Quantstamp

FDFI Governance Token On EVM

Executive Summary

This audit report was prepared by Quantstamp, the leader in blockchain security.

Type	ERC20
Timeline	2025-11-10 through 2025-11-12
Language	Solidity
Methods	Architecture Review, Unit Testing, Functional Testing, Computer-Aided Verification, Manual Review
Specification	None
Source Code	• https://github.com/1stdigital/fd-fdfi-token #a0426e0
Auditors	• Cameron Biniamow Auditing Engineer • Ibrahim Abouzied Auditing Engineer

Documentation quality	Medium
Test quality	Medium
Total Findings	2 Fixed: 2
High severity findings	0
Medium severity findings	1 Fixed: 1
Low severity findings	1 Fixed: 1
Undetermined severity findings	0
Informational findings	0

Summary of Findings

The FDFI governance token is an upgradeable ERC20 token that supports voting functionality, EIP-2612 gasless approvals, a fixed maximum supply, and disabled token transfers by default. Additionally, the governance token integrates with LayerZero for cross-chain bridging. Governance delegation and voting are only supported on the source chain (Ethereum). After transfers are enabled, token holders may bridge their tokens between the source chain and supported remote chains within rate-limited configurations.

During the review, the audit team discovered two vulnerabilities. The first issue (FDFI-1) pertains to the `FDFIAdapter`, which inherits from the `RateLimiter` but fails to invoke the necessary `_inflow()` and `_outflow()` functions, rendering rate limiting ineffective. The second issue (FDFI-2) involves the `FDFIOFT` contract, where burned tokens from remote chains do not decrease the `totalSupply` on the source chain. This discrepancy could lead to situations where the total supply on the source chain reflects a value greater than the actual number of tokens in circulation. In addition to these vulnerabilities, minor code improvements were suggested to enhance the code quality.

Update: The FDFI team has addressed and fixed all issues listed in this report.

ID	DESCRIPTION	SEVERITY	STATUS
FDFI-1	The `RateLimiter` Has No Effect for the `FDFIAdapter`	• Medium	Fixed
FDFI-2	Tokens May Be Capped Below the `MAX_SUPPLY`	• Low	Fixed

Assessment Breakdown

Quantstamp's objective was to evaluate the repository for security-related issues, code quality, and adherence to specification and best practices.

Disclaimer

Only features that are contained within the repositories at the commit hashes specified on the front page of the report are within the scope of the audit and fix review. All features added in future revisions of the code are excluded from consideration in this report.

Possible issues we looked for included (but are not limited to):

- Transaction-ordering dependence
- Timestamp dependence
- Mishandled exceptions and call stack limits
- Unsafe external calls
- Integer overflow / underflow
- Number rounding errors
- Reentrancy and cross-function vulnerabilities
- Denial of service / logical oversights
- Access control
- Centralization of power
- Business logic contradicting the specification
- Code clones, functionality duplication
- Gas usage
- Arbitrary token minting

Methodology

1. Code review that includes the following
   1. Review of the specifications, sources, and instructions provided to Quantstamp to make sure we understand the size, scope, and functionality of the smart contract.
   2. Manual review of code, which is the process of reading source code line-by-line in an attempt to identify potential vulnerabilities.
   3. Comparison to specification, which is the process of checking whether the code does what the specifications, sources, and instructions provided to Quantstamp describe.
2. Testing and automated analysis that includes the following:
   1. Test coverage analysis, which is the process of determining whether the test cases are actually covering the code and how much code is exercised when we run those test cases.
   2. Symbolic execution, which is analyzing a program to determine what inputs cause each part of a program to execute.
3. Best practices review, which is a review of the smart contracts to improve efficiency, effectiveness, clarity, maintainability, security, and control based on the established industry and academic practices, recommendations, and research.
4. Specific, itemized, and actionable recommendations to help you take steps to secure your smart contracts.

Scope

Files Included

Repo: `https://github.com/1stdigital/fd-fdfi-token`

Included Paths: `contracts`

Operational Considerations

1. The `FDFIToken` implements `ERC20VotesUpgradeable` while the `FDFIOFT` token does not. We assume that all votes will be done on the source chain.
2. The `FDFIToken` is initialized with transfers disabled. The contract owner can later call `enableTransfers()`. We assume that no future functionality will be added to allow disabling transfers after they've been enabled, as the contracts do not currently allow pausing transfers across all chains.
3. Both the `FDFIToken` and the `FDFIOFT` token are upgradeable. All contract upgrades should be audited to ensure that the token implementations don't create any irreversible desynchronizations.

