verichains

DEPLOYMENT REVIEW OF XPOOL & XPOOL TOKEN

Xpool

PUBLIC REPORT

JUNE 02, 2021

Verichains Lab

info@verichains.io

https://www.verichains.io

Driving Technology > Forward

ACRONYMS AND ABBREVIATIONS

NAME	DESCRIPTION
Ethereum	An open source platform based on blockchain technology to create and distribute smart contracts and decentralized applications.
Ether (ETH)	A cryptocurrency whose blockchain is generated by the Ethereum platform. Ether is used for payment of transactions and computing services in the Ethereum network
Binance Chain	Binance Chain is a blockchain software system developed by Binance and its community.
Binance Smart Chain (BSC)	Binance Smart Chain (BSC) is a blockchain network built for running smart contract-based applications. BSC runs in parallel with Binance's native Binance Chain (BC), which allows users to get the best of both worlds: the high transaction capacity of BC and the smart contract functionality of BSC.
Binance Coin (BNB)	Binance Coin (BNB) is a cryptocurrency that can be used to trade and pay fees on the Binance cryptocurrency exchange.
Smart contract	A computer protocol intended to digitally facilitate, verify or enforce the negotiation or performance of a contract.
Xpool (XPO)	Xpool is an open-source Decentralized protocol - built on Binance Smart Chain that enables users to borrow assets from the platform, deploy your assets and earn yield with ease. XPO is the native token of Xpool.

EXECUTIVE SUMMARY

Aave is an open-source decentralized non-custodial liquidity protocol where users can participate as depositors or borrowers. Depositors provide liquidity to the market to earn a passive income, while borrowers are able to borrow in an overcollateralized (perpetually) or undercollateralized (one-block liquidity) fashion.

Xpool is a decentralized protocol built from the Aave v2 source codes, but instead of running on Ethereum (ETH), it was ported by Xpool team to run on Binance Smart Chain (BSC) to make use of many advantages of BSC such as cheap transaction fees or high-performance network.

At this point of writing, the Xpool system was deployed and tested on BSC Mainnet. The purpose of this report is to review the differences between the original Aave system (on ETH) and the new Xpool system (on BSC Mainnet), to ensure all the business logic still remains after the porting effort.

TABLE OF CONTENTS

1. MANAGEMENT SUMMARY

1.1. Reviewing scope

All Aave's smart contracts were audited by many independent auditors. The audit reports are listed at https://docs.aave.com/developers/security-and-audits#audits. Hence in this report, we do not focus on the security and logic of the smart contracts. We only reviewed the changes between Aave's smart contracts on Ethereum Mainnet and Xpool's smart contracts on Binance Smart Chain Mainnet.

The list of contracts to be reviewed was provided by the Xpool team, which categorized into four components:

COMPONENT	NETWORK	COMPARE WITH
XPO Token	BSC Mainnet	Aave Token v2
Xpool Core Protocol	BSC Mainnet	Aave Core Protocol v2
Xpool Governance	BSC Mainnet	Aave Governance v2
Xpool Staking	BSC Mainnet	Aave Staking v2

All the contract addresses are listed in the Section 2 of this document.

1.2. Reviewing methodology

During the review process, for each smart contract, we will compare it to the corresponding smart contract in the Aave system to find out if there is any difference between these two contracts. The modified contract will be checked in depth to identify any change in logic and new security problems.

In the review, the audit team used several tools for viewing, comparing and testing smart contracts, such as following:

#	NAME	NOTE
1	The Ethereum Blockchain Explorer	https://etherscan.io
2	Binance Smart Chain Explorer	https://bscscan.com
3	Visual Studio Code	https://code.visualstudio.com/, version 1.55.2
4	Remix IDE	https://remix.ethereum.org
5	Hardhat	https://hardhat.org, version 2.0.8

1.3. Disclaimer

As mentioned in the **Reviewing scope** section, we only reviewed the changes between Aave's smart contracts on Ethereum Mainnet and Xpool's smart contracts on Binance Smart Chain, corresponding changes in logic and security surface, and do not focus on the security and business logic of the original Aave contracts.

2. REVIEWED CONTRACTS

This section contains a detailed analysis of all the contracts that were reviewed by the team during the review process.

Some smart contracts (in both Aave and Xpool systems) were deployed using OpenZepplin Upgrades Plugins[1], so we reviewed both proxy and current implementation (as of report writing time) for each such contract.

For the sake of brevity, some minor changes which obviously does not affect the contract logic were not listed in this report, such as:

- Renaming, e.g., from *_aaveGovernance* to *_xpoolGovernance*.
- Changing function keywords order, e.g., from *"internal override pure"* to *"internal pure override"*.
- Changing revision number in upgradable contract.
- Changing string literals from double quote (") to single quote (') and vice versa.
- Updating/adding/removing comments.
- Code formatting.

All the detailed diff checking results are provided in attachment files along with this report.

Table 1 shows the summary review result for comparison.

