BITRES SYSTEM DESIGN DOCUMENT: FUNCTIONAL ARCHITECTURE AND WEAK GOVERNANCE

Bitres Labs
https://github.com/bitres-labs

December 16, 2025

1 Document Objectives

This document is written based on the Bitres whitepaper: "Bitres: A Decentralized Stablecoin System Collateralized by Bitcoin". The document elaborates on the overall architecture and design objectives of Bitres, as well as the economics of three tokens, including the stablecoin BTD, bond token BTB, and governance token BRS. The document focuses on token supply, price pegging, collateral safety thresholds, and incentive parameters, aiming to provide a requirements specification consistent with the whitepaper for subsequent economic analysis and implementation iterations, without relying on any specific implementation's function or variable names.

1.1 Design Philosophy: Functional and Weak Governance

The system adopts a design philosophy of **functional programming** and **weak governance**, aiming to achieve immutability and predictability similar to Bitcoin:

- **Functional Architecture**: Core business logic is abstracted into pure function libraries, with minimal contract state. Price calculation, deviation validation, IUSD adjustment, and other algorithms are solidified as mathematical formulas, independent of mutable state, ensuring behavioral determinism.
- **Weak Governance Principle**: The system hard-codes the vast majority of parameters as immutable constants at launch, retaining only minimal governance capability for a very small number of necessary parameters. Governable parameters must satisfy four strict security mechanisms: boundary checks, cooldown periods, whitelist restrictions, and event logging.
- **Parameter Classification System**:
  - *Static Constants* (constant): Mathematical constants, precision, time parameters, etc., determined at compile time
  - *Immutables* (immutable): Core addresses, oracle IDs, inflation parameters, etc., set at deployment and never changeable
  - *Governable Parameters* (governable): A very small number of parameters requiring flexible adjustment, subject to strict constraints

This design ensures the system behaves as close to Bitcoin's immutability as possible while retaining minimal flexibility to handle extreme situations.

2 Overall Structure Overview

The system contains a custody module, a minting and redemption module, a parameter configuration module, plus two incentive modules for interest rates and mining.

- **Collateral Custody Module**: Stores all WBTC collateral and manages the BTD and BRS inventory accumulated within the system;
- **Minting and Redemption Module**: Responsible for BTC $\leftrightarrow$ BTD exchange, also issues new bond token BTB when collateral is insufficient, and can use governance token BRS stored in the custody module for compensation;

- **Configuration and Oracle Module**: Maintains asset addresses, decentralized price pools, off-chain price sources, interest rates, and price threshold parameters;
- **Interest Rate Control Module**: Updates the annualized yield rates for BTD and BTB staking, following the federal funds rate and secondary market price respectively;
- **Mining Module**: Releases governance tokens according to a four-year halving system and distributes rewards to the ecosystem fund, team, and liquidity providers.

3 Price Pegging and Collateral Ratio

3.1 Target Unit of Account

The stablecoin BTD maintains a peg to "Ideal USD" (IUSD):

$$1\,\text{BTD} = 1\,\text{IUSD} = \frac{\text{PCE}_n}{\text{PCE}_0 \times 1.02^{n/12}}\,\text{USD}. \tag{1}$$

IUSD is initially priced at 1 USD, with a target annual inflation rate of 2%, i.e., monthly inflation rate $\approx 0.165\%$. The initial month's Personal Consumption Expenditure price index is PCE_0, and the current month's Personal Consumption Expenditure price index is PCE_n, where n represents the nth month after the initial month. When PCE data is higher than the target inflation, IUSD will depreciate accordingly, and vice versa, thereby achieving a long-term monetary policy of 2% annual inflation.

3.1.1 Inflation Parameter Solidification

The 2% annual inflation rate of IUSD is the system's core monetary policy commitment, fully solidified at deployment:

- **Immutable Constants**: The annual inflation rate (2% = 0.02) and monthly growth factor ($1.02^{1/12} \approx 1.00165158130192017 4$) are set as immutable parameters in the constructor, never modifiable after deployment.
- **PCE Oracle Solidification**: The data source address and precision configuration for the US PCE (Personal Consumption Expenditure) price index are solidified at deployment, preventing data source tampering.
- **Functional Adjustment Factor Calculation**: The calculation logic for the IUSD adjustment factor is encapsulated as a pure function library, deterministically calculated based on on-chain PCE data and solidified inflation targets through mathematical formulas, independent of mutable state.
- **Emergency Override Mechanism**: Only in extreme situations such as PCE oracle failure, authorized addresses are allowed to manually override the IUSD value, but subject to strict restrictions: deviation not exceeding 5%, interval of at least 7 days, and complete event log recording.