Key Actors And Their Capabilities

`FDFIAdapter.sol`
Owner

- Update the rate limiter address.
- Upgrade the implementation contract.

Rate Limiter and Owner

- Update the rate limit configuration.

`FDFIOFT.sol`
Owner

- Update the rate limiter address.

Rate Limiter and Owner

- Update the rate limit configuration.

`FDFIToken.sol`
Owner

- Mint tokens up to the maximum supply.
- Enable token transfers.
- Upgrade the implementation contract.

Findings

FDFI-1 The `RateLimiter` Has No Effect for the `FDFIAdapter`

• Medium ⓘ Fixed

Update
The client fixed the issue in commits `284f6591bd30902002af040cfaf0ff6682aa4abc` and `07d40b2fd72bb35e6a71c8c5049d6db841931b8c` .

File(s) affected: `contracts/FDFIAdapter.sol`

Description: The `FDFIAdapter` inherits from the `RateLimiter` contract, which helps rate-limit the inflow and/or outflow of tokens between chains.

However, neither the `_inflow()` or `_outflow()` function is ever called. The `RateLimiter` has no effect on the contract.

Recommendation: Depending on the rate-limiting requirements for the source chain, do one of the following:
- Override `_debit()` to include a call to `_outflow()` .
- Remove the `RateLimiter` if its not required for the source chain.

FDFI-2 Tokens May Be Capped Below the `MAX_SUPPLY`

• Low ⓘ Fixed

Update
The client fixed the issue in commit `8d2f30f0671f669c280e69f9e9be32e3b27a0122` and provided the following explanation:

```
remove remote burnability (satellites) to keep canonical totalSupply authoritative
```

File(s) affected: `contracts/FDFIToken.sol` , `contracts/FDFIOFT.sol`

Description: The `FDFIToken.mintTo()` function enforces an upper bound of `MAX_SUPPLY` on the token's `totalSupply()` .

However, if a user burns their `FDFIOFT` token on a remote chain, the decrease in tokens is undetectable by the `FDFIToken` on the source chain. The `totalSupply()` will return a value greater than the true number of tokens in global circulation, which may prevent mints if this value is close to the `MAX_SUPPLY` .

Recommendation: Depending on the business requirements, consider removing `ERC20BurnableUpgradeable` from `FDFIOFT` and instead requiring any burns to be made on the source chain. Alternatively, consider using LayerZero to send a cross-chain message from remote chains to the source chain, indicating that tokens have been burned, and thereby reducing the total supply on the source chain.

Auditor Suggestions

S1 Code Improvements

Fixed

Update

The client fixed the suggestion in commits `8d2f30f0671f669c280e69f9e9be32e3b27a0122` and `fd0b8f753e6312c893aa2736d931c5014dfbb5b8`.

File(s) affected: `contracts/FDFIToken.sol`, `contracts/FDFIOFT.sol`

Description:

1. Remove the check `require(to != address(0))` from `FDFIToken.mintTo()` as the same check is performed in `_mint()`.
2. Remove the unused parameter `_lzEndpoint` in the `FDFIOFT.initialize()` function.

Recommendation: Consider implementing the improvements.

Definitions

- **High severity** – High-severity issues usually put a large number of users' sensitive information at risk, or are reasonably likely to lead to catastrophic impact for client's reputation or serious financial implications for client and users.
- **Medium severity** – Medium-severity issues tend to put a subset of users' sensitive information at risk, would be detrimental for the client's reputation if exploited, or are reasonably likely to lead to moderate financial impact.
- **Low severity** – The risk is relatively small and could not be exploited on a recurring basis, or is a risk that the client has indicated is low impact in view of the client's business circumstances.
- **Informational** – The issue does not pose an immediate risk, but is relevant to security best practices or Defence in Depth.
- **Undetermined** – The impact of the issue is uncertain.
- **Fixed** – Adjusted program implementation, requirements or constraints to eliminate the risk.
- **Mitigated** – Implemented actions to minimize the impact or likelihood of the risk.
- **Acknowledged** – The issue remains in the code but is a result of an intentional business or design decision. As such, it is supposed to be addressed outside the programmatic means, such as: 1) comments, documentation, README, FAQ; 2) business processes; 3) analyses showing that the issue shall have no negative consequences in practice (e.g., gas analysis, deployment settings).

