



Security Review For ODEI



Collaborative Audit Prepared For: **ODEI**
Lead Security Expert(s): **0x73696d616f**
Date Audited: **June 5 - June 8, 2026**

Introduction

\$ODAI aims to become the #1 verifiable AI agent-created asset on-chain.

It is built around a simple standard: trust should come from live on-chain evidence, not from promises.

This Sherlock security review gives holders an independent view of the actual contracts, permissions, control surfaces, and trust assumptions behind the ODEI on-chain system.

The review focuses on mint and burn behavior, transfer mechanics, treasury and Flaunch action permissions, RevenueManager relevance, Progressive Bid Wall assumptions, and the public claims that can be supported by current on-chain state.

For ODEI, transparency is not a marketing layer. It is part of the product.

ODEI cares about its holders, and this report is part of ODEI's commitment to clear and verifiable security communication.

Scope

Repository: `odei-ai/odei-sherlock-scope`

Audited Commit: `a682d142511167957516979a1e19b4f3da6bda69`

Files:

- `agent-registry-implementation-0x7274e874ca62410a93bd8bf61c69d8045e399c02`
 - `/project/contracts/IdentityRegistryUpgradeable.sol`
- `agent-registry-proxy-0x8004A169FB4a3325136EB29fA0ceB6D2e539a432`
 - `/project/contracts/ERC1967Proxy.sol`
- `bidwall-0x66681f10BA90496241A25e33380004f30Dfd8aa8`
 - `/src/contracts/bidwall/BidWall.sol`
 - `/src/contracts/hooks/FairLaunch.sol`
 - `/src/contracts/hooks/FeeDistributor.sol`
 - `/src/contracts/hooks/FeeExemptions.sol`
 - `/src/contracts/hooks/InternalSwapPool.sol`
 - `/src/contracts/hooks/Notifier.sol`
 - `/src/contracts/libraries/CurrencySettler.sol`
 - `/src/contracts/PositionManager.sol`
 - `/src/contracts/referrals/ReferralEscrow.sol`
 - `/src/contracts/treasury/ActionManager.sol`

- /src/contracts/treasury/MemecoinTreasury.sol
- /src/contracts/types/MemecoinFinder.sol
- /src/contracts/types/StoreKeys.sol
- /src/contracts/types/TickFinder.sol
- /src/contracts/zaps/PoolSwap.sol
- /src/contracts/bidwall/BidWall.sol
- /src/contracts/escrows/FeeEscrow.sol
- /src/contracts/escrows/ReferralEscrow.sol
- /src/contracts/Flaunch.sol
- /src/contracts/hooks/FairLaunch.sol
- fee-escrow-0x72e6f7948b1B1A343B477F39aAbd2E35E6D27dde
 - /src/contracts/hooks/FeeDistributor.sol
 - /src/contracts/hooks/FeeExemptions.sol
 - /src/contracts/hooks/InternalSwapPool.sol
 - /src/contracts/hooks/Notifier.sol
 - /src/contracts/libraries/CurrencySettler.sol
 - /src/contracts/libraries/ProtocolRoles.sol
 - /src/contracts/libraries/TokenSupply.sol
 - /src/contracts/libraries/UniswapHookEvents.sol
 - /src/contracts/PositionManager.sol
 - /src/contracts/price/FlaunchFeeExemption.sol
 - /src/contracts/subscribers/Indexer.sol
 - /src/contracts/treasury/ActionManager.sol
 - /src/contracts/treasury/MemecoinTreasury.sol
 - /src/contracts/types/MemecoinFinder.sol
 - /src/contracts/types/StoreKeys.sol
 - /src/contracts/types/TickFinder.sol
 - /src/contracts/zaps/PoolSwap.sol
- flaunch-revenue-streams-0x516af52D0c629B5E378DA4DC64Ecb0744cE10109
 - /src/contracts/bidwall/BidWall.sol
 - /src/contracts/escrows/FeeEscrow.sol