This design ensures that IUSD's monetary policy is as predictable and immutable as Bitcoin's fixed supply.

3.2 Price Sources and Validation

BTC price references both:

1. Off-chain oracle (using Chainlink BTC/USD feed as benchmark);
2. Decentralized exchange pool (WBTC/USDC trading pair).

If the deviation between the two exceeds 1%, the minting process will terminate directly. The prices of bond and governance tokens are also converted to USD through their respective DEX trading pairs for calculating compensation, interest rates, and buyback amounts.

3.2.1 Oracle Parameter Solidification Strategy

To prevent price manipulation, oracle configuration is fully solidified at deployment:

- **Immutable Parameters**: Pyth price ID, Redstone data source ID, precision configuration, TWAP enable flag, etc., are never changeable after being set in the constructor.
- **Deviation Threshold Governance**: Price deviation tolerance is governable, but can only be *unidirectionally tightened* (from 1% to stricter values), and is constrained by cooldown period (1 day) and range limits (0.5%–5%).
- **Functional Price Aggregation**: Price calculation logic is encapsulated as a pure function library, including multi-source median calculation, deviation validation, etc., independent of mutable state, ensuring the algorithm is transparent and immutable.

3.3 Collateral Ratio Definition

The system measures security boundaries using Collateral Ratio (CR):

$$\text{CR} = \frac{\text{BTC Holdings} \times P_{\text{BTC}}}{\text{BTD Equivalent Total Amount} \times P_{\text{IUSD}}}. \tag{2}$$

Where **BTD Equivalent Total Amount** is defined as:

$$\text{BTD Equivalent Total Amount} = \text{BTD Supply} + \text{All stBTD converted to BTD Amount.} \tag{3}$$

P_{BTC} represents the price of BTC, and P_{IUSD} represents the price of IUSD. When CR $\geq 100\%$, the system is fully backed by BTC; when CR $< 100\%$, the gap is absorbed through newly issued BTB and treasury BRS.

4 BTD: Stablecoin

4.1 Token Overview

Name	Bitcoin Dollar
Symbol	BTD
Token Standard	ERC20, extended to support upgradability, burnable, pausable, Permit, custody and blacklist control
Total Supply	Unlimited, dynamically adjusted based on collateral demand and interest distribution
Precision	18 decimals
Production Mechanism	Issued by the minting and redemption module when WBTC is collateralized, and minted when the staking interest module distributes interest
Issuer	Bitres
Collateral	WBTC
Use Case	Stable settlement asset pegged to IUSD, used for payments, liquidation, and intra-system collateral

4.2 Supply and Fees

After a user deposits 1 WBTC, Q_{BTD} stablecoins are minted according to the validated price:

$$Q_{\text{BTD}} = \frac{P_{\text{BTC}}}{P_{\text{IUSD}}}. \tag{4}$$

$$\text{Fee} = Q_{\text{BTD}} \times 1\%. \tag{5}$$

The fee is additionally minted and not collected from the user, and will be fully transferred to treasury inventory. BTD has no total supply cap; its supply is entirely driven by collateral and interest rate policy.

4.3 Redemption and Gap Handling

When a user deposits 1 BTD into the system for redemption, the user's BTD will first be burned, then assets are returned based on the current collateral ratio:

- **CR** $\geq 100\%$: All converted to Q_{BTC} WBTC and returned:

$$Q_{\text{BTC}} = \frac{P_{\text{IUSD}}}{P_{\text{BTC}}}. \tag{6}$$

$$\tag{7}$$

- **CR** $< 100\%$:
  1. First return x WBTC,

$$x = \frac{\text{BTC Holdings}}{\text{BTD Equivalent Total Amount}}. \tag{8}$$

  2. Then newly issue y BTB,

$$y = \frac{(1 - \text{CR}) \times P_{\text{IUSD}}}{max(P_{\text{BTB}}, P_{\text{BTB,min}})}. \tag{9}$$