Appendix

File Signatures

The following are the SHA-256 hashes of the reviewed files. A file with a different SHA-256 hash has been modified, intentionally or otherwise, after the security review. You are cautioned that a different SHA-256 hash could be (but is not necessarily) an indication of a changed condition or potential vulnerability that was not within the scope of the review.

Files

Repo: `https://github.com/1stdigital/fd-fdfi-token`

- `57e...f2c ./contracts/FDFIAdapter.sol`
- `c7a...a9b ./contracts/FDFIOFT.sol`
- `3f8...ec6 ./contracts/FDFIToken.sol`
- `232...63e ./contracts/libs/RateLimiterUpgradeable.sol`

Test Suite Results

```
$ npx hardhat test

  FDFI OFT and Adapter Basic Coverage
    FDFIOFTAdapter Basic Tests
      ✔ Should deploy adapter with correct configuration (91ms)
      ✔ Should handle rate limit configurations (56ms)
```

```
      ✔ Should allow owner to update rate limits (59ms)
      ✔ Should handle rate limiter role (71ms)
    Integration Tests
      ✔ Should allow token approvals for adapter (81ms)
      ✔ Should verify adapter token compatibility (81ms)

  FDFIToken
    ✔ initializes with zero supply and transfers disabled (114ms)
    ✔ owner can mint within cap and non-owner cannot (136ms)
    ✔ should revert when minting to zero address (110ms)
    ✔ enforces transfer gating then enables once (115ms)
    ✔ only owner can enable transfers (81ms)
    ✔ should allow transferFrom after transfers enabled (104ms)
    ✔ burn reduces supply and balance (129ms)
    ✔ permit sets allowance and prevents replay (173ms)
    ✔ delegation and vote movement after transfers (237ms)
    ✔ past votes snapshot remains accessible (115ms)
    ✔ ownership two-step transfer (100ms)
    - upgrade preserves state and restricts non-owner upgrade
    ✔ should handle multiple sequential mints correctly (96ms)
    ✔ should correctly handle approve and allowance (88ms)
    ✔ should handle burn from delegated address (110ms)
    ✔ should handle delegation changes correctly (113ms)
    ✔ should correctly track delegatee after token transfer (122ms)
    ✔ should enforce cap at exactly MAX_SUPPLY (111ms)
    ✔ should handle nonces correctly for permit (92ms)
    ✔ should allow burning tokens after enabling transfers (100ms)


  25 passing (5s)
  1 pending
```

Code Coverage

```
$ npx hardhat coverage

-----------------------------|----------|----------|----------|----------|----------------|
File                         |  % Stmts | % Branch |  % Funcs |  % Lines |Uncovered Lines |
-----------------------------|----------|----------|----------|----------|----------------|
 contracts/                  |    59.46 |    55.88 |       50 |    60.47 |                |
  FDFIAdapter.sol            |      100 |    66.67 |      100 |      100 |                |
  FDFIOFT.sol                |        0 |        0 |        0 |        0 |... ,89,100,105 |
  FDFIToken.sol              |      100 |    83.33 |    85.71 |       95 |             73 |
 contracts/libs/             |        0 |        0 |        0 |        0 |                |
  RateLimiterUpgradeable.sol |        0 |        0 |        0 |        0 |       16,24,25 |
 contracts/mocks/            |        0 |      100 |    33.33 |       20 |                |
  FDFITokenV2.sol            |        0 |      100 |        0 |        0 |          11,15 |
  MockLayerZeroEndpoint.sol  |        0 |      100 |       50 |    33.33 |          21,28 |
-----------------------------|----------|----------|----------|----------|----------------|
All files                    |       50 |       50 |    42.31 |    52.94 |                |
-----------------------------|----------|----------|----------|----------|----------------|
```

Changelog

- 2025-11-13 - Initial report
- 2025-11-21 - Final report

About Quantstamp

Quantstamp is a global leader in blockchain security. Founded in 2017, Quantstamp's mission is to securely onboard the next billion users to Web3 through its best-in-class Web3 security products and services.

Quantstamp's team consists of cybersecurity experts hailing from globally recognized organizations including Microsoft, AWS, BMW, Meta, and the Ethereum Foundation. Quantstamp engineers hold PhDs or advanced computer science degrees, with decades of combined experience in formal verification, static analysis, blockchain audits, penetration testing, and original leading-edge research.

