Trading System Security Analysis

> **Scope & purpose**: An **end-to-end security analysis** of the Quantum Dream Fund I (QDFI) / NovaMarket trading and settlement system, covering three layers — on-chain contracts, off-chain execution/signing services, and the platform/API. Intended for security review, third-party audit preparation, and the mainnet go-live decision.
>
> **Status**: Phase 1 (Arbitrum Sepolia testnet), pending third-party audit. This is an internal security analysis, **not** an investment solicitation and **not** an audit opinion.
>
> **Related documents**: `docs/08_SECURITY.md` (security policy & checklist) · `qdf-contracts/audit-package/AUDIT.md` + `INVARIANTS.md` · `fund-platform/docs/FUND_CUSTODY_SEPARATION_AND_SECURITY.md` + `AUDIT_READINESS.md` + `SELF_AUDIT_REPORT.md`. The companion remediation tracker is `docs/SECURITY_REMEDIATION_TRACKER.md`.
>
> **Last updated**: 2026-06 (based on a code reconnaissance at that time). Code evolves — verify cited files/functions against the current repository state.

1. Executive Summary

The QDFI / NovaMarket trading system uses a **"on-chain custody + off-chain execution" separation of duties**: investor capital is locked in on-chain contracts (FundVault / CapitalPool), trades execute on centralized venues (Hyperliquid / Polymarket / Uniswap) via isolated off-chain services, and the chain only draws bounded allowances, books P&L, and writes net asset value (NAV) back through a multi-signature oracle.

Overall rating: the architecture is **above average**, with defense-in-depth in place (multisig + timelock + guardian pause + draw limits + circuit breakers + credential sealing + least privilege). The core residual risks concentrate at two points: **off-chain trust** and **key single points of failure**.

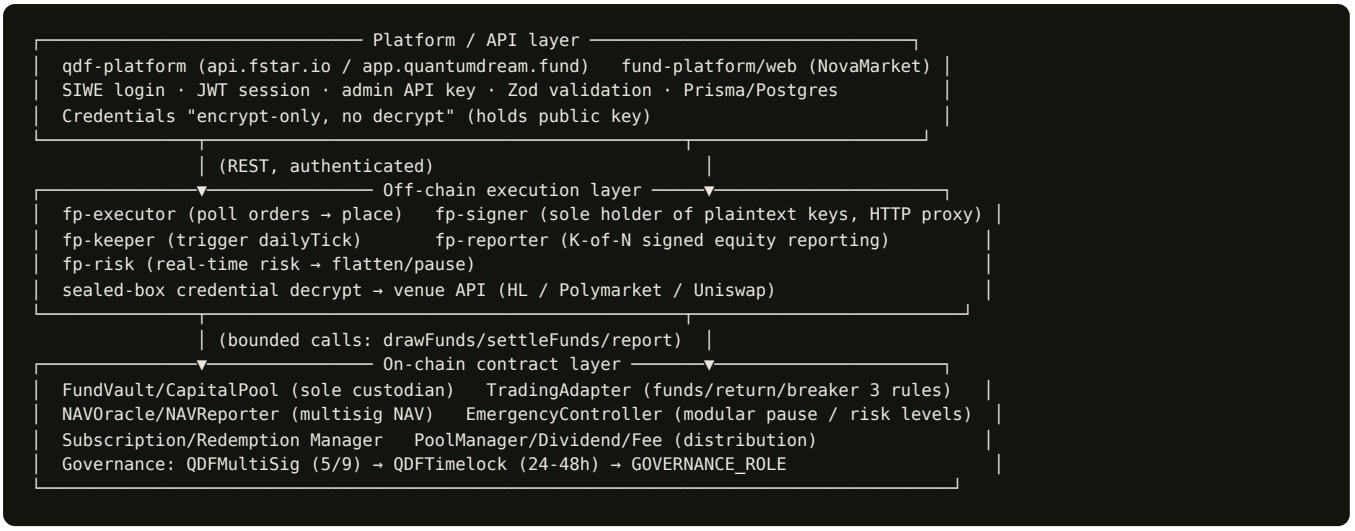
Top risks (by severity):