  Where P_{BTB} is the market price of BTB, and $P_{\text{BTB,min}}$ is the system-specified minimum price of BTB.
  3. If the market price P_{BTB} is below the bond floor price $P_{\text{BTB,min}}$, the difference is compensated with z BRS,

$$z = \frac{(1 - \text{CR}) \times P_{\text{IUSD}} \times (P_{\text{BTB,min}} - P_{\text{BTB}})}{P_{\text{BTB,min}} \times P_{\text{BRS}}}. \tag{10}$$

The BRS used for compensation comes from treasury inventory and is not newly minted. Thus, each 1 BTD redemption request is split into "immediately returned BTC + newly minted BTB + final backstop BRS". In the initial setup, the system defaults to $P_{\text{BTB,min}} = 0.5$ BTD.

4.4 Staking Interest

BTD can earn deposit interest in the staking pool:

- The annualized rate is updated by the interest rate feed or directly by governance, defaulting to reference the upper limit of the US federal funds target range;
- Interest accumulates per second and is paid to stakers through BTD minting;
- The operator can levy an additional fee rate on interest income (default 10%, adjustable by governance).

5 BTB: Bond Token

5.1 Token Overview

Name	Bitcoin Bond
Symbol	BTB
Token Standard	ERC20, extended to support upgradability, burnable, pausable, custody and blacklist control
Total Supply	Unlimited, dynamically changes based on collateral gap and interest rate incentives
Precision	18 decimals
Production Mechanism	Automatically minted during redemption processes when collateral ratio is insufficient, and minted when the staking interest module distributes interest
Issuer	Bitres
Collateral	BTD
Use Case	Deferred payment debt certificate, stabilizes collateral ratio and recovers system risk premium

5.2 Issuance and Pricing

BTB represents the system's deferred payment promise to stablecoin holders:

- When the collateral ratio is below 100%, bonds are minted at the real-time IUSD/BTB price during the redemption process;
- If the market price falls below 0.5 BTD, the system still issues BTB at the 0.5 BTD face value and compensates the discount with BRS, preventing losses from being further amplified due to insufficient market depth;
- Outside of redemption scenarios, BTB can only be minted for staking interest in the staking pool.

5.3 Redemption Conditions

When the collateral ratio recovers to above 100%, bond holders can exchange stablecoins at a rate of 1 BTB $=$ 1 BTD:

$$\text{Redeemable BTD Upper Limit} = (\text{CR} - 1) \times \text{BTD Equivalent Total Amount}. \tag{11}$$

Redemption burns the corresponding BTB until the system returns to exactly 100% collateralization.

5.4 Staking Interest Rate Policy

The annualized rate for the bond staking pool is adjusted at most once per day, following this interval strategy:

BTB Price (in BTD)	Intraday Price Trend	Policy Action
< 0.99	Falling	Increase rate
< 0.99	Rising	Maintain rate
$[0.99, 1.01]$	Any	Maintain rate
> 1.01	Rising	Decrease rate
> 1.01	Falling	Maintain rate

Adjustment Magnitude Algorithm: When rate adjustment is needed, the adjustment magnitude equals the daily price change of BTB (expressed in basis points). Specifically:

- When price < 0.99 and falling, rate increase magnitude = daily decline;
- When price > 1.01 and rising, rate decrease magnitude = daily increase.

For example, if BTB price falls from 1.00 BTD to 0.95 BTD (5% daily decline), the annualized rate will increase by 5 percentage points (500 basis points). This dynamic adjustment mechanism ensures the rate can promptly respond to market price signals, effectively regulating bond supply and demand.

The system defaults the maximum annualized rate for bond tokens to 20%, with the rate cap adjustable by the governance module. Interest is distributed by minting BTB.

6 BRS: Governance Token

6.1 Token Overview

Name	Bitcoin Reserve System Token
Symbol	BRS
Token Standard	ERC20, extended to support upgradability, burnable, pausable, Permit, custody, blacklist and vote accounting
Total Supply	Capped at 2.1 billion, four-year halving mining mechanism
Precision	18 decimals
Production Mechanism	Issued by mining module with four-year halving
Issuer	Bitres
Collateral	Treasury net assets after deducting total BTD and BTB liabilities
Use Case	Governance voting, risk buffer and ecosystem incentives, supporting treasury buybacks and system upgrades

6.2 Mining Mechanism

BRS is used for governance voting, compensation in extreme situations, and ecosystem incentives. Core production parameters are as follows:

- **Total Supply Cap:** 2.1×10^9 tokens;
- **Halving Cycle:** Every 4 years;
- **First Cycle Release:** 1.05×10^9 tokens, corresponding to initial production of approximately 8.33 tokens/second;
- **Fund Allocation:** By default, three addresses receive Treasury 20%, Foundation 10%, Team 10% of new production (totaling 40%, adjustable by governance), with the remaining 60% used to reward stakers.

