Disclaimer: Information published below represents the official position of the Collateral Engineering Core Unit as a team.

Risk Summary/Key Takeaways

- The technical smart contract risk is considered MEDIUM

, as it will be using existing and already live in production MIP21 contracts and MIP22 Centrifuge Tinlake conduits. However the fact that the integration utilizes contracts outside of Maker's scope and has not been extensively reviewed by Maker increases the technical risk. Furthermore, the increased operational overhead of managing 4 vaults instead of a single vault, results in increased maintenance directly in executive spells, which also increases the technical risk.

- Whilst technical risk is considered medium the proposed solution requires manual/human monitoring, which will require clear processes for assessing the ongoing performance of the collateral based on the reporting required by the various actors.
- Due to the manual operations of RWA vaults, CES recommends that for future deals consideration is made to limit the amount of maintenance work required in executive spells. This analysis is based on a future perspective where if more of these types of deals get submitted to Maker, it could potentially dramatically increase the amount of vault facilities that would need to be managed directly in executive spells, which increases the overall technical risk in the Maker Protocol.
- Any technology required for the monitoring and uploads of data to off-chain infrastructure is not in scope for this technical assessment.
- The counterparty custodian infrastructure and security is out of scope of this technical assessment.
- The Centrifuge Tinlake platform infrastructure and its security is out of scope of this technical assessment. Readers should note that notwithstanding Centrifuge Drop Tokens are underlying collateral as these are not held directly by Maker Protocol smart contracts. Any enforcement of liquidation or rights would require enforcement in the real world through TACO - the real world trust that has entered into a secured loan agreement with Blocktower Drop SPV the holder of the Drop tokens.

General Information

- Symbol:

RWA010, RWA011, RWA012, RWA013

- Token Names:

RWA-010, RWA-011, RWA-012, RWA-013

- Ilk Registry Name:

RWA010,11,12,13-A: BlockTower Credit

- Total Supply:

1

- Relevant MIP Information:
- MIP6 Collateral Onboarding Application: BlockTower Credit (Arranger)
- MIP21: Real World Assets – Off-Chain Asset Backed Lender
- MIP22: Centrifuge Direct Liquidation Module
- MIP6 Collateral Onboarding Application: BlockTower Credit (Arranger)
- MIP21: Real World Assets – Off-Chain Asset Backed Lender
- MIP22: Centrifuge Direct Liquidation Module
- BlockTower Website:

https://www.blocktower.com/

- Github Repository:
- GitHub - makerdao/mip21-toolkit: SW Repo; Content Manager: CES-001; MIP21 Toolkit: Equipment for Off-chain

Asset Backed Lending in MakerDAO

- GitHub - centrifuge/tinlake-maker-lib
- GitHub - makerdao/mip21-toolkit: SW Repo; Content Manager: CES-001; MIP21 Toolkit: Equipment for Off-chain Asset Backed Lending in MakerDAO
- GitHub - centrifuge/tinlake-maker-lib
- Collateral Type Adapter:

The collateral will use the MIP21 authed join and exit functions.

Technical Information

- Implements ERC20 Token Standard:

Yes

- Compiler Version:

solidity:0.6.12

- Decimals:

18

- Overflow checks:

Yes

- Mitigation against overflow race-condition:

No

- Upgradeable contract patterns:

No

- Access control or restriction lists:

No.

- Non-standard features or behaviors:

No.

- Key addresses:
- The auth governance address for RwaLiquidationOracle, RwaUrn and RwaConduit contracts.
- The operator address that is permitted to operate on the RwaUrn and RwaConduit contracts. This can be multiple addresses, however, each address must be approved by governance.
- The auth governance address for RwaLiquidationOracle, RwaUrn and RwaConduit contracts.
- The operator address that is permitted to operate on the RwaUrn and RwaConduit contracts. This can be multiple addresses, however, each address must be approved by governance.