- /src/contracts/escrows/ReferralEscrow.sol
- /src/contracts/Flaunch.sol
- /src/contracts/hooks/FairLaunch.sol
- /src/contracts/hooks/FeeDistributor.sol
- /src/contracts/hooks/FeeExemptions.sol
- /src/contracts/hooks/InternalSwapPool.sol
- /src/contracts/hooks/Notifier.sol
- /src/contracts/libraries/CurrencySettler.sol
- /src/contracts/libraries/ProtocolRoles.sol
- /src/contracts/libraries/TokenSupply.sol
- /src/contracts/libraries/UniswapHookEvents.sol
- /src/contracts/PositionManager.sol
- /src/contracts/price/FlaunchFeeExemption.sol
- /src/contracts/subscribers/Indexer.sol
- /src/contracts/treasury/ActionManager.sol
- /src/contracts/treasury/MemecoinTreasury.sol
- /src/contracts/types/MemecoinFinder.sol
- /src/contracts/types/StoreKeys.sol
- /src/contracts/types/TickFinder.sol
- /src/contracts/zaps/PoolSwap.sol
- memecoin-implementation-0xF1EEeeeeECd95E9Eb2df58484ceed175AcBD945C
 - /src/contracts/bidwall/BidWall.sol
 - /src/contracts/Flaunch.sol
 - /src/contracts/hooks/FairLaunch.sol
 - /src/contracts/hooks/FeeDistributor.sol
 - /src/contracts/hooks/FeeExemptions.sol
 - /src/contracts/hooks/InternalSwapPool.sol
 - /src/contracts/hooks/Notifier.sol
 - /src/contracts/libraries/CurrencySettler.sol
 - /src/contracts/libraries/TokenSupply.sol
 - /src/contracts/Memecoin.sol

- /src/contracts/PositionManager.sol
- /src/contracts/referrals/ReferralEscrow.sol
- /src/contracts/treasury/ActionManager.sol
- /src/contracts/treasury/MemecoinTreasury.sol
- /src/contracts/types/MemecoinFinder.sol
- /src/contracts/types/StoreKeys.sol
- /src/contracts/types/TickFinder.sol
- /src/contracts/zaps/PoolSwap.sol
- memecoin-treasury-implementation-0x7397390360Bd9D559D9277E60d47b99933791232
 - /src/contracts/bidwall/BidWall.sol
 - /src/contracts/escrows/FeeEscrow.sol
 - /src/contracts/escrows/ReferralEscrow.sol
 - /src/contracts/Flaunch.sol
 - /src/contracts/hooks/FairLaunch.sol
 - /src/contracts/hooks/FeeDistributor.sol
 - /src/contracts/hooks/FeeExemptions.sol
 - /src/contracts/hooks/InternalSwapPool.sol
 - /src/contracts/hooks/Notifier.sol
 - /src/contracts/libraries/CurrencySettler.sol
 - /src/contracts/libraries/ProtocolRoles.sol
 - /src/contracts/libraries/TokenSupply.sol
 - /src/contracts/libraries/UniswapHookEvents.sol
 - /src/contracts/PositionManager.sol
 - /src/contracts/price/FlaunchFeeExemption.sol
 - /src/contracts/subscribers/Indexer.sol
 - /src/contracts/treasury/ActionManager.sol
 - /src/contracts/treasury/MemecoinTreasury.sol
 - /src/contracts/types/MemecoinFinder.sol
 - /src/contracts/types/StoreKeys.sol
 - /src/contracts/types/TickFinder.sol
 - /src/contracts/zaps/PoolSwap.sol

