# Meridion Risk Assessment: USDS (Sky Dollar)

| FIELD | VALUE |
|---|---|
| Digital Asset | USDS (Sky Dollar) |
| Risk Areas | Smart Contract, Operations, Financials |
| Chains | Ethereum Mainnet (Chain ID: 1) |
| Report Version | 1.0.0 |
| Assessment Period | 2026-04-23 to 2026-05-06 |
| Date of Publication | 2026-05-08 |
| Requested by | Non-issuer |

## 1. Methodology

### 1.1 Rating Standard and Assessment Framework

This report is produced under the **Meridion Risk Rating Standard v1**. The engagement was conducted by a team combining smart contract security specialists, financial risk analysts, and formal methods engineers.

Each independent domain receives a risk rating of **Low**, **Medium**, or **High**:

- **Low** applies when no material adverse condition was identified within the assessed scope and residual risks are ordinary for the asset type.
- **Medium** applies when a material weakness, dependency, opacity, concentration, or stress scenario exists but is contingent, mitigated, not currently causing holder harm, or primarily a future-risk driver.
- **High** applies when a direct or credible path exists to material holder loss, unauthorized issuance, severe depeg, impaired transferability, asset shortfall or misrepresentation, governance or administrator capture, or operational failure.

The composite risk rating is derived from the three domain ratings using the following labels:

- **Minimal:** all three domain ratings are Low, with no material identified weakness beyond ordinary residual risk for the asset type.
- **Low:** no domain rating reaches High, and at most one reaches Medium; any Medium condition is isolated or future-oriented, with no immediate consequence for holders.
- **Moderate:** no domain rating reaches High, but multiple domains carry a Medium rating, or one Medium domain carries direct consequence, structural importance, or meaningful interaction with another domain.
- **Elevated:** at least one domain rating reaches High, but the condition remains contingent, is not actively causing holder harm, and does not currently show immediate adverse consequences.
- **High:** immediate adverse consequences are present, or an identified High domain condition bears directly on holders.

- **Severe:** multiple domains are critically compromised, or a single compromised domain produces cascading failure across the asset.

Each domain and the composite conclusion also receive a confidence rating of **Low**, **Medium**, or **High**. Confidence measures the reliability, completeness, independence, recency, and reproducibility of the evidence supporting the risk rating. Confidence does not reduce or soften the risk rating; instead, it tells the reader how strongly the rating is supported and whether material scope limitations reduce certainty.

## 1.2 Review Techniques Applied

### SMART CONTRACTS: FUNCTIONALITY AND SECURITY

- **Runtime entry-point catalogue:** reconstruction of the complete externally callable runtime surface from deployed bytecode and verified source artifacts
- **Manual code review:** line-by-line inspection of all in-scope source files by independent security specialists
- **Edge-case and exploit-negative review:** targeted analysis of protocol-specific invariants, failure modes, access-control boundaries, replay protections, and upgrade assumptions
- **Formal verification:** direct deployed-bytecode verification of hidden calldata surface, covering selectors outside the complete runtime catalogue and malformed calldata shorter than 4 bytes
- **Fork-based behavioral testing:** live-chain execution of representative positive and negative behavioral tests against deployed bytecode using Foundry; test counts, entry-point coverage, and PASS/FAIL status are reported in the bytecode assurance section
- **AI-assisted analysis:** automated pattern detection and anomaly scanning across the full codebase to supplement human review

### OPERATIONS: KEY MANAGEMENT AND ADMINISTRATIVE CONTROL

- **Role mapping:** reconstruction of deployment and administrative authority from on-chain state and historical events
- **Key management review:** assessment of HSM, MPC, and key ceremony controls based on SOC reports and public disclosures
- **Monitoring and alerting:** assessment of on-chain event alerting coverage and anomaly detection arrangements for privileged operations and administrative actions
- **Recovery assessment:** review of signer loss, compromise, and administrative recovery procedures

### FINANCIALS: UNDERLYING ASSETS AND COUNTERPARTIES

- **Reserve attestation analysis:** comparison of attestation disclosures against on-chain liabilities and supply metrics
- **Counterparty profiling:** identification of issuer, custodian, banking, and attestation dependencies and their risk posture
- **Liquidity analysis:** assessment of redemption capacity and market depth across DEX and CEX venues
- **Scenario analysis:** review of economic stress scenarios and their implications for solvency, redemptions, and price stability.

# 2. Executive Summary

### Subject Overview

USDS (Sky Dollar) is the native USD-pegged stablecoin of Sky Protocol, formerly known as MakerDAO, the largest decentralised collateralised debt position protocol on Ethereum. USDS is minted permissionlessly by users who deposit qualifying collateral into over-collateralised vault positions (ETH, wBTC, stETH, USDC, and real-world asset tokens), or by swapping USDC through the Peg Stability Module at a 1:1 rate, creating a crypto-backed and partially RWA-backed stable asset rather than a fiat-reserve model. Sky Protocol renamed DAI to USDS in 2024 as part of its broader rebranding, adding on-chain blacklisting capability to align with OFAC compliance practices. As of block 24,941,472 (2026-04-23), the Ethereum on-chain circulating supply was approximately 7.53 billion USDS; the CoinGecko-reported all-chain figure was approximately 11.36 billion USDS, with the gap attributable to deployments on Solana, Arbitrum, Base, and Sui. USDS is used as the primary stable asset across Sky Protocol's own lending products (SparkLend, sUSDS), and is broadly integrated in third-party DeFi protocols including Morpho, Aave v3, and Pendle.

### KYC Gating

USDS is a permissionless protocol-issued asset. Secondary-market holding requires no KYC from any issuer or protocol. Minting via CDP vaults or the Peg Stability Module is permissionless and requires no identity verification at the protocol level; any party with qualifying collateral can mint USDS on-chain. Redemption of the underlying collateral for vault holders is similarly permissionless: a vault holder repays USDS debt in a single on-chain transaction to recover their collateral, with no minimum size, settlement delay, or KYC requirement. Non-vault holders access liquidity through secondary markets or the PSM, not through a dedicated redemption desk. The USDS token contract includes a `deny(address)` function callable by the ward role (a governance-controlled privileged address), which can block specific Ethereum addresses from calling token functions; this mechanism has been applied for OFAC-sanctioned addresses in a manner consistent with industry practice for USDC and USDT. No allowlisting mechanism exists for secondary trading.

### Overall Risk Rating: Moderate

#### Composite Confidence: Medium

USDS presents a Moderate composite risk profile, driven by Medium ratings in both Operational Security and Financial Integrity, with Smart Contract Security rated Low. The operational risk is the flat ward role that can control minting, role management, and UUPS contract upgrades at the token-contract level. Supplemental Rely/Deny event reconstruction verified two active wards at the assessment snapshot and at the follow-up latest block: `0x3c0f895007ca717aa01c8693e59df1e8c3777feb` ( `UsdsJoin` ) and `0xbe8e3e3618f7474f8cb1d074a26affef007e98fb` ( `DSPauseProxy` ). `UsdsJoin` is an accounting adapter whose mint path is constrained by Vat balance movement; `DSPauseProxy` is the delayed governance execution path owned by `DSPause` , authorized through `Chief` , and subject to a 24-hour pause delay before scheduled governance actions execute. Financially, the CDP over-collateralisation model provides continuous on-chain-verifiable reserve adequacy, but meaningful risks remain: limited secondary-market executable DEX liquidity ($181.8M) relative to the $11.4 billion circulating supply, an unverified multi-chain supply gap of approximately 3.83 billion USDS, RWA vault counterparty dependencies with 30- to 90-day settlement timelines, and evolving US and EU regulatory frameworks that could restrict distribution or impose

compliance obligations. The smart contract surface is lean and well-controlled, with only one Low-severity finding and formal verification confirming the absence of hidden callable bytecode surfaces. Fork-based behavioral tests were executed against the deployed bytecode, covering all 11 state-mutating entry points, and all required cases passed. Composite confidence is Medium because the active ward identities and first-layer control paths are verified, while deeper governance token concentration, spell process, emergency authority, off-chain operational controls, and RWA vault composition remain outside independently verified evidence.

### Smart Contract Security Summary

| SEVERITY | COUNT | RESOLVABLE |
|---|---|---|
| Critical | 0 | 0 |
| High | 0 | 0 |
| Medium | 0 | 0 |
| Low | 1 | 1 |

The USDS smart contract system consists of a minimal ERC1967 UUPS proxy and a Usds implementation contract, both compiled with Solidity 0.8.21 and deployed with verified source code. The review identified one Low-severity finding: SEC-01, a theoretical ward lockout condition arising from the absence of a minimum ward count check in the `deny()` function, which could permanently disable all administrative functions if all wards are revoked. No Critical, High, or Medium findings were identified. The contract's attack surface is narrow: most critical operations are gated by the `auth` modifier, arithmetic is safe under Solidity 0.8 built-in overflow protection with justified unchecked blocks, signature replay is prevented via per-owner EIP-712 nonces with chain ID binding, and the UUPS upgrade pattern is correctly implemented with `_disableInitializers()` and `onlyProxy` guards. Formal verification of the deployed bytecode returned CONFIRMED on all hidden-surface absence and proxy-routing cases. Fork-based behavioral tests covering all 11 state-mutating entry points were executed against the deployed bytecode; all 21 required test cases passed. Confidence is rated High.