When cumulative issuance reaches the cap, mining rewards terminate.

6.2.1 Initial Mining Pool Configuration

The 60% portion of BRS mining rewards distributed to stakers is implemented through multiple mining pools to incentivize different types of liquidity provision and token staking behavior. Initial mining pool configuration is as follows:

1. LP Liquidity Mining Pools (45%):

- **BRS/BTD LP Pool:** 15% allocation weight, incentivizes BRS and BTD liquidity pairing
- **BTD/USDC LP Pool:** 15% allocation weight, provides mainstream on-ramp for stablecoin
- **BTB/BTD LP Pool:** 15% allocation weight, promotes bond token and stablecoin swap liquidity

2. Single Token Staking Mining Pools (15%):

- **USDC Pool:** 1% allocation weight, provides additional yield for USDC holders
- **USDT Pool:** 1% allocation weight, supports another mainstream stablecoin
- **WBTC Pool:** 1% allocation weight, encourages Bitcoin asset participation
- **ETH Pool:** 1% allocation weight, supports Ethereum asset participation
- **stBTD Pool:** 3% allocation weight, users deposit BTD into ERC4626 vault to receive stBTD share tokens, staking stBTD yields dual returns (BTD interest through automatic share price growth + BRS mining rewards)
- **stBTB Pool:** 3% allocation weight, users deposit BTB into ERC4626 vault to receive stBTB share tokens, staking stBTB yields dual returns (BTB interest through automatic share price growth + BRS mining rewards)
- **BRS Pool:** 5% allocation weight, encourages long-term holding of governance tokens

stToken Dual Yield Mechanism Explanation:

BTD and BTB implement dual yield mechanisms using the ERC4626 tokenized vault standard. Users participate through the following process:

1. **Deposit Phase:** Users deposit BTD/BTB into the corresponding vault contract and receive stBTD/stBTB share tokens at the current exchange rate;
2. **Interest Accumulation:** The stToken share price automatically grows over time, reflecting accumulated interest earnings. Interest earnings come from:
    - stBTD: Deposit interest following the federal funds rate

- stBTB: Bond interest dynamically adjusted based on market price

3. **Further Staking:** Users can stake stBTD/stBTB into the BRS mining pool (FarmingPool) to earn additional BRS mining rewards;
4. **Dual Yield:** Ultimately achieving "one deposit, dual yields":
   - BTD → stBTD → BTD interest (auto-compounding) + BRS mining rewards
   - BTB → stBTB → BTB interest (auto-compounding) + BRS mining rewards

The advantage of stToken design: Funds only need to be transferred once, interest auto-compounds through share price appreciation, and users can redeem principal plus interest at any time without manual claiming.

Initial Mining Pool Allocation Weight Table:

Pool ID	Type and Token	Allocation Weight	Percentage
0	BRS/BTD LP	15	15%
1	BTD/USDC LP	15	15%
2	BTB/BTD LP	15	15%
3	USDC Single Token	1	1%
4	USDT Single Token	1	1%
5	WBTC Single Token	1	1%
6	ETH Single Token	1	1%
7	stBTD Single Token Staking	3	3%
8	stBTB Single Token Staking	3	3%
9	BRS Single Token	5	5%
Total (Stakers)		**60**	**60%**
Treasury		–	**20%**
Foundation		–	**10%**
Team		–	**10%**
Total		–	**100%**

This configuration can be dynamically adjusted through the governance module to adapt to market changes and ecosystem development needs.

6.3 Price Support and Buyback

The treasury can use accumulated BTD to buy back BRS in the secondary market, with purchased BRS retained in the treasury to form BRS inventory. This process does not rely on token burn functions but adjusts circulating supply through treasury management. Treasury BRS inventory mainly comes from:

- **Mining Allocation:** 20% of mining output is automatically allocated to the treasury address;
- **Fee Buyback:** Using BTD fees collected by the system to buy back BRS in the market.