- odai-token-0x0086cff0c1e5d17b19f5bcd4c8840a5b4251d959
 - /src/contracts/bidwall/BidWall.sol
 - /src/contracts/Flaunch.sol
 - /src/contracts/hooks/FairLaunch.sol
 - /src/contracts/hooks/FeeDistributor.sol
 - /src/contracts/hooks/FeeExemptions.sol
 - /src/contracts/hooks/InternalSwapPool.sol
 - /src/contracts/hooks/Notifier.sol
 - /src/contracts/libraries/CurrencySettler.sol
 - /src/contracts/libraries/TokenSupply.sol
 - /src/contracts/Memecoin.sol
 - /src/contracts/PositionManager.sol
 - /src/contracts/referrals/ReferralEscrow.sol
 - /src/contracts/treasury/ActionManager.sol
 - /src/contracts/treasury/MemecoinTreasury.sol
 - /src/contracts/types/MemecoinFinder.sol
 - /src/contracts/types/StoreKeys.sol
 - /src/contracts/types/TickFinder.sol
 - /src/contracts/zaps/PoolSwap.sol
- odai-treasury-0x14926224168492CdC06461713b3011cd25d8C901
 - /src/contracts/bidwall/BidWall.sol
 - /src/contracts/escrows/FeeEscrow.sol
 - /src/contracts/escrows/ReferralEscrow.sol
 - /src/contracts/Flaunch.sol
 - /src/contracts/hooks/FairLaunch.sol
 - /src/contracts/hooks/FeeDistributor.sol
 - /src/contracts/hooks/FeeExemptions.sol
 - /src/contracts/hooks/InternalSwapPool.sol
 - /src/contracts/hooks/Notifier.sol
 - /src/contracts/libraries/CurrencySettler.sol
 - /src/contracts/libraries/ProtocolRoles.sol

- /src/contracts/libraries/TokenSupply.sol
- /src/contracts/libraries/UniswapHookEvents.sol
- /src/contracts/PositionManager.sol
- /src/contracts/price/FlaunchFeeExemption.sol
- /src/contracts/subscribers/Indexer.sol
- /src/contracts/treasury/ActionManager.sol
- /src/contracts/treasury/MemecoinTreasury.sol
- /src/contracts/types/MemecoinFinder.sol
- /src/contracts/types/StoreKeys.sol
- /src/contracts/types/TickFinder.sol
- /src/contracts/zaps/PoolSwap.sol
- revenue-manager-clone-0x3Bc08524d9DaaDEC9d1Af87818d809611F0fD669
 - /src/contracts/bidwall/BidWall.sol
 - /src/contracts/escrows/FeeEscrow.sol
 - /src/contracts/escrows/ReferralEscrow.sol
 - /src/contracts/Flaunch.sol
 - /src/contracts/hooks/FairLaunch.sol
 - /src/contracts/hooks/FeeDistributor.sol
 - /src/contracts/hooks/FeeExemptions.sol
 - /src/contracts/hooks/InternalSwapPool.sol
 - /src/contracts/hooks/Notifier.sol
 - /src/contracts/libraries/CurrencySettler.sol
 - /src/contracts/libraries/ProtocolRoles.sol
 - /src/contracts/libraries/TokenSupply.sol
 - /src/contracts/libraries/UniswapHookEvents.sol
 - /src/contracts/PositionManager.sol
 - /src/contracts/price/FlaunchFeeExemption.sol
 - /src/contracts/subscribers/Indexer.sol
 - /src/contracts/treasury/ActionManager.sol
 - /src/contracts/treasury/managers/RevenueManager.sol
 - /src/contracts/treasury/managers/TreasuryManagerFactory.sol