**Smart Contract Risk Score: Low | Confidence: High**

### Operational Security Summary

The USDS token contract employs a flat ward role that grants any ward holder token-level authority over minting, access control, and UUPS contract upgrades. Supplemental Rely/Deny event reconstruction identified two active wards at block `24,941,472`: `UsdsJoin` (`0x3c0f895007ca717aa01c8693e59df1e8c3777feb`) and `DSPauseProxy` (`0xbe8e3e3618f7474f8cb1d074a26affef007e98fb`). `UsdsJoin` is not an arbitrary administrator; it is an accounting adapter whose `exit` flow requires a corresponding `Vat.move` before minting USDS. `DSPauseProxy` is the governance execution ward: its owner is `DSPause`, `DSPause` is authorized by `Chief`, and the active `Chief.hat` spell schedules execution through a 24-hour delay. No multisig threshold is enforced at the USDS token level and no SOC 2 or equivalent independent operational security attestation has been published. Sky Protocol has operated its DAI and USDS token contracts since 2017 without a

publicly documented key compromise or operational control failure. The operational security risk is rated Medium because governance can still execute privileged actions, while confidence is Medium because first-layer ward identity is verified but deeper governance and off-chain operational controls were not fully audited.

**Opsec Risk Score: Medium | Confidence: Medium**

### Financial Summary

USDS is backed by a diversified over-collateralised CDP system covering ETH-family vaults, USDC via the Peg Stability Module, and RWA vault exposures in US Treasuries and structured credit. The structural over-collateralisation enforced at vault level prevents under-collateralisation without triggering on-chain liquidations, and the Surplus Buffer provides a first-loss capital layer. Key financial risks are: simultaneous severe crypto collateral price declines (tail risk analogous to March 2020); approximately 3.83 billion USDS in multi-chain deployments not independently verified per chain; RWA vault exposure via Monetalis and BlockTower with 30- to 90-day wind-down timelines; thin secondary-market executable DEX liquidity ($181.8 million conservative estimate) relative to total circulating supply; and regulatory uncertainty under US GENIUS Act and EU MiCA that could restrict distribution. The Ethereum on-chain supply and market price are independently verified; exact collateral composition was not directly queried at snapshot. S&P Global assigned a 'B-' issuer credit rating to Sky Protocol in August 2025, the first major-agency rating for a decentralised CDP protocol. No third-party financial attestor is appointed, which is a structural feature of the CDP model rather than an omission. Financial risk is rated Medium; confidence is Medium.

**Financial Risk Score: Medium | Confidence: Medium**

### Scope Limitations

- **Cross-chain deployments (Solana, Arbitrum, Base, Sui) excluded from scope.** Approximately 3.83 billion USDS is deployed on non-Ethereum chains; per-chain supply totals were not independently verified. Affects confidence in the financial domain.
- **RWA vault composition (Monetalis Clydesdale, BlockTower Credit) not independently audited.** Underlying asset quality and liquidity were assessed using Sky governance forum disclosures only, not independently confirmed documents or third-party attestation. Affects both the risk rating and confidence in the financial domain.
- **Broader minting module and governance ecosystem not fully assessed.** The active `UsdsJoin` adapter and `DSPauseProxy` first-layer control path were reviewed, but the broader Sky governance process, token-holder concentration, spell review workflow, emergency authority, and off-chain operations were not fully audited. Affects confidence in the operational security domain.
- **Sky Protocol governance system (SKY token contracts, Pause Proxy, executive spell executor) outside scope.** Governance attack vectors and governance contract security are not assessed in this report. Affects confidence in the operational security domain.

## 3. Part I: Smart Contract Security Analysis

**Smart Contract Security Rating: Low | Confidence: High**

### 3.1 Scope and Execution Environment

**Chain:** Ethereum Mainnet

**Assessment snapshot block:** 24,941,472 (2026-04-23)

| CONTRACT | ROLE | ADDRESS | COMPILER | V |
|---|---|---|---|---|
| ERC1967Proxy | Proxy | `0xdC035D45d973E3EC169d2276DDab16f1e407384F` | `v0.8.21+commit.d9974bed` | Y |
| Usds | Implementation | `0x1923dfee706a8e78157416c29cbccfde7cdf4102` | `v0.8.21+commit.d9974bed` | Y |

Both contracts are compiled with Solidity 0.8.21, optimized with 200 runs, targeting the `paris` EVM version. Source code is verified on Blockscout (Ethereum Mainnet). The proxy is a minimal ERC1967Proxy with no admin address or named upgrade functions; upgrade authority resides entirely in the implementation via the UUPS pattern. The implementation ( `Usds` ) extends OpenZeppelin Contracts Upgradeable v5.0.0, specifically `UUPSUpgradeable` , `ERC1967Utils` , and `Initializable` , in addition to a custom ERC-20 implementation derived from the MakerDAO DSS auth pattern.

### 3.2 Entry Point Catalogue

*Proxy contract entry points:*

| ID | SIGNATURE | ACCESS GATED? | CRITICALITY | DESCRIPTION |
|---|---|---|---|---|
| EP-PR-001 | `fallback()` | Public (delegates all calls to implementation) | High | Routes all calldata to the USDS implementation via EIP-1967 delegatecall. As the sole runtime entry point of the proxy, every state-modifying operation on the token flows through this handler. Misconfiguration of the implementation slot or delegatecall target would affect all implementation functions. |

The proxy contract includes no additional functions that are not the fallback; there are no named functions in the proxy ABI. ETH sent with empty calldata is forwarded through the fallback delegatecall to the implementation, which lacks a `receive()` handler and reverts, so no ETH can be deposited to the proxy.

*Implementation contract entry points:*

| ID | SIGNATURE | ACCESS GATED? | CRITICALITY | DESCRIPTION |
|---|---|---|---|---|
| EP-IM-001 | `burn(address,uint256)` | Public; requires `from` == `msg.sender` or sufficient allowance from `from` | High | Destroys `value` USDS tokens from `from`'s balance and reduces `totalSupply`. No ward gate. Supply integrity depends on correct allowance enforcement. Infinite allowance (uint256.max) bypasses allowance decrement. |
| EP-IM-002 | `deny(address)` | `auth` modifier (ward only) | High | Revokes ward status from `usr` by setting `wards[usr] = 0`. Any ward can deny any other ward, including themselves. No minimum ward count is enforced; denial of the last ward permanently locks all admin-gated functions. (SEC-01) |
| EP-IM-003 | `initialize()` | OZ `initializer` modifier (one-time only) | High | One-time initializer executed at proxy deployment. Sets `wards[msg.sender] = 1`, granting ward status to the deployer. Protected against replay by OZ's `initializer` slot mechanism. The implementation constructor calls `_disableInitializers()`, preventing direct initialization. |
| EP-IM-004 | `mint(address,uint256)` | `auth` modifier (ward only) | High | Creates `value` new USDS tokens and assigns them to `to`. Access-gated. Rejects zero address and the contract itself as recipient. Supply integrity depends entirely on ward access control. |
| EP-IM-005 | `rely(address)` | `auth` modifier (ward only) | High | Grants ward status to `usr` by setting `wards[usr] = 1`. Access-gated. Allows expansion of the privileged set with no upper bound on ward count. No zero-address check; `rely(address(0))` is harmless because |

| ID | SIGNATURE | ACCESS GATED? | CRITICALITY | DESCRIPTION |
|---|---|---|---|---|
| | | | | `address(0)` cannot sign transactions. |
| EP-IM-006 | `upgradeToAndCall(address,bytes)` | `auth` modifier + `onlyProxy` guard (ward only, must call via proxy) | High | UUPS upgrade function. Replaces the EIP-1967 implementation slot with `newImplementation` in a single transaction. If `data` is non-empty, delegatecalls into the new implementation for initializer execution. The new implementation must return the correct `proxiableUUID()` or the upgrade reverts. Any ward can upgrade immediately with no timelock. |
| EP-IM-007 | `approve(address,uint256)` | Public | Medium | Sets the caller's allowance for `spender` to `value`. Standard ERC-20 approve. No increaseAllowance or decreaseAllowance variants; the known ERC-20 approve race condition applies. |
| EP-IM-008 | `permit(...) [5 args]` | Public; requires valid EIP-712 signature from `owner` (compact bytes form, supports ERC-1271) | Medium | EIP-2612 permit variant accepting a compact signature bytes blob. Increments `nonces[owner]` before ERC-1271 staticcall, preventing replay. Chain ID is bound in the domain separator. |
| EP-IM-009 | `permit(...) [7 args]` | Public; requires valid EIP-712 signature from `owner` (v/r/s form) | Medium | EIP-2612 permit variant accepting split v/r/s signature components. Shares nonce and domain separator logic with EP-IM-008. Single-use per nonce per owner. |
| EP-IM-010 | `transfer(address,uint256)` | Public | Medium | Transfers `value` USDS from `msg.sender` to `to`. Requires sufficient balance. Rejects zero address and self-transfer to the contract address as recipient. No external calls. |

| ID | SIGNATURE | ACCESS GATED? | CRITICALITY | DESCRIPTION |
|---|---|---|---|---|
| EP-IM-011 | `transferFrom(address,address,uint256)` | Public; requires `from` == `msg.sender` or sufficient allowance from `from` | Medium | Transfers `value` USDS from `from` to `to` on behalf of `msg.sender`. Infinite allowance (uint256.max) bypasses allowance decrement. No external calls. |

The implementation contract includes 14 additional functions that are not state-modifying (view or pure): `DOMAIN_SEPARATOR`, `PERMIT_TYPEHASH`, `UPGRADE_INTERFACE_VERSION`, `allowance`, `balanceOf`, `decimals`, `getImplementation`, `name`, `nonces`, `proxiableUUID`, `symbol`, `totalSupply`, `version`, and `wards`.