#	CONTRACT	SUMMARY
1	XpoolTokenV1	Modified, does not affect the functional logic of the original AaveTokenV2 contract
2	StakedToken	Modified, changed reward claiming logic
3	AaveGovernanceV2	Not modified
4	Executor	Not modified
5	GovernanceStrategy	Not modified
6	LendingPoolAddressesProviderRegistry	Not modified
7	LendingPoolAddressesProvider	Not modified
8	LendingPool	Not modified
9	LendingPoolConfigutor	Not modified
10	AaveCollector	Not modified

[1] https://docs.openzeppelin.com/learn/upgrading-smart-contracts

11	AaveOracle	Not modified
12	LendingRateOracle	Not modified
13	AaveProtocolDataProvider	Not modified
14	LendingPoolCollateralManager	Not modified
15	AToken	Modified, does not affect the functional logic of the original contract
16	StableDebtToken	Modified, does not affect the functional logic of the original contract
17	VariableDebtToken	Modified, does not affect the functional logic of the original contract
18	InterestRateStrategy	Not modified

Table 1: Summary review result

2.1. XpoolTokenV1

INFORMATION

Name	XpoolTokenV1
Component	Xpool Token
Network	BSC Mainnet
Explorer	https://bscscan.com/address/0xeBB59CeBFb63f218db6B5094DC14AbF34d56D35D
Corresponding Aave Contract	AaveTokenV2 https://github.com/aave/aave-token-v2/blob/6ebf51ddbdfb6ae66de0b4c191b978ef5149a9ce/contracts/token/AaveTokenV2.sol https://etherscan.io/address/0xc13eac3b4f9eed480045113b7af00f7b5655ece8

REVIEW RESULT

The `XpoolTokenV1` has 4 differences from the `AaveTokenV2`.

Firstly, `XpoolTokenV1` has modified and added some constants and variables in the contract:

```
string internal constant NAME = 'Xpool';
string internal constant SYMBOL = 'XPO';
uint256 public constant REVISION = 1;

uint256 internal constant MINTED_AMOUNT = 210000000 ether;
```

The second difference is that `XpoolTokenV1` has a hard-coded owner address:

```
  // on BSC Mainnet
  address public owner = 0x0B29814cf0d28b3F89C58FBEb3667fD1FF797de8;
```

The third is that `XpoolTokenV1` has added a new *`setNewGorvernance`* function:

```
  /// @notice External function called by the Xpool governance to set or replace sources of
assets
  /// @param gov The addresses of the gorvernance
  function setNewGorvernance(ITransferHook gov)
    external
  {
    require(address(msg.sender) == owner, 'INVALID_ACTION');
    _xpoolGovernance = gov;
  }
```

Finally, the `XpoolTokenV1` has modified the *`initialize`* function. The function's code is from the first `AaveToken` contract, and this code was removed when Aave team upgraded that contract to `AaveTokenV2`.

```
  /**
   * @dev initializes the contract upon assignment to the
InitializableAdminUpgradeabilityProxy
   */
  function initialize(
    ITransferHook xpoolGovernance
  ) external initializer {
    uint256 chainId;

    //solium-disable-next-line
    assembly {
      chainId := chainid()
    }

    DOMAIN_SEPARATOR = keccak256(
      abi.encode(
        EIP712_DOMAIN,
        keccak256(bytes(NAME)),
        keccak256(EIP712_REVISION),
        chainId,
        address(this)
      )
    );
    _name = NAME;
```

```
    _symbol = SYMBOL;
    _setupDecimals(DECIMALS);
    _xpoolGovernance = xpoolGovernance;
    _mint(msg.sender, MINTED_AMOUNT);
  }
```

These modifications in `XpoolTokenV1` contract does not affect the functional logic of the original `AaveTokenV2` contract.

2.2. StakedToken

INFORMATION

Name	StakedToken
Component	Xpool Staking
Network	BSC Mainnet
Explorer	Proxy: https://bscscan.com/address/0xC4C1e78b14FC7B4dA7ae17328Bc2e83f7ee9AadF Current implementation: https://bscscan.com/address/0x1dee5031d9f745362529f23342b521b5083cf7f3
Corresponding Aave Contract	StakedTokenV2 https://github.com/aave/aave-stake-v2/blob/aa4c9cbf4d05762ed33f2d6da2e955b2d27459cb/contracts/stake/StakedTokenV2.sol https://etherscan.io/address/0x4da27a545c0c5b758a6ba100e3a049001de870f5 https://etherscan.io/address/0xe42f02713aec989132c1755117f768dbea523d2f