- /src/contracts/treasury/managers/TreasuryManager.sol
- /src/contracts/treasury/MemecoinTreasury.sol
- /src/contracts/types/MemecoinFinder.sol
- /src/contracts/types/StoreKeys.sol
- /src/contracts/types/TickFinder.sol
- /src/contracts/zaps/PoolSwap.sol
- /src/contracts/bidwall/BidWall.sol
- /src/contracts/escrows/FeeEscrow.sol
- /src/contracts/escrows/ReferralEscrow.sol
- /src/contracts/Flaunch.sol
- /src/contracts/hooks/FairLaunch.sol
- /src/contracts/hooks/FeeDistributor.sol
- /src/contracts/hooks/FeeExemptions.sol
- /src/contracts/hooks/InternalSwapPool.sol
- /src/contracts/hooks/Notifier.sol
- /src/contracts/libraries/CurrencySettler.sol
- /src/contracts/libraries/ProtocolRoles.sol
- /src/contracts/libraries/TokenSupply.sol
- /src/contracts/libraries/UniswapHookEvents.sol
- /src/contracts/PositionManager.sol
- /src/contracts/price/FlaunchFeeExemption.sol
- /src/contracts/subscribers/Indexer.sol
- /src/contracts/treasury/ActionManager.sol
- /src/contracts/treasury/managers/RevenueManager.sol
- /src/contracts/treasury/managers/TreasuryManagerFactory.sol
- /src/contracts/treasury/managers/TreasuryManager.sol
- /src/contracts/treasury/MemecoinTreasury.sol
- /src/contracts/types/MemecoinFinder.sol
- /src/contracts/types/StoreKeys.sol
- /src/contracts/types/TickFinder.sol
- /src/contracts/zaps/PoolSwap.sol

- treasury-action-manager-0xFB5c20c4E60c9c64648DD3692437E3E313Add4A4
 - /src/contracts/treasury/ActionManager.sol

Findings

Each issue has an assigned severity:

- High issues are directly exploitable security vulnerabilities that need to be fixed.
- Medium issues are security vulnerabilities that may not be directly exploitable or may require certain conditions in order to be exploited. All major issues should be addressed.
- Low/Info issues are non-exploitable, informational findings that do not pose a security risk or impact the system's integrity. These issues are typically cosmetic or related to compliance requirements, and are not considered a priority for remediation.

1. Executive Summary

\$ODAI is a Flaunch Memecoin clone with a clean launch profile and a holder-facing surface that contains **no blacklist, no wallet-freeze, no creator-controlled mint, and no perpetual transfer restriction**. Launch integrity is strong and now proven at the transaction level: the full 100,000,000,000 supply was minted to the PositionManager (not the creator), there was **no pre-mine and no team allocation**, and selling is hard-blocked during the fair-launch window.

The material risk is **centralization / third-party trust**:

- Essentially all Flaunch protocol infrastructure (PositionManager, FeeEscrow, BidWall, TreasuryActionManager, the FLAUNCH ERC-721) is owned by a single **Flaunch 3-of-5 Safe**, which can set the pool swapFee up to 100% and controls the approved-action allowlist of the Treasury.
- The **RevenueManager clone** that *would* route ODAI creator fees is a shared, multi-tenant contract owned by an **unrelated third party** (Axiom-bot EOA) – though it is **currently inactive for ODAI** because the position NFT sits with the creator, not the manager.
- The creator can route treasury funds to arbitrary recipients via an approved treasury action (DistributeAction).

No critical/holder-fund-loss vulnerability was identified in the in-scope code. Findings are centralization and accuracy-of-claims-driven (see §13).

2. Verified Claims

V1 — No pre-sale. Full supply minted to PositionManager at flaunch (Flaunch.sol:190); no pre-sale path.

V2 — No pre-mine. Creation tx 0xffa1a30f... mints **100B → PM, 0 → creator; zero PoolPremium events** for ODAI's poolId (launch block 42,057,936).

V3 — No team allocation. No supply slice reserved; unsold fair-launch supply is burned (FairLaunch.sol:178,188); creator receives only the ERC-721.

V4 — Fair-launch window (uniform pricing). FairLaunch.createPosition opens a fixed-price single-sided position for the window (:118-147).

V5 — Anti-dump during launch window. Sells revert with CannotSellTokenDuringFairLaunch while the window is open (PositionManager.sol:480). Window-scoped; expires at close.