#	Risk	Layer	Severity	Status
R-1	NAV / equity oracle signers collude to forge net value	On-chain + off-chain	🔴 Critical	Mitigated by K-of-N multisig + deviation breaker; testnet uses Mock ZK PoR
R-2	Signer process compromise / `FP_CREDENTIAL_SECRET_KEY` leak → all venue keys lost	Off-chain	🔴 Critical	Internal-network isolation + in-memory-only; pending KMS/HSM + HA
R-3	Governance multisig key compromise (testnet uses plaintext test addresses)	On-chain	🟠 High (testnet)	Must migrate to Gnosis Safe / institutional custody before mainnet
R-4	Keeper single point of failure stalls settlement / redemption queue	Off-chain	🟠 High	Needs redundancy + monitoring/alerting
R-5	Admin auth is a single API key (no hash / 2FA / RBAC)	Platform	🟠 Med-High	Pending account-system upgrade

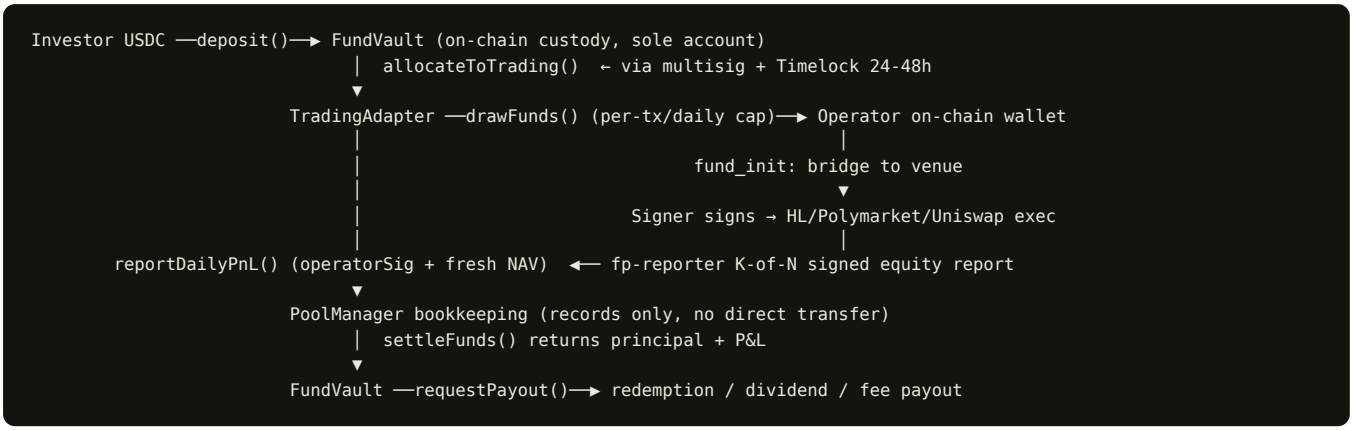
Key conclusion: the security design is reasonable for the testnet phase. **Before mainnet**, the system must complete a third-party contract audit, key-custody migration (HSM/KMS + Gnosis Safe), a real ZK Proof-of-Reserves circuit, Signer high availability, and admin-auth hardening (see § 10 checklist).

2. System Scope & Architecture Overview

2.1 Three-layer architecture



2.2 Core fund flow & trust boundary



Trust boundary highlights:
- **Custody vs execution separation**: contracts custody the funds; off-chain services execute trades; the two connect only through `TradingAdapter`'s bounded interface, so off-chain services **cannot bypass limits to touch the vault directly**.
- **Funds are off-chain after drawFunds**: this is the core custody/execution boundary. `FundVault._deployed[asset]` books "drawn but not yet returned"; real P&L is realized via the reporter multisig — **the single largest off-chain trust assumption** (see R-1).