REVIEW RESULT

There are several modifications in the Xpool's `StakedToken` contracts, with the detailed changes are listed in Figure 1.

```
+ event NewLockReward(address indexed lockContract, address indexed receiver);
+
  /**
   * @dev Claims an `amount` of `REWARD_TOKEN` to the address `to`
   * @param to Address to stake for
   * @param amount Amount to stake
   **/
  function claimRewards(address to, uint256 amount) external override {
    uint256 newTotalRewards =
      _updateCurrentUnclaimedRewards(msg.sender, balanceOf(msg.sender), false);
    uint256 amountToClaim = (amount == type(uint256).max) ? newTotalRewards : amount;
-
+   require(amountToClaim > 0, 'ZERO_CLAIM_AMOUNT');
    stakerRewardsToClaim[msg.sender] = newTotalRewards.sub(amountToClaim, 'INVALID_AMOUNT');
-
-   REWARD_TOKEN.safeTransferFrom(REWARDS_VAULT, to, amountToClaim);
-
+   uint256 directReturn = amountToClaim / 5;
+   uint256 lockAmount = amountToClaim.sub(directReturn);
+   REWARD_TOKEN.safeTransferFrom(REWARDS_VAULT, to, directReturn);
+   if (lockAmount > 0) {
+     TokenTimeLockV2 claimContract = new TokenTimeLockV2(address(REWARD_TOKEN), to, 365 days);
+     REWARD_TOKEN.safeTransferFrom(REWARDS_VAULT, address(claimContract), lockAmount);
+     emit NewLockReward(address(claimContract), to);
+   }
    emit RewardsClaimed(msg.sender, to, amountToClaim);
  }
```

Figure 1: Changes in StakedToken contract

These modifications change the current logic of the `claimRewards` function. Instead of transfering all claimable rewards to address `` `to` ``, this function now:

1) transfers only 20% amount of claimable rewards to address `` `to` ``;
2) creates a new `TokenTimeLockV2` contract to lock the remain 80% amount of claimable rewards. The rewards in this new contract can be partially unlocked in 365 days. After 365 days, all locked rewards can be fully unlocked. All unlocked rewards will be transferred to the address `` `to` `` when the `claim` function is called.

2.3. AaveGovernanceV2

INFORMATION

Name	AaveGovernanceV2
Component	Xpool Governance
Network	BSC Mainnet
Explorer	https://bscscan.com/address/0x0BE1064A23e4091E9223816b44138e28A6d8e94c
Corresponding Aave Contract	AaveGovernanceV2 https://github.com/aave/governance-v2/blob/7b0e1e254a4bae161ca65886681366a002f94d4a/contracts/governance/AaveGovernanceV2.sol

	https://etherscan.io/address/0xEC568fffba86c094cf06b22134B23074DFE2252c

REVIEW RESULT

There are no differences among the Xpool's `AaveGovernanceV2` contract and the Aave's `AaveGovernanceV2` contract that could affect the contract logic.

2.4. Executor

INFORMATION

Name	Executor
Component	Xpool Governance
Network	BSC Mainnet
Explorer	https://bscscan.com/address/0x53948138C6a84826dc0fb7f5e383e0026eeD8636
Corresponding Aave Contract	Executor https://github.com/aave/governance-v2/blob/612833ab825f9cea57d82a5d214590ce16c7723d/contracts/governance/Executor.sol https://etherscan.io/address/0xee56e2b3d491590b5b31738cc34d5232f378a8d5

REVIEW RESULT

There are no differences among the Xpool's `Executor` contract and the Aave's `Executor` contract that could affect the contract logic.

2.5. GovernanceStrategy

INFORMATION

Name	GovernanceStrategy
Component	Xpool Governance
Network	BSC Mainnet
Explorer	https://bscscan.com/address/0xd7E0CdbC0f0681c3242555FCaF5cd6398b8e4378
Corresponding Aave Contract	GovernanceStrategy https://github.com/aave/governance-v2/blob/612833ab825f9cea57d82a5d214590ce16c7723d/contracts/governance/GovernanceStrategy.sol https://etherscan.io/address/0xb7e383ef9b1e9189fc0f71fb30af8aa14377429e

REVIEW RESULT

There are no differences among the Xpool's `GovernanceStrategy` contract and the Aave's `GovernanceStrategy` contract that could affect the contract logic.

2.6. LendingPoolAddressesProviderRegistry

INFORMATION

Name	LendingPoolAddressesProviderRegistry
Component	Xpool Core Protocol
Network	BSC Mainnet
Explorer	https://bscscan.com/address/0x4Af0DDc7606de5726c257dbFFe8253d32b48309e
Corresponding Aave Contract	LendingPoolAddressesProviderRegistry https://github.com/aave/protocol-v2/blob/30a2a19f6d28b6fb8d26fc07568ca0f2918f4070/contracts/protocol/configuration/LendingPoolAddressesProviderRegistry.sol https://etherscan.io/address/0x52D306e36E3B6B02c153d0266ff0f85d18BCD413

REVIEW RESULT

There are no differences among the Xpool's `LendingPoolAddressesProviderRegistry` contract and the Aave's `LendingPoolAddressesProviderRegistry` contract that could affect the contract logic.