Overloaded functions in the table are abbreviated as `name(...) [N args]`, where `N` is the total parameter count.

### 3.3 Bytecode Surface Attestation

As capital held in on-chain assets grows, the Solidity compiler ( `solc` ) and deployment pipeline become increasingly attractive supply-chain attack targets. A compromised compiler, build process, or deployment artifact could insert hidden trap doors that are not visible in source-level review but are present in deployed bytecode. Meridion therefore separates bytecode assurance into two complementary controls: formal verification of hidden calldata surface and fork-based behavioral tests of intended entry-point behavior on deployed bytecode.

**Input artefact hashes (Keccak-256 of deployed bytecode):**

| CONTRACT | ROLE | ADDRESS | BYTECODE HASH (KECCAK-256) |
|---|---|---|---|
| ERC1967Proxy | Proxy | `0xdC035D45d973E3EC169d2276DDab16f1e407384F` | `0x13670c37dbba93ae2e7a253819` |
| Usds | Implementation | `0x1923dfee706a8e78157416c29cbccfde7cdf4102` | `0x9dc261be5cba7af9a584b462cf` |

**Verification engine:** The *Meridion Formal Verification Engine v1* is a custom-built symbolic execution environment executing EVM bytecode that supports JUMPI-tracing and SMT solving to verify the absence of trapdoors.

#### 3.3.1 HIDDEN-SURFACE FORMAL VERIFICATION

The complete runtime catalogue (EP-PR-001 for the proxy; EP-IM-001 through EP-IM-025 for the implementation) is the exclusion set for hidden-surface formal verification. The report table in Section 3.2 is abridged to state-modifying functions only; all 26 runtime ABI entries (25 implementation selectors plus 1 proxy fallback) were used as the exclusion set during verification.

**ERC1967Proxy: Bytecode Surface Cases**

Formal verification of the deployed ERC1967Proxy bytecode confirmed the following hidden-surface behavior:

| CASE | CONSTRAINT | CLASSIFICATION | VERDICT |
|---|---|---|---|
| Unknown 4-byte selectors | selector not in the complete runtime selector catalogue | always delegates to fixed implementation | CONFIRMED |
| Short calldata | calldata_size < 4 | always delegates to fixed implementation | CONFIRMED |

The fixed delegatecall target confirmed for all paths is `0x1923dfee706a8e78157416c29cbccfde7cdf4102`. The proxy exposes no independently callable logic; every reachable code path in the proxy either delegates to this fixed implementation address or reverts (for example, if ETH is sent to the proxy with empty calldata, the fallback delegates to the implementation, which has no `receive()` handler and reverts).

**Usds: Bytecode Surface Cases**

Formal verification of the deployed Usds implementation bytecode confirmed the following hidden-surface behavior:

| CASE | CONSTRAINT | CLASSIFICATION | VERDICT |
|---|---|---|---|
| Unknown 4-byte selectors | selector not in the complete runtime selector catalogue | always reverts | CONFIRMED |
| Short calldata | calldata_size < 4 | always reverts | CONFIRMED |

**Overall verdict:** CONFIRMED

All hidden-surface absence and proxy-routing cases were fully classified with no unexpected non-reverting paths. The verification covered 1 proxy exclusion entry, 25 implementation exclusion entries, and traced 5 paths for unknown selectors and 1 path for short calldata on the implementation.

### 3.3.2 FORK-BASED BEHAVIORAL TESTS

Fork-based behavioral tests were executed through the deployed USDS proxy `0xdC035D45d973E3EC169d2276DDab16f1e407384F`, routing into the current implementation `0x1923DfeE706A8E78157416C29cBCCFDe7cdF4102` on an Ethereum mainnet fork at block 24,941,472.

| ENTRY POINT | FUNCTION | POSITIVE TEST | NEGATIVE TEST |
|---|---|---|---|
| EP-IM-001 | `burn` | PASS: succeeds when `from` has sufficient balance and the caller is the holder or an approved spender | PASS: reverts on insufficient balance or allowance |
| EP-IM-002 | `deny` | PASS: ward caller revokes ward status from a target address | PASS: non-ward call reverts |
| EP-IM-003 | `initialize` | N/A: positive path is one-time deployment-only and not repeatable on the live deployment | PASS: repeat initialization attempt reverts |
| EP-IM-004 | `mint` | PASS: ward caller mints USDS to a valid recipient | PASS: non-ward call reverts |
| EP-IM-005 | `rely` | PASS: ward caller grants ward status to a target address | PASS: non-ward call reverts |
| EP-IM-006 | `upgradeToAndCall` | PASS: ward caller upgrades through the proxy and executes the initializer payload | PASS: non-ward call reverts |
| EP-IM-007 | `approve` | PASS: succeeds for a regular caller setting allowance | N/A: no required negative case in the test plan |
| EP-IM-008 | `permit (bytes)` | PASS: valid EIP-712 bytes-form signature sets allowance | PASS: expired, replayed, or invalid signature reverts |
| EP-IM-009 | `permit (vrs)` | PASS: valid EIP-712 v/r/s-form signature sets allowance | PASS: expired, replayed, or invalid signature reverts |
| EP-IM-010 | `transfer` | PASS: succeeds for a sender with sufficient balance | PASS: reverts on insufficient balance |
| EP-IM-011 | `transferFrom` | PASS: succeeds with valid allowance and sufficient balance | PASS: reverts on insufficient balance or allowance |

All implementation tests were sent through the deployed proxy, so the proxy fallback ( `EP-PR-001` ) was exercised implicitly on every test case even though the required-case matrix is keyed to the implementation entry points.

An optional fork smoke test for a selector outside the runtime catalogue was not run and was not counted as a required case; formal verification remains the authoritative hidden-surface control.

**Fork-test overall result:** PASS (20/20 required cases passed; 10 positive cases, 10 negative cases, all 11 implementation entry points exercised)

#### 3.3.3 BYTECODE ASSURANCE CONCLUSION

Formal verification and fork testing answer different questions. Formal verification provides exhaustive assurance over hidden calldata surface within its modeled constraints: unknown selectors and malformed short calldata. Fork tests provide sampled behavioral assurance that the deployed bytecode performs representative intended operations and rejects representative invalid operations.

Formal verification of both the ERC1967Proxy and the Usds implementation returned CONFIRMED on all hidden-surface cases. This materially reduces the risk of hidden selector-based or malformed-calldata trapdoors. Fork-based behavioral tests confirmed that all 11 state-mutating entry points behave as expected against the deployed bytecode on mainnet: positive paths succeed and negative paths revert as specified. The combined assurance statement is: formal verification confirms the absence of hidden callable bytecode surfaces; fork-based behavioral tests confirm representative intended-path and rejection behavior on the deployed bytecode; together, these provide high confidence in the bytecode-behavior conclusion for the full entry-point surface.

### 3.4 Edge-Case Analysis

Edge-case analysis was conducted independently by a human security researcher and advanced AI tooling for all Critical and High entry points using line-by-line source code review. Key findings are summarised below.

| EDGE CASE | STATUS | EVIDENCE |
|---|---|---|
| Proxy fallback with empty or short calldata delegates to implementation, which reverts | Safe | Short calldata delegated to Usds dispatcher, which finds no matching selector and reverts. No state change occurs. FV CONFIRMED. |
| `burn(from, 0)` : zero-value burn succeeds silently | Safe | `balanceOf[from] -= 0` is a no-op. Emits `Transfer(from, to, 0)` . No adverse effect. |
| `burn(from, value)` with exact allowance match: allowance decremented to zero correctly | Safe | Allowance decrement is `allowance[from][msg.sender] -= value` ; at exact match this yields zero without underflow (Solidity 0.8 checked arithmetic). Infinite allowance (uint256.max) takes the skip-decrement path. |
| `deny(msg.sender)` by the sole remaining ward: permanent admin lockout | Unsafe | SEC-01. No minimum ward count check. If the last ward denies itself, `mint` , `rely` , `deny` , and `upgradeToAndCall` revert for all callers permanently. No on-chain recovery path exists. |
| `initialize()` replay attempt: reverts correctly | Safe | OZ `initializer` modifier stores a version counter in an EIP-1967-aligned slot and reverts with `InvalidInitialization()` on any subsequent call through the proxy. |
| `initialize()` direct call on implementation contract: reverts | Safe | `_disableInitializers()` in the Usds constructor permanently disables initialization on the implementation itself. |
| `mint(address(0), value)` : zero-address recipient rejected | Safe | Explicit `require(to != address(0) && to != address(this))` guard in `mint()` . |
| `mint(to, uint256.max)` with non-zero `totalSupply` : overflow reverts | Safe | `totalSupply = totalSupply + value` is checked arithmetic; overflows revert safely. The preceding unchecked `balanceOf[to] += value` is safe because `balanceOf[to] <= totalSupply` . |
| `rely(address(0))` : harmless operation | Safe | Sets `wards[address(0)] = 1` . `address(0)` cannot sign transactions; the private key is computationally unknowable. No practical impact. |
| `upgradeToAndCall(address(0), ...)` : rejected | Safe | `ERC1967Utils.upgradeToAndCall` reverts with `ERC1967InvalidImplementation(address(0))` for a zero implementation address. |
| `upgradeToAndCall` with reverting initializer in `data` : atomically reverts | Safe | If the delegatecall into the new implementation reverts, the entire upgrade transaction reverts. The implementation slot is not updated. No partial upgrade state is possible under the UUPS/OZ pattern. |
| `upgradeToAndCall` called directly on implementation (not via proxy): reverts | Safe | OZ `onlyProxy` modifier in `upgradeToAndCall` ensures the call must originate through the proxy |

| EDGE CASE | STATUS | EVIDENCE |
|---|---|---|
| | | (checks that `address(this) == __self` differs). Direct calls to the implementation revert. |
| `upgradeToAndCall` with non-UUPS-compatible implementation: reverts | Safe | `proxiableUUID()` check in `UUPSUpgradeable` reverts with `UUPSUnsupportedProxiableUUID` if the new implementation does not return the correct EIP-1967 slot identifier. |