3. Trust Model & Roles

Role	Key/credential held	Privileges	Impact if compromised	Mitigation
Investor	Own wallet key	Subscribe/redeem/claim (on-chain self-service)	Self-borne	Non-custodial; 30-day lockup + T+3 redemption
Operator	`FP_OPERATOR_PRIVATE_KEY`	`drawFunds` / `returnFunds`, issue order commands	Bounded by per-tx/daily caps + breaker	Limits, first-loss margin, KMS
Platform Web	`NM_CREDENTIAL_PUBLIC_KEY` (**public key only**), session secret	Store/show credential status, bookkeeping	**Cannot decrypt credentials**; Web breach only leaks useless ciphertext	Public/private key split (key design)
Executor	`FP_CREDENTIAL_SECRET_KEY` (for decryption)	Decrypt credentials, call Signer, read equity	Can decrypt all venue credentials	Internal-net isolation, KMS, in-memory only

Role	Key/credential held	Privileges	Impact if compromised	Mitigation
Signer	All venue private keys + decryption key (in memory)	Sign orders/transfers/init	**Single-point compromise → total loss of funds**	localhost bind, KMS/HSM, HA (pending)
Reporter	`FP_REPORTER_PRIVATE_KEYS` (K-of-N)	Multisig equity/NAV reporting	One key leak insufficient to forge	K-of-N (currently 3-of-N), deviation cap
Keeper	`FP_KEEPER_PRIVATE_KEY`	Trigger `dailyTick` / `processQueue` (gas only)	Least privilege, moves no funds	On-chain gating; permissionless design
Governance multisig	QDFMultiSig 5/9 owners	Change params/roles via Timelock	Protected by multisig threshold	Migrate to Gnosis Safe on mainnet
Guardian	GUARDIAN_ROLE (5/9 multisig)	**Real-time** pause, freeze redemptions	Can maliciously pause (DoS), but cannot steal funds	"Easy to pause, hard to resume" (resume via Timelock)
KYC Admin	KYC_ADMIN_ROLE multisig	Whitelist / jurisdiction blocking	Can wrongly allow/block	No PII on-chain; multisig

Trust-minimization design: Web "encrypt-only", Reporter K-of-N, Keeper permissionless+limits, Guardian pause-but-not-transfer — these four shrink single-point power. **Remaining concentration points**: the Signer (plaintext keys in a single process) and the NAV oracle (off-chain truthfulness).

4. On-chain Contract Layer Security

> Repo: `qdf-contracts/` (12 core + governance, ~1,879 lines of Solidity); 97 tests passing, 90.1% line / 61.9% branch coverage. NovaMarket-side contracts are in `fund-platform` (CapitalPool / PoolSettlement / NAVReporter, with Foundry invariant tests).

4.1 Fund custody — FundVault (`contracts/core/FundVault.sol`)

- **Sole USDC/USDT custody account**; all subsystems pay out via `requestPayout()` (`onlyAuthorized`).
- Allocate to trading: `allocateToTrading()` (GOVERNANCE_ROLE, via Timelock) increments `_deployed[asset]`; `returnFromTrading()` (TRADING_ADAPTER_ROLE only) decrements.
- **Controls**: ReentrancyGuard (3 functions), SafeERC20 throughout, checks-effects-interactions (CEI).
- **Gap**: **no first-loss margin layer** (account-level isolation; trading loss directly erodes principal); GOVERNANCE_ROLE is a single point of payout authority — constrained by the 24-48h Timelock.
- Note: the NovaMarket side has a `FirstLossPolicy` first-loss waterfall (operator margin absorbs losses first), a different model from the QDFI FundVault.

4.2 Trading boundary — TradingAdapter (`contracts/core/TradingAdapter.sol`), the "three iron rules"

1. **Fund right**: `drawFunds()` (onlyOperator) bounded by `maxDrawPerTx` (Sepolia 1M USDC) + `dailyDrawLimit` (3M), with daily counter reset by `block.timestamp/1 days`.
2. **Return right**: `reportDailyPnL()` requires the operator's ECDSA signature over `(chainid, this, epoch, pnl)`, with NAV being current-epoch and fresh (within 26h); **records only, no transfer** (calls `poolManager.settleDailyPnL()`).
3. **Cross-check / breaker**: `|pnl| > AUM * maxDailyPnLBps` (Sepolia 20%) → `emergency.pause(ALL)`.
   - **Controls**: digest includes chainId (cross-chain replay protection); ReentrancyGuard.
   - **Risk**: no in-chain nonce — the same epoch's same PnL could in theory be re-reported (relies on Keeper idempotency); `tradingOperator` is a single address whose key leak allows draw/return within limits.

4.3 NAV oracle — NAVOracle (`contracts/core/NAVOracle.sol`) / NAVReporter (NovaMarket)

- **3-of-3 signatures** (custodian + auditor + Chainlink, must be distinct) over `(chainid, this, epoch, navPerShare, totalAUM, porProofHash)`; **ZK Proof-of-Reserves (PoR)**; adjacent-epoch deviation > 2% → breaker (requires GOVERNANCE `resumeAfterDeviation`).

