHA-256 DS records matching live DNSKEYs). However, a direct `NSEC3PARAM` query intermittently returns `SERVFAIL` with **Extended DNS Error 12 ("NSEC missing")** on the validating resolver path, indicating an inconsistency in how authenticated denial-of-existence is served across the authoritative nameservers (`ns77/`[redacted]) rather than a trust-chain break.
IMPACT	A validating resolver may fail to prove non-existence of a name — for a subset of clients this degrades to resolution failures (availability), and it weakens the assurance that a forged "subdomain does not exist" answer cannot be injected. Not a compromise of the signed positive answers.
RECOMMENDATION	Confirm with the DNS provider that NSEC3 is served completely and consistently across all authoritative nameservers with matching NSEC3PARAM salt/iterations; move to **NSEC3 with 0 extra iterations (RFC 9276)** or plain NSEC; re-validate with DNSViz / `delv` until EDE-12 clears; add authenticated-denial to DNS-change monitoring.
CONFIDENCE	Medium — reproduced via authenticated DoH; root cause (opt-out vs NS-set inconsistency) needs provider-side confirmation.

INFO PROTO-I-01 … I-12 **Informational & hardening items**

ALL LAYERS

I-01 **Dust redeem for zero collateral** — [redacted]`.redeem` lacks a zero-collateral guard symmetric to `mint`'s `ZeroMintAmount`; sub-unit redemptions burn [redacted] for 0 collateral (self-griefing; protocol-favorable). Add a `ZeroRedeemAmount` revert / non-dust `minRedeemAmount`.

I-02 **Mint fee rounding** — a fee that rounds to 0 in collateral decimals is skipped while the per-block counter is decremented by the 18-dec fee; track actual minted [redacted]`Amount`. Dust-scale only.

I-03 **removeSupportedAsset leaves a stale feed** — re-adding an asset silently re-enables the old `collateralFeed` / `feedMaxAge`; clear or re-validate on re-add.

I-04 **collateralBalance strands stray transfers** — tokens sent directly (not via mint/deposit) are unrecoverable; accept or add a surplus sweep.

I-05 **vestingPeriod == 0 re-opens reward-sandwich MEV** — the anti-MEV guarantee rests on `vestingPeriod > 0`; enforce a nonzero minimum, or never combine `vestingPeriod=0` with `cooldownDuration=0`.

I-06 **burnAssets centralization** — an unbounded (down to 1e18) discretionary NAV haircut on stakers; multisig-gated and profitless, but consider a per-epoch cap/timelock.

I-07 **Cross-chain blacklist non-atomicity** — hub and spoke restriction sets are independent; underlies M-01. Document the trust boundary; enforce at the OFT layer on all chains if compliance is hard-required.

I-08 **MINTER_ROLE singularity** — [redacted] has no independent pause; the kill-switch relies on MINTER_ROLE being held by exactly one pausable contract ([redacted]). Monitor that member count stays at 1.

I-09 **Compiler margins** — [redacted] sits ~110 bytes under the EIP-170 limit (viaIR/runs=1); any future logic re-breaks deployability. Foundry (runs=20,000, no per-file override) cannot build a deployable [redacted] and diverges from the Hardhat artifact — treat Hardhat as the sole deploy source; track the margin as an upgrade gate; pin `evmVersion` globally.

I-10 **Role concentration** — `rewardsOperator` == admin multisig (confirm intent vs a dedicated bot key); pausers are hot EOAs (consider a small pauser multisig); the multisig holds MINTER_ROLE (unbacked-mint capable, capped).

I-11 **Frontend RPC hygiene** — `NEXT_PUBLIC_*_RPC_URL` ships to the browser; use keyless/public endpoints or proxy keyed RPC server-side. Remove the `console.log('upstreamUrl', …)` on the proxy hot path. [redacted] admin `verify()` is EOA-only (no EIP-1271) — confirm single-key admin is intended.

I-12 **Deploy runbook drift** — example commands still reference an Arbitrum/Base/Optimism topology, not the live Ethereum-hub / BSC-spoke; a blind copy-paste could target the wrong chain. Update all examples to `ethereum` / `bsc`.

05 CONFIGURATION & DEPLOYMENT NOTES

Topology & roles. Hub = Ethereum (LZ EID [redacted]), spoke = BSC ([redacted]). `DEFAULT_ADMIN`, `rewardsOperator`, and treasury flow to the Gnosis Safe `0xaB05…fe6F` (3-of-4, verified deployed byte-identically on BSC per internal records); pausers are two EOAs. Upgrade authority = the multisig, **no timelock** (H-02).