Transaction Outline

Parameter

Value

Vault 1:

Token name:

RWA-010

Debt Ceiling (line):

20,000,000 (20 Million)

Stability Fee (duty):

4 %

Onchain Liquidation Ratio (mat):

100 %

Term length:

60 months

Oracle price (pip/ds-value):

24,333,058

Debt write-off timelock (tau):

0

Hash of borrower’s final term sheet with MakerDAO (doc):

QmRqsQRnLfaRuhFr5wCfDQZKzNo7FRVUyTJPhS76nfz6nX

Liquidation Process:

MIP21c3 & MIP22c2

Vault 2:

Token name:

RWA-011

Debt Ceiling (line):

30,000,000 (30 Million)

Stability Fee (duty):

4 %

Onchain Liquidation Ratio (mat):

100 %

Term length:

60 months

Oracle price (pip/ds-value):

36,499,587

Debt write-off timelock (tau):

0

Hash of borrower’s final term sheet with MakerDAO (doc):

QmRqsQRnLfaRuhFr5wCfDQZKzNo7FRVUyTJPhS76nfz6nX

Liquidation Process:

MIP21c3 & MIP22c2

Vault 3:

Token name:

RWA-012

Debt Ceiling (line):

30,000,000 (30 Million)

Stability Fee (duty):

4 %

Onchain Liquidation Ratio (mat):

100 %

Term length:

60 months

Oracle price (pip/ds-value):

36,499,587

Debt write-off timelock (tau):

0

Hash of borrower’s final term sheet with MakerDAO (doc):

QmRqsQRnLfaRuhFr5wCfDQZKzNo7FRVUyTJPhS76nfz6nX

Liquidation Process:

MIP21c3 & MIP22c2

Vault 4:

Token name:

RWA-013

Debt Ceiling (line):

70,000,000 (70 Million)

Stability Fee (duty):

4 %

Onchain Liquidation Ratio (mat):

100 %

Term length:

60 months

Oracle price (pip/ds-value):

85,165,703

Debt write-off timelock (tau):

0

Hash of borrower’s final term sheet with MakerDAO (doc):

QmRqsQRnLfaRuhFr5wCfDQZKzNo7FRVUyTJPhS76nfz6nX

Liquidation Process:

MIP21c3 & MIP22c2

Implementation Design

Operational Security Considerations

This assessment does not take into consideration the security of the Centrifuge systems, such as Tinlake, TIN and DROP tokens, or the Centrifuge parachain. In case such external systems fail, this assessment recommends that that primary reliance be on the legal documentation for ensuring the obligations between the parties are met.

Modifications to standard MIP21

MIP22 Centrifuge Tinlake Conduits

This vault utilizes the Tinlake Manager contract instead of a conduit, to facilitate the Dai flows between the RwaUrn and the Tinlake Pools. This contract is the same that is live in production for RWA-002, 3, 4, 5.

Setting up the vault

- Deploy contracts:
- 4xAuthGemJoin
- 4xRwaUrn
- 4xRwaToken
- 4xTinlakeManager (Done by Centrifuge)
- 4xAuthGemJoin
- 4xRwaUrn
- 4xRwaToken
- 4xTinlakeManager (Done by Centrifuge)
- Deploy executive spell which:
- Sets all risk parameters for the vault (see table in “Transaction Outline” above)
- Sets BlockTower TinlakeManager contracts as operator

of the RwaUrn contracts

- Sets DsPauseProxy as auth

role for the RwaUrn conracts.

- Send the RWA tokens into the respective TinlakeManagers.
- Sets all risk parameters for the vault (see table in “Transaction Outline” above)
- Sets BlockTower TinlakeManager contracts as operator

of the RwaUrn contracts

- Sets DsPauseProxy as auth

role for the RwaUrn conracts.

- Send the RWA tokens into the respective TinlakeManagers.

Drawing Dai/Paying back Dai

Dai is automatically supplied to/withdrawn from the Tinlake Pools to provide liquidity using the Tinlake Manager which is granted the operator

role on the RwaUrn, and is thus authorized to call draw

/wipe

.

Paying vault fees

Stability fees accrue automatically to the surplus buffer.