- NovaMarket NAVReporter: **K-of-N multisig** (sorted-signature dedup) + strictly increasing epoch (`NoFreshReport`) + `maxDeviationBps` deviation cap + chainId replay protection.
- ⚠️ **Testnet uses** `MockZKVerifier` ; production requires a real ZK circuit and an audit.
- **Risk (R-1, critical)**: colluding signers can forge arbitrary NAV → over-priced subscription / under-priced redemption; NAV stale > 26h halts subscription/redemption (availability).

4.4 Emergency & risk — EmergencyController (`contracts/core/EmergencyController.sol`)

- Modular pause: SUBSCRIPTION / REDEMPTION / DIVIDEND / TRADING / ALL.
- **Dual power**: `pause()` triggered **in real time** by the Guardian (5/9); `unpause()` requires GOVERNANCE via Timelock (24h) — "easy to pause, hard to resume".
- Risk levels: NORMAL → HALT_NEW_TRADE (daily -3%) → FULL_LIQUIDATION (daily -5%) → DAO_VOTE (monthly -10%). Guardian can freeze redemptions ⩽ 7 days.

4.5 Subscription / Redemption

- **Subscription** (`SubscriptionManager.sol`): KYC + stablecoin whitelist + min 10,000 USDC + fresh NAV; mints shares at the day's NAV and locks them for 30 days.
- **Redemption** (`RedemptionManager.sol`): two phases (enqueue → Keeper `processQueue`); **T+3 settlement delay**, **daily ⩽ 10% of AUM** (first item always allowed to avoid deadlock), **tiered redemption fee** (<90d 2% / 90-180d 1% / ⩾180d 0%, fee routed to the Dream Reserve, Pool C).
- **Design intent**: 30-day lockup + T+3 + daily cap = leaves a window to detect NAV manipulation and prevents bank runs. **Risk**: under extreme conditions the redemption queue may congest (some users delayed > a week).

4.6 Share token — QDFToken (`contracts/core/QDFToken.sol`)

- ERC20 (6-decimal) + **FIFO Lot mechanism** (each subscription forms a Lot with a lockup); transfers require recipient KYC-approved + sufficient unlocked shares.
- **Risk (K-2)**: Lot queue O(n) iteration — thousands of subscriptions from a single address could OOM gas (AUDIT.md remediation pending: Lot cap / merge).

4.7 Governance — QDFMultiSig (5/9) → QDFTimelock (24-48h)

- All GOVERNANCE param/role changes go through 5/9 multisig proposal → Timelock delay → execution; emergency `pause` does **not** go through Timelock.
- **Risk (R-3)**: on testnet the 9 signer keys are in `keys/multisig-signers.json` (plaintext test addresses) — **must migrate to Gnosis Safe / institutional custody before mainnet** (AUDIT.md K-4).

4.8 Reentrancy & safety-pattern matrix

Contract	ReentrancyGuard	SafeERC20	CEI
FundVault / TradingAdapter / Subscription / Redemption / Dividend	✓	✓	✓
NAVOracle / PoolManager / EmergencyController	(state-write only, no external transfer)	N/A	✓
QDFToken	(ERC20, no reentrancy point)	✓	✓
FeeManager	relies on FundVault's guard	⚠️ external call to FundVault	⚠️

On-chain invariants (Foundry / INVARIANTS.md): fund conservation `cashInvestor + marginStaked + marginForfeited + surplus == poolBalance` , share-NAV consistency, NAV monotonicity, governance constraints, single-claim dividends, pause reverts.

5. Off-chain Execution & Key Security

> Services: `fund-platform/services/` (Python): executor / signer / keeper / reporter / risk_engine / connectors / fund_init.

5.1 Execution flow & atomicity

- **fp-executor** (`services/executor/server.py`) polls each pool's PENDING orders ~every 30s, atomically claims them with `UPDATE ... FOR UPDATE SKIP LOCKED` (prevents double processing), executes each (place/flatten/cancel) → `mark_result` writes back venueOrderId/raw.
- **fp-keeper** scans settlement conditions ~every 600s and triggers on-chain `dailyTick` ; **fp-reporter** reads venue equity and reports it with K-of-N signatures; **fp-risk** monitors thresholds in real time → `flatten` + `pause` .