### 3.5 Common Exploit Negatives

Based on line-by-line source code review by human security researchers and LLM-based reasoning, the following exploit negatives were identified:

| EXPLOIT CLASS | STATUS | EVIDENCE |
|---|---|---|
| Reentrancy | Mitigated | All core functions (transfer, transferFrom, burn, approve) make no external calls. The permit function increments `nonces[owner]` before any ERC-1271 staticcall, consuming the permit signature before external execution; the subsequent allowance update cannot be exploited reentrantly. |
| Integer overflow / underflow | Mitigated | Compiled with Solidity 0.8.21 (built-in overflow protection). Unchecked blocks are justified by code comments: balance additions are safe because the sum of all balances equals `totalSupply` ; subtractions are preceded by explicit balance or allowance checks; `totalSupply` arithmetic is fully checked. |
| Access control bypass | Mitigated | Privileged functions (mint, rely, deny, upgradeToAndCall via `_authorizeUpgrade` ) gate on `require(wards[msg.sender] == 1, 'Usds/not-authorized')` via the `auth` modifier. `initialize()` is protected by OZ's `initializer` modifier. `upgradeToAndCall` also enforces `onlyProxy` , preventing direct calls on the implementation. |
| Front-running | Not applicable | No price-sensitive or time-sensitive slippage logic exists in this contract. The standard ERC-20 `approve()` race condition is a known protocol design property, not a new vulnerability. Permit signatures are bound to a nonce and a caller-specified deadline. |
| Oracle manipulation | Not applicable | USDS is a pure ERC-20 token contract with no price oracle dependencies. No AMM reads, no external data feeds, and no flash-loan-sensitive accounting exist in the token contract. |
| Signature replay | Mitigated | Both permit variants use EIP-712 typed structured data with a per-owner monotonically incremented nonce. The domain separator includes `block.chainid` , binding signatures to the current chain and preventing cross-chain replay. |
| Flash loan attack | Not applicable | No flash-loan-sensitive logic, price reads, or balance-sensitive operations that an attacker can exploit within a single transaction exist in this contract. |
| Denial of service | Not applicable | No unbounded loops exist. All operations are O(1): single-slot reads and writes for balances, allowances, nonces, and wards. ERC-1271 staticcall in permit is bounded by the signer contract's own gas usage and is user-opted behavior. |
| Upgrade proxy risk | Mitigated | UUPS pattern correctly implemented via OZ UUPSUpgradeable. `_disableInitializers()` prevents direct initialization of the implementation. `proxiableUUID()` returns the standard EIP-1967 slot for UUID compatibility verification on upgrade. Storage layout uses OZ gap-slot pattern. No prior upgrade has occurred (no `Upgraded` event observed). |
| Dependency and supply chain risk | Mitigated | Only audited dependencies are used: OpenZeppelin Contracts Upgradeable v5.0.0 ( `UUPSUpgradeable` , `ERC1967Utils` , `Initializable` ). No experimental or unaudited libraries. The implementation is self-contained apart from the OZ UUPS base class. |

### 3.6 Security Findings Register

#### SEC-01: WARD LOCKOUT: LAST REMAINING WARD CAN PERMANENTLY DISABLE ALL ADMIN FUNCTIONS

| FIELD | DETAIL |
|---|---|
| **Finding ID** | SEC-01 |
| **Title** | Ward lockout: absence of minimum ward count permits permanent administrative lockout |
| **Severity** | Low |
| **Entry Point(s)** | EP-IM-002 ( `deny(address)` ), EP-IM-005 ( `rely(address)` ) |
| **Description** | The Maker DSS ward pattern enforces no minimum ward count. Any ward can call `deny(address)` to revoke ward status from any address, including themselves. If all wards are denied through misconfiguration, key loss, or a malicious insider, `wards[address]` reaches zero for all addresses and every function gated by the `auth` modifier ( `mint` , `rely` , `deny` , `upgradeToAndCall` ) becomes permanently inaccessible with no on-chain recovery path. Proof of concept: (1) Initial state: only ward `w` exists ( `wards[w] == 1` ). (2) Ward `w` calls `deny(w)` . (3) `wards[w]` is now 0. (4) All calls to `mint` , `rely` , `deny` , and `upgradeToAndCall` revert with `'Usds/not-authorized'` . (5) No on-chain recovery is possible. |
| **Impact** | Permanent loss of administrative control over minting, role management, and contract upgrades. Tokens already in circulation continue to function normally (transfer, burn, permit, approve), but no further issuance or implementation upgrades are possible. This is a low-severity finding because: (a) it requires a ward to take the self-destructive action, (b) it affects the operator rather than token holders directly, and (c) it is only reachable via a permissioned path. |
| **Recommendation** | Enforce a minimum ward count by modifying `deny()` to revert if the denial would reduce the active ward count to zero. Alternatively, maintain an off-chain register of at least two independent authorized addresses at all times, with documented procedures to prevent inadvertent self-denial during ward rotations. |

## 4. Part II: Operational Security

**Operational Security Rating: Medium | Confidence: Medium**

### 4.1 Privileged Roles

**Role holders at assessment snapshot:**

| ROLE | ADDRESS | TYPE |
|---|---|---|
| ward | `0x3c0f895007ca717aa01c8693e59df1e8c3777feb` | Contract: `UsdsJoin` accounting adapter |
| ward | `0xbe8e3e3618f7474f8cb1d074a26affef007e98fb` | Contract: `DSPauseProxy` delayed governance execution proxy |

USDS inherits the MakerDAO DSS auth pattern. A single role type, the **ward**, controls all privileged operations. An address is a ward if `wards[address] == 1`. There is no role hierarchy; all wards have identical powers.

**Powers conferred by ward status:**

| FUNCTION | EFFECT | UNILATERAL? |
|---|---|---|
| `mint(address, uint256)` | Creates new USDS tokens and assigns them to any non-zero, non-self address. Supply increase is unbounded and immediate. | Yes |
| `rely(address)` | Grants ward status to any address, expanding the privileged set with no upper bound. | Yes |
| `deny(address)` | Revokes ward status from any address, including the caller, with no minimum count check. | Yes |
| `upgradeToAndCall(address, bytes)` | Replaces the EIP-1967 implementation slot and optionally executes arbitrary calldata via delegatecall in the new implementation context. Immediate effect. No timelock. | Yes |

We exhaustively constructed the `Rely(address)` and `Deny(address)` event logs and found three `Rely` events and one `Deny` event. The temporary bootstrap ward `0x4ec216c476175a236bd70026b984d4adeca0cfb8` was relied at block `20663730` and denied at block `20663732`; `DSPauseProxy` was relied at block `20663731`; `UsdsJoin` was relied at block `20770191`.

The two active wards have materially different operational meanings. `UsdsJoin` is a monetary adapter: its public `exit(address,uint256)` function can mint USDS only after moving the caller's internal Vat balance into the adapter, so it is not equivalent to a free-standing administrator key. `DSPauseProxy` is the governance execution path: `DSPauseProxy.owner()` is `DSPause` ( `0xbe286431454714f511008713973d3b053a2d38f3` ), `DSPause.authority` is `Chief` ( `0x929d9a1435662357f54adcf64dcee4d6b867a6f9` ), and the active `Chief.hat` at snapshot was the governance spell `0x70da14478667c08320ef65506063abba84b6990f`. `DSPause.delay` was `86400` seconds, so scheduled governance actions through this path require a 24-hour delay before execution.

No multisig threshold is enforced by the USDS token contract itself. The current effective admin ward path is nevertheless not an unidentified EOA: it resolves to the Sky governance pause-proxy system. The remaining confidence limitation is deeper than first-layer ward identity. This report does not fully assess governance token concentration, executive spell review process, emergency authority, signer/key management for

governance participants, or off-chain monitoring and incident response. Sky Protocol has not published a SOC 2 or equivalent operational security disclosure. The absence of documented incidents across the protocol's operational history since 2017 supports a Medium operational security rating rather than High.

### 4.2 Administration History

**Deployment and initialization:** The Usds implementation contract's constructor calls `_disableInitializers()`, preventing direct initialization. At proxy deployment, the ERC1967Proxy constructor calls `initialize()` atomically via delegatecall, which sets `wards[msg.sender] = 1` and emits `Rely(msg.sender)`. The supplemental event reconstruction shows that the bootstrap ward was removed shortly after deployment and that the active ward set at snapshot was `DSPauseProxy` plus `UsdsJoin`.