V6 — Post-window AMM (Uniswap V4, ODAI/fiETH). FairLaunch.closePosition creates an immutable V4 position (:159-201); live pool 0xbf0f7169... on PoolManager 0x498581ff...

V7 — No creator-controlled mint. Memecoin.mint is onlyFlaunch (:90,307); creator has no mint authority.

V8 — No blacklist / freeze / transfer restriction. Memecoin is a standard ERC-20 (mint/permit/crosschain only); no `_beforeTokenTransfer` hook, cooldown, or pause.

V9 — PBW accumulates fee share. `deposit()` (only `PositionManager`) increments `pendingETHFees/cumulativeSwapFees` (`BidWall.sol:168`); ODAI lifetime about 25.66 ETH.

V10 — PBW places limit buys below spot. `_reposition` opens a single-sided native position below spot (`:227`); ODAI live band 188580–188640.

V11 — PBW destination not redirectable. All exits hardcode `memecoin.treasury()`; `BidWall` is non-upgradeable (plain runtime, not a proxy).

3. Rejected / Overstated Claims

R1 — Anti-whale cap (0.25%): Not in code. No per-wallet / max-buy / 0.25% limit anywhere in fair-launch or token code. Fair-launch buys fill from supply at uniform price with no holder cap.

R2 — In-scope BidWall 0x66681f10 is ODAI's: Wrong contract. That `BidWall` is wired to a *different* `PositionManager` (`0x51Bba152...`) and is **legacy**. ODAI's live `BidWall` is `0x7f22353d...` (from `positionManager.bidWall()`).

R3 — Memestream NFT in RevenueManager custody: Not current. `ownerOf(8917)` = creator EOA `0x9349...`, not the RM. The NFT was rescued out; RM holds only a stale registration.

R4 — RevenueManager actively routes/taxes ODAI creator fees: Inactive for ODAI. Because the NFT isn't in the RM, `memecoin.creator()` = the EOA, so `creatorFee` is allocated directly to it; the RM's 10% does not apply at current state.

R5 — Treasury cannot dump via approved actions: Currently inaccurate. `DistributeAction` (`0x8FFf...`) is **currently approved** (`approvedActions = true`), so the `poolCreator` can route treasury tokens to **arbitrary recipients today**. Approved set also includes `ClaimFees`, `BuyBack`, and a `Burn` action; all four `approved = true`.

R6 — Anti-dump = ERC-20 transfer restriction: Imprecise. The protection is a *swap-level sell-block during the window*, not a token-transfer freeze; wallet-to-wallet transfers always work.

4. Role Map

`0x0646441c3eC556c1994420C3F9B0f31279BEfb6e` — **Flaunch 3-of-5 Safe.** Owner of `PositionManager` (`0x23321f11`), `FeeEscrow` (`0x72e6f794`), `BidWall` (`0x7f22353d`), `TreasuryActionManager` (`0xFB5c20`), `FLAUNCH ERC-721` (`0x516af52D`), `TreasuryManagerFactory` (`0x48af8b28`).

0x93499d74Af0485D1dAe23250992f55972fd744e0 — **EOA**. ODAI creator() + ownerOf(NFT #8917) → the poolCreator for treasury actions and the live creatorFee recipient.

0x0c8b9d0a7e5Bd2E66270aD02FF1E4EFa6BADface — **EOA (EIP-7702 → Axiom)**. RevenueManager clone managerOwner + protocolRecipient.

0x510a08954fD7096F67b58Ed210703b323a5f405E — **EOA (EIP-7702 → same Axiom impl)**. RM internal end-creator label for ODAI (stale).

0x547289319C3e6aedB179C0b8e8aF0B5ACd062603 — **EOA**. ERC-8004 Agent Registry proxy owner.

SUPERCHAIN_TOKEN_BRIDGE (**predeploy**) — **system**. Only address allowed to crosschainMint / crosschainBurn ODAI.