5.2 Venue integration (Hyperliquid / Polymarket / Uniswap)

- **Isolated signing mode** (production): the executor holds **no private key** and calls the Signer over HTTP (`/hl-order` , `/sign-and-post` , etc.); keys never leave the Signer process. Local signing is a dev-only fallback.
- **Hyperliquid**: main wallet (holds funds / can withdraw) + agent wallet (`approve_agent` , trade rights, **no withdrawal rights**).
- **Polymarket**: CCTP cross-chain (Arbitrum→Polygon native USDC) → funder wallet; EIP-712 order signing.
- **Bridge limit guards**: `FP_BRIDGE_MAX_USDC6_PER_TX/PER_DAY` , `FP_VENUE_MAINNET_ENABLED=false` , `FP_POLYMARKET_MAINNET_ENABLED=false` (real-fund switches default off).

5.3 Credential sealing (sealed-box) — key design

- **X25519 sealed-box** (Web `tweetnacl-sealedbox-js` ↔ Python `PyNaCl SealedBox` , standard libsodium X25519 + XSalsa20-Poly1305).
- Flow: operator enters venue API key/private key in Web → `sealSecret()` seals with the **public key** → stored in `VenueCredential.sealed` (Web never SELECTs the plaintext back) → Executor/Signer `unseal()` with the **private key**, held **in memory only**, discarded at the end of the tick.
- **Trust model**: Web holds the public key and **cannot decrypt**; a Web vulnerability only leaks useless ciphertext.
- **Known pitfall**: `libsodium-wrappers` breaks under Turbopack bundling → must use the pure-JS `tweetnacl-sealedbox-js` (already mitigated, recorded in `08_SECURITY.md §3.2`).

5.4 Key sources & storage

- `resolve_secret()` (`services/common/config.py`) supports `plain` (dev only) / `env:` / `file:` (chmod 600) / `awskms:` / `vault:` .
- Signer (`signer/store.py`): on first `/keygen` , generates an EOA → immediately seals and stores it → caches in memory only; the sole process holding plaintext private keys.
- **Mainnet requirement** (`08_SECURITY.md §2.1`): all `*_PRIVATE_KEY` must not be plaintext in .env — use KMS/Vault/HSM; K-of-N reporter keys split across different hosts; NAV verifiers → HSM/cold wallet.

5.5 Network exposure

Service	Listen	Exposure	Risk	Mitigation
Signer	127.0.0.1:8088	Internal only	HTTP, no auth	localhost bind + private net/VPN/mTLS (pending)
Executor/Keeper/Reporter/Risk	No inbound	Outbound DB+RPC+venue	—	Firewall + VPC isolation + RPC TLS

5.6 Injection & replay protection

- **SQL**: asyncpg/Prisma fully parameterized — **no injection surface**.
- **Command injection**: no shell/subprocess concatenation.
- **Order replay**: `Order.clientOrderId @unique` (Web idempotency, 409 on duplicate); venue SDKs have built-in nonce/timestamp.
- **Equity-report replay**: digest includes chainId (cross-chain) + strictly increasing epoch + deviation cap.

5.7 Off-chain risks

Risk	Impact	Severity	Mitigation
`FP_CREDENTIAL_SECRET_KEY` leak	Decrypts all venue credentials	🔴 Critical	KMS/HSM, access control, audit logs
Signer compromise (R-2)	Total loss of funds	🔴 Critical	Internal-net isolation, HSM, HA, audit logs
`sealed` column leak (DB)	Trading/withdrawal on that venue	🔴 Critical	PG TLS+password+IP allowlist, encrypted backups
Reporter single-key leak	Insufficient to forge (K-of-N)	🟠 High	K-of-N, HSM, anomaly detection

Risk	Impact	Severity	Mitigation
Operator key leak	Draw within limits	🟠 High	On-chain limits + breaker, KMS, rotation
`.env` accidentally in git	All keys exposed	🔴 Critical	.gitignore, secret scanner, CI gate

6. Platform & API Layer Security

> `qdf-platform` (api.fstar.io / app.quantumdream.fund) and `fund-platform/web` (NovaMarket).

