TENSET

Smart Contract Audit

Spot

Table of Content

Executive Summary

SPOT is an innovative decentralized flatcoin, designed to be an inflation-resistant, stable asset capable of withstanding black swan events. SPOT serves as a refuge from inflation, a peer-to-peer digital cash system, operating effectively across all market scenarios.

SPOT's stability depends on its collateral, the Senior AMPL Tranches, derived from AMPL through a two-step process of tranching and bundling. This process helps to resegment and manage volatility, creating senior tranches as stable, debt-like instruments, and junior tranches as more volatile, equity-like instruments.

This audit evaluates SPOT's security features, potential vulnerabilities, and risk mitigation strategies, underpinning the key assumptions about the SPOT protocol's functionality, interaction with external smart contracts, and the overall trusted and secure environment in which it operates.

Project Summary

Language	Codebase	Commits
Solidity	https://github.com/ampleforth/spot	42e7a19ff969591efcef5d6822f4e34301943cdc

Audit Scope

ID	File	SHA256
PT	contracts/PerpetualTranche.sol	12763bcde6c4ee48f44049b87acccc8e95f9884f28b63e1b0cbc3e427e5b1619

General security assumptions

In conducting our security audit of the SPOT Protocol, we have made a number of important security assumptions. These assumptions are fundamental to our analysis and should be considered when evaluating the overall security and potential vulnerabilities of the system:

Security of the Underlying Blockchain
Our security analysis is predicated on the assumption that the underlying blockchain (for example, Ethereum) is secure and functions as intended. Potential vulnerabilities or flaws in the blockchain protocol are outside the scope of this audit.

Trusted Environment
We assume that users of the smart contract will interact with it in a secure and trusted environment. This encompasses the use of secure, up-to- date software and hardware, which is free from malware or any potential interference from malicious third parties.

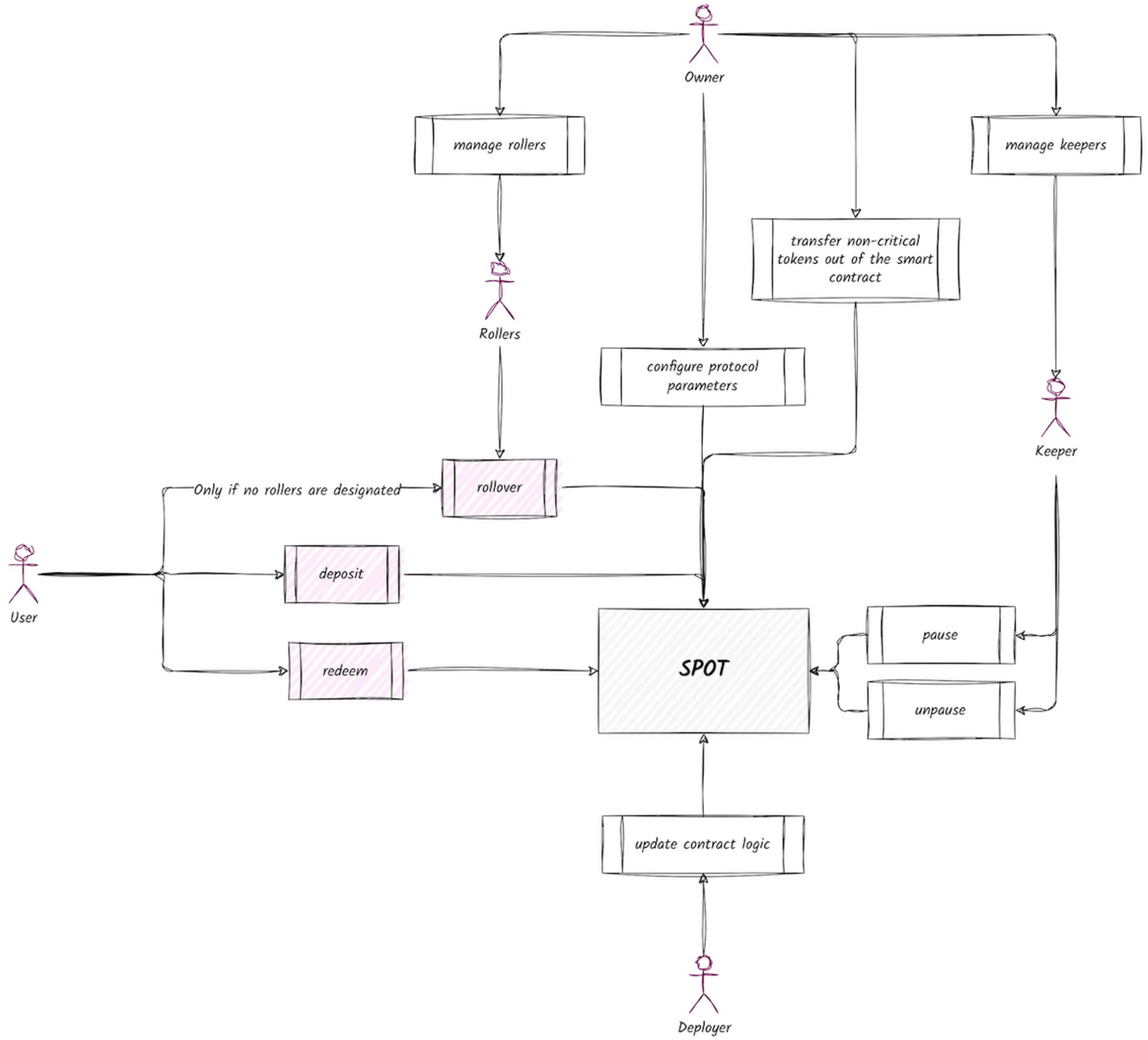
Private Key Security
Users of the smart contract are assumed to safeguard their private keys effectively. This involves not sharing private keys, securely storing key backups, and using hardware wallets or other secure means for key management.