2.7. LendingPoolAddressesProvider

INFORMATION

Name	LendingPoolAddressesProvider
Component	Xpool Core Protocol
Network	BSC Mainnet
Explorer	https://bscscan.com/address/0xe7c6Bc809fB3e8970e0F02b180d2669d64b6065D
Corresponding Aave Contract	LendingPoolAddressesProvider https://github.com/aave/protocol-v2/blob/30a2a19f6d28b6fb8d26fc07568ca0f2918f4070/contracts/protocol/configuration/LendingPoolAddressesProvider.sol https://etherscan.io/address/0xb53c1a33016b2dc2ff3653530bff1848a515c8c5

REVIEW RESULT

There are no differences among the Xpool's `LendingPoolAddressesProvider` contract and the Aave's `LendingPoolAddressesProvider` contract that could affect the contract logic.

2.8. LendingPool

INFORMATION

Name	LendingPool
Component	Xpool Core Protocol

Network	BSC Mainnet
Explorer	Proxy: https://bscscan.com/address/0x618BD91EBe2224B7CD433D92532730da10032e08 Current implementation: https://bscscan.com/address/0xf20f8898f51ccfc98d59ef2e7799d469ab01d015
Corresponding Aave Contract	LendingPool https://github.com/aave/protocol-v2/blob/30a2a19f6d28b6fb8d26fc07568ca0f2918f4070/contracts/protocol/lendingpool/LendingPool.sol https://etherscan.io/address/0x7d2768de32b0b80b7a3454c06bdac94a69ddc7a9 https://etherscan.io/address/0xc6845a5c768bf8d7681249f8927877efda425baf

REVIEW RESULT

There are no differences among the Xpool's `LendingPool` contract and the Aave's `LendingPool` contract that could affect the contract logic.

2.9. LendingPoolConfigurator

INFORMATION

Name	LendingPoolConfigurator
Component	Xpool Core Protocol
Network	BSC Mainnet
Explorer	Proxy: https://bscscan.com/address/0x289aa513180f10A9370d4489025E2289cE27425f Current implementation: https://bscscan.com/address/0xfda01a5e973b531c402455d3f18be9a2e2c27b02
Corresponding Aave Contract	LendingPoolConfigurator https://github.com/aave/protocol-v2/blob/4b8fabfca79445882aef6985c1ce8771f2bf105e/contracts/protocol/lendingpool/LendingPoolConfigurator.sol https://etherscan.io/address/0x311bb771e4f8952e6da169b425e7e92d6ac45756 https://etherscan.io/address/0x3a95ee42f080ff7289c8b4a14eb483a8644d7521

REVIEW RESULT

There are no differences among the Xpool's `LendingPoolConfigurator` contract and the Aave's `LendingPoolConfigurator` contract that could affect the contract logic.

2.10. AaveCollector

INFORMATION

Name	AaveCollector
Component	Xpool Core Protocol

Network	BSC Mainnet
Explorer	Proxy: https://bscscan.com/address/0xA007aCF323a87C44E94cCDB0cd227dAF9fdEca38 Current implementation: https://bscscan.com/address/0xf3887f8f2ce9c03ef44f01acb725c21b404ac5d1
Corresponding Aave Contract	AaveCollector https://etherscan.io/address/0x464c71f6c2f760dda6093dcb91c24c39e5d6e18c https://etherscan.io/address/0xe7cbd5b000958e19e6ca37e20aca499f83021469

REVIEW RESULT

There are no differences among the Xpool's `AaveCollector` contract and the Aave's `AaveCollector` contract that could affect the contract logic.

2.11. AaveOracle

INFORMATION

Name	AaveOracle
Component	Xpool Core Protocol
Network	BSC Mainnet
Explorer	https://bscscan.com/address/0xCc11D1eF0570FD599A43515Fd0924231655cBDF9
Corresponding Aave Contract	AaveOracle https://github.com/aave/protocol-v2/blob/30a2a19f6d28b6fb8d26fc07568ca0f2918f4070/contracts/misc/AaveOracle.sol https://etherscan.io/address/0xa50ba011c48153de246e5192c8f9258a2ba79ca9

REVIEW RESULT

There are no differences among the Xpool's `AaveOracle` contract and the Aave's `AaveOracle` contract that could affect the contract logic.

2.12. LendingRateOracle

INFORMATION

Name	LendingRateOracle
Component	Xpool Core Protocol
Network	BSC Mainnet
Explorer	https://bscscan.com/address/0xD4faCc4B20D5836675cE52b208919fC1f8ABaA37
Corresponding Aave Contract	LendingRateOracle https://github.com/aave/protocol-v2/blob/30a2a19f6d28b6fb8d26fc07568ca0f2918f4070/contracts/mocks/oracle/LendingRateOracle.sol

REVIEW RESULT

The Xpool's `LendingRateOracle` contract was deployed using the source code from the simple mocking `LendingRateOracle` contract in the Aave's github repo (stored at https://github.com/aave/protocol-v2/blob/30a2a19f6d28b6fb8d26fc07568ca0f2918f4070/contracts/mocks/oracle/LendingRateOracle.sol).