6.1 Authentication

- **SIWE (wallet login)**: nonce (qdf 10-min TTL; NovaMarket persisted in DB `authNonce`, single-use consumption with `updateMany` race protection) → `SiweMessage.verify()` → HS256 JWT (qdf 24h, cookie `qdf_session`; NovaMarket 7d, cookie `nm_session` + Bearer for mobile). Cookies are httpOnly + secure + SameSite=lax.
- **Admin (qdf)**: **single `ADMIN_API_KEY` plaintext string comparison** → HS256 JWT 8h (cookie `qdf_admin`). **Risk (R-5)**: no hash/salt, no 2FA, no RBAC (all admins equal).

6.2 Authorization — on-chain is the source of truth

- NovaMarket `requireOperator()` (`web/src/lib/auth/requireOperator.ts`): resolve caller (Bearer/JWT/cookie) → API-key pool-scope check → **read on-chain `CapitalPool.operator()`** and compare → 403 if mismatch. Credential/account/strategy writes are **session-only (no API key)**.
- **Weak points**: qdf admin has no RBAC; NovaMarket API-key `scopes.pools` is optional, empty scope defaults to all pools.

6.3 Custody model

- **QDFI (qdf-platform)**: **non-custodial** — funds in the on-chain FundVault / ERC-4626 vault; the platform only records on-chain events + KYC; no user private keys.
- **NovaMarket**: investor funds locked in CapitalPool (contract-controlled); trading credentials sealed-box (Web has no decryption rights) → **non-custodial credentials + contract-custodied funds**; operators cannot directly touch investor funds.

6.4 Input validation / rate limiting / CORS / errors

- **Zod everywhere** (body/query/param), wallet `/^0x[0-9a-fA-F]{40}$/`, BigInt `^\d+$` overflow protection; **Prisma ORM, no raw SQL**.
- **Rate limiting incomplete**: only `/api/reservations` is rate-limited by email (5/hr); **login/order/redemption have none** (to be added, R-10).
- **CORS**: explicit allowlists (fstar.io / app.quantumdream.fund; novamarket.io), but with a `*` fallback — NovaMarket should remove the fallback for POST/DELETE (R-11).
- **CSRF**: relies on SameSite=lax + Bearer (no automatic cookie send); trading endpoints have no explicit CSRF token.
- **Errors**: mostly generic; some endpoints return `zod.issues` (leaks field names, no sensitive data).

6.5 Key trading endpoints (excerpt)

Platform	Endpoint	Method	Auth	Risk
qdf	`/api/admin/nav/publish`	POST	admin session	Med (critical finance)
qdf	`/api/admin/login`	POST	`ADMIN_API_KEY`	Med-High (no hash/2FA)
qdf	`/api/me/redemptions/quote`	GET	session	Low
NM	`/api/pools/[addr]/credentials`	POST	operator + **session only**	Med-High (sensitive credential)
NM	`/api/pools/[addr]/orders`	POST/DELETE	operator	Med (trade execution)
NM	`/api/pools/[addr]/accounts/provision`	POST	operator + session only	High (key generation)

6.6 Deployment / operations

- pm2 (`manzi-web` / `novamarket-web` 127.0.0.1:3105, qdf-platform `/opt/apps/qdf-platform`) + nginx SSL reverse proxy; Docker `node:22-alpine` non-root (`adduser -S nextjs`) multi-stage build.
- Postgres 16 local 127.0.0.1:5432; `NEXT_PUBLIC_*` inlined at build time (do not use that prefix for secrets).
- **Recommendations**: move env to a secrets manager; enable DB SSL + IP allowlist; SSH key-only.

