bitSmiley White Paper

(Initial Draft)

bitSmiley Team

December 26,2023

Introduction

BitSmiley is a protocol based on the Bitcoin blockchain under the Fintegra framework. It consists of three main components: a decentralized overcollateralized stablecoin protocol, a native trustless lending protocol, and a derivatives protocol. These components work together to provide a comprehensive financial ecosystem on the Bitcoin blockchain, enhancing its functionality and utility in the decentralized finance (DeFi) space.

1 Decentralized Stablecoin Protocol--bitUSD

A native overcollateralized stablecoin protocol on the Bitcoin blockchain is crucial due to the inherent volatility of Bitcoin, which limits its use in everyday transactions and as a stable store of value. A native stablecoin would expand Bitcoin's utility, allowing for a wider range of financial activities like lending, borrowing, and yield farming directly on its network. This would enhance the decentralization and trust aspects fundamental to Bitcoin, aligning with its ethos of trustless and permissionless finance. Moreover, such a protocol would provide much-needed liquidity and accessibility, making the Bitcoin ecosystem more approachable, especially for those in economically unstable regions or looking to hedge against fiat currency inflation. Overall, this move could transform Bitcoin from merely a store of value to a versatile medium of exchange in the decentralized finance space.

1.1 Stablecoin bitUSD

bitUSD is a cryptocurrency on the Bitcoin blockchain, generated from overcollateralized assets and softly pegged to the US Dollar. The issuance of bitUSD is decentralized and permissionless, allowing any Bitcoin holder to generate bitUSD by depositing Bitcoin into the 'bitSmiley Treasury' smart contract. Every bitUSD in circulation is backed by excess collateral, and all bitUSD transactions are publicly visible on the Bitcoin blockchain. The monetary attributes of bitUSD are as follows:

(1) Store of Value: bitUSD maintain their value over time, making them a reliable store of value. This is especially significant in the cryptocurrency market known for its volatility.

(2) Medium of Exchange: bitUSD is widely accepted for trade and payments, primarily due to their stability compared to other cryptocurrencies.

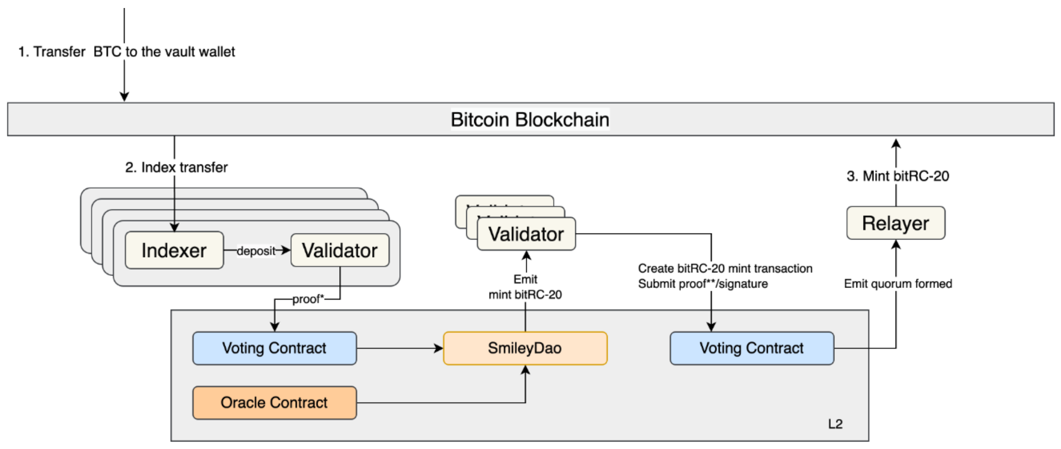
(3) Unit of Account: bitUSD can be used to price goods and services and to keep financial records, as their value is stable and predictable.

(4) Standard of Deferred Payment: bitUSD can be used in contracts for future payments because their stable value ensures that both parties can anticipate the future worth accurately, reducing the risk associated with volatility. In the bitSmiley protocol, bitUSD can be used to settle debts, such as paying stability fees or interest on loans.