What is configured correctly. Collateral and Stargate pool addresses match canonical mainnet USDC/USDT and their Stargate V2 pools (no wrong-token risk); the AccessControl handover grants admin to the multisig and revokes the deployer with `DEFAULT_ADMIN` revoked last (correct ordering, no self-lockout); the LZ delegate is migrated before ownership transfer; a `DEPLOY_ENV` guard prevents a silent testnet default; `.env.example` carries no secrets; no sequencer feed is required (neither chain is an L2).

What needs action before the next deploy. Arm oracle feeds for every funded collateral (H-03); reconcile the [redacted] config to live (M-02); verify silo `lockWiring()` and all two-step ownership acceptances on-chain (M-03); confirm LZ confirmations and enforced compose gas (M-04); adopt an upgrade timelock (H-02); arm the collateral-withdrawal allowlist + rate cap (H-01).

LIVE ON-CHAIN VERIFICATION — ETHEREUM MAINNET, 2026-08-11

Read-only `eth_call` against the live proxies. These confirm the live contracts are the [redacted]-audited baseline (the branch's oracle, unbacked-cap, [redacted]-pause, and silo-lock features are **not** live yet), and pin the current risk posture.

CHECK (LIVE)	RESULT	MEANING
[redacted].[redacted]Address()	0x0	KYC/[redacted] is **OFF** on live, as stated (M-02 risk is a config-drift-on-redeploy concern, not a live issue).
[redacted].paused()	false	Live and active.
[redacted].totalSupply()	≈ 4,457,941	~$[redacted]M [redacted] outstanding — real funds at stake.
[redacted].isOraclePriced(USDC)	revert (absent)	Live [redacted] has **no per-collateral oracle** → all collateral priced **1:1**. Confirms H-03 for live: depeg is mitigated only by pause (there is also no unbacked cap live).
[redacted].maxUnbackedMint()	revert (absent)	The unbacked-mint cap is a branch feature, **not live**.
[redacted].paused()	revert (absent)	Live [redacted] has no Pausable (matches design); kill-switch relies on [redacted].pause() + single MINTER.
RedeemSilo / PlusSilo.wiringLocked()	revert (absent)	`lockWiring` is a branch feature, not live.
RedeemSilo.owner() · PlusSilo.owner()	0xaB05…fe6F	Both silos owned by the **multisig** ✓ — live silo security rests on this (sound).

Net: the live system is the audited baseline holding ~$[redacted]M with **1:1 collateral pricing and no unbacked cap** — so on live, a stablecoin depeg is mitigated only by the pause switch. The branch's oracle + cap materially improve this; H-03/H-01 are the reasons to arm feeds and withdrawal guardrails as that branch deploys. (Definitive role-divestment and implementation-hash checks require a full node; our GET-proxy read path returned reliable results for value/revert reads but not for `hasRole` /storage-slot reads.)

06 CROSS-CHAIN (LAYERZERO V2) NOTES

Supply integrity is **sound**: we found no path to mint on a spoke without a matching hub lock, and no double-spend in the composer flows. Spoke OFTs mint only via `_credit` from the owner-configured hub peer; the quarantine override conserves supply; the `MCTOFTAdapter` is hard-locked so [redacted] can never bridge (protecting the 1:1 backing). The reentrancy model — endpoint-gated `lzCompose` with a self-called handler that is deliberately not `nonReentrant` — is safe because every downstream call is to trusted, callback-free contracts. The cross-chain issues are the compliance asymmetry (M-01), reorg/gas config (M-04), the unprotected outbound leg (L-06), and seizure reconciliation (L-05). One documentation fix: a [redacted] `Composer` docstring claims the inner deposit/redeem are `nonReentrant` — they are not; the real protection is endpoint/self gating. Correct the comment to match.

07 FRONTEND / BACKEND / DNS NOTES

Frontend/backend. No committed secrets; server secrets are correctly non-public. Hardcoded contract addresses were byte-checked against canonical mainnet values and are correct (no fund-routing/typo risk) — the highest-value frontend check for a stablecoin. No SSRF/open-proxy-to-arbitrary-host (upstreams are fixed/env-derived); no XSS sinks; transaction wrappers correctly throw on reverted receipts; anti-framing/transport headers enforced. The real items are the API proxy abuse surface (M-05), the [redacted] admin session hardening (M-06), report-only CSP (L-02), and the client-only wallet attestation (L-09). The two admin routes (`/api/[redacted]/admin/{nonce,verify}`) should be reviewed to confirm nonce single-use and httpOnly/secure/strict cookies.