Treasury BRS inventory is used both for price support and as the funding source for BRS compensation functionality.

6.4 Compensation Function

When collateral is insufficient and bond prices are below face value, redeemers receive additional BRS compensation. The compensation amount depends on the degree of bond price deviation and the BRS/BTD exchange rate, serving as the last line of defense in the BTD price pegging mechanism's three-layer defense system.

Important Mechanism Explanation:

- **Compensation Source:** BRS compensation is not from newly minted tokens, but transferred from treasury BRS inventory, ensuring the 2.1 billion total supply cap is not breached;
- **Natural Limitation:** Compensation is constrained by available treasury balance. When treasury BRS inventory is insufficient, compensation amount automatically reduces to inventory balance; when inventory is depleted, compensation stops;
- **Risk Buffer:** The treasury's BRS inventory represents the system's extreme risk tolerance capacity. During normal periods, the treasury accumulates BRS; during crisis periods, BRS is released for compensation, forming a natural risk buffer mechanism.

7 Interest

7.1 stToken Vault Mechanism

The system uses the ERC4626 tokenized vault standard to implement interest accumulation and distribution. User participation process is as follows:

7.1.1 Deposit and Share Minting

- Users deposit BTD or BTB into the corresponding vault contract (stBTD Vault or stBTB Vault);
- The vault calculates and mints the corresponding amount of share tokens (stBTD or stBTB) at the current exchange rate and returns them to the user;
- Initial exchange rate is 1:1, i.e., first deposit of 1 BTD yields 1 stBTD share.

7.1.2 Interest Accumulation Mechanism

Interest is not implemented by minting additional tokens to user accounts, but automatically accumulated through share price appreciation:

- **stBTD Vault:** Interest Pool accumulates BTD interest per block, interest enters vault total assets;
- **stBTB Vault:** Interest Pool accumulates BTB interest per block, interest enters vault total assets;
- **Share Price Appreciation:** Total assets increase but total shares remain unchanged, causing each share to correspond to more assets, achieving auto-compounding;
- **Claim Method:** When users redeem stToken, they receive principal plus interest in BTD/BTB at the current exchange rate.

Example: User deposits 100 BTD and receives 100 stBTD. After one year (assuming 5% annual rate), redemption yields 105 BTD, while stBTD quantity remains 100.

7.1.3 Interest Rates

- **BTD Rate:** Synchronized with the federal funds rate, with risk premium added if necessary, adjustable by community governance; additionally, when depositors collect interest, an extra 10% BTD is minted as system fee and deposited into the treasury;
- **BTB Rate:** Dynamically adjusted through price intervals, encouraging buying when bonds are at discount and exiting when at premium;
- Both interest pools' rewards mint new tokens, which enter vault total assets and are distributed to stakers through share price appreciation.

7.1.4 Further Staking for Dual Yields

After obtaining stBTD/stBTB, users can further stake them into the BRS mining pool (FarmingPool):

- Stake stBTD $\rightarrow$ Continue enjoying BTD interest (through stBTD price appreciation) + Earn BRS mining rewards;
- Stake stBTB $\rightarrow$ Continue enjoying BTB interest (through stBTB price appreciation) + Earn BRS mining rewards;
- Dual yields require no additional operations, stToken share price grows automatically, BRS rewards can be claimed at any time.

This mechanism ensures users' funds only need to be transferred once to simultaneously receive interest earnings and mining rewards, greatly improving capital efficiency and user experience.

7.2 Operation Delay and Security

To prevent short-term interest rate attacks or reentrancy, there must be at least 5 blocks between reward withdrawals. Staking, withdrawal, and claiming operations are all protected by reentrancy guards and ownership controls, combined with pause, custody, and blacklist permissions in token contracts, enhancing compliance and security.

8 Risk Hierarchy and Return Comparison

Token	Risk Level	Primary Backing Assets	Return Source
BTD	Low	BTC, BTB, BRS	Deposit rate (linked to FFR)
BTB	Medium	BTD reserves	Bond rate (dynamically adjusted)
BRS	High	System residual equity	Mining output and treasury buyback

BTD enjoys the highest priority redemption order; BTB has priority redemption after collateral ratio recovery; BRS, as the residual claimant, bears the greatest risk but gains appreciation space through governance rights, fee buybacks, and ongoing mining incentives.