Users deposit a specific amount of BTC into the bitSmiley Treasury to generate bitUSD. To retrieve their deposited BTC, users need to repay the generated bitUSD and also pay a certain amount of stability fee.

1.2 Process Flow

The mint flow is shown below:



* The proof that validator must submit is a merkle path proof of the transaciton
** The proof for mint operation on BTC is a merkle proof of the transaction recipt

Figure1:Mint bitUSD flow

When the user wants to withdraw the original BTC, the following flow is proposed in Figure 2.

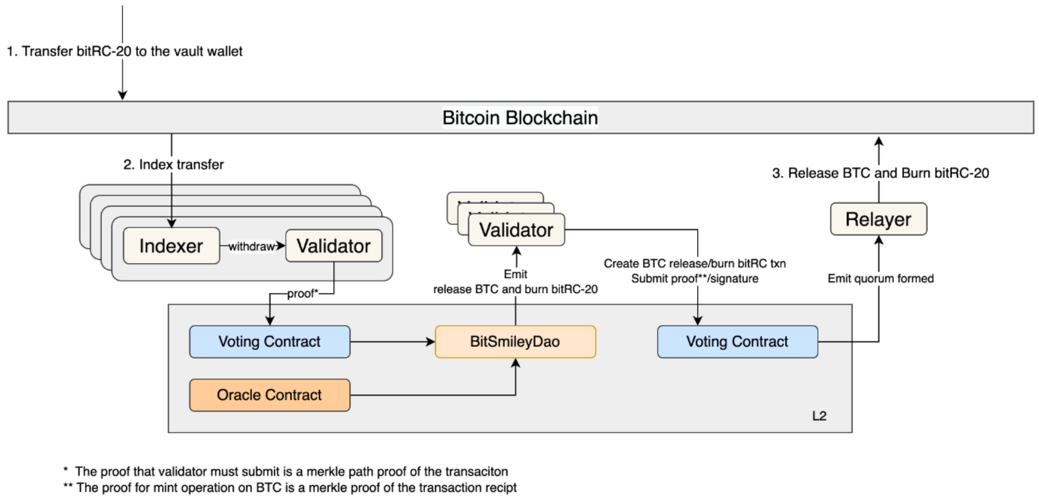

1. Transfer bitRC-20 to the vault wallet
Bitcoin Blockchain
2. Index transfer
Indexer
withdraw
Validator
proof*
Validator
Emit
release BTC and burn bitRC-20
Create BTC release/burn bitRC txn
Submit proof**/signature
3. Release BTC and Burn bitRC-20
Relayer
Emit quorum formed
Voting Contract
BitSmileyDao
Voting Contract
Oracle Contract
L2
* The proof that validator must submit is a merkle path proof of the transaciton
** The proof for mint operation on BTC is a merkle proof of the transaction recipt

Figure2:Withdraw bitUSD flow

If the price of the collateral drops below certain threshold, liquidation will be triggered and it happens over L2. The proposed flow is shown in Figure 3.