2.13. AaveProtocolDataProvider

INFORMATION

Name	AaveProtocolDataProvider
Component	Xpool Core Protocol
Network	BSC Mainnet
Explorer	https://bscscan.com/address/0x92484Abf038FdA08A539a6De2f1d0f6bF57B7851
Corresponding Aave Contract	AaveProtocolDataProvider https://github.com/aave/protocol-v2/blob/30a2a19f6d28b6fb8d26fc07568ca0f2918f4070/contracts/misc/AaveProtocolDataProvider.sol https://etherscan.io/address/0x057835Ad21a177dbdd3090bB1CAE03EaCF78Fc6d

REVIEW RESULT

There are no differences among the Xpool's `AaveProtocolDataProvider` contract and the Aave's `AaveProtocolDataProvider` contract that could affect the contract logic.

2.14. LendingPoolCollateralManager

INFORMATION

Name	LendingPoolCollateralManager
Component	Xpool Core Protocol
Network	BSC Mainnet
Explorer	https://bscscan.com/address/0xE7DFEa7fDaA415157cd6c7E07171764AB3B21866
Corresponding Aave Contract	LendingPoolCollateralManager https://github.com/aave/protocol-v2/blob/30a2a19f6d28b6fb8d26fc07568ca0f2918f4070/contracts/protocol/lendingpool/LendingPoolCollateralManager.sol https://etherscan.io/address/0xbd4765210d4167ce2a5b87280d9e8ee316d5ec7c

REVIEW RESULT

There are no differences among the Xpool's `LendingPoolCollateralManager` contract and the Aave's `LendingPoolCollateralManager` contract that could affect the contract logic.

2.15. AToken

INFORMATION

Name	AToken
Component	Xpool Core Protocol
Network	BSC Mainnet
Explorer	N/A
Corresponding Aave Contract	AToken https://github.com/aave/protocol-v2/blob/30a2a19f6d28b6fb8d26fc07568ca0f2918f4070/contracts/protocol/tokenization/AToken.sol https://etherscan.io/address/0xFFC97d72E13E01096502Cb8Eb52dEe56f74DAD7B https://etherscan.io/address/0xda5e8e1c3596d3cc11a4dd5ad66b8f03b5410f8c

All AToken contracts that were deployed in Xpool system on BSC Mainnet are listed in Table 2:

#	NAME	ADDRESSES
1	aXPO	Proxy: https://bscscan.com/address/0xFaeA32854E5e85D44bB6763860ABc02785f94115 Current implementation: https://bscscan.com/address/0xc80ab27f9c7b3589acb4d0a7181066b89837a537
2	aUSDT	Proxy: https://bscscan.com/address/0x5f612Ba87d8EDA251e6584D2AA85E7ceDc175348 Current implementation: https://bscscan.com/address/0xf1a38d2782c6e7657136cb9d7347ec08558ef8fb
3	aDAI	Proxy: https://bscscan.com/address/0xBb391B4F25caD95eB29A32a3254c4BfEA09716fe Current implementation: https://bscscan.com/address/0xd06b58d3fe9e6e1314ca1421d1db320c4f0d43f4
4	aUSDC	Proxy: https://bscscan.com/address/0x89Daf30F71Eb42E662D04eA4DE404072c82f29a4 Current implementation: https://bscscan.com/address/0x3f255aa1feca7f2505052f6d50922b2b51ef5c9e
5	aBUSD	Proxy: https://bscscan.com/address/0xd54B28Ee0f5E1F6Cf96D451e947bC57847aD1a2e Current implementation: https://bscscan.com/address/0x6e4da41b80ed6a5a41eb0b01c045873de99b5e8e
6	aBNB	Proxy: https://bscscan.com/address/0xFA563999aD4A34C57e3A2Cc1d821Ca722656F44D Current implementation: https://bscscan.com/address/0xd8dabb2cf242a76a451655527cfab37a48d8232b
7	aETH	Proxy: https://bscscan.com/address/0x250De5B6586e168baf5ccc7aC4E72871afDdcaCE Current implementation: https://bscscan.com/address/0x17ff563248aafe646d9febac74eef0076553e441

8	aBTC	Proxy: https://bscscan.com/address/0x6A5578525807c9B757f6DCb72d7776Cd66A31e36 Current implementation: https://bscscan.com/address/0xf36d7168c8845862bca3834e3670a7ef8521be60
9	aDOT	Proxy: https://bscscan.com/address/0xfd2470feff3E374A69bc6185E2F810Da8192c05D Current implementation: https://bscscan.com/address/0xdf2272aff791426220985489965645539edc2f3c
10	aLINK	Proxy: https://bscscan.com/address/0x14045e8D6428C332Dbf44405fAf0d1f8a097f573 Current implementation: https://bscscan.com/address/0x45c2571ac224f122d0f75ac9d543e046c1da55d4
11	aYFI	Proxy: https://bscscan.com/address/0x4088409032Fce0751c0ceaC1436A29648f34a7CA Current implementation: https://bscscan.com/address/0x6fd09729ea0c25fbc8cebeefb4145aec6914cb7b
12	aCAKE	Proxy: https://bscscan.com/address/0xd81FE17Df3fDD632A2AD48bc0f725D819deA180F Current implementation: https://bscscan.com/address/0x3283013112778758b4193261e5d5cc3a7928f7a1

Table 2: List of AToken contracts

REVIEW RESULT

All `AToken` contracts in Table 2 were deployed using the same source code. This source code has some differences from the current Aave's `AToken` smart contract source code.