Safe signers (3-of-5): 0xDaeA0a29...37f0, 0x722cfd9F...641E, 0xB8525477...2660, 0x36421d50...b88B, 0x3CBbB5E8...92Cc.

Key contracts: ODAI 0x0086cff0...; Memecoin impl 0xF1EEeeee...; Treasury clone 0x14926224... (impl 0x73973903...); RevenueManager clone 0x3Bc08524... (impl 0x712FA8...); f1ETH 0x0000...7Cf8; V4 PoolManager 0x498581ff...; ODAI poolId 0xbf0f7169...b3fb.

5. Privileged Function Map

setFeeDistribution / setPoolFeeDistribution — **PositionManager**. Gate: onlyOwner (Safe). Effect: set swapFee up to **100%**.

setProtocolFeeDistribution — **PositionManager**. Gate: flayGovernance. Effect: protocol fee only, capped 10%.

approveAction / unapproveAction — **TreasuryActionManager**. Gate: onlyOwner (Safe). Effect: control the global approved-action allowlist.

executeAction — **MemecoinTreasury**. Gate: poolCreator + approved action. Effect: run an approved action with max allowance on both pool tokens.

setSwapFeeThreshold / setStaleTimeWindow — **BidWall**. Gate: onlyOwner (Safe). Effect: tune PBW trigger threshold / stale window (config, not redirect).

setDisabledState — **BidWall**. Gate: poolCreator. Effect: disable PBW → liquidity withdrawn to treasury.

closeBidWall / deposit — **BidWall**. Gate: onlyPositionManager. Effect: internal fee/liquidity ops.

setCreator / setProtocolRecipient / rescue / transferManagerOwnership — **RevenueManager**. Gate: onlyManagerOwner (3rd-party EOA). Effect: reassign end-creator, redirect protocol fee, pull NFT out, transfer ownership.

mint — **Memecoin**. Gate: onlyFlaunch. Effect: mint ODAI (observed: one-time INITIAL_SUPPLY).

`crosschainMint` / `crosschainBurn` — **Memecoin**. Gate: `onlySuperchain`. Effect: cross-chain supply move (supply-neutral).

`approveManager` / `unapproveManager` — **TreasuryManagerFactory**. Gate: `onlyOwner` (Safe). Effect: control approved manager implementations.

6. Mint / Burn / Bridge Analysis

- **Mint authority** is `onlyFlaunch` (`Memecoin.sol:90,307`): only the Flaunch contract can call `mint`. **The creator has no mint path**. Observed usage is a single `mint(positionManager, 100e27)` at `flaunch` (`Flaunch.sol:190`); the creation tx shows exactly that one mint and nothing to the creator.
- **Burn**: `burn`/`burnFrom` operate on the caller's own balance/allowance (standard). Unsold fair-launch supply is burned to `BURN_ADDRESS` at window close.
- **Bridge**: `crosschainMint`/`crosschainBurn` are restricted to the **Superchain token bridge predeploy** (`onlySuperchain, :317`). These are supply-neutral cross-chain moves, not a discretionary inflation path. Flaunch also has a `MAX_BRIDGING_WINDOW = 1 hour` constraint on bridging flows.

Holder takeaway: no unrestricted/creator inflation; cross-chain mint is bridge-gated.

7. Transfer Restriction Analysis

- The `Memecoin` is a **standard ERC-20** (`OZ` + `permit` + `Superchain crosschain`). There is **no `_beforeTokenTransfer`/`_update` override, no cooldown, no max-wallet, no pause, no blacklist**. Wallet-to-wallet transfers are unrestricted at all times.
 - The **only** trading restriction is the **fair-launch sell-block** (`CannotSellTokenDuringFairLaunch, PositionManager.sol:480`): during the launch window the hook reverts sell swaps (buys allowed). It is **window-scoped and expires** at window close.
 - **Net**: genuine anti-dump *during launch*, but not a perpetual transfer restriction. Phrase publicly as "sells were blocked during the fair-launch window," not "transfers are restricted."
-

8. Liquidity / Pool Control Analysis

- `ODAI` trades on a **Uniswap V4** pool (`f1ETH/ODAI`, poolId `0xbf0f7169...`) with the `PositionManager` (`0x23321f11`) as the hook; the post-window position is created **immutable** by `closePosition`.