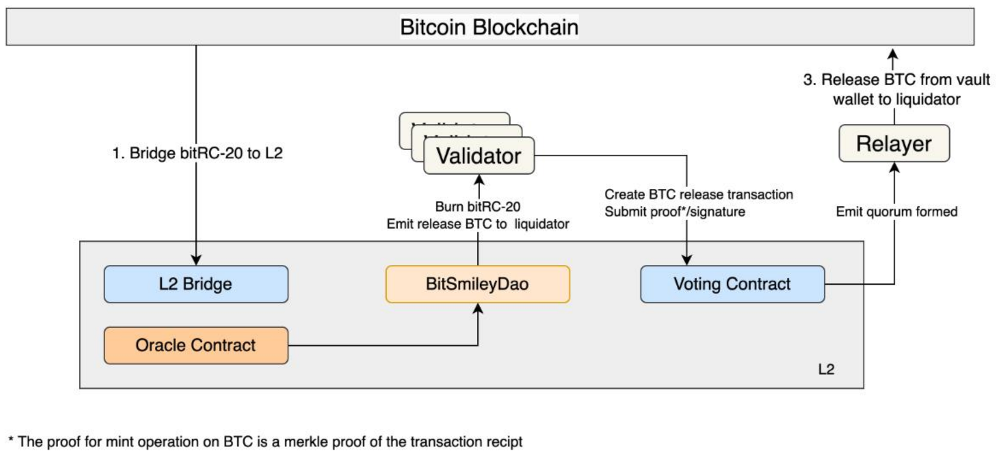

Bitcoin Blockchain
1. Bridge bitRC-20 to L2
Validator
Burn bitRC-20
Emit release BTC to liquidator
Create BTC release transaction
Submit proof*/signature
3. Release BTC from vault wallet to liquidator
Relayer
Emit quorum formed
L2 Bridge
BitSmileyDao
Voting Contract
Oracle Contract
L2
* The proof for mint operation on BTC is a merkle proof of the transaction recipt

Figure3:Liquidate bitUSD flow

1.3 Collateral Liquidation Design

To ensure that each bitUsd is adequately backed by collateral, any Treasury with a Col-lateral to Debt Ratio falling below the threshold will enter the liquidation process. The liquidation formula is as follows：

$$Liquidation\ Price = \frac{bitUSDGenerated * LiquidationRatio}{Quantity\ of\ Collateral}$$

The liquidated party will incur a liquidation penalty, and depending on the duration of the liquidation process, the distribution ratio of the liquidation penalty between bitSmiley and the liquidator will change. The longer the liquidation takes, the more liquidation penalty the liquidator will receive. The bitSmiley liquidation will adopt a Dutch auction format, where the bidding starts at a high price and decreases in steps. As the auction progresses, the price will automatically float downwards over time. If a bidder is willing to bid at a certain price during the fluctuation, then the auction is considered successful at that price. Specifically, in the bitSmiley protocol, the system will set an initial starting bid price, and the quantity of collateral to be acquired will be determined based on a time-price curve. Once a liquidator participates within this time window, the transaction is executed immediately. The parameters involved in the time-price curve are as follows:

（1） Price Buffer Coefficient: Determines the starting bid price for the collateral.

（2） Price Function: Determines the percentage decrease every N seconds.

（3） Maximum Auction Price Drawdown: The percentage of price drop.

（4） Maximum Auction Period: Determines the auction time window.

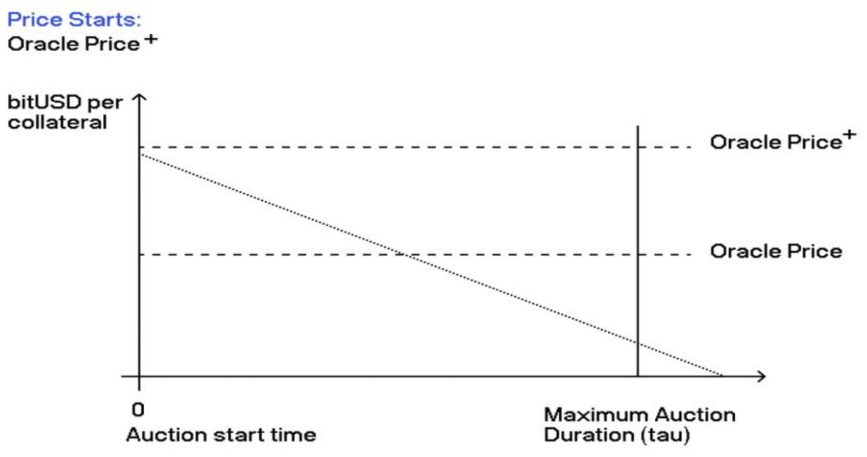


Figure 4:Collateral auction design

1.4 Surplus Auction

The money earned through the bitSmiley stablecoin protocol will be stored in a surplus account, including stability fees and auction proceeds. Of this, 90% will be used as a liquidation buffer. In cases where collateral auction proceeds are insufficient to cover the expenses of the bitSmiley Treasury, the liquidation buffer will be utilized to compensate. The remaining 10% will be allocated for team operational expenses. The auction will utilize an English auction mechanism. The trigger condition is when the balance in the surplus account, after deducting debts, exceeds a certain threshold. The excess amount will be auctioned off, and the proceeds from the auction will be used to incentivize users of the bitSmiley community.