DNS ([redacted], [redacted]). Strong posture overall: **DNSSEC enabled** (DS at registry, ECDSA-P256, authenticated), **DMARC** `p=reject`, SPF present, Google Workspace mail; [redacted]/www on [redacted]; registrar/DNS [redacted]. Gaps: **no CAA record** (L-01), SPF softfail + unconfirmed DKIM selector (L-11), and one authenticated-denial anomaly (below).

DNSSEC CHAIN-OF-TRUST ANALYSIS (DNSVIZ-STYLE)

We validated the full delegation chain from the root to [redacted] the way DNSViz does — every link, key, signature, and denial-of-existence proof — and report it as a standard section of every Vari review. The chain is **complete and cryptographically valid end-to-end**: the root KSK signs the `.io` DS, the `.io` zone signs the delegation to [redacted], and the registry-published DS records match the live DNSKEY set at the zone apex. There is no broken link and no insecure downgrade.

CHAIN LINK	STATUS	DETAIL
`.` (root) → `io`	SECURE	Root trust anchor (KSK-2024) signs the `io` DS set. Algorithm 8 (RSA/SHA-256) at root/TLD.
`io` → [redacted]	SECURE	**2 DS records** published at the `.io` registry, digest type 2 (SHA-256), referencing the apex KSK. Both DS digests match a live DNSKEY.
[redacted] DNSKEY	SECURE	**Algorithm 13 (ECDSA-P256-SHA256)** — modern, compact, recommended. **2 KSK + 2 ZSK** present (clean key-rollover posture; pre-published successor keys).
Zone RRSIGs (A / MX / TXT / SOA)	SECURE	Apex records are signed and validate against the ZSKs; signature validity windows current at review time.
Authenticated denial (NSEC3)	ANOMALY	An explicit `NSEC3PARAM` query returned **SERVFAIL with EDE code 12 (NSEC missing)** from the validating path, while negative answers otherwise resolve. See finding below.

> **PROTO-L-13 (Low) — NSEC3PARAM / authenticated-denial anomaly.** Direct `NSEC3PARAM` resolution intermittently returns `SERVFAIL` with **Extended DNS Error 12 ("NSEC missing")** on the validating resolver path, even though the zone is otherwise correctly signed and positive answers validate. This most often indicates an **inconsistency in how the authoritative set ([redacted]) serves authenticated denial-of-existence** — e.g. NSEC3 opt-out or parameter mismatch across the nameserver set, or a resolver-visible gap in the NSEC3 chain — rather than a trust-chain break. The practical effect is that **a validating resolver can fail to prove non-existence of a name**, which for a subset of clients degrades to resolution failures (availability), and weakens the guarantee that "this subdomain does not exist" cannot be forged.

> **Fix / improve:** (1) confirm with the DNS provider that **NSEC3 is served consistently and completely across all authoritative nameservers** (ns77/[redacted]), with matching NSEC3PARAM salt/iterations; (2) prefer **NSEC3 with 0 extra iterations** (current best practice — RFC 9276) or plain NSEC if the zone has nothing to hide, since high iteration counts add cost with little benefit; (3) re-run a DNSViz / `delv` validation after the change and confirm the EDE-12 clears; (4) add authenticated-denial to the DNS-change monitor so a future signing regression is caught.

DNSSEC — IMPROVE & HARDEN

The signed zone is already ahead of most protocols. To make it best-in-class: **(a)** resolve the NSEC3 denial anomaly above (PROTO-L-13); **(b)** pair DNSSEC with **CAA** (L-01) — a CAA record is DNSSEC-signed and therefore tamper-evident, so the two compound: signed CAA is a genuinely strong issuance control; **(c)** consider publishing **TLSA / DANE** records for the mail host now that the zone is signed, hardening SMTP against downgrade; **(d)** document a **key-rollover runbook** (the 2-KSK/2-ZSK layout suggests one is in flight — make it explicit and monitored) and set **DS/DNSKEY monitoring** so an expired signature or failed rollover pages the on-call before resolvers start returning SERVFAIL; **(e)** keep signature validity windows short enough to limit replay but long enough to survive a signing outage (typical 2–4 week RRSIG lifetime with daily re-signing).