All the modifications are in the `IncentivizedERC20` contract. The detailed modifications are:

- in *`_transfer`* function, changes from

`_incentivesController.handleAction(sender, currentTotalSupply, oldSenderBalance)` to
`_incentivesController.handleAction(sender, oldSenderBalance, currentTotalSupply)`

- in *`_transfer`* function, changes from

`_incentivesController.handleAction(recipient, currentTotalSupply, oldRecipientBalance)` to
`_incentivesController.handleAction(recipient, oldRecipientBalance, currentTotalSupply)`

- in *`_mint`* function, changes from

`_incentivesController.handleAction(account, oldTotalSupply, oldAccountBalance)` to
`_incentivesController.handleAction(account, oldAccountBalance, oldTotalSupply)`

- in *`_burn`* function, changes from

`_incentivesController.handleAction(account, oldTotalSupply, oldAccountBalance)` to
`_incentivesController.handleAction(account, oldAccountBalance, oldTotalSupply)`

These modifications are the swapping of agruments when calling then *`handleAction`* function of incentive controller. As of report writing time, all `AToken` contracts in Xpool system were deployed without using incentive controller, so these modifications will not affect to the

current contract execution. In the future, all deployed incentive controlllers must follow this arguments order in *handleAction* function.

2.16. StableDebtToken

INFORMATION

Name	StableDebtToken
Component	Xpool Core Protocol
Network	BSC Mainnet
Explorer	N/A
Corresponding Aave Contract	StableDebtToken https://github.com/aave/protocol-v2/blob/30a2a19f6d28b6fb8d26fc07568ca0f2918f4070/contracts/protocol/tokenization/StableDebtToken.sol

All StableDebtToken contracts that were deployed in Xpool system on BSC Mainnet are listed in Table 3:

#	NAME	ADDRESSES
1	stableDebtXPO	Proxy: https://bscscan.com/address/0x500Ed9Fd05b1fCF8506Fd85B33F12F32C2deBeE0 Current implementation: https://bscscan.com/address/0x9dd0b349fb077cae00c706f03b25433efc5c62f5
2	stableDebtUSDT	Proxy: https://bscscan.com/address/0x96126120497eC150A98745Ee6731bfbD8dc93405 Current implementation: https://bscscan.com/address/0x9dfdff5eee3da32e932479f584dbbfb9e9a29355
3	stableDebtDAI	Proxy: https://bscscan.com/address/0xb043CC262F6FC801117413aBbd29eA8C64401B03 Current implementation: https://bscscan.com/address/0x55bf5f90c68b38a62d498e16f5b872ed2b21cbdb
4	stableDebtUSDC	Proxy: https://bscscan.com/address/0x8b152871eD1C732856Ca9E0C30cCBfb0ff6437Ea Current implementation: https://bscscan.com/address/0x17f6a5747ee2f6e2e9592d38228b4447f4bbaa5f
5	stableDebtBUSD	Proxy: https://bscscan.com/address/0x6d51D1DDcCc0074fCDFB0f384F0ed964f0a6591B Current implementation: https://bscscan.com/address/0xaecac53df6a2032c4f5e2ab66b1291a6481260d7
6	stableDebtBNB	Proxy: https://bscscan.com/address/0xF401d476c125949bC099EF9981dBca28fc048E9C Current implementation: https://bscscan.com/address/0xdcc9074469ac79db3a25fcde274b9a1aa2029fc7
7	stableDebtETH	Proxy: https://bscscan.com/address/0xa9061fC14a59A8B97CdBf5e45cbCEFD8aa9024a6 Current implementation: https://bscscan.com/address/0xa7bc9ed69576390f5836880327ba37c25580a5bc

8	stableDebtBTC	Proxy: https://bscscan.com/address/0xf33eE9f6be555e34074E7775A942b06fF6dAfcd8 Current implementation: https://bscscan.com/address/0x84c4e3a7479826a156de29a88a333f3423b4751b
9	stableDebtDOT	Proxy: https://bscscan.com/address/0x7b12cD1672B55EFb95de2337BC2B1E9C0156EB45 Current implementation: https://bscscan.com/address/0x464ef5e27c13caafbe87e8f5c85967a4201c9238
10	stableDebtLINK	Proxy: https://bscscan.com/address/0x99602c75Ab9A68c5CAd3B80FB494a9885Add4c82 Current implementation: https://bscscan.com/address/0xd46ebd11cfbad7596371aa4aa4cf3b2225f2b822
11	stableDebtYFI	Proxy: https://bscscan.com/address/0x8c4D0B93786977f6efb1b2F123169433A564756c Current implementation: https://bscscan.com/address/0xfe79f6e9808eb0031fbbe574331629f90da0cc96
12	stableDebtCAKE	Proxy: https://bscscan.com/address/0x85e230D413CC5dCB71B2CB57026cA732Cd892439 Current implementation: https://bscscan.com/address/0x931670532fb1c3854addcf1d78721c2c5beafab9

Table 3: List of StableDebtToken contracts

REVIEW RESULT

All `StableDebtToken` contracts in Table 3 were deployed using the same source code. All the modifications are the agruments swapping when calling `handleAction` function as describe in 2.15.