1.5 Debt Auction

When the assets in the surplus account are insufficient to cover bad debts, the platform will conduct a debt auction using future income as collateral. This auction will follow an English auction mechanism, where participants are required to bid using BTC. The platform guarantees that once future income is received and after deducting basic operational expenses, creditors will be prioritized for repayment.

1.6 Emergency Shutdown Mechanism

Emergency shutdown is the last resort for bitSmiley to respond to malicious attacks, illegal intrusions, and security vulnerabilities. The decision to initiate an emergency shutdown will be made by a vote of an emergency governance committee composed of selected voters. This committee could freeze the oracle to shut down the protocol. The entire shutdown process is divided into 4 steps:

1. Protocol Closure: Once shutdown is triggered, the oracle freezes, preventing users from operating or creating new Treasuries, while allowing them to withdraw collateral ex- ceding the amount required for debt collateralization.
2. Auction Mechanism Activation: Collateral auctions are triggered and are mandated to complete within a set time frame.
3. Redemption of Remaining Collateral: After the auction ends, bitUSD holders can redeem the collateral at a fixed exchange rate.
4. Protocol Restart: The decision to restart the protocol is made by the Economic Governance Committee.

2 Native Peer-to-Peer Lending Protocol--bitLending

bitLending is a fully decentralized, peer-to-peer lending protocol developed by bitSmiley, based on the native Bitcoin blockchain. Users can choose to lend out any bitRC-20 tokens, including bitUSD, using high-value and highly liquid cryptocurrencies like Bitcoin as collateral. Due to the technical limitations of the Bitcoin blockchain, bitLending will adopt a peer-to-peer loan matching mechanism. Once this mechanism is mature, bitLending will introduce order splitting and merging function to enhance the capital efficiency of the protocol.

2.1 Borrow And Repay

For borrow and repay to work trustless, bitLending relies on existing Bitcoin opcodes and generates a lending transaction that ensures:

(1) Borrowers receive the token they wish to borrow.

(2) Borrowers' collateral is locked during the lending period.

(3) The borrowers' collateral will be released once borrowers repay the token.

Do note that the collateral provided by borrowers must be approved by bitLending.

2.2 bitLending Process

To simplify the notations, imagine Alice wants to borrow bitRC-20 and provide BTC as collateral. Bob wants to lend his bitRC-20 and earn some interests. The process is as follows:

- Bob posts on BitLending with a certain amount of bitRC-20 he is willing to lend and sets the expected interest rate.
- Alice requests to borrow from bitLending. bitLending matches Bob’s listing with Alice. Alice agrees with the interest rate and duration. Once matched, the expected repay block number with principal plus interest will be calculated and confirmed.
- bitLending matches Alice and Bob and creates a multisig wallet.
- bitLending generates fund transfer transactions for Alice and Bob to the multisig wallet and also a lending transaction for the multisig wallet, asking both Alice and Bob to sign.

- Bob and Alice execute the fund transfer transactions.
- bitLending broadcasts the lending transaction.
- Alice withdraws the bitRC-20 from the lending transaction.
- Alice repays the bitRC-20 by publishing a repay transaction.
- When Bob receives the bitRC-20, Alice will atomically receive her BTC.

If Alice does not repay the bitRC-20 before the deadline, Bob will be entitled to take all the collateral.

2.3 Fund Protection---bitInsurance

In collateral-based lending protocols like Compound, a liquidation protocol is backed by an on-chain Oracle. However, this model may not be directly applicable to Bitcoin. One primary consideration is the comparatively slower block time in Bitcoin, approximately 10 minutes. This delay might pose challenges for a liquidation process similar to Compound's, where triggering at the first minute may render execution invalid by the 10th minute. Additionally, Bitcoin, relying solely on Unspent Transaction Outputs (UTXO) for transactions, lacks the capability to support on-chain price feeds.