9 Governance

After the official genesis of the BRS system, there will be a beta testing period. During this period, to quickly fix bugs and add necessary features based on market demand, governance rights for the entire system will be controlled by the development team. Once the system develops to a certain stage and enters a stable period, the development team will transfer governance rights to the community. From then on, the system adopts an on-chain governance mechanism based on BRS tokens, ensuring the community can democratically participate in system parameter adjustments and protocol upgrade decisions. The governance architecture follows the OpenZeppelin Governor standard, providing transparent and auditable proposal and voting processes.

9.1 Weak Governance Design Principles

The system adopts a "weak governance" style, hard-coding the vast majority of parameters at launch to make its behavior as close to Bitcoin's immutability as possible. Governance capability is strictly limited to the minimum necessary scope:

9.1.1 Parameter Solidification Strategy

- **Core Constants Fully Solidified** (immutable):
  - All core contract addresses (BTD, BTB, BRS, Treasury, Minter, Config, etc.)
  - All oracle configurations (Pyth/Redstone/Chainlink feed IDs, data source addresses, precision)
  - IUSD inflation parameters (2% annual, monthly growth factor)
  - Core trading pool addresses (BTD/USDC, BTB/BTD, BRS/BTD, WBTC/USDC)
  - BRS total supply cap (2.1 billion) and halving mechanism
- **System Constants** (static constant):
  - Mathematical precision constants (18 decimal base)
  - Time constants (seconds/days/years conversion)
  - Safety ceilings (maximum deviation threshold 5%, minimum deviation threshold 0.5%)

9.1.2 Governable Parameters and Their Constraints

A very small number of parameters retain governance capability, but are strictly constrained by **four security mechanisms**:

1. **Price Deviation Tolerance** (default 1%)
   - *Boundary Check*: Can only be adjusted within 0.5%–5% range
   - *Unidirectional Tightening*: Can only lower the threshold, not raise it
   - *Cooldown Period*: At least 1 day between adjustments
   - *Event Log*: Complete record of each change
2. **BTD Minting Fee** (default 1%)
   - *Range Limit*: 0%–2%
   - *Cooldown Period*: At least 3 days between adjustments
3. **BTB Minimum Price Protection** (default 0.5 BTD)
   - *Range Limit*: 0.3–0.7 BTD
   - *Cooldown Period*: At least 7 days between adjustments
4. **BTB Maximum Annual Rate** (default 20%)
   - *Range Limit*: 10%–50%
   - *Cooldown Period*: At least 7 days between adjustments
5. **Mining Allocation Weights**
   - *Whitelist Restriction*: Can only adjust weights within predefined pool set
   - *Total Conservation*: Sum of all weights always equals 100%
   - *Cooldown Period*: At least 14 days between adjustments

6. Emergency Operations (contract pause, blacklist)

- *Fast Track*: Can be quickly executed during security events, but requires multi-sig confirmation
- *Timelock Exemption*: Pause operations are exempt from timelock, resume operations must go through complete governance process
- *Transparency*: All operations are fully recorded in on-chain events

9.1.3 Anti-Governance Abuse Mechanisms

- **Timelock Delay**: All governance proposals (except emergency pause) must go through at least 2 days of timelock delay before execution
- **Voting Power Snapshot**: Voting weight is determined at proposal creation, preventing flash loan attacks
- **Proposal Threshold**: Creating a proposal requires holding sufficient BRS (initially 250,000 BRS)
- **Quorum**: Proposal passage requires at least 4% of total supply participating in voting
- **Proposal Cancellation Right**: When malicious proposals are discovered, creators or timelock administrators can cancel before execution
- **Whitelist Constraint**: All address-type parameters (such as backup oracles) can only switch within pre-approved whitelists

9.1.4 Design Objectives

Weak governance design pursues the following objectives:

- **Predictability**: Users and developers can rely on the system's core behavior not being changed by governance
- **Immutability**: Key economic parameters (inflation rate, total supply cap) are as unmodifiable as Bitcoin
- **Minimal Trust**: Reduces trust assumptions about governors, lowers governance attack surface
- **Emergency Flexibility**: Retains minimum necessary capability to handle extreme market situations and security events

This design ensures the system is as stable and reliable as the Bitcoin protocol while retaining minimum flexibility to handle unknown risks.