To date, Quantstamp has performed more than 500 audits and secured over $200 billion in digital asset risk from hackers. Quantstamp has worked with a diverse range of customers, including startups, category leaders and financial institutions. Brands that Quantstamp has worked with include Ethereum 2.0, Binance, Visa, PayPal, Polygon, Avalanche, Curve, Solana, Compound, Lido, MakerDAO, Arbitrum, OpenSea and the World Economic Forum.

Quantstamp's collaborations and partnerships showcase our commitment to world-class research, development and security. We're honored to work with some of the top names in the industry and proud to secure the future of web3.

Notable Collaborations & Customers:

- Blockchains: Ethereum 2.0, Near, Flow, Avalanche, Solana, Cardano, Binance Smart Chain, Hedera Hashgraph, Tezos
- DeFi: Curve, Compound, Maker, Lido, Polygon, Arbitrum, SushiSwap
- NFT: OpenSea, Parallel, Dapper Labs, Decentraland, Sandbox, Axie Infinity, Illuvium, NBA Top Shot, Zora
- Academic institutions: National University of Singapore, MIT

Timeliness of content

The content contained in the report is current as of the date appearing on the report and is subject to change without notice, unless indicated otherwise by Quantstamp; however, Quantstamp does not guarantee or warrant the accuracy, timeliness, or completeness of any report you access using the internet or other means, and assumes no obligation to update any information following publication or other making available of the report to you by Quantstamp.

Notice of confidentiality

This report, including the content, data, and underlying methodologies, are subject to the confidentiality and feedback provisions in your agreement with Quantstamp. These materials are not to be disclosed, extracted, copied, or distributed except to the extent expressly authorized by Quantstamp.

Links to other websites

You may, through hypertext or other computer links, gain access to web sites operated by persons other than Quantstamp. Such hyperlinks are provided for your reference and convenience only, and are the exclusive responsibility of such web sites' owners. You agree that Quantstamp are not responsible for the content or operation of such web sites, and that Quantstamp shall have no liability to you or any other person or entity for the use of third-party web sites. Except as described below, a hyperlink from this web site to another web site does not imply or mean that Quantstamp endorses the content on that web site or the operator or operations of that site. You are solely responsible for determining the extent to which you may use any content at any other web sites to which you link from the report. Quantstamp assumes no responsibility for the use of third-party software on any website and shall have no liability whatsoever to any person or entity for the accuracy or completeness of any output generated by such software.

Disclaimer

The review and this report are provided on an as-is, where-is, and as-available basis. To the fullest extent permitted by law, Quantstamp disclaims all warranties, expressed implied, in connection with this report, its content, and the related services and products and your use thereof, including, without limitation, the implied warranties of merchantability, fitness for a particular purpose, and non-infringement. You agree that access and/or use of the report and other results of the review, including but not limited to any associated services, products, protocols, platforms, content, and materials, will be at your sole risk. FOR AVOIDANCE OF DOUBT, THE REPORT, ITS CONTENT, ACCESS, AND/OR USAGE THEREOF, INCLUDING ANY ASSOCIATED SERVICES OR MATERIALS, SHALL NOT BE CONSIDERED OR RELIED UPON AS ANY FORM OF FINANCIAL, INVESTMENT, TAX, LEGAL, REGULATORY, OR OTHER ADVICE. This report is based on the scope of materials and documentation provided for a limited review at the time provided. You acknowledge that Blockchain technology remains under development and is subject to unknown risks and flaws and, as such, the report may not be complete or inclusive of all vulnerabilities. The review is limited to the materials identified in the report and does not extend to the compiler layer, or any other areas beyond the programming language, or programming aspects that could present security risks. The report does not indicate the endorsement by Quantstamp of any particular project or team, nor guarantee its security, and may not be represented as such. No third party is entitled to rely on the report in any way, including for the purpose of making any decisions to buy or sell a product, service or any other asset. Quantstamp does not warrant, endorse, guarantee, or assume responsibility for any product or service advertised or offered by a third party, or any open source or third-party software, code, libraries, materials, or information to, called by, referenced by or accessible through the report, its content, or any related services and products, any hyperlinked websites, or any other websites or mobile applications, and we will not be a party to or in any way be responsible for monitoring any transaction between you and any third party. As with the purchase or use of a product or service through any medium or in any environment, you should use your best judgment and exercise caution where appropriate.

Quantstamp