bitSmiley protects user funds in a different way. It is possible that Alice does not repay the bitRC-20 in time, and this could be due to either a drastic increase in the price of the borrowed bitRC-20 or a significant drop in the price of BTC. In the first scenario, Bob will not incur any loss and the value of BTC still holds. However, if the value of BTC drops significantly, Bob may choose to buy insurance. Both Alice and Bob will pay a certain amount of BTC for insurance so that, if Alice does not repay, the insurance will completely refund Bob. The sequence diagram is as follows:

- bitSmiley matches Alice, Bob, and Charlie, and creates a multisig wallet.
- Bob and Alice transfer tokens to the multisig wallet.
- Bob and Alice transfer the insurance fees to the multisig wallet.
- bitSmiley broadcasts the lending transaction."
- Charlie withdraws the insurance fee.
- Alice withdraws the BTC from the lending transaction.

- Alice does not repay the BTC.
- Bob will receive the original BTC back from Charlie, and Charlie will receive the collateral.

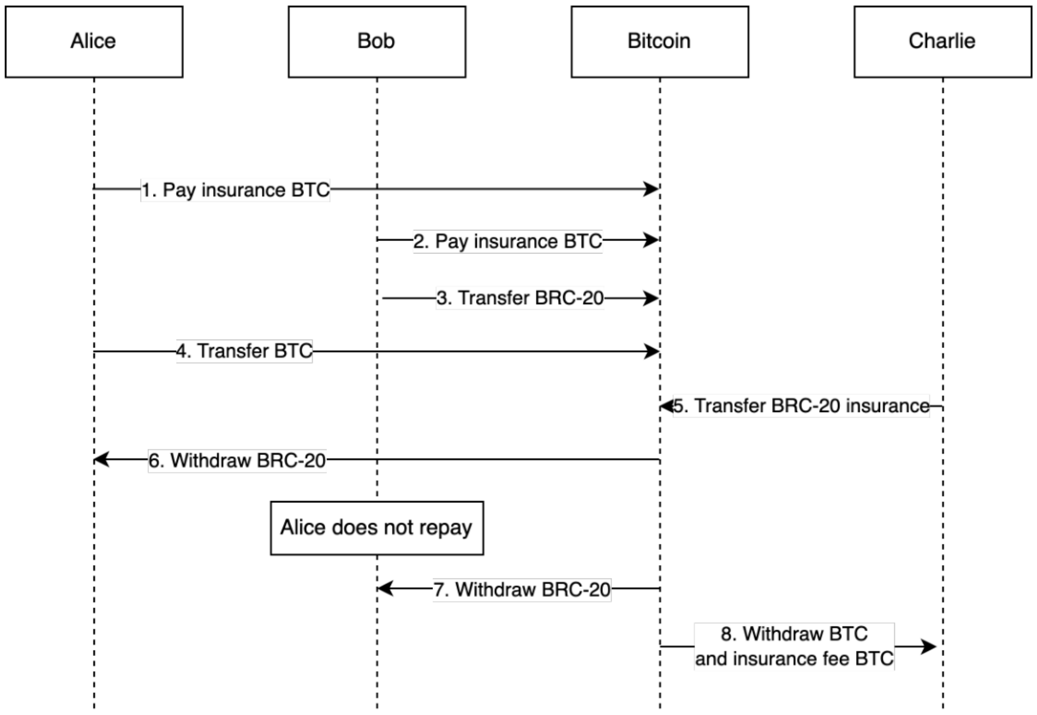


Figure5:bitInsurance flow

2.4 Pricing Design of bitInsurance

Alice and Bob need to pay for the insurance if Bob requests it. The price for Alice and Bob will be different. bitSmiley proposes two insurance pricing models: Extreme Value Theory and T-Copula.

For Extreme Value Theory, it works as follows:

1. Analyze Tail Data: Obtain historical tail data for BTC and bitRC-20 tokens, analyzing the frequency, duration, and magnitude of extreme fluctuations.
2. Establish a Tail Risk Model: Consider both Peak Over Threshold (POT) and Block Maxima methods.
3. Quantify Risk Indicators: Based on the historical data analysis, estimate the probability of market fluctuations for two potential loss scenarios for the insurer (BTC price decrease or bitRC-20 price increase), using Tail Value at Risk (TVaR) to measure the size of extreme losses.