9.2 Governance Token Voting Rights

BRS token integrates ERC20Votes extension, granting holders voting weight:

- Voting weight equals the amount of BRS held (1 BRS = 1 vote);
- Voting power is implemented through delegation mechanism, holders can delegate voting power to themselves or other addresses;
- Voting power snapshot is determined at proposal creation, preventing token transfers during voting from affecting results;
- Supports offline signature voting (Permit feature), reducing participation costs.

9.3 Governance Process

The complete proposal lifecycle includes the following phases:

9.3.1 Proposal Creation

Any address holding sufficient BRS can create a proposal:

- **Proposal Threshold:** Hold at least 250,000 BRS (adjustable by governance);
- **Proposal Content:** Includes target contract address, call data, and detailed description;
- **Proposal Types:** Supports parameter adjustment, protocol upgrade, fund allocation, and any other on-chain operations.

9.3.2 Voting Delay Period

After proposal creation, the system enters a 1-day voting delay period:

- Determines voting power snapshot, preventing temporary token purchases to manipulate voting;
- Gives the community sufficient time to review proposal content;
- Automatically enters voting period after delay period ends.

9.3.3 Voting Period

Voting period lasts 7 days, addresses with voting power can vote:

- **Voting Options:** For, Against, Abstain;
- **Voting Methods:** On-chain transaction or offline signature (EIP-712);
- **Vote Changes:** Can modify choice within voting period after voting.

9.3.4 Quorum and Passing Conditions

Proposal passage requires meeting both:

- **Quorum:** At least 4% of total supply participating in voting (based on supply at snapshot);
- **Majority For:** For votes $>$ Against votes (Abstain votes not counted).

9.3.5 Timelock Queue

Passed proposals are not executed immediately, but enter the timelock queue:

- Proposal queues in timelock contract with minimum delay period (e.g., 2 days);
- During delay period, community can review upcoming operations and cancel through emergency proposal if issues are found;
- Timelock protects system from rapid attacks by malicious proposals.

9.3.6 Proposal Execution

After timelock delay period ends, anyone can trigger proposal execution:

- Executor calls Governor contract's `execute()` function;
- Timelock contract verifies delay period has passed, executes on-chain operations specified in proposal;
- Execution results are recorded in on-chain events, fully transparent and auditable.

9.4 Governance Scope

System parameters and operations adjustable through on-chain governance include (but are not limited to):

Economic Parameters:

- BTD minting fee ratio (default 1%)
- BTB minimum price protection (default 0.5 BTD)
- BTB maximum annual rate cap (default 20%)
- Interest fee ratio (default 10%)
- Price deviation tolerance threshold (default 1%)

Mining Allocation:

- Allocation weights for each mining pool
- Fund allocation ratios (Treasury, Foundation, Team)
- Adding or removing mining pools

Protocol Upgrades:

- Upgrade operations for UUPS upgradeable contracts (must go through timelock)
- Oracle contract address updates
- Addition of new feature modules

Emergency Operations:

- Contract pause/resume (responding to security events)
- Blacklist management (compliance requirements)
- Treasury fund management (buyback, transfer, etc.)

9.5 Governance Security Mechanisms

To ensure governance security, the system employs multi-layer protection measures:

- **Voting Power Snapshot:** Prevents flash loan attacks and temporary voting power purchases;
- **Timelock Delay:** Gives community sufficient time to discover and respond to malicious proposals;
- **Proposal Cancellation:** Proposal creators or timelock administrators can cancel proposals before execution;
- **Emergency Pause:** When serious security issues are discovered, contracts can be paused through fast governance process;
- **Upgradability:** Governor contract itself is upgradeable, supporting iterative optimization of governance mechanisms.

9.6 Governance Parameters Overview

Parameter	Initial Value
Proposal Threshold	1 BRS
Voting Delay	1 day
Voting Period	7 days
Quorum	4% of total supply
Timelock Delay	2 days (configurable)
Voting Method	Simple majority (For $>$ Against)

All governance parameters can be adjusted through governance proposals themselves, enabling system self-evolution and optimization.