- **No classic founder LP rug:** the creator holds no privileged LP withdrawal path; fair-launch liquidity is protocol-placed and unsold supply burned.
 - **Fee lever:** the pool `swapFee` is owner-settable up to 100% by the Flaunch Safe (per-pool). This is a market-quality/centralization lever, not a supply or LP-withdrawal lever.
 - **PBW liquidity** is single-sided and owned by the BidWall contract; see §11.
-

9. Treasury Action Analysis

- ODAI treasury `0x14926224...` (EIP-1167 clone of `0x73973903...`). `executeAction` requires (a) the action be approved in the TreasuryActionManager (Safe-controlled) and (b) `msg.sender == poolCreator`. During execution the treasury grants `type(uint).max` allowance on both pool tokens, then revokes.
 - **Approved action set (live, all approved = true):** `ClaimFeesAction` (`0x3db1d2d3...`), a Burn-type action (`0x8696a1f2...`), `BuyBackAction` (`0xda4866c9...`, `poolSwap` → `0x4c211268...`), and `DistributeAction` (`0x8FFfDd52...`).
 - **Drainability — active, not hypothetical:** because `DistributeAction` is approved right now, the `poolCreator` can call `executeAction(DistributeAction, ...)` today and send the treasury's ~4.65B ODAI to **arbitrary recipients** (decoded from calldata). So "treasury cannot dump" is **false at current state**, not merely "depends on what the Safe approves." The gate that remains is `msg.sender == poolCreator` (`0x9349...`).
 - **Centralization:** the approved-action set is a Safe-controlled trusted-code registry; a malicious/compromised Safe approving a hostile action, combined with a creator (or compromised creator) executing it, is the worst case. 3-of-5 threshold mitigates single-key compromise.
 - Treasury currently holds ~4.65B ODAI (approximately 4.65% of supply, from PBW buyback rewards) and 0 FIETH/ETH.
-

10. RevenueManager / Creator Path Analysis

- The clone `0x3Bc08524...` is a **logic-less EIP-1167 proxy** (`delegatecalls impl 0x712FA8...`), **shared/multi-tenant** (~3,466 tokens), owned by a **third-party Axiom-bot EOA** (`0x0c8b...BADface`), with a **10% protocolFee**.
- **Mechanism:** the RM only earns ODAI fees if it is `memecoin.creator()` — i.e. if it custodies the position NFT. The PositionManager then routes the `creatorFee` branch to it (`PositionManager.sol:953,959` → `feeEscrow.allocateFees`).
- **Current state: INACTIVE for ODAI.** NFT #8917 is held by the creator EOA, so `creatorFee` is allocated directly to the EOA and the RM's 10% does not apply.

- **Two distinct "creator" notions:** `memecoin.creator()` = live NFT owner (0x9349...); `RevenueManager.creator[]` = a **set-once storage label** (0x510a...) written at `deposit()` and changeable by the manager owner via `setCreator()`, **not cleared by `rescue()`** (hence the stale value). The RM's "creator" is operator-controlled and non-authoritative.
 - **If re-deposited**, the third-party owner could `setCreator/setProtocolRecipient` and skim 10% but not currently in effect.
-