Interactions with External Smart Contracts
The SPOT token implementation interacts with external smart contracts, which were beyond the scope of this audit. We have assumed that these external smart contracts function correctly. Any vulnerabilities or flaws in these external smart contracts could have a direct impact on the security of the SPOT token.

Dependence on Collateral Value
The value of the SPOT token is inherently dependent on the value of the collateral senior AMPL tranches. Although fluctuations in their value have historically been unlikely, any such fluctuations could directly impact the value of the SPOT token.

These security assumptions form the basis of our audit. Deviations from these assumptions could potentially lead to risks or vulnerabilities not covered in this analysis. Therefore, the effective management of these factors is critical to ensure the ongoing security of the SPOT Protocol.

Actors in the system


Owner
manage rollers
manage keepers
transfer non-critical tokens out of the smart contract
Rollers
configure protocol parameters
Keeper
Only if no rollers are designated
rollover
User
deposit
redeem
SPOT
pause
unpause
update contract logic
Deployer

In our examination of the SPOT ecosystem, we've identified several key actors, each with their distinct roles, responsibilities, and potential influence within the protocol. These actors are as follows:

User
This encompasses any individual or entity interacting with the protocol in a permissionless manner. Users drive the basic operations within the protocol.

Rollers
These are entities authorized to call the rollover function. If no rollers are designated, users are permitted to call the rollover function.

Deployer
The deployer has the power to deploy new versions of the protocol, an ability that can facilitate the introduction of new features and the fixing of bugs. However, the implications of this role are double-edged. While beneficial for system upgrades, it does require absolute trust from other actors in the system as deployers have the ability to deploy malicious updates, potentially leading to a loss of tokens for all parties involved.

Owner
The owner carries significant administrative authority within the system. They can designate rollers and keepers, transfer non-critical tokens out of the smart contract, and configure various protocol parameters. These parameters include:
- bond issue
- fee strategy
- pricing strategy
- discount strategy
- tolerable tranche maturity
- minting limits
- mature value target

Keeper
The keeper has the power to pause and resume certain features of the smart contract. These include:
- deposit
- redeem
- rollover

Thread model for each actor

User

In the process of our smart contract security audit, we did not identify any major security threats directly impacting users' interactions with the system. However, the safety of users' tokens within the SPOT ecosystem fundamentally relies on a number of crucial security assumptions, each of which carries inherent risks:

Trust in the Deployer
Users place implicit trust in the deployer of the smart contract. As outlined in the 'Actors in the system' section, if the deployer were to act maliciously or become compromised, they could potentially deploy a harmful update at any moment, which could lead to a substantial loss of funds for all users. Similarly, a well-intentioned deployer could inadvertently deploy a flawed update, which could also lead to a devastating loss of funds.

Trust in the Owner
Users are dependent on the owner to correctly configure protocol parameters and to act in good faith for the benefit of users and the wider ecosystem. If the owner were to act maliciously, become compromised, or accidentally misconfigure the protocol, users could lose partial or total access to their funds.

Trust in the Keeper
Users trust the keeper to only pause the redeem feature of a smart contract in emergency situations and for short periods. If the keeper were to pause the redeem feature permanently, users would be unable to redeem their SPOT tokens for the underlying collateral. It is important to note that for this scenario to occur, the owner, the keeper, and the deployer would need to collude.

Dependency on AMPL Token Value
The value of the SPOT token is fundamentally underpinned by the value of the AMPL token, which acts as collateral. Any significant fluctuation in the value of the AMPL token could therefore directly impact the value of SPOT tokens.

These assumptions underline the importance of careful and thorough scrutiny of the roles and behaviours of key actors within the SPOT ecosystem, alongside diligent monitoring of the AMPL token value. Recognising and addressing these inherent risks is vital to ensure the ongoing security and stability of the SPOT ecosystem.

Rollers

Rollers, within the SPOT ecosystem, are responsible for executing rollover transactions of tranche tokens. In doing so, they may either incur a fee or receive a reward, a configuration determined by the owner.