7. Threat Model & Attack Surface

Threat scenario	Attack path	Existing controls	Residual risk
Oracle manipulation	Collude to forge NAV/equity → arbitrage subs/redemptions	3-of-3 / K-of-N, deviation breaker, PoR, T+3	🔴 Collusion still possible (trust assumption); testnet Mock PoR
Key theft	Steal Signer/decryption key → sign malicious orders/withdrawals	Internal isolation, in-memory-only, limits, first-loss	🔴 Signer single point; pending HSM + HA
Governance takeover	Control ⩾5/9 multisig → change roles / drain	Multisig + Timelock 24-48h (reaction window)	🟠 testnet plaintext keys (migrate to Safe on mainnet)
Availability / DoS	Keeper down, NAV stale, malicious Guardian pause	permissionless dailyTick, monitoring	🟠 Keeper single point, pause DoS
Bank run	Mass redemptions drain liquidity	30d lockup, T+3, 10% daily cap	🟠 queue congestion → delay
Cross-chain bridge	CCTP burn/mint tamper/stall	Circle standard, bridge limit guards, mainnet switch	🟠 depends on Circle Iris availability
Replay	Replay orders/reports	unique clientOrderId, chainId+epoch, SDK nonce	🟢 Low
Web intrusion	Compromise the Web layer	public/private key split (Web cannot decrypt), on-chain auth	🟢 only ciphertext/bookkeeping leak, no fund theft
Application layer	Injection/privilege-escalation/weak admin auth	Zod, Prisma, on-chain authority	🟠 admin no 2FA/RBAC, incomplete rate limiting

8. Risk Register

ID	Risk	Layer	Severity	Likelihood	Mitigation / status
R-1	Oracle signers collude to forge net value	On-chain + off-chain	🔴 Critical	Low	K-of-N + deviation breaker + PoR; mainnet needs real ZK + independent audited signers
R-2	Signer compromise / decryption key leak → total loss	Off-chain	🔴 Critical	Low-Med	Internal net + in-memory-only; **pending HSM/KMS + HA + audit logs**
R-3	Governance multisig keys (plaintext on testnet)	On-chain	🟠 High (testnet)	Med	**Migrate to Gnosis Safe / institutional custody before mainnet**
R-4	Keeper single point → settlement/redemption stall	Off-chain	🟠 High	Med	**Redundant Keeper + monitoring/alerting**
R-5	Admin single API key (no hash/2FA/RBAC)	Platform	🟠 Med-High	Med	**Upgrade to account + 2FA + RBAC**
R-6	No first-loss buffer (QDFI FundVault)	On-chain	🟠 Med	Low	NovaMarket has first-loss waterfall; QDFI relies on limits + breaker
R-7	QDFToken Lot O(n) gas OOM	On-chain	🟠 Med	Low	AUDIT.md K-2: Lot cap / merge
R-8	NAV stale > 26h → subscription/redemption halt (availability)	On-chain	🟠 Med	Med	Keeper reliability + alerting

ID	Risk	Layer	Severity	Likelihood	Mitigation / status
R-9	Redemption queue congestion in extreme conditions	On-chain	🟡 Med-Low	Med	10% daily cap + T+3 (design trade-off)
R-10	No rate limiting on login/order/redemption endpoints	Platform	🟡 Med-Low	Med	**Add rate limiting**
R-11	CORS `*` fallback (POST/DELETE)	Platform	🟡 Low	Low	Remove fallback, strict allowlist
R-12	Bridge depends on Circle CCTP availability	Off-chain	🟡 Low	Low	Limit guards + mainnet switch

9. Implemented Defensive Controls

On-chain: 5/9 multisig + Timelock 24-48h, Guardian real-time modular pause, TradingAdapter per-tx/daily draw limits, PnL anomaly breaker, NAV deviation breaker, K-of-N / 3-of-3 signed oracle, ReentrancyGuard + SafeERC20 + CEI, fund-conservation and other invariants (Foundry / Slither 0 critical), first-loss waterfall (NovaMarket), permissionless dailyTick.

Off-chain: sealed-box credentials (Web cannot decrypt), Signer isolation on localhost, private keys in memory only, `resolve_secret` supports KMS/Vault, atomic order claim via `FOR UPDATE SKIP LOCKED`, bridge limit guards + mainnet switch, least privilege (executor has no signing rights, keeper only triggers).

Platform: SIWE + single-use nonce, JWT httpOnly/secure/SameSite, on-chain `operator()` authorization, Zod validation throughout, Prisma parameterization, CORS allowlist, public/private key split custody model.

10. Pre-Mainnet Security Checklist

> Consolidated from `docs/08_SECURITY.md`, `qdf-contracts/audit-package/AUDIT.md`, `fund-platform/docs/AUDIT_READINESS.md`.
> Tracked in `docs/SECURITY_REMEDIATION_TRACKER.md`.