08 PROTOCOL-SPECIFIC ECONOMIC ANALYSIS

Backing invariant. Every mint/redeem/burn path preserves `totalSupply([redacted]) == [redacted].balanceOf([redacted])`, with rounding always protocol-favorable — no path over-mints vs backing or over-pays on redeem. The only ledger break is admin-driven: `withdrawCollateral` (H-01) and the `redistributeLockedAmount(to=0)` desync (L-07).

Peg / depeg. The clamp design (mint at `min(price,$1)`, redeem at real price) is value-conservative **with feeds armed** — we verified an attacker cannot extract value across any asset pair in a [0.5, 2] price band. The exposure is entirely the no-feed 1:1 fallback (H-03): one un-fed depeggable collateral reopens the classic mint-cheap / redeem-healthy drain against the whole reserve. Mitigations present: pause, per-block rate caps, and the unbacked-mint cap.

Unbacked mint. The cap (net-of-burn counter, 2-day timelocked raises, `CAP_GUARDIAN` veto, floor-at-zero) is well-constructed with no bypass found — a genuine strength.

Staking / yield. Share-inflation/first-depositor theft is defended by MIN_SHARES (1e18) + the OZ virtual offset — donations degrade to a low-TVL DoS (L-03), never theft. Reward-sandwich MEV is prevented by the 8h linear vesting plus cooldown, except under the `vestingPeriod=0` misconfig (I-05). The `redistributeLockedAmount` vesting-burn was verified NAV-monotonic.

Cross-chain supply. Hub-locked == spoke-minted holds across the composer/OFT paths; the only invariant break is the operator-driven `burnQuarantine` (L-05), which is over-collateralizing, not inflationary.

09 SEAL FRAMEWORK ALIGNMENT

Assessed against the applicable SEAL (Security Alliance) frameworks. Status reflects observable evidence; items requiring off-chain/operational confirmation are marked accordingly.

SEAL DOMAIN	STATUS	NOTES
External Security Reviews (Smart Contracts)	GOOD	[redacted] audit + multiple internal remediation rounds — strong. Keep the same external-review discipline for future core changes, and keep the deployed commit matched to a fix-reviewed, published report.
Multisig for Protocols	GOOD	3-of-4 Safe, byte-identical on both chains, admin non-renounceable, self-revoke protection. Gap: no upgrade timelock (H-02); pausers are EOAs.
Operational Security / Key Mgmt	PARTIAL	Deployer correctly divested (verify on-chain). Powerful uncapped levers (H-01) and instant upgrade (H-02) concentrate operational risk on keys with no on-chain brake.
Monitoring	ACTION	Events exist (`CollateralWithdrawn` , `Upgraded` , quarantine). Recommend active alerting on: collateral withdrawals, upgrades/queued ops, unbacked-cap changes, hub-locked vs spoke-supply divergence, MINTER_ROLE membership, oracle staleness.
Incident Management	ACTION	Pause hatches exist. Recommend a written IR runbook and an on-call rotation, and keep a record of the exact deployed commit so responders reason about the right bytecode.
Supply Chain / DevSecOps	GOOD	Dependabot patches tracked; deploy artifacts committed. Pin the deploy toolchain and treat Hardhat as the canonical [redacted] build (I-09).
Wallet Security (users)	GOOD	Frontend avoids blind-signing pitfalls; correct addresses. Harden admin session (M-06) and CSP (L-02).
Safe Harbor	CONSIDER	Adopting the SEAL Whitehat Safe Harbor would give a legal path for good-faith rescue during an incident. Recommended for a live stablecoin.

SEAL OPERATIONAL SECURITY SCORECARD (FRONTEND / BACKEND / DNS / OPS)

A per-dimension operational checklist complementing the protocol-level alignment above. Status reflects **Vari's verified assessment**; where it differs from a self-assessment, that is noted. PASS no action · ACTION fix/harden · VERIFY confirm live.

SEAL AREA	STATUS	NOTES
Well-maintained framework	PASS	Next.js (patched) / React 19 / wagmi / viem current; no framework-level issue found.
Dependency hygiene	PASS	Lockfile pinned; Dependabot patches tracked. (No full SCA run this pass — recommend continuous dependency scanning.)
Third-party scripts	PASS	No external scripts/analytics/CDN imports; only the public WalletConnect projectId. No wallet-SDK telemetry observed.