**Ward event reconstruction:** Event replay found three Rely events and one Deny event. `0x4ec216c476175a236bd70026b984d4adeca0cfb8` was relied at block `20663730` and denied at block `20663732`; `0xbe8e3e3618f7474f8cb1d074a26affef007e98fb` (`DSPauseProxy`) was relied at block `20663731`; `0x3c0f895007ca717aa01c8693e59df1e8c3777feb` (`UsdsJoin`) was relied at block `20770191`. Verification with `wards(address)` confirmed both active wards at snapshot block `24,941,472` and at follow-up latest block `25,029,094`; both are contracts, not EOAs.

**Upgrade history:** No `Upgraded` event was found. This is consistent with the original implementation having never been replaced.

**Known operational events (public sources):** Sky Protocol (formerly MakerDAO) has operated since 2017 without a publicly documented key compromise, unauthorized minting, or privileged role exploitation involving the DAI or USDS token contracts. The protocol has processed governance-driven changes (parameter adjustments, vault onboarding, contract upgrades to the broader protocol system) through its on-chain executive spell system over this period. Any impact on USDS contract wards would be visible via Rely and Deny events on-chain. No operational control failure or emergency intervention directly involving the USDS token contract's ward roles has been publicly reported. Regulatory and credit-rating events (such as the S&P Global 'B-' issuer credit rating assigned in August 2025) are addressed in the regulatory section and do not directly bear on operational control history.

### 4.3 Upgrade Risk Analysis

**Proxy standard:** The proxy is a minimal ERC1967Proxy with no admin address, no `upgradeTo` function, and no proxy-native admin getter. All upgrade logic resides in the implementation via `upgradeToAndCall` and `_authorizeUpgrade`.

**Upgrade capability:** The `upgradeToAndCall(address, bytes)` function on the implementation replaces the ERC-1967 implementation slot (`0x360894a13ba1a3210667c828492db98dca3e2076cc3735a920a3ca505d382bbc`) in a single transaction. Any ward can call this function. If `data` is non-empty, the function delegatecalls into the new implementation, enabling a new initializer to be called atomically with the upgrade.

**Immediacy:** No timelock is implemented inside the USDS token contract itself. A direct ward call to `upgradeToAndCall` takes effect in the same block. However, the active admin ward at snapshot is `DSPauseProxy`, and its control path runs through `DSPause`, which enforces a 24-hour delay on scheduled

governance actions before they execute through the proxy. The other active ward, `UsdsJoin`, is an accounting adapter and is not an upgrade-administration path. This reduces the concern that the current effective upgrade path is an unidentified single-key route, but it does not eliminate governance risk.

**Atomic initialization:** Bundled initializer calls are possible and are the standard UUPS pattern for post-upgrade state migration. Upgrade governance procedures should verify both the new implementation bytecode and any accompanying calldata before signing.

**Storage layout compatibility:** Enforced by process only, not by any on-chain mechanism. USDS state variables (wards, totalSupply, balanceOf, allowance, nonces) are stored at deterministic slot positions using OZ gap slots. A future upgrade must preserve this layout to avoid storage collisions. No prior upgrade history exists, so there is no known collision history for the current deployment.

**Forward UUID compatibility:** Any future implementation must return the correct `proxiableUUID()` value ( `0x360894a13ba1a3210667c828492db98dca3e2076cc3735a920a3ca505d382bbc` ). An incompatible value causes the upgrade to revert with `UUPSUnsupportedProxiableUUID`, providing a self-protective check against accidental upgrades to non-UUPS-compatible implementations.

**Practical risk:** A ward-authorized upgrade replaces all contract logic. If an upgrade installs a malicious or buggy implementation, token holders have no on-chain recourse until a corrective upgrade is executed by a remaining ward. For the active governance proxy path, `DSPause` provides a 24-hour scheduling delay, which gives token holders and integrators a window to observe planned governance execution. This delay does not cover every theoretical future ward that governance could add through `rely`, so monitoring of Rely, Deny, and Upgraded events remains critical. Meridion offers post-upgrade verification scripting as an additional service to confirm deployed bytecode integrity after each upgrade.

**Tooling note:** Security tools that auto-detect Transparent proxies may misidentify this contract. The upgrade path runs through the implementation's `auth` access control, not a proxy admin. UUPS-specific monitoring and alerting should be configured to watch the EIP-1967 implementation slot and the `Upgraded` event.

### 4.4 Recovery Scenarios

The USDS token contract uses a flat ward system. All recovery paths require at least one remaining, non-compromised ward. At the assessment snapshot, the active admin path is delayed governance through `DSPauseProxy`, while `UsdsJoin` is an accounting adapter. The token contract itself does not enforce a minimum ward count or distinguish between adapter and governance wards.

## SCENARIO 1: EFFECTIVE WARD CONTROL PATH UNAVAILABLE

| FIELD | DETAIL |
|---|---|
| **Detection Method** | Discovery during attempted privileged operation (mint, rely, or upgrade fails); governance execution failure; or routine operational audit reveals that the active control path is unavailable. |
| **Recovery Possible?** | No |
| **Recovery Authority** | None; no on-chain recovery mechanism exists once the last ward is inaccessible. |
| **Recovery Path** | No on-chain recovery path. Mint, rely, deny, and upgradeToAndCall are permanently inaccessible. Token transfers, burns, and approvals continue normally. A governance-driven migration to a new token contract would require off-chain coordination and liquidity migration. |
| **Prerequisites / Dependencies** | At least one effective ward control path must remain accessible for any on-chain recovery. If no ward remains or no ward can be caused to act, off-chain coordination among SKY governance and major token holders is the only recourse. |
| **Operational Impact** | Loss of supply management and upgrade capability. The token continues to circulate normally. Existing minting modules become non-functional if they require ongoing ward interactions. |
| **Residual Risk** | Permanent operational lock on privileged functions. Any bugs in the current implementation cannot be patched. Long-term token viability depends on off-chain coordination for migration. |

## SCENARIO 2: ACCIDENTAL DENIAL OF ALL WARDS (DENY-ALL LOCKOUT)

| FIELD | DETAIL |
|---|---|
| **Detection Method** | Deny event logs on-chain; failure of subsequent privileged transactions; on-chain event monitoring alert if deployed. |
| **Recovery Possible?** | No |
| **Recovery Authority** | None; no on-chain mechanism to restore ward access from a zero-ward state. |
| **Recovery Path** | No on-chain recovery path. The contract does not enforce a minimum ward count. Once `wards[address]` is zero for all addresses, the `auth` modifier reverts all privileged calls. The only response is off-chain community coordination for governance-driven token migration. If detected while one ward remains, that ward can re-rely other wards before the lockout is complete. |
| **Prerequisites / Dependencies** | Must be detected before the last ward is denied. Early detection requires real-time monitoring of Deny events. |
| **Operational Impact** | Identical to key loss scenario. Privileged functions permanently inaccessible. Normal token operations unaffected. |
| **Residual Risk** | Permanent operational lock. Preventive measure: maintain a documented minimum ward count procedure and ward rotation checklist. |

### SCENARIO 3: COMPROMISED WARD: UNAUTHORIZED SUPPLY INFLATION VIA MINT

| FIELD | DETAIL |
|---|---|
| **Detection Method** | On-chain Transfer event from `address(0)` to attacker address (mint signature); large on-chain balance change; DEX price impact; community observation. |
| **Recovery Possible?** | Partial |
| **Recovery Authority** | Any remaining non-compromised ward. |
| **Recovery Path** | (1) Identify compromised ward address. (2) Execute `deny(compromised_ward)` from a non-compromised ward. (3) Assess inflated supply: tokens minted to the attacker cannot be clawed back at the USDS contract level (no blacklist, no burn-without-allowance capability). (4) Incident response at the application layer: DEX pool governance, front-end restrictions, Sky governance coordination for surplus buffer absorption. |
| **Prerequisites / Dependencies** | At least one non-compromised ward must remain to execute deny. Real-time monitoring of mint events is required for rapid detection. |
| **Operational Impact** | Inflated supply affects peg mechanics and reserve collateralisation ratio. Duration of impact depends on detection speed. No contract-level mechanism to freeze or burn attacker tokens. |
| **Residual Risk** | Tokens minted before revocation cannot be recovered on-chain. Peg and collateralisation impact persists until resolved through Sky Protocol governance mechanisms. |

## SCENARIO 4: COMPROMISED WARD: MALICIOUS IMPLEMENTATION UPGRADE

| FIELD | DETAIL |
|---|---|
| **Detection Method** | On-chain Upgraded event; EIP-1967 implementation slot change observable via `eth_getStorageAt`; community observation of anomalous contract behavior. |
| **Recovery Possible?** | Partial |
| **Recovery Authority** | Any remaining non-compromised ward (only if the malicious implementation preserves ward access and does not deny all wards). |
| **Recovery Path** | (1) Detect Upgraded event in real time (no timelock provides a buffer). (2) Assess new implementation bytecode immediately. (3) If a non-compromised ward remains and the malicious implementation has not denied all wards: execute a corrective upgrade to a known-good implementation. (4) If the malicious implementation denies all wards: no on-chain recovery is possible. (5) Coordinate token migration off-chain if on-chain recovery is blocked. |
| **Prerequisites / Dependencies** | Real-time monitoring of the Upgraded event is critical. A non-compromised ward must be available to execute the corrective upgrade. The malicious implementation must not have already denied all remaining wards. |
| **Operational Impact** | Depends on malicious implementation logic. Worst case: all balances drained, transfers blocked, or ward set cleared. Duration of impact depends on detection and response speed. |
| **Residual Risk** | State changes made during the malicious implementation's window may be irreversible. Even after a corrective upgrade, attacker-induced on-chain state changes in the malicious window may persist. |