Audit & contracts
- [ ] Third-party contract audit passed (Trail of Bits / OpenZeppelin / equivalent) + remediation closed
- [ ] Real ZK PoR circuit replaces `MockZKVerifier` and is audited
- [ ] QDFToken Lot cap / forced merge (K-2)
- [ ] Bug bounty program live

Keys & governance
- [ ] After deploy, deployer `renounce DEFAULT_ADMIN_ROLE`, roles migrated to Timelock
- [ ] 9 governance multisig owners → Gnosis Safe / institutional custody (drop test addresses, K-4)
- [ ] 3 NAV verifiers → HSM/cold wallet, signers independent & auditable
- [ ] All `*_PRIVATE_KEY` → `awskms:` / `vault:` / `file:` (chmod600), no plaintext .env
- [ ] K-of-N reporter keys split across different hosts/HSM
- [ ] Key-loss drill (4-of-9 unavailable still governable) + rotation procedure

Off-chain services
- [ ] Signer on an isolated physical host, bound to 127.0.0.1 + VPN/mTLS, **HA primary/standby**
- [ ] Signer audit logs persisted (who/when/what was signed)
- [ ] Redundant Keeper + settlement/NAV-stall alerting (R-4/R-8)
- [ ] All outbound RPC TLS + cert verification; `FP_*_MAINNET_ENABLED` enabled with care

Platform & ops
- [ ] Admin auth upgrade: account + password (hashed) + 2FA + RBAC (R-5)
- [ ] Rate limiting on login/order/redemption/credential endpoints (R-10)
- [ ] Remove CORS `*` fallback (R-11)
- [ ] Postgres SSL + password + IP allowlist + encrypted backups; SSH key-only
- [ ] Sentry + Slack/Telegram alerting live; standby restore drill (RPO ⩽ 24h)

11. Recommendations (by priority)

P0 (mainnet blockers): ① third-party contract audit + real ZK PoR; ② key-custody migration (Signer→HSM/KMS, multisig→Gnosis Safe, verifiers→cold wallet); ③ Signer HA + audit logs; ④ admin-auth hardening (2FA/RBAC).

P1 (strongly recommended): ⑤ redundant Keeper + full-stack monitoring/alerting (NAV/settlement/redemption/equity deviation); ⑥ rate limiting on key endpoints + remove CORS fallback; ⑦ evaluate adding a first-loss buffer for QDFI or explicitly disclose its absence.

P2 (enhancements): ⑧ QDFToken Lot cap; ⑨ add a nonce to TradingAdapter reporting to prevent same-epoch repeats; ⑩ evaluate DB row-level security (RLS); ⑪ user-facing visibility and prioritization for redemption-queue congestion.

12. Appendix: Key File Quick Reference

Domain	Path
On-chain core contracts	`qdf-contracts/contracts/core/*.sol` (FundVault/TradingAdapter/NAVOracle/EmergencyController/Subscription/Redemption/PoolManager/QDFToken/Fee/KYC)
On-chain governance	`qdf-contracts/contracts/governance/QDFMultiSig.sol`, `core/QDFTimelock.sol`
On-chain audit package	`qdf-contracts/audit-package/AUDIT.md` · `INVARIANTS.md` · `DEPLOYMENT.md`
NovaMarket contracts/invariants	`fund-platform/` (CapitalPool/PoolSettlement/NAVReporter/FirstLossPolicy), `fund-platform/docs/AUDIT_READINESS.md` · `SELF_AUDIT_REPORT.md` · `FUND_CUSTODY_SEPARATION_AND_SECURITY.md`
Off-chain execution	`fund-platform/services/{executor,signer,keeper,reporter,risk_engine,connectors,fund_init,common}`
Credential sealing	web `src/lib/{credentials,crypto}.ts` · services `common/crypto.py` · `signer/store.py`
Platform auth	qdf `src/lib/{siwe,session,admin-auth}.ts` · NovaMarket `src/lib/auth/*`
Security policy	`docs/08_SECURITY.md` (this document is its trading-system-specific supplement)

This document is based on a code reconnaissance and is intended to support security review and audit preparation; it does not constitute an audit opinion or a security guarantee. All amounts/parameters reflect the Arbitrum Sepolia testnet deployment; mainnet parameters are subject to final governance.