Liquidations & Losses

- Monitoring of the RWA investments happens off chain.
- Maker Governance can manually trigger a liquidation event by calling the Liquidation Oracle in an executive spell.
- In case a liquidation is triggered the Trustee will begin selling collateral assets and pay back Dai to the RwaUrn.
- If the RwaUrn has not been fully repaid after a liquidation event, Maker Governance can write off the debt to the surplus buffer using the LiquidationOracle through an executive spell.

Emergency Shutdown

- Dai holders will be able to redeem Dai for RWA010-13 tokens through the End module.
- Subsequently RWA010-13 token holders can swap RWA010-13 tokens for a proportionate amount of USDC in the RwaCageSettlement contract.
- The RwaCageSettlement is still in development, but very close to being completed. When this contract is finished, it can also be reused for other RWA vaults.
- The fact that this contract is not live in production yet is not a blocker to initiate the vault, as existing RWA also lacks this easy onchain redeem functionality in case of ES. The plan is to deploy this contract as soon as it is ready.

Architecture

MIP21 Contracts

In order to onboard Blocktower Credit Vault the MIP21 standard contracts and Tinlake Adapter (MIP22) utilized in Centrifuge vaults will be utilized. 4 vaults with different debt ceilings will be setup to integrate with 4 Tinlake Pools. MIP21 turns off automatic liquidation of the vaults and ensures that the borrower is prevented from minting more DAI than the Debt Ceiling (line). Any liquidation of a vault would require that Maker Governance trigger a liquidation.

For the proposed implementation, the following MIP21 and MIP22 contracts will be utilized:

- RwaToken
- RwaUrn
- RwaLiquidationOracle
- TinlakeManager

As part of the spell, after the Urns has been set up the spell will lock the ERC20 tokens into the RwaUrn, to allow of Dai minting up to the specified debt ceilings of the vaults.

Below is a summary of the contracts which will be utilized.

RwaToken

Source code

A standard implementation of the ERC20 token standard, with the balanceOf(address) of the deployer of the contract being set to 1 WAD at deployment. There are 18 decimals of precision.

There are three state changing functions, that are all available to the token holder, and are specific to the ERC20 token standard:

- transfer(address dst, uint wad) external returns (bool)
- transferFrom(address src, address dst, uint wad) public returns (bool)
- approve(address usr, uint wad) external returns (bool)

RwaUrn

Source code

The RwaUrn is unique to each MIP21 collateral type. Aside from the core DSS wards, can, rely(address), deny(address), hope(address), and nope(address) functions, there are five functions:

- file(bytes32, address)
- lock(uint256)
- free(uint256)
- draw(uint256)
- wipe(uint256)
- exit(uint256)

The file

function can only be called by governance (via the auth modifier)

The rest of the functions can only be called by those who have been given the operator permission (hope

'd or nope

'd) on the RwaUrn contract. And any Dai drawn by the RwaUrn can only be sent to the RwaOutputConduit address defined by governance when deploying the contract. In this case the RwaOutputConduit address will be the custody address of Monetalis.

Tinlake Manager

Source code

The Tinlake Manager contract integrates the RwaUrn with the Tinlake Platform. It takes the place of the conduits in a standard MIP21 setup, and routes Dai between the Maker Protocol and the Tinlake Platform.

RwaLiquidationOracle

Source code

The RwaLiquidationOracle contract consists of six state-changing functions (besides the usual DSS rely(address)

, deny(address)

), all protected by the auth

modifier and can only be called by governance:

- file

can be called by governance to change the vow address (used in cull

).

- init

is the initialization function. It takes 4 parameters:

- ilk

: name of the vault, in this case, RWA010.

- val

: estimated value of the collateral token.

- doc

: link to legal documents representing the underlying legal scheme.

- tau

: minimum delay between the soft-liquidation and the hard-liquidation/write-off.

- ilk

: name of the vault, in this case, RWA010.

- val

: estimated value of the collateral token.

- doc

: link to legal documents representing the underlying legal scheme.

- tau

: minimum delay between the soft-liquidation and the hard-liquidation/write-off.

- bump

can be called by governance to increase or decrease the estimated value of the collateral.

- tell

can be called by governance to start a soft-liquidation.

- cure

can be called by governance after a soft-liquidation has been triggered to stop it.