In additional to four modifications in `IncentivizedERC20` contract, there are also two modifications in the StableDebtToken contract:

- in *`_mint`* function, changes from

`_incentivesController.handleAction(account, oldTotalSupply, oldAccountBalance)` to
`_incentivesController.handleAction(account, oldAccountBalance, oldTotalSupply)`

- in *`_burn`* function, changes from

`_incentivesController.handleAction(account, oldTotalSupply, oldAccountBalance)` to
`_incentivesController.handleAction(account, oldAccountBalance, oldTotalSupply)`

As of report writing time, all `StableDebtToken` contracts in Xpool system were deployed without using incentive controller, so these modifications will not affect to the current contract execution. In the future, all deployed incentive controlllers must follow this arguments order in *`handleAction`* function.

2.17. VariableDebtToken

INFORMATION

Name	VariableDebtToken

Component	Xpool Core Protocol
Network	BSC Mainnet
Explorer	N/A
Corresponding Aave Contract	VariableDebtToken https://github.com/aave/protocol-v2/blob/30a2a19f6d28b6fb8d26fc07568ca0f2918f4070/contracts/protocol/tokenization/VariableDebtToken.sol

All `VariableDebtToken` contracts that were deployed in Xpool system on BSC Mainnet are listed in Table 4:

#	NAME	ADDRESSES
1	variableDebtXPO	Proxy: https://bscscan.com/address/0x83492D7ff42c0Ca74ff1A8BBF763A3414bdE6837 Current implementation: https://bscscan.com/address/0xe7aeba483a0ef79d4f815787beec7c212ed76e80
2	variableDebtUSDT	Proxy: https://bscscan.com/address/0xA16878a810AD706cDDb6AdfD0E05E3E081641B43 Current implementation: https://bscscan.com/address/0x03420d36f42550c65815c9df72aba7900a7a554f
3	variableDebtDAI	Proxy: https://bscscan.com/address/0xd46c6Da40B2A3BdBdFb86b64C91e37B888a8f5d5 Current implementation: https://bscscan.com/address/0xcb60261a3710102ed228bd40354363a585c0959a
4	variableDebtUSDC	Proxy: https://bscscan.com/address/0xF48407b4DB1048da915a8641A54f0C9EB8F73629 Current implementation: https://bscscan.com/address/0x2812c012a39f1fc0d5bfe1d1b845483333e538dd
5	variableDebtBUSD	Proxy: https://bscscan.com/address/0x7a9843c5D6FC343395188d12fa61beB3708d80Cc Current implementation: https://bscscan.com/address/0x6bf552a1ca1038356e363a841b9257eee7d20bc9
6	variableDebtBNB	Proxy: https://bscscan.com/address/0x8E57724912D02cA60c0f751b47a00C6A1551E8Ae Current implementation: https://bscscan.com/address/0x3327702db3a1e33bc1a73ef86a8bdf6446ea3596
7	variableDebtETH	Proxy: https://bscscan.com/address/0x5A1D621Fb19D4aee247Cc52362Ca3aeE7225D871 Current implementation: https://bscscan.com/address/0x45b113123b77785a8b69d9293a57f700ef0780e3
8	variableDebtBTC	Proxy: https://bscscan.com/address/0x0aF338dc2c33096c8D3Ffff53451Fd0bDc80d93C Current implementation: https://bscscan.com/address/0x5ba8b207af1f696b636aac86e63377d82ff1b0b3
9	variableDebtDOT	Proxy: https://bscscan.com/address/0x55aCCeC7b86903CDC99409B0df1DAE93f7631E5E Current implementation: https://bscscan.com/address/0x00d4e70526665da41f821480c8d70fe91bef51cb

10	variableDebtLINK	Proxy: https://bscscan.com/address/0x1D9E33451bc466CFC9Fb391f90Ca0f74dDD27d06 Current implementation: https://bscscan.com/address/0x3e016f1cff65ff28835e00caf3e6144163f32895
11	variableDebtYFI	Proxy: https://bscscan.com/address/0x50F067a651D615432951D8b99A797818bCCA7099 Current implementation: https://bscscan.com/address/0x028b58d12464b78ef952601014938f9eae0ea511
12	variableDebtCAKE	Proxy: https://bscscan.com/address/0x9246D1677a2a4dC573a3Fc149400605bf7C82b9a Current implementation: https://bscscan.com/address/0xac764d77ea530eecfa7976e81f22cbbd70671d99

Table 4: List of VariableDebtToken contracts

REVIEW RESULT

All `VariableDebtToken` contracts in Table 4 were deployed using the same source code. All the modifications are the agruments swapping when calling `handleAction` function as describe in 2.15.