The TVaR calculation formula is:

$$TVaR_{\alpha} = \frac{1}{1-\alpha}\int_{VaR\alpha}^{\infty} xdF(x)$$

where F represents the cumulative distribution function of losses, and $VaR\alpha$ is the Value at Risk at the confidence level α.

4. Risk Assessment and Pricing: With quantified indicators in hand, set a reasonable risk premium and guarantee fee.

For T-Copula, it works as follows:

1. Collect historical price data for BTC and bitRC-20, including data under different market conditions.
2. Determine Marginal Distributions: Find suitable marginal distributions for BTC and bitRC-20, fitting the data to a specific probability distribution model, such as the log-normal distribution.
3. Choose T-Copula: Mainly used to describe and simulate the dependency between multiple random variables, connecting multiple distribution functions into a multidimensional joint distribution function without changing their marginal distributions.
4. Parameter Estimation: Estimate parameters for the T-Copula model, including correlation parameters and degrees of freedom.
5. Simulation and Analysis: Use T-Copula to simulate the joint behavior of BTC and bitRC-20 prices under different market conditions, analyzing the mutual impact of these two assets under extreme market conditions.
6. Risk Assessment and Pricing: Use T-Copula correlation information to evaluate the overall transaction risk and determine a reasonable guaranteed fee.

2.5 Black Box Purchase Mechanism

When Alice and Bob decide to buy insurance, the platform calculates a reference quote for the policy using its model. Alice and Bob each make an offer for the insurance, but their bids must not be lower than 25% of the reference quote. The platform takes the lower of the two bids as the final buyer's offer. The insurance seller, unaware of Alice and Bob's bids, makes an offer within a set time. If the seller's lowest offer is less than the final bid of Alice and Bob within this time, the insurance is sold. The three lowest bidders receive varying amounts of platform tokens as rewards. If the seller's offer is higher than Alice and Bob's final bid, the final bid is revealed after the set time. If a seller accepts this final offer, they can receive the reward alone.

2.6 bitLending Technical Details

The initial version of the transaction requires bitSmiley to generate a secret exclusively for Bob, which Bob should not disclose to anyone until Alice has repaid. The following are the inputs and outputs from the multisig wallet:

- Inputs:

  1. Alices's BTC

  2. Bob's bitRC-20
- Outputs:
  1. Alice can withdraw the bitRC-20
  2. If

     current block number > deadline, Bob can withdraw the BTC

     Else

     hash(preimage) == bytes, Alice can withdraw the BTC

     The corresponding repay transaction would be:
- Inputs:
  1. Alice's bitRC-20
- Outputs:
  1. hash(preimage) == bytes, Bob can spend the bitRC-20

In this case, as long as Bob withdraws the bitRC-20, the secret will be revealed and Alice can withdraw the collateral.

3 CDS(Credit Default Swap) Based bitLending

bitSmiley's loan model naturally aligns with the characteristics of CDS. Packaging multiple loans with similar risks and characteristics can form a CDS asset portfolio. Loan-based CDS can ensure the security of the guarantor's funds while having the speculative attributes of a derivative.

3.1 CDS process

The process for the CDS product based on bitLending is as follows:

1. Determine the conditions for exercising the CDS based on current market conditions. If the default rate is less than or equal to the estimated default rate, the seller does not have to compensate; if the default rate is higher than the estimate, the seller must pay compensation to the buyer, calculated as the difference between the actual price of each loan and the collateral value.
2. Issue the CDS.
3. Settle the CDS on the expiration date. Considering the potential large number of claims in loans, it’s impractical for a single individual to be the seller of a CDS. Therefore, the platform will use blockchain characteristics to divide a full CDS into NFTs. Given the high volatility of cryptocurrencies, it’s impossible for CDS sellers to bear unlimited risk.

