BitSmiley Lending Protocol White Paper (Initial Draft)

BitSmiley Team

December 7, 2023

1 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.

2 Decentralized Stablecoin Protocol

A stablecoin protocol on the Bitcoin blockchain is crucial due to the 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.

2.1 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 attibutes of bitUSD is 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.

2.2 Process Flow

The mint flow is shown below:

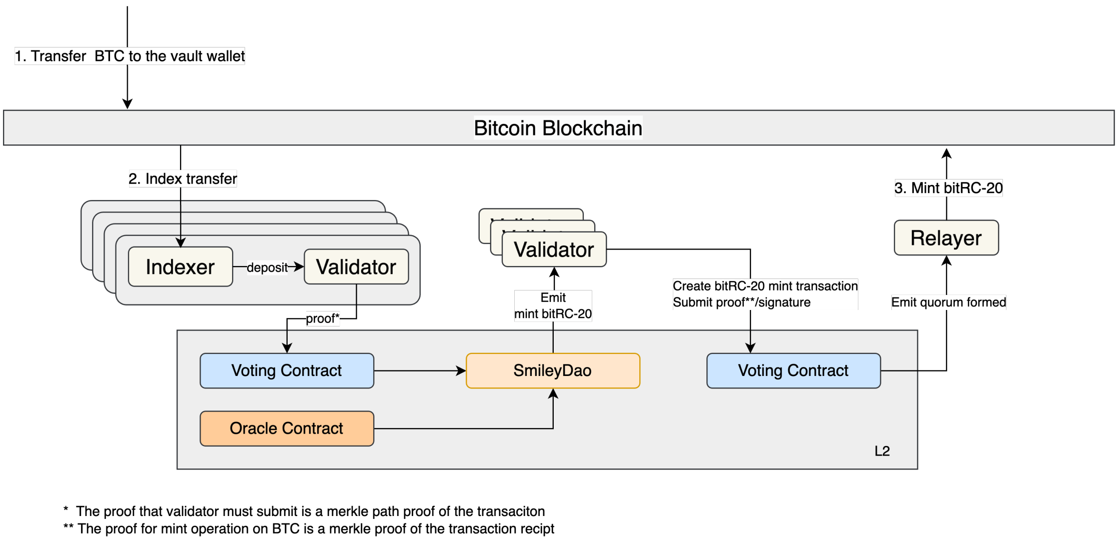

Figure 1: Mint bitUSD flow

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

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.

2.3 Liquidation Design

To ensure that each bitUSD is adequately backed by collateral, any Treasury with the "Collateral to Debt Ratio" falling below a predefined threshold will trigger the liquidation process. The liquidation formula is as follows:

$$LiquidationPrice = (bitUSDGenerated * LiquidationRatio)/QuantityOfCollateral$$

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,

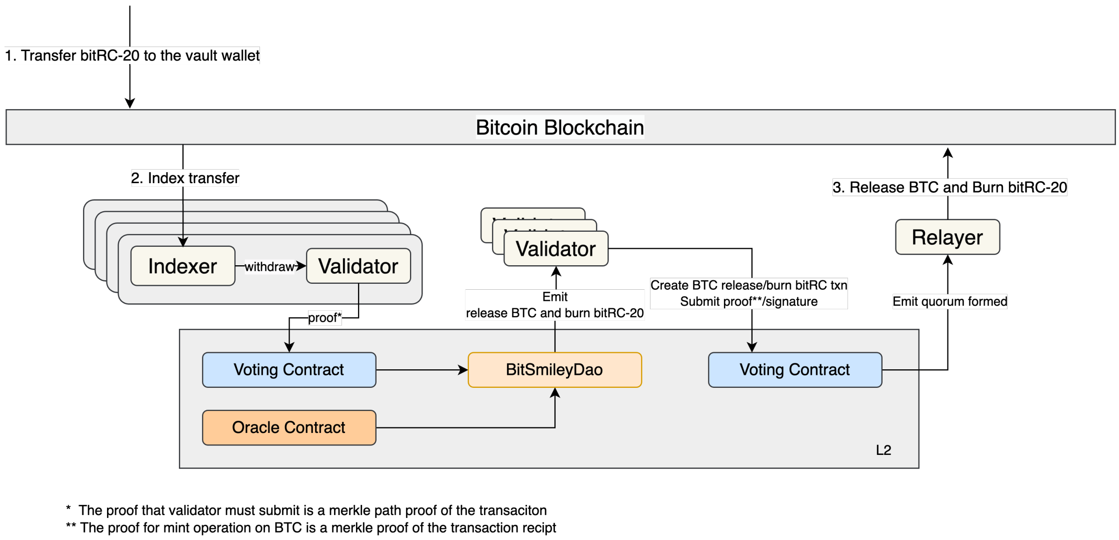

Figure 2: Withdraw bitUSD flow

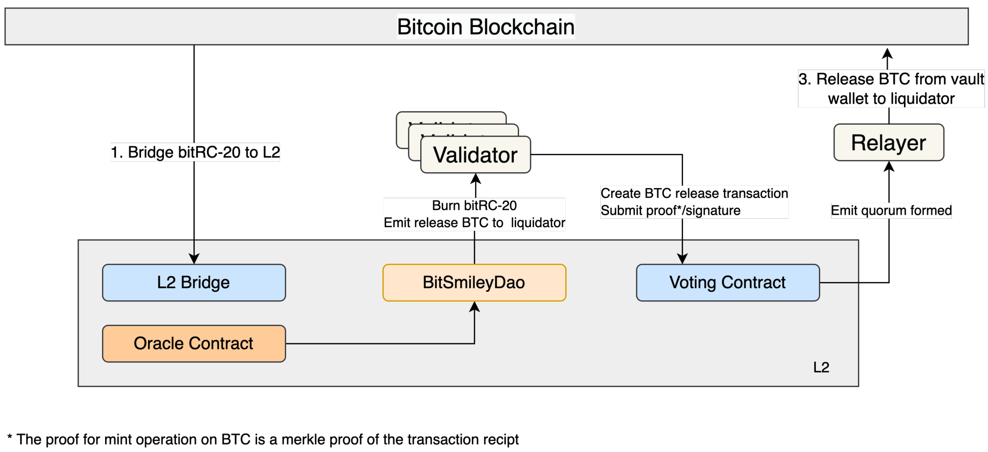

Figure 3: Liquidate bitUSD flow

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

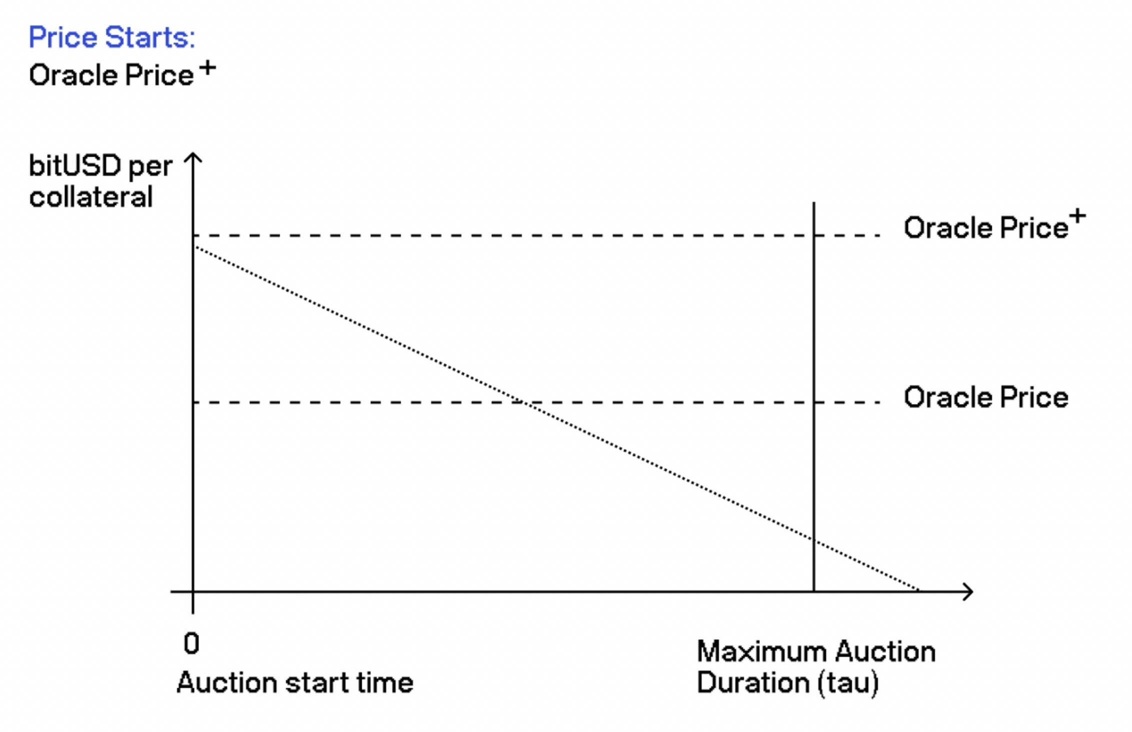

Figure 4: Auction Design

2.4 Auction Design

The auction mechanism in BitSmiley primarily encompasses collateral auction, surplus auction, and debt auction, with the collateral auction already explained in the previous sectio.

2.4.1 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's community.

2.4.2 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.

2.5 BitSmiley 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 has the ability to 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 exceeding 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.

3 Borrow And Repay

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

- Borrowers receive the token he/she wish to borrow

- Borrowers' collateral is locked during the lending period

- Borrowers' collateral will be released once borrowers repay the token

Do note that the collateral provided by borrowers must be approved by BitSmiley. To simplify the notations, imaging 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 BitSmiley with certain amount of bitRC-20 he is willing to lend, also he sets the expected interest rate.
- Alice requests borrowing from BitSmiley, BitSmiley matches Bob's listing with Alice. Alice agrees with the interest rate and duration. Once matched, the expected repay block number with principle plus interest will be calculated and confirmed.
- BitSmiley matches Alice and Bob, creates a multisig wallet.
- BitSmiley will generate fund transfer transactions for Alice and Bob to the multisig wallet and also a lending transaction for the multisig wallet and will ask both Alice and Bob to sign.
- Bob and Alice executes the fund transfers transactions.
- BitSmiley 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 receive her BTC atomically.

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

4 Fund Protection

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 relies solely on Unspent Transaction Outputs (UTXO) for transactions, lacking 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, either the price of the borrowed bitRC-20 increases drastically, or the price of BTC drops significantly. For 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 insurance certain amount of BTC so that when Alice does not repay, insurance will completely refund Bob. The sequence diagram is as follows:

The process is as follows:

- BitSmiley matches Alice, Bob and Charlie, creates a multisig wallet.
- Bob and Alice transfers tokens to mutisig wallet.
- Bob and Alice transfers insurance fees to mutisig wallet.
- BitSmiley broadcasts the lending transaction.