As of report writing time, all `VariableDebtToken` contracts in Xpool system were deployed without using incentive controller, so these modifications will not affect to the current contract execution. In the future, all deployed incentive controlllers must follow this arguments order in *`handleAction`* function.

2.18. InterestRateStrategy

INFORMATION

Name	InterestRateStrategy
Component	Xpool Core Protocol
Network	BSC Mainnet
Explorer	N/A
Corresponding Aave Contract	DefaultReserveInterestRateStrategy https://github.com/aave/protocol-v2/blob/30a2a19f6d28b6fb8d26fc07568ca0f2918f4070/contracts/protocol/lendingpool/DefaultReserveInterestRateStrategy.sol https://etherscan.io/address/0xd4cA26F2496195C4F886D464D8578368236bB747

All `InterestRateStrategy` contracts that were deployed in Xpool system on BSC Mainnet are listed in Table 5:

#	NAME	ADDRESS
1	strategyXPO	https://bscscan.com/address/0xE8425AbD844A320e67bbe763D182683f459F1248
2	strategyUSDT	https://bscscan.com/address/0xeB1BDa7f3C6d63743d9991286866664181b0D703

3	strategyDAI	https://bscscan.com/address/0xB6848300AFae78ca9663C53298Ba339D9B6bE694
4	strategyUSDC	https://bscscan.com/address/0x72CCbBF3638FE332f9dEe1E53d3e1ee61FFb1E53
5	strategyBUSD	https://bscscan.com/address/0x6c57f272ce616e025Be36fDd81Fd38D1e1B4C7f3
6	strategyBNB	https://bscscan.com/address/0xB14A2d3c02666E328E5da65C4Af629CDf60AD529
7	strategyETH	https://bscscan.com/address/0xBD487D646d1c5EE33d068009Dfd3Fe77139521D1
8	strategyBTC	https://bscscan.com/address/0x38F03D3a4E3746B4d188d73C34772779220daD24
9	strategyDOT	https://bscscan.com/address/0x816DD92e0A079087ad2c494b852Ed2538A2455a1
10	strategyLINK	https://bscscan.com/address/0x4B704C5717A3398647EB60E49f5ffcec15bA0f98
11	strategyYFI	https://bscscan.com/address/0xa9f4dE92d145B55cc459b9CC28AFcD0C0de28890
12	strategyCAKE	https://bscscan.com/address/0x17F7A0B7Ac8366972f438820Fc14dFa50Dc573e0

Table 5: List of InterestRateStrategy contracts

REVIEW RESULT

All `InterestRateStrategy` contracts listed in Table 5 were deployed using the same source code with the `DefaultReserveInterestRateStrategy` contract in the current Aave system.

3. CONCLUSION

After the review process, we have a final conclusion as follows:

- All Xpool smart contracts that were deployed on BSC Mainnet (listed in this report) are forked from the Aave smart contracts.
- Almost all the modifications are for the change of network adaption and do not affect the main business logic (except the addition of time locking logic in `StakedToken`).
- The Xpool system that was deployed on BSC Mainnet (with all smart contracts listed in this report) will work similarly to the Aave system that was deployed on Ethereum Mainnet.

4. VERSION HISTORY

Version	Date	Status/Changes	Created by
1.0	2021/06/01	Initial report	Verichains Lab
2.0	2021/06/02	Public report	Verichains Lab

APPENDIX A: TOKENTIMELOCKV2 CONTRACT

This is the source code of `TokenTimeLockV2` contract that was developed by Xpool team and was used in `StakedToken` contract.

```
contract TokenTimeLockV2 {
  using SafeMath for uint256;
  address public token;

  address public beneficier;
  uint256 lockedLength;
  uint256 lockedUntil;
  uint256 lastClaim;

  constructor(address rewardToken, address _beneficier, uint256 locked_length) public {
    beneficier = _beneficier;
    lockedLength = locked_length;
    lastClaim = block.timestamp;
    lockedUntil = block.timestamp.add(locked_length);
    token = rewardToken;
  }

  function balance() public view returns (uint256 claimableBalance, uint256 totalBalance)
{
    totalBalance = IERC20(token).balanceOf(address(this));
    if (block.timestamp >= lockedUntil) {
      claimableBalance = totalBalance;
    } else {
      claimableBalance =
totalBalance.mul(block.timestamp.sub(lastClaim)).div(lockedLength);
    }
  }

  function claim() external {
    (uint256 claimableBalance, ) = balance();
    if (claimableBalance > 0) {
      lastClaim = block.timestamp;
      require(IERC20(token).transfer(beneficier, claimableBalance));
    }
  }
}
```