In the event that the owner, deployer, or 'feeStrategy' owner becomes compromised or acts with malicious intent, they could preemptively front-run the rollover transaction. By altering the configuration, they could set the fee to be paid by the roller at an exorbitant rate of 100%, thus resulting in substantial roller griefing.

To mitigate the risk of an unexpected hike in fees during the rollover process, rollers could implement a defensive coding strategy. They can call the rollover function from a smart contract that includes a condition to automatically revert the transaction if the received token amount does not match the expected value.

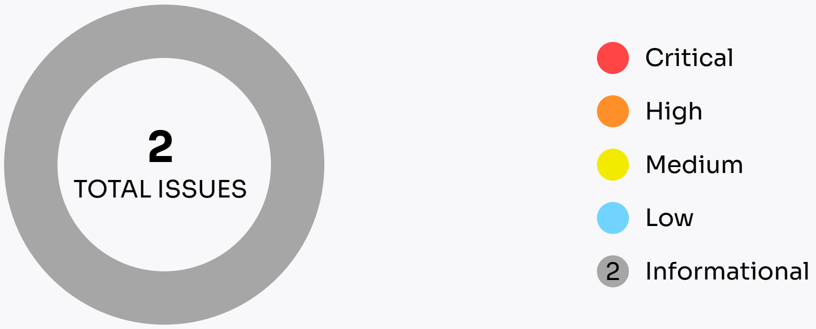
However, this mitigation strategy requires the roller role to be assigned to the address of this protective smart contract, rather than to an Externally Owned Account (EOA). In doing so, the smart contract effectively acts as a protective barrier, verifying transaction conditions and shielding the roller from potentially excessive fees.

Deployer, Owner, and Keeper

In the context of our smart contract security audit, the roles of the deployer, owner, and keeper within the SPOT ecosystem do not inherently carry any direct risk to tokens of these actors.

Findings

Audit Overview



Issues

Severity	Found	Resolved	Partially Fixed	Acknowledged
● Critical	0	0	0	0
● High	0	0	0	0
● Medium	0	0	0	0
● Low	0	0	0	0
● Informational	2	0	0	0
Total	**2**	**0**	**0**	**0**

PT-01
Pragma not locked.

● Informational   Found

Description

Using a floating pragma like '^0.8.19' introduces risks of unexpected behavior and potential vulnerabilities. This is because a newer compiler version within the declared range could introduce significant changes, bugs, or alter how certain code is executed. Hence, this could lead to unforeseen issues or vulnerabilities in the smart contract that the developers didn't account for when writing their code.

Recommendation

Lock the pragma to a specific Solidity version to ensure consistency and reliability across all deployments and environments. When choosing which Solidity version to use, carefully consider potential benefits of new features against the possible risks associated with undiscovered issues.

PT-02

Potential for Denial of Service Attack through Misconfiguration

● Informational   Found

Description

Our analysis has revealed a potential vulnerability within the redeem function of the SPOT protocol that, if the system is misconfigured, could enable a Denial of Service (DoS) attack. This would disrupt service availability, preventing all users from accessing the protocol's features.

The 'redeem' function contains the following code segment:

```
for (uint256 i = 0; i < tokensOuts.length; i++) {
       if (tokenOutAmts[i] > 0) {
           _transferOutOfReserve(msg.sender, tokensOuts[i], tokenOutAmts[i]);
      }
}
```

This iterative mechanism, given a sufficiently extended 'tokensOuts' array, could lead to a situation where the execution of the 'redeem' function becomes infeasible due to high gas costs, thereby resulting in a DoS attack.

Recommendation

Presently, a carefully crafted configuration of the SPOT protocol guards against this scenario. However, the protocol's owner must exercise extreme caution when updating this configuration to continuously prevent this potential vulnerability from becoming a real threat. It is imperative to ensure that the system's configuration always limits the size of the 'tokensOuts' array to a manageable length.

Disclaimer

The smart contracts given for audit have been analyzed by the best industry practices at the date of this report, with cybersecurity vulnerabilities and issues in smart contract source code, the details of which are disclosed in this report (Source Code); the Source Code compilation, deployment, and functionality (performing the intended functions).

The report contains no statements or warranties on the identification of all vulnerabilities and security of the code. The report covers the code submitted to and reviewed, so it may not be relevant after any modifications. Do not consider this report as a final and sufficient assessment regarding the utility and safety of the code, bug-free status, or any other contract statements.

While we have done our best in conducting the analysis and producing this report, it is important to note that you should not rely on this report only — we recommend proceeding with several independent audits and a public bug bounty program to ensure the security of smart contracts. English is the original language of the report. The Consultant is not responsible for the correctness of the translated versions.

Smart contracts are deployed and executed on a blockchain platform. The platform, its programming language, and other software related to the smart contract can have vulnerabilities that can lead to hacks. Thus, Consultant cannot guarantee the explicit security of the audited smart contracts.