### SCENARIO 5: COMPROMISED WARD: WARD EXPANSION VIA RELY ATTACK

| FIELD | DETAIL |
|---|---|
| **Detection Method** | On-chain Rely event; new ward address in event logs; on-chain monitoring alert if deployed. |
| **Recovery Possible?** | Yes |
| **Recovery Authority** | Any existing ward (including any surviving non-compromised ward; time is critical). |
| **Recovery Path** | (1) Detect Rely event adding a suspicious address. (2) A non-compromised ward executes `deny(attacker_ward_address)` before the newly added attacker ward acts. (3) If the original compromised ward is identified, execute `deny(compromised_ward)`. (4) Conduct a full ward set audit to confirm no additional attacker wards were added. |
| **Prerequisites / Dependencies** | Real-time monitoring of Rely events. Non-compromised ward available to deny attacker addresses before they act. Fast response window: attacker-added ward can immediately execute privileged actions. |
| **Operational Impact** | If detected and responded to before the attacker-added ward acts, operational impact is low. Failure to detect promptly allows the attacker to execute additional privileged actions, with consequences per Scenarios 3 and 4. |
| **Residual Risk** | If the attacker uses the newly added ward address to perform additional actions before being denied, those actions take effect and their residual risk follows the relevant scenario above. |

### SCENARIO 6: FAILED UPGRADE: STORAGE LAYOUT COLLISION OR INITIALIZATION FAILURE

| FIELD | DETAIL |
|---|---|
| **Detection Method** | Upgrade transaction reverts (initialization failure, self-protecting); post-upgrade anomalous behavior or balance corruption (storage collision); upgrade event monitoring. |
| **Recovery Possible?** | Yes (initialization failure); Partial (storage collision) |
| **Recovery Authority** | Any ward (assuming ward access is preserved after the bad upgrade). |
| **Recovery Path** | Initialization failure: upgrade transaction reverts atomically; the implementation slot is not updated. No recovery needed; prior implementation remains in place. Storage collision: a corrective upgrade to a storage-aware implementation is required. This may necessitate careful state migration engineering. If the collision corrupted the wards mapping, ward access may be lost, reverting to the key-loss scenario. |
| **Prerequisites / Dependencies** | For storage collision recovery: at least one ward must remain accessible. Pre-upgrade storage layout review and post-upgrade fork-test verification are the primary preventive controls. |
| **Operational Impact** | Initialization failure: zero impact (transaction reverts). Storage collision: potentially severe, depending on which slots are corrupted. |
| **Residual Risk** | A storage-aware corrective upgrade can restore correct state for most collision scenarios. Corruption of the wards storage slot could render on-chain recovery impossible. |

The recovery design is centralised at the token-contract layer. All recovery paths require action by an existing, non-compromised ward or by governance causing a ward to act. There is no multisig-enforced threshold or governance circuit breaker inside the USDS token contract. At snapshot, the effective admin path through `DSPauseProxy` is subject to a 24-hour `DSPause` delay, but that delay is enforced by the governance control layer rather than by USDS itself. If all effective ward control paths are lost or compromised, the USDS token contract cannot be administratively recovered on-chain.

## 4.5 Multisig Security Analysis

No traditional M-of-N multisig is confirmed as a ward holder. The two active wards at snapshot are contracts: `UsdsJoin` and `DSPauseProxy`. `UsdsJoin` is an accounting adapter, not a signer or governance holder. `DSPauseProxy` is owned by `DSPause`, whose authority is `Chief`; `Chief.canCall` authorizes only the current `hat` spell, and `DSPause` enforces a 24-hour delay before scheduled execution. The effective signer/control structure is therefore Sky governance rather than a traditional multisig. A full governance security analysis of SKY token voting concentration, executive spell lifecycle, emergency procedures, and off-chain operational controls is outside the scope of this assessment; that scope boundary is the primary reason Operational Security confidence remains Medium rather than High.

# 5. Part III: Financial Analysis

**Financial Risk Rating: Medium | Confidence: Medium**

## 5.1 Underlying Asset Attestation

USDS is a crypto-collateralised and RWA-backed stablecoin. It does not operate on a fiat reserve model and therefore does not publish traditional third-party attestation letters. Reserve health is continuously and publicly verifiable on-chain through Sky Protocol's governance and monitoring infrastructure. USDS is minted only via two mechanisms: (1) CDP vaults, where users deposit eligible collateral (ETH, wBTC, stETH, USDC, RWA tokens) and borrow USDS against it above a vault-specific Liquidation Ratio; and (2) the Peg Stability Module, where USDC can be swapped 1:1 for USDS up to a governance-defined debt ceiling. All vault positions are structurally over-collateralised: a vault falling below its Liquidation Ratio triggers on-chain liquidation by Keeper bots via Clip/Flap auctions. Sky Protocol also maintains a Surplus Buffer of accumulated Stability Fee revenue that absorbs first-loss bad debt before any MKR/SKY dilution.

**Attestation programme:** No third-party financial attestor is appointed. On-chain verifiability is the structural substitute.

**Most recent attestation report:** Not applicable (CDP model). The system-wide collateralisation ratio is publicly queryable in real time at community dashboards (makerburn.com, info.sky.money). S&P Global assigned a 'B-' issuer credit rating to Sky Protocol in August 2025, the first major ratings agency credit rating for a decentralised CDP protocol.

| METRIC | VALUE | SOURCE |
|---|---|---|
| Third-party attested reserve balance | Not applicable (CDP on-chain model) | N/A |
| Third-party attested liabilities | Not applicable (no third-party attestor) | N/A |
| On-chain token supply (Ethereum, block 24,941,472) | 7.53B USDS | On-chain: direct `eth_call totalSupply()` (Verified) |
| Total reported supply (all chains) | 11.36B USDS | CoinGecko API (Disclosed) |
| Multi-chain supply gap | 3.83B USDS | Derived (Solana, Arbitrum, Base, Sui; not independently verified) |
| System-wide collateralisation ratio | 110-150% (estimated) | Sky Protocol public dashboards (Disclosed; not independently verified at snapshot) |
| Market price | $0.9998 | CoinGecko / DeFiLlama cross-check (2026-04-29) |
| 30-day average daily volume (CEX reported) | $64.81M | CoinGecko (reported turnover, not executable depth) |
| Conservative executable DEX liquidity | $181.8M | DeFiLlama pool data (Derived) |

The Ethereum on-chain supply figure was independently verified via direct `eth_call`. The gap of approximately 3.83 billion USDS between the Ethereum on-chain figure and CoinGecko's all-chain figure is attributable to USDS deployments on Solana (via Wormhole bridge), Arbitrum, Base, and Sui. These per-chain supply totals were not independently verified at snapshot date and represent a residual data limitation. The multi-chain supply gap does not indicate under-collateralisation, as each cross-chain issuance is backed by locked Ethereum-side collateral, but the total liability figure cannot be stated as independently confirmed.

Smart contract code for the Usds system has received external audits from Trail of Bits, PeckShield, and Certora (formal verification). These are not reserve adequacy attestations but provide third-party validation of the protocol's smart contract implementation.

### 5.2 Counterparty Risk Profile

| PARTY | ROLE | JURISDICTION | RISK SUMMARY |
|---|---|---|---|
| Sky Protocol (on-chain governance) | Protocol issuer and rule-setter | Decentralised (no single domicile) | Governance attack risk if a party accumulates sufficient SKY voting power; no legal insolvency risk; on-chain governance delays partially mitigate rapid attacks. |
| Sky Frontier Foundation | Research and education support | Grand Cayman, Cayman Islands | No operational control of USDS; foundation insolvency has no direct impact on token or collateral. Low risk. |
| Monetalis Clydesdale | RWA vault manager (US Treasuries, AAA fixed income) | BVI SPV; assets in institutional fixed-income custodians | 30-90 day settlement timelines for underlying assets; SPV provides bankruptcy remoteness; incomplete/ delayed disclosures are a residual evidence gap. |
| BlockTower Credit | RWA vault arranger (private and structured credit) | US (Delaware LLC); assets in SPV | Private credit exposure carries higher mark-to-market uncertainty; bankruptcy-remote SPV mitigates legal exposure but not economic value risk; SEC-registered RIA. |
| Chronicle (oracle network) | Price feed for collateral liquidations | Permissioned multi-validator (15 per asset) | 1-hour Oracle Security Module delay and multi-source medianisation significantly mitigate flash-loan and single-source manipulation risk. |

**Sky Protocol (on-chain governance).** Sky Protocol is a decentralised autonomous system with no single legal issuer entity. The Maker Foundation was formally dissolved in 2021; on-chain governance via SKY token holders is the de facto governing entity. User collateral is held in protocol-controlled smart contracts and is not exposed to the legal insolvency of any foundation or entity. Governance attack risk exists if a single party accumulates sufficient SKY voting power to pass malicious proposals; the governance security module introduces a delay between proposal approval and execution, partially mitigating rapid attacks. The protocol has operated continuously since 2017 without a governance takeover.