XSS sinks	PASS	No `dangerouslySetInnerHTML`, `eval`, or `new Function`; no raw innerHTML writes in app code.
Security headers	PASS	HSTS, X-Frame-Options DENY + enforced `frame-ancestors 'none'`, nosniff, Referrer-Policy, Permissions-Policy all enforced.
Content Security Policy	ACTION	(L-02) CSP is **report-only**, not enforcing, and allows `script-src 'unsafe-inline'`. Promote to enforcing with nonce/hash.
DNS & domain security	ACTION	(L-01, L-13) DNSSEC chain-of-trust validates end-to-end (alg-13, 2 KSK + 2 ZSK, 2 SHA-256 DS) and DMARC `p=reject` confirmed — strong. Gaps: **no CAA record** (contradicts a "CAA restricts issuance" self-claim) and an **NSEC3 authenticated-denial anomaly (EDE-12)**. Publish CAA; fix NSEC3 serving. Full DNSViz chain analysis in §07.
DDoS & availability	VERIFY	(M-05) Static frontend sits behind [redacted]'s CDN, but the API proxy has **no app-level rate/complexity limiting**. Add throttling + GraphQL cost limits.
Backend access control	ACTION	(M-05, M-06) Proxy gated only by a **spoofable Referer** with no rate limit; [redacted] admin session lacks domain binding/expiry and is a client-stored bearer. Server secrets are correctly vaulted (good).
Wallet interaction safety	PASS	Simulation + receipt-revert checks sound; addresses byte-verified; RPC failover. Caveat (L-09): the sanctions/T&C attestation is a client-side UX gate, not a control.
Vulnerability disclosure	ACTION	(L-12) No `security.txt` at [redacted] (404). Publish RFC 9116 + adopt the SEAL Safe Harbor.
Monitoring	ACTION	Frontend/uptime/DNS-change monitoring per self-assessment; add **on-chain protocol monitoring**: collateral withdrawals, upgrades/queued ops, unbacked-cap changes, oracle staleness, hub-locked vs spoke-supply.
Account security	ACTION	(M-07) Enforce hardware-key 2FA + lockdown across registrar / DNS / hosting / source / infra — the frontend-hijack vector behind major 2026 losses.

10 WHAT'S GENUINELY SOLID

A fair report names strengths, and [redacted] has many: value-conservative mint/redeem math with no over-mint/over-redeem path; a well-built unbacked-mint cap (timelock + guardian); consistent `nonReentrant` discipline and state-before-external-calls ordering; MIN_SHARES + virtual-offset inflation defense; a verified-monotonic vesting-burn; the hard-locked [redacted] adapter that keeps [redacted] off-chain-bridgeable and preserves 1:1 backing; role self-revoke protection and non-renounceable admin; disciplined, append-only storage-gap management for UUPS; canonical, byte-checked collateral and frontend addresses; no committed secrets; and a domain security posture (DNSSEC + DMARC `p=reject` + enforced transport headers) stronger than most live protocols. The team's own extensive internal audit trail reflects real security maturity. The findings above are about closing the governance/configuration and provenance gaps that a contract-only audit structurally cannot reach — not about a broken core.

11 ACTION POINTS & FIX-PRIORITY CHECKLIST

Consolidated, prioritized remediation list. **P0** = do before the next deployment / urgent for the live system · **P1** = high priority · **P2** = hardening. The **Status** column is filled at re-review: ◦ OPEN · ✔ FIXED · ~ ACK (risk accepted) · ✗ NOT FIXED.

P0 — BEFORE DEPLOY / URGENT FOR LIVE

REF	ACTION	STATUS
H-01	Arm `withdrawCollateral` guardrails on [redacted] (allowlist + per-epoch rate cap); move `COLLATERAL_MANAGER` to a timelocked multisig distinct from admin.	◦ OPEN
H-02	Route UUPS upgrade authority through a `TimelockController` (24–72h) with the multisig as proposer.	◦ OPEN
H-03	Arm a validated Chainlink feed for every funded collateral (USDC, USDT); gate deploy on `isOraclePriced == true`. (Applies to the branch deploy; live-audited version has no per-collateral oracle — mitigate via pause + unbacked cap.)	◦ OPEN
M-02	Reconcile [redacted]`Address` config to the live KYC posture (live confirmed `0x0` / KYC off); gate the [redacted] write behind an explicit flag so a re-run can't silently flip KYC.	◦ OPEN
M-03	Silo ownership confirmed = multisig (live) ✓. For the branch deploy: verify `wiringLocked()==true`, all Ownable2Step ownerships accepted, deployer holds no roles (full-node check).	◦ OPEN
M-07	Enforce hardware-key 2FA + account lockdown across registrar ([redacted]), DNS, hosting ([redacted]), source, CI/infra; add DNS-change + deploy alerting.	◦ OPEN