- cull

can be called by governance to start a hard-liquidation/write-off. This will mark all the remaining debt of the vault as bad debt and impact the Surplus Buffer (vow).

There is one externally accessible view function called good(bytes32)

that anyone can use to check the liquidation status of the position. This function does not change contract state.

This is not a typical Maker Oracle. It will only report on the liquidation status of RwaUrn, and can only be acted upon by governance. This oracle is not vulnerable to flash loan attacks or any manipulation aside from a governance attack.

Contract Risk Summary

The reader should note that his assessment is solely with respect to the smart contract transactions and interfaces required to effect the on-chain state changes required under the proposed architecture and excludes any technical functionality related to the technical infrastructure required for monitoring and uploading data to MakerDAO offchain storage and ongoing monitoring. Furthermore, this assessment does not take into consideration the operational and technical security of the counterparties and it systems of BlockTower or Centrifuge.

Risk Analysis Conclusion: Medium technical risk

The RWA code implementation resides within a sandbox-like environment, and any operation not related to locking, freeing, drawing, or wiping in the RwaUrn must be voted on by governance. The code itself is lightweight. This implementation uses simplified Oracle and Urn contracts to achieve the functionality required for this specific instance of RWA. Furthermore, MIP21 contracts and the Tinlake Manager contracts have been live in production for over a year. However the fact that this vault integration utilizes smart contracts from external sources, outside of the Maker Protocol scope, and has deep integrations into the Centrifuge platform that has not been reviewed, increases the technical risk of this implementation.

In terms of technical maintenance of the proposed system, utilizing numerous vaults to facilitate a single deal increases the complexity and required technical maintenance in executive spells, and therefore slightly increases the technical risks. Therefore the technical risk of this implementation is considered MEDIUM

.

Supporting Materials

Sūrya’s Description Report

Legend

Symbol

Meaning

Function can modify state

Function is payable

Contracts Description Table

File Name

SHA-1 Hash

RwaLiquidationOracle.sol

88c2b4fac899d39af0198c1fb4776171e4249c19

RwaUrn.sol

f8746cc5cd0fd44f4616d0452f013545c08d45eb

RwaToken.sol

8d75732d93e0ad82a7bf3e0faf34550082291775

Contract

Type

Bases

∟

Function Name

Visibility

Mutability

Modifiers

RwaLiquidationOracle

Implementation

∟

rely

External

auth

∟

deny

External

auth

∟

add

Internal

∟

mul

Internal

∟

Constructor

Public

NO❗

∟

file

External

auth

∟

init

External

auth

∟

bump

External

auth

∟

tell

External

auth

∟

cure

External

auth

∟

cull

External

auth

∟

good

External

NO❗

RwaUrn

Implementation

∟

rely

External

auth

∟

deny

External

auth

∟

hope

External

auth

∟

nope

External

auth

∟

add

Internal

∟

sub

Internal

∟

mul

Internal

∟

divup

Internal

∟

Constructor

Public

NO❗

∟

file

External

auth

∟

lock

External

operator

∟

free

External

operator

∟

draw

External

operator

∟

wipe

External

NO❗

∟

quit

External

NO❗

RwaToken

Implementation

∟

add

Internal

∟

sub

Internal

∟

Constructor

Public

NO❗

∟

transfer

External

NO❗

∟

transferFrom

Public

NO❗

∟

approve

External

NO!

Inheritance Graph

[

1184×156 27.1 KB

](//makerdao-forum-backup.s3.dualstack.us-east-1.amazonaws.com/original/3X/9/6/96378b1bebccec8c4c4899a4f6ddc6d629621f1f.png)

Call Graph

[

579×3608 287 KB

](//makerdao-forum-backup.s3.dualstack.us-east-1.amazonaws.com/original/3X/2/4/2424b8b45a2a6e59c4a4cb52c959c3ca7d87731c.png)

Interaction Graph

[

1072×1394 204 KB

](//makerdao-forum-backup.s3.dualstack.us-east-1.amazonaws.com/original/3X/f/b/fba7c6ee8d1884d554ad908219ba3e480437730a.jpeg)