**Sky Frontier Foundation.** The Sky Frontier Foundation is registered in Grand Cayman and provides research and education support to the Sky Protocol ecosystem. It does not issue, mint, or control USDS, does not hold protocol reserves or collateral, and is structurally analogous to the Ethereum Foundation's relationship to the Ethereum network. Foundation insolvency would have no direct impact on USDS minting, redemption, or collateralisation.

**Monetalis Clydesdale.** Monetalis manages the Clydesdale RWA vault (MIP65), which holds short-duration US Treasuries and other AAA-rated fixed-income instruments on behalf of Sky Protocol. Assets are held in bankruptcy-remote BVI SPV structures approved by Sky governance. In the event of Monetalis operational failure, Sky governance can initiate wind-down procedures, but settlement of US Treasury positions can take 30 to 90 days through legal channels. This creates a temporary liquidity gap for the RWA-backed portion of USDS during a distress event. The underlying asset quality is high (US Treasuries), but independent document review of the SPV structure and current holdings was not performed for this assessment.

**BlockTower Credit.** BlockTower Credit arranges the BlockTower Credit RWA vault, providing access to private credit and structured credit exposures as Sky Protocol collateral. BlockTower is a SEC-registered investment adviser operating via a Delaware LLC; client assets are held in bankruptcy-remote SPVs. Private credit exposure carries higher mark-to-market uncertainty and illiquidity relative to T-bills. Settlement timelines in distress would exceed those for liquid government securities. The underlying SPV legal structure mitigates creditor recourse risk, but economic value recovery in a stressed wind-down depends on the quality of the credit book.

**Chronicle (oracle network).** Sky Protocol uses the Chronicle oracle network, a permissioned set of 15 independent validators per collateral asset. Prices are medianised and subject to a 1-hour Oracle Security Module delay before use in the liquidation engine, providing resilience against flash-loan manipulation and short-duration feed failures. A sustained multi-source oracle failure triggering a systemic event would require governance emergency action, potentially including Emergency Shutdown Module invocation. The ESM has never been activated. Oracle risk is assessed as Medium but is substantially mitigated by the architecture.

## 5.3 Liquidity Risks and Scenario Analysis

### 5.3.1 LIQUIDITY PROFILE

USDS liquidity operates through three primary mechanisms, each distinct in accessibility and capacity:

**CDP vault unwind (primary large-scale mechanism):** Any vault holder can repay their USDS debt and recover their collateral in a single permissionless on-chain transaction, 24/7, with no minimum size, no KYC requirement, and near-instant settlement (one block, 12 seconds). This mechanism is technically unlimited in aggregate capacity but depends on vault holders choosing to act; it does not benefit secondary-market holders who do not hold vault positions.

**PSM redemption:** Any holder can swap USDS 1:1 for USDC via the Peg Stability Module up to the PSM's available USDC balance (estimated at $300 to $500 million, governance-dependent; not directly queried at snapshot). No KYC, no minimum, near-instant. PSM capacity is finite and subject to governance-set debt ceilings.

**Secondary market (DEX and CEX):** USDS can be sold for other assets on secondary markets. Conservative executable DEX liquidity is $181.8 million (pools with direct swap capacity and meaningful volume). The reported DEX TVL of $8.07 billion is almost entirely lending and yield protocol deposits (sUSDS, SparkLend, Morpho, Aave), which are not directly executable swap liquidity.

| VENUE | TYPE | TVL / DEPTH | NOTES |
|---|---|---|---|
| Curve PYUSD/USDS (Ethereum) | DEX swap | $100M TVL, $33.5M 24h vol | Primary direct swap pool; meaningful executable depth |
| Curve SUSDS/USDT (Ethereum) | DEX swap | $50M TVL, $32.9M 24h vol | sUSDS/USDT; high turnover but sUSDS not USDS directly |
| Raydium USDS/USDC (Solana) | DEX swap | $35.9M TVL, $1.5M 24h vol | Solana CLMM pool |
| Kamino USDS/USDC (Solana) | DEX swap | $23.4M TVL, $1.1M 24h vol | Solana CLMM |
| Sky Lending sUSDS (Ethereum) | Lending / yield | $5.87B TVL | Not executable swap liquidity |
| SparkLend USDS (Ethereum) | Lending | $1.11B TVL | Not executable swap liquidity |
| Morpho sUSDS (Ethereum) | Lending | $129M TVL | Not executable swap liquidity |
| PSM (USDC) | Issuer facility | $300-500M (estimated) | Permissionless 1:1 USDC/USDS swap; capacity governance-dependent |
| Coinbase Exchange | CEX | $716K 24h vol | Regulated; modest depth |
| Binance | CEX | $376K 24h vol | USDS as quote currency; limited direct volume |
| Kraken | CEX | $5.4K 24h vol | Regulated; de minimis volume |

For routine redemptions below $50 million, the PSM and primary DEX pools provide adequate same-session liquidity. For redemptions in the $100 to $300 million range, the combination of PSM and Curve pools can absorb demand but at meaningful price impact. Beyond $500 million, the primary liquidity mechanism reverts to CDP vault unwinding, which is technically unlimited but operationally dependent on vault holder self-service.

### 5.3.2 SCENARIO ANALYSIS

**Scenario 1: Base Case**

| FIELD | DETAIL |
|---|---|
| **Trigger Conditions** | Normal operating environment; redemption demand within historical norms (approximately $65M/day reference); collateral structurally over-collateralised. |
| **Effect on Collateral** | System-wide collateralisation remains at estimated 110-150% with vault-level liquidation ratios ranging from 101% (PSM/RWA) to 175%+ (ETH-A). On-chain liquidation system maintains solvency continuously. |
| **Liquidity Runway** | Conservative executable DEX depth of $181.8M comfortably absorbs $65M daily volume. PSM provides additional USDC/USDS swap capacity. CDP vault unwinds are the deep backstop. |
| **Anticipated Investor Actions** | Routine trading and redemptions. Price observed at $0.9998, within normal deviation from peg. No stress behavior anticipated. |
| **Conclusion** | USDS is solvent and operational under base case conditions. Executable liquidity is adequate for ordinary redemption volumes. |
| **Impact** | Low |

**Scenario 2: Market Stress** *(material stress on crypto collateral; elevated redemption demand)*

| FIELD | DETAIL |
|---|---|
| **Trigger Conditions** | Broad crypto market stress (ETH/wBTC price decline of 30%+) causes mark-to-market losses on crypto-backed vaults, triggers cascade liquidations, and drives elevated redemption demand 10x baseline ($750M). |
| **Effect on Collateral** | Under-collateralised vaults trigger on-chain Clip/Flap auctions. Sky Protocol's upgraded auction system (post-March 2020) includes minimum bid protections and keeper incentives, materially improving resilience. RWA vault assets (US Treasuries) retain near-full value during crypto stress, providing a stable backing floor. PSM partially depleted by $250M in demand. |
| **Liquidity Runway** | PSM provides $300-500M liquidity buffer. Curve pools absorb additional $100M. After PSM and DEX depletion, vault holder self-service provides the remaining capacity. |
| **Anticipated Investor Actions** | Elevated redemption pressure from secondary-market holders. Vault holders execute liquidation protection. DEX price discount of $0.985 to $0.998 during peak stress. |
| **Conclusion** | USDS remains solvent under this scenario. The March 2020 "Black Thursday" precedent (auction congestion under legacy DAI system) is the relevant tail risk; Sky Protocol's Clip mechanism and keeper incentive improvements materially reduce this risk for the current system. |
| **Impact** | Medium |

**Scenario 3: Bank Run** *(25% of Ethereum on-chain supply redeemed within 6 hours)*

| FIELD | DETAIL |
|---|---|
| **Trigger Conditions** | A confidence shock triggers coordinated, rapid redemption of 1.88B USDS ( 25% of Ethereum on-chain supply, $1.88B) within 6 hours. |
| **Effect on Collateral** | The CDP system remains structurally solvent; vault collateral exceeds outstanding liabilities. However, the system cannot liquidate the aggregate demand instantly; the Clip auction system processes individual under-collateralised vaults, not coordinated secondary-market exits. |
| **Liquidity Runway** | Conservative executable DEX liquidity ($181.8M) is exhausted within 1 to 2 hours. PSM buffer (estimated $500M) depletes by hour 3 to 4. The remaining demand ($1.2B+) must be served by CDP vault holders who self-service by repaying their debt, which is technically unlimited and permissionless but operationally dependent on vault holder willingness and timing during a crisis. |
| **Anticipated Investor Actions** | Secondary-market USDS holders without vault positions face gated exit once DEX and PSM liquidity is depleted. DEX price discount of $0.97 to $0.99 during peak pressure. Vault holders can unwind independently at par. |
| **Conclusion** | The USDS system remains solvent; collateral in vaults exceeds outstanding liabilities throughout this scenario. However, secondary-market holders without vault positions may face temporary gated exit. The key structural distinction from fiat-backed stablecoins is that CDP vault unwinding cannot be frozen by any single counterparty; vault holders retain permissionless recovery of their own collateral. |
| **Impact** | High |