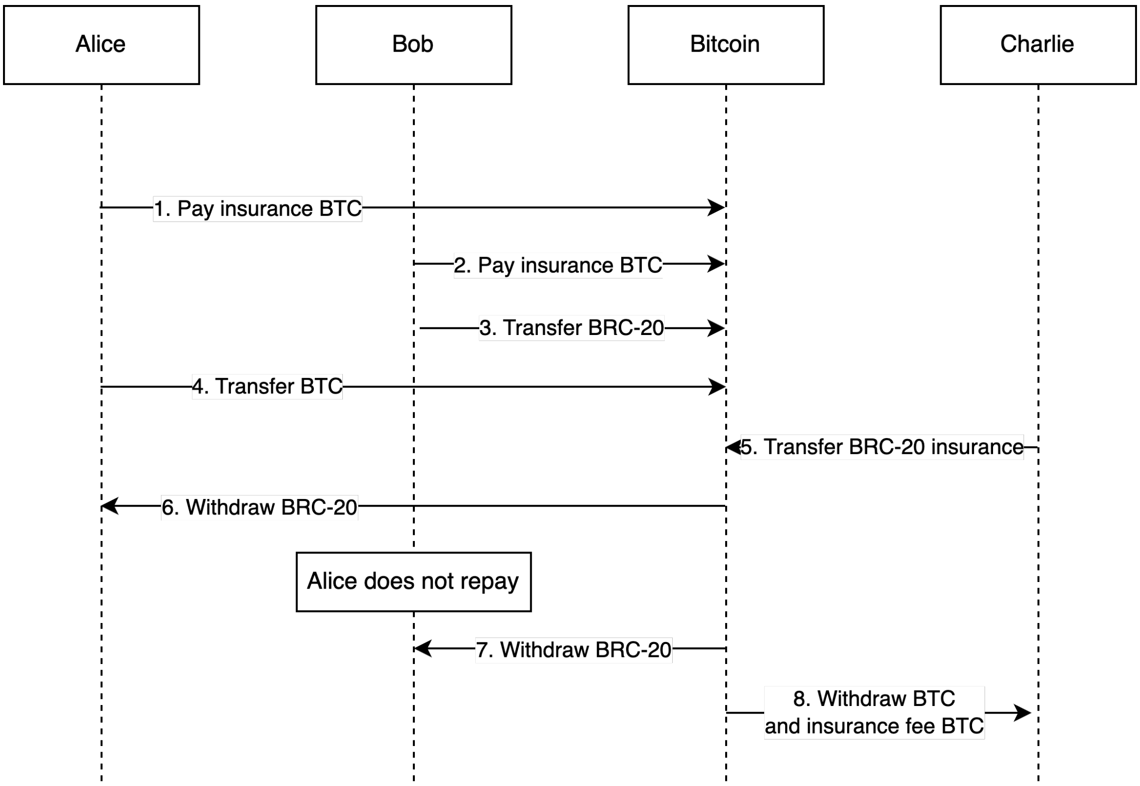

Figure 5: Insurance flow

- 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, Charlie will receive the collateral.

4.1 Pricing Design

Alice and Bob need to pay for the insurance if Bob requests for it. The price for Alice and Bob to pay 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 tail historical data for BTC and bitRC-20 tokens, analyzing the frequency, duration, and magnitude of extreme fluctuations in these data.
2. Establish a Tail Risk Model: Consider both Peak Over Threshold (POT) and Block Maxima methods.
3. Quantify Risk Indicators: Based on historical data analysis, estimate the probability of market fluctuations for two potential loss scenarios for the insurer (BTC price decreases or bitRC-20 price increases), using Tail Value at Risk (TVaR) to measure the size of extreme losses. The TVaR calculation formula is:

$$TVaR_\alpha = \frac{1}{1-\alpha}\int_{VaR_\alpha}^{\infty} x\,dF(x)$$

F represents the cumulative distribution function of losses, and VaR is the VaR at confidence level .

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 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 entire transaction risk and determine a reasonable guarantee fee.

4.2 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.

5 Future Work

The future development of BitSmiley will revolve around three aspects: first, the development of derivatives based on lending; second, transitioning from "peer-to-peer" lending to "group-to-group" through merging and splitting methods; third, leveraging the platform's Total Value Locked (TVL) to issue an over-collateralized native stablecoin on the BTC blockchain. Ultimately, the goal is to transform BitSmiley into a one-stop decentralized DeFi platform.

5.1 CDS (Credit Default Swap)

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.

1. Identify multiple loan orders with similar characteristics.
2. Estimate the default probability of borrowers based on historical data from the platform.
3. 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.
4. Issue the CDS.
5. 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 split a full CDS into Nfts. Given the high volatility of cryptocurrencies, it' s impossible for CDS sellers to bear unlimited risk.

5.1.1 Pricing Design

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.

5.2 Loan Splitting and Order Merging Function

The splitting and merging of loan orders will be key to increasing the capital utilization efficiency of BitSmiley and transitioning the platform from "peer-to-peer" to "group-to-group" lending. Consider a scenario where there are five requests to collateralize 1 BTC for 1000 bitRC-20 and one lender wanting to lend 5000 bitRC-20. The platform can merge the five loan orders and match them with the lender's order. Once this matching mechanism is established, it will greatly improve the platform's capital utilization efficiency and TVL. This "group-to-group" mechanism will offer capital efficiency not inferior to Ethereum's lending protocols and will be able to satisfy users' financial needs with greater precision.

6 Appendix.A

The initial version of the transaction requires BitSmiley generating a secret exclusively for Bob and Bob should not disclose to anyone until Alice has repayed. The following inputs and outputs from the multisig wallet are shown below:

- 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.

7 Appendix.B - 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 needs 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 offchain 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"
    }
}
```