P1 — HIGH PRIORITY

REF	ACTION	STATUS
M-01	Add the destination-beneficiary blacklist check to [redacted] `Composer` (mirror the [redacted]+ composer).	◦ OPEN
M-04	Confirm on-chain LZ `ULNConfig.confirmations` = [15,20] both directions; verify OFT peers + enforced compose gas.	◦ OPEN
M-05	Add rate + GraphQL cost/depth limiting to the API proxy routes; stop treating Referer as auth.	◦ OPEN
M-06	Harden the [redacted] admin session: bind `domain` + short `expirationTime`, single-use nonce, server-side session (not a client-stored bearer).	◦ OPEN
Mon.	Stand up on-chain monitoring/alerting: collateral withdrawals, upgrades/queued ops, unbacked-cap changes, oracle staleness, hub-locked vs spoke-supply, MINTER_ROLE membership.	◦ OPEN
L-01	Publish a CAA record restricting certificate issuance to the CA(s) in use.	◦ OPEN
L-02	Promote CSP from report-only to enforcing; remove `script-src 'unsafe-inline'` (nonce/hash).	◦ OPEN
L-12	Publish `/.well-known/security.txt`; adopt the SEAL Whitehat Safe Harbor.	◦ OPEN

L-13	Fix NSEC3 authenticated-denial serving across both nameservers (EDE-12); move to NSEC3 0-iterations (RFC 9276); re-validate with DNSViz.	◦ OPEN

P2 — HARDENING

REF	ACTION	STATUS
L-03	Enforce MIN_SHARES on `unstake` (allow only 0 or ≥ floor), or seed dead shares at genesis.	◦ OPEN
L-04	Allow `cancelRedeem` / decrease while paused (return user's own escrow).	◦ OPEN
L-05	Add a hub-side burn + on-chain reconciliation for `burnQuarantine` .	◦ OPEN
L-06	Carry a destination minimum through the compose message for the collateral leg (or restrict to 1:1 no-fee OFTs).	◦ OPEN
L-07	On `redistributeLockedAmount(to=0)` , also burn the matching [redacted] to preserve the 1:1 ratio.	◦ OPEN
L-08	Confirm composer `DEFAULT_ADMIN` is the multisig; consider event-monitored/timelocked `rescueTokens` .	◦ OPEN
L-09	Server-verify (or explicitly label as non-security) the wallet sanctions/T&C attestation.	◦ OPEN
L-10 / I-01…I-04	[redacted]: measured-delta on `depositCollateral` ; zero-collateral redeem guard; clear feed on `removeSupportedAsset` ; mint-fee rounding; stray-token sweep.	◦ OPEN
I-05 / I-06	Enforce nonzero `vestingPeriod` ; consider a cap/timelock on `burnAssets` .	◦ OPEN
I-08 / I-10	Monitor MINTER_ROLE = 1; confirm `rewardsOperator` intent; consider a pauser multisig.	◦ OPEN
I-09 / I-11 / I-12	Treat Hardhat as canonical [redacted] build + track the EIP-170 margin; keyless `NEXT_PUBLIC_` RPC + remove proxy console.log; fix deploy-runbook chain examples.	◦ OPEN

> At the fix-review pass this report becomes **v1.1**: each row's Status is updated to FIXED (with the fixing commit/tx), ACK (risk formally accepted, with rationale), or NOT FIXED, and finding blocks gain a matching status line.

12 RESIDUAL RISK & LIMITATIONS

This is a point-in-time review of the reviewed branch as staged. Several findings' live exposure depends on live on-chain state (whether oracle feeds are armed, the [redacted] value, silo wiring-lock, composer admin, and LZ confirmations); each is flagged with the exact on-chain check to run, and we recommend the client execute those checks before deploying the reviewed branch. Not covered: formal verification; exhaustive novel economic modeling beyond the surfaces analyzed in §8; the LayerZero wiring config files and the two [redacted] admin API routes; off-chain key custody and operational practices; and the base LayerZero/OpenZeppelin library code (assumed correct at pinned versions). Accepted by-design residuals remain the trusted collateral lever (H-01) and instant upgradeability (H-02) until governance bounds them.

13 DISCLAIMER

This report describes a time-boxed technical security review of the materials identified in the Scope of Review as they existed during the review window. It is a point-in-time assessment: subsequent changes to code, configuration, infrastructure, or dependencies are not covered, and the live on-chain state must be independently confirmed as described herein.

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 except where expressly stated.

[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.