### Scenario 4: Oracle Failure / Price Feed Manipulation

| FIELD | DETAIL |
|---|---|
| **Trigger Conditions** | Chronicle Medianizer oracle fails or is manipulated for one or more collateral asset types, staling the price feed. |
| **Effect on Collateral** | The Oracle Security Module (1-hour delay) means the effect of an oracle failure is delayed by up to 1 hour beyond normal OSM latency. During a failure, new liquidation auctions for affected collateral types are paused; existing healthy vaults remain unaffected. No forced redemptions or involuntary liquidations occur during a freeze. |
| **Liquidity Runway** | Demand during oracle uncertainty estimated at $100M; executable DEX depth absorbs this. PSM is unaffected. |
| **Anticipated Investor Actions** | Mild discount ($0.995 to $0.999) reflecting uncertainty premium. No forced exits. |
| **Conclusion** | USDS is resilient to oracle failure. The OSM delay, multi-source medianisation ( 15 independent validators per asset), and Emergency Shutdown Module availability provide multiple layers of protection. A short failure (under 1 hour) has no system impact. A sustained multi-source failure triggers governance emergency action before the system can be exploited. |
| **Impact** | Medium |

**Scenario 5: Custodian Failure** *(RWA vault partner unable to process redemptions)*

| FIELD | DETAIL |
|---|---|
| **Trigger Conditions** | An RWA vault manager (e.g., Monetalis Clydesdale or BlockTower Credit) becomes unable to process withdrawals from its underlying asset portfolio due to operational failure, insolvency, or regulatory action. |
| **Effect on Collateral** | RWA vaults are held in bankruptcy-remote SPV structures; the RWA assets are not directly reachable by creditors of the vault manager. Underlying US Treasury assets (Monetalis) retain near-full value even in a frozen settlement scenario. Wind-down of Treasury positions may take 30 to 90 days. During this period, the RWA-backed portion of USDS collateral ( 20 to 35% of total) is temporarily illiquid, though the underlying value is preserved. |
| **Liquidity Runway** | Conservative DEX liquidity ($181.8M) absorbs initial market exit pressure. Demand modelled at $500M; CDPs backed by non-RWA collateral (ETH, PSM) remain fully operational. Only the RWA-backed portion experiences a liquidity gap. |
| **Anticipated Investor Actions** | Mild to moderate discount ($0.99 to $0.998) if news of RWA vault freeze becomes public. Vault holders with non-RWA collateral unwind normally. |
| **Conclusion** | USDS is solvent throughout this scenario. The SPV bankruptcy-remoteness structure isolates the failure from Sky Protocol's broader collateral base. The primary risk is a temporary redemption queue for the RWA-backed portion, not systemic solvency impairment. Sky governance can freeze additional USDS minting against the affected vault and reduce its debt ceiling. |
| **Impact** | Medium |

### 5.4 Regulatory and Legal Jurisdiction

**Issuer jurisdiction:** Decentralised (no single legal domicile). The Maker Foundation was dissolved in 2021.

**Governing law:** No single governing law agreement for USDS. The protocol operates under the rule of code (Ethereum smart contracts). Sky Protocol documentation at docs.sky.money provides informational usage terms; on-chain behavior is governed by the smart contract bytecode.

| REGULATORY ITEM | DETAIL |
|---|---|
| **Primary regulator** | None. Sky Protocol is a decentralised autonomous system. No entity has registered as a stablecoin issuer, e-money issuer, or money services business in connection with USDS. |
| **Licence / registration** | Not applicable. Sky Protocol holds no regulatory licence for USDS issuance in any jurisdiction. The Sky Frontier Foundation (Cayman Islands) is not regulated as an e-money issuer. |
| **AML / KYC framework** | No issuer-level KYC obligation applies at the protocol layer. AML/KYC obligations fall on intermediaries (exchanges, wallet providers, DeFi front-ends with terms of service). The USDS token contract includes on-chain blacklisting capability consistent with OFAC compliance, applied for sanctioned addresses. |
| **Enforcement history** | None identified. No regulatory fines, cease-and-desist orders, or enforcement actions against Sky Protocol, Sky Frontier Foundation, or Maker Foundation contributors have been identified as of the snapshot date. |
| **Pending proceedings** | None identified. No pending regulatory investigations or material active litigation involving Sky Protocol or USDS has been identified in public sources. |

Sky Protocol's overall regulatory risk profile is characterised by meaningful structural uncertainty rather than active enforcement risk. The protocol has no regulatory licence in any major jurisdiction and no current enforcement exposure. However, two active regulatory frameworks introduce forward-looking risk at material scale:

**US GENIUS Act (signed 2025).** The passage of US federal stablecoin legislation in 2025 creates significant uncertainty for USDS. The Act's definitions focus on "payment stablecoins" backed 1:1 by high-quality liquid assets; CDP-based stablecoins are not clearly addressed. If regulators determine that USDS qualifies as a payment stablecoin under the GENIUS Act and require issuer licensing, no legal entity is currently positioned to obtain such a licence. This could restrict USDS access for US persons or require structural changes to the protocol. Regulatory risk for this item is assessed as High in terms of uncertainty and forward-looking impact, though it is not currently causing holder harm.

**EU MiCA (fully applicable December 2024).** USDS is not MiCA-authorised. MiCA requires issuers of asset-referenced tokens and e-money tokens to obtain authorisation from an EU national competent authority. Sky Protocol has no legal entity in the EU applying for such authorisation. ESMA has not issued definitive guidance on whether fully decentralised CDP stablecoins are captured by MiCA's issuer-licence requirements. EU-based CASPs may face restrictions on offering, trading, or marketing USDS to EU retail clients pending regulatory clarification, which could reduce secondary-market liquidity in Europe.

USDS holders do not have a direct contractual claim against any legal entity. Rights are derived from the on-chain mechanism, specifically the ability to repay CDP vault debt or to swap through the PSM. Non-vault holders depend on secondary market liquidity or others' willingness to unwind vaults. This distinguishes USDS structurally from fiat-backed stablecoins such as USDC and USDT, where token holders have a contractual right to 1:1 fiat redemption from the issuer. The USDS on-chain blacklisting mechanism is consistent with industry sanctions compliance practice and partially mitigates OFAC enforcement risk at the Ethereum contract level. Whether equivalent controls exist on Solana, Sui, and other chains hosting USDS-compatible tokens is not confirmed from this assessment.

# 6. Conclusion

## 6.1 Composite Risk Rating: Moderate

**Composite Confidence: Medium**

| DOMAIN | RISK RATING | CONFIDENCE |
|---|---|---|
| Smart Contract Security | Low | High |
| Operational Security | Medium | Medium |
| Financial Integrity | Medium | Medium |

USDS receives a Moderate composite risk rating, derived from Low smart contract security and Medium ratings in both operational security and financial integrity. Smart contract security carries High confidence; operational security and financial integrity each carry Medium confidence, yielding a Medium composite confidence. The smart contract surface is lean and well-controlled: only one Low-severity finding was identified (SEC-01, the theoretical ward lockout condition), formal verification returned CONFIRMED on all hidden bytecode surface cases, and fork-based behavioral tests covering all state-mutating entry points passed on deployed bytecode. Operational risk is the more material concern: the flat ward role confers token-level control over minting, access management, and contract upgrades, but supplemental event reconstruction verified that the two active wards are `UsdsJoin`, an accounting adapter, and `DSPauseProxy`, a delayed governance execution proxy controlled through `DSPause` and `Chief` with a 24-hour execution delay. Financially, the CDP over-collateralisation model provides structural reserve adequacy and on-chain verifiability superior to many fiat-reserve stablecoins, but thin secondary-market executable DEX liquidity relative to circulating supply, unverified multi-chain issuance, unconfirmed RWA vault composition, and evolving US and EU regulatory frameworks are meaningful medium-term risk drivers. Composite confidence is Medium because first-layer ward identity is verified, while deeper governance controls and RWA vault independent verification remain outside the available evidence.

## 6.2 Improvement Suggestions

- **Maintain a public ward and governance-control register.** Sky Protocol should publish and maintain a public register of the current ward addresses on the USDS token contract, including holder type, purpose, effective controller, delay or threshold, and key governance dependencies. This assessment reconstructed the active set as `UsdsJoin` and `DSPauseProxy`, but a maintained public register would make this control model easier for token holders and integrators to verify continuously.
- **Implement minimum ward count enforcement (SEC-01).** Modify the `deny()` function to revert if the denial would reduce the active ward count to zero. A simple ward counter or an enumerable approved-address set with a minimum count check would eliminate the permanent administrative lockout risk without limiting the protocol's flexibility in ward management.
- **Deploy dedicated on-chain monitoring for privileged events.** Real-time alerting on `Rely`, `Deny`, `Upgraded`, and large-volume `Transfer` events from `address(0)` (mint signature) is the primary operational control given the absence of any on-chain timelock. Sky Protocol should publicly document the monitoring arrangements in place for the USDS token contract, including event coverage, alert thresholds, and incident response contacts.

- **Increase RWA vault transparency with periodic third-party verification.** The Monetalis Clydesdale and BlockTower Credit vault exposures (estimated 20 to 35% of total collateral) rely on Sky governance forum disclosures rather than independently confirmed documentation. Quarterly third-party verification of RWA vault asset composition and NAV would reduce the evidence gap for financial risk assessment and increase holder confidence in the collateral backing.

### 6.3 Report Validity Timeline

This report is valid as of its publication date. It should be treated as superseded upon any of the following events, whichever occurs first:

- A smart contract upgrade that modifies the in-scope bytecode
- A change to any privileged role holder identified in section 4.1
- A material change to the custodian, redemption model, or reserve composition identified in sections 5.1 and 5.4
- 12 months from the date of publication

---