11. PBW (Progressive Bid Wall) Analysis

- ODAI's live BidWall 0x7f22353d... (owner = Safe; **non-upgradeable**). Enabled for the pool (`isBidWallEnabled = true`), initialized, single-sided position at ticks 188580–188640.
 - **Fee accrual & trigger:** fees accumulate as `pendingETHFees`; the wall (re)deploys once `pendingETHFees` is greater than or equal to `swapFeeThreshold` (default **0.1 ETH**, `BidWall.sol:121`). The threshold is `onlyOwner` (Safe), **not** creator-settable.
 - **ETH does not pile up:** the wall is placed just below spot, so ETH continuously converts to ODAI; on each reposition the **bought ODAI is skimmed to the treasury** (: 294–298) and only unfilled ETH is redeployed. Live: ~0.00664 ETH standing + ~0.00036 pending = ~0.007 ETH in the wall, vs lifetime 25.66 ETH routed through and ~4.65B ODAI now in the treasury.
 - **Drain by founder:** funds always exit to `memecoin.treasury()`. The creator can `setDisabledState(true)` (creator-gated) to collapse the wall into the treasury, then extract via an approved treasury action. So the drain path is **through the treasury**, gated to the creator role, and event-emitting. Still, the Bidwall continuously sells ETH for \$ODAI, so ETH draining is not really applicable.
 - **Redirect — confirmed not possible:** destination is hardcoded; owner levers are config-only.
-

12. Future Reward Vault Concerns

A "Reward Vault / payout rail" is referenced as a **future** ODEI component. **It is not deployed and not in the current address set** – nothing on-chain to verify today. Before any live distribution, it must be reviewed separately for:

- **Custody model:** where reward funds sit, and who can withdraw (owner/multisig/timelock vs. permissionless claim).
- **Payout authority:** can an admin redirect, pause, or drain the vault? Upgradeability/proxy admin.

- **Accounting:** claim math, double-claim/reentrancy on payout, snapshot integrity vs. on-chain balances.
- **Funding source:** whether it pulls from the treasury (and thus inherits the `poolCreator`/Safe trust assumptions) or from an independent source.
- **Token interactions:** any new mint/allowance grants to the vault.

Do not publish any safety assurance about reward distribution until this contract exists and is reviewed.

13. Severity-Ranked Findings

[Low] Flaunch Safe can set pool swapFee to 100%. `setFeeDistribution/setPoolFeeDistribution` (`onlyOwner`), no on-chain cap below 100%, no timelock observed. Per-pool targetable. (§5, §8)

[High] Treasury funds drainable by creator today via approved `DistributeAction`. `DistributeAction` (`0x8FFf...`) is **currently approved**; `poolCreator` can send the treasury's ~4.65B ODAI to arbitrary recipients now. Only gate is `msg.sender == poolCreator`. (§9)

Recommended mitigation: place the `poolCreator` / NFT #8917 control behind a timelock; gating treasury actions this way removes the ability to drain on demand and reduces this finding's severity.

[Info] Single 3-of-5 Safe owns all protocol infra. `PositionManager`, `FeeEscrow`, `BidWall`, `ActionManager`, FLAUNCH ERC-721, `Factory`. (§4)

[Low] `RevenueManager` is third-party-owned with 10% fee + `setCreator` power. Currently **inactive** for ODAI (NFT not in custody); becomes a live trust dependency only if NFT is (re)deposited. (§10)

[Info] Stale RM `creator/internalId` mapping after `rescue`. Mapping not cleared on withdrawal; non-authoritative leftover (`0x510a...`). (§10)

[Info] Scope lists inactive instances for \$ODAI. The in-scope `BidWall` (`0x66681f10`) and its `PositionManager` (`0x51Bba152...`) are **legacy**; ODAI's live `BidWall` is `0x7f22353d...`. The in-scope `RevenueManager` (`0x3Bc08524...`) has **no live fee rights over ODAI**: the Memestream NFT was rescued out, so `memecoin.creator()` = creator EOA `0x9349...` and the RM's 10% does not apply at current state. It becomes relevant only if the NFT is deposited.

Disclaimers

Sherlock does not provide guarantees nor warranties relating to the security of the project.

Usage of all smart contract software is at the respective users' sole risk and is the users' responsibility.