Given the potentially large number of claims in loans, it's impractical for a single individual to be the seller of a CDS. Therefore, the platform will leverage blockchain characteristics to split a full CDS and generate NFTs. Considering the high volatility of cryptocurrencies, unlike traditional finance, CDS sellers cannot assume unlimited risk. Additionally, due to the high level of expertise required for derivative pricing, the platform will set the maximum compensation amount for the CDS based on market conditions.

For example, consider packaging a loan portfolio of 50 collaterals with a total deviation within 0.1 BTC, a total collateral value of 100 BTC, and a maturity of 7 days, where BTC is mortgaged to borrow bitRC-20. The CDS issued will have a maximum payout of 10 BTC, with an estimated default rate of 15%. Users who want to become sellers of this CDS contribute BTC to the CDS contract, sellers will receive mint fees from the buyers. After obtaining the NFT, buyers can also choose to trade on the secondary market before the expiration date. If the final default rate is less than or equal to 15%, buyers will receive compensation, with the payout per NFT being [8 BTC - {(actual default compensation - collateral value) * collateral coefficient %}] /

CDS Sales Volume. If this value is negative or zero, the maximum compensation of 0.1 BTC can be obtained. If there is no default, the seller's guarantee fee will be fully refunded.

3.2 Pricing Of CDS

Fairly pricing Credit Default Swaps (CDS) in different scenarios solely through mathematical models poses challenges in both traditional and decentralized finance. Therefore, bitSmiley will adopt an aggregated bidding method to determine the selling price of CDS. The process is initiated one hour before the CDS sale, with an aggregated bidding round lasting 40 minutes. Both buyers and sellers can freely post prices and quantities. After bidding concludes, the platform calculates a price from valid orders that maximizes current transactions and fulfills all qualifying orders. If the transaction volume is less than the CDS issuance quantity, a second round of aggregated bidding starts 15 minutes before the sale, lasting for 10 minutes. Like the first round, this round calculates a price that maximizes transactions for that bidding round.

Appendix.A-bitRC-20

bitRC-20 is an inscription based fungible token standard. It is an extension of BRC-20 and bitRC-20 is compatible with BRC-20. It supports the following operations: deploy, upgrade, mint, burn and transfer. bitRC-20 supports access control and upgrading. The framework used is on-chain validation and off-chain computation.

To deploy a bitRC-20, one needs to inscribe:

```
{
    "p": "bitRC-20",
    "tick": "bitUSD",
    "op": "deploy",
    "init": "21000000",  // the initial token amount
    "dec": 18,
    "v": 1,              // the version number
    "scripts": {
        "mint": "hash256 or empty",
        "burn": "hash256 or empty",
        "transfer": "hash256 or empty"
    }
}
```

The scripts are extra validation required to perform off-chain in the indexer. The scripts need to be disclosed by the deployer. If any of the fields are empty or null, it implies no extra validations required. Only the deployer can perform upgrade operation, with every upgrade operation, the version number increases by 1.

The data for mint is as follows:

```
{
    "p": "bitRC-20",
    "tick": "bitUSD",
    "op": "mint",
    "amt": "10000",
    "n": 10,
    "p": "hex encoded bytes"
}
```

Aside from the usual fields, one needs to attach "p" for "proof" and "n" for "nonce" fields. The "p" is the proof of mint for the off-chain indexer that tracks the state. If the token does not have a "mint" script, "p" should be empty. Nonce is tracked per address.

The data for burn is as follows:

```
{
    "p": "bitRC-20",
    "tick": "bitUSD",
    "op": "burn",
    "amt": "10000",
    "n": 1,
    "p": "hex encoded bytes"
}
```

It shares the same idea as mint.

The data for transfer is similar:

```
{
    "p": "bitRC-20",
    "tick": "bitUSD",
    "op": "transfer",
    "to: "address",
    "amt": "10000",
    "n": 1,
    "p": "hex encoded bytes"
}
```

If the deployer needs to perform upgrade operation, one can only upgrade the scripts. The payload is as follows:

```
{
    "p": "bitRC-20",
    "tick": "bitUSD",
    "op": "upgrade",
    "v": 2,              // the version number
    "scripts": {
        "mint": "hash256 or empty",
        "burn": "hash256 or empty",
        "transfer": "hash256 or empty"
    }
}
```