10 Summary

Chapters 3–6 of the whitepaper propose a three-layer defense architecture of "BTC collateral + bond adjustment + governance token backstop". The formulas, parameters, and sequence given in this document are fully aligned: BTD minting and redemption follow the pegging principle of $1\,\text{BTD} = 1\,\text{IUSD}$; BTB regulates debt interest rates through price intervals; BRS is generated through a four-year halving mining mechanism while using treasury buybacks to absorb long-term risks.

10.1 Core Advantages of Functional and Weak Governance

The system design follows two core principles to ensure long-term stability and predictability:

Functional Architecture achieves clear separation of algorithms and state:

- Core business logic (price calculation, collateral ratio calculation, IUSD adjustment, interest accumulation, etc.) encapsulated as pure function libraries
- Contract layer state minimized, storing only necessary addresses, balances, and accumulated values
- All calculation logic is deterministic, testable, and auditable
- Gas efficiency optimization: immutable parameter reading costs only 3 gas (vs 2100 gas for storage), saving 99.86%

Weak Governance Principle anchors system behavior to Bitcoin-like immutability:

- **47% of parameters fully solidified**: Core addresses, oracle configurations, IUSD inflation rate, BRS total supply, etc., set as immutable at deployment, never changeable
- **20% are static constants**: Mathematical precision, time conversion, safety ceilings, etc., determined at compile time
- **Only 33% of parameters are governable**: A very small number of parameters like fees, deviation thresholds, rate caps retain governance capability, but are strictly constrained by four security mechanisms (boundary checks, cooldown periods, whitelist restrictions, event logs)
- **Unidirectional Tightening Principle**: Price deviation threshold can only be gradually tightened, not loosened, preventing governance from lowering security standards
- **Timelock Protection**: All governance operations (except emergency pause) must go through at least 2 days of timelock delay, giving the community sufficient reaction time

10.2 Design Objectives Achieved

Through the combination of functional and weak governance, the system achieves:

- **Predictability**: Core economic parameters (2% inflation rate, 2.1 billion BRS total supply) are as immutable as Bitcoin's 21 million cap
- **Transparency**: All algorithms presented in mathematical formula form, behavior is completely deterministic
- **Security**: Attack vectors like price manipulation and inflation rate tampering are completely eliminated
- **Minimal Trust**: Users need not trust that governors won't change core rules
- **Emergency Flexibility**: Retains minimum necessary capability to handle extreme situations (limited parameter adjustment, contract upgrades, emergency pause)

This design philosophy ensures the BRS system maintains the same level of stability and trustworthiness as the Bitcoin protocol while providing decentralized stablecoin services. Through on-chain governance mechanisms, the community can democratically adjust necessary parameters and drive protocol evolution within strict constraints. Therefore, as long as implementations follow the mechanisms and parameters specified in this document, they can remain consistent with the whitepaper and expand into various implementation schemes in the future.

A Technical Appendix

A.1 Functional Architecture and Pure Function Library Layer

The system adopts a functional programming paradigm, abstracting core business logic into pure function libraries, achieving separation of state and computation.

A.1.1 Library Layer Design Principles

- **Pure Functionality**: All library functions do not read or write storage, only compute based on input parameters, ensuring determinism and testability
- **State Independence**: Library functions do not depend on contract state, can be independently tested and formally verified
- **Algorithm Transparency**: All calculation logic presented in mathematical formula form, behavior is completely predictable
- **Gas Optimization**: Library functions compile to inline code or delegatecall, avoiding state access overhead

A.1.2 Core Function Libraries

PriceBlend (Price Aggregation Library):

- `median3(p1, p2, p3)`: Three-number median calculation
- `blendMultiSource(prices[], maxDeviation)`: Multi-source price median aggregation, supports 5+ oracle sources
- `validateSpotAgainstRef(spot, ref, maxBps)`: Spot price vs reference price deviation validation
- `validateAllWithinBounds(prices[], maxBps)`: Comprehensive deviation verification ($O(n^2)$ all price pair checks)

IUSDMath (IUSD Calculation Library):

- `adjustmentFactor(currentPCE, basePCE, monthlyFactor, periods)`: Calculate adjustment factor based on PCE data
- `calculateIUSD(basePCE, currentPCE, targetRate, periods)`: Complete IUSD price calculation
- Boundary checks: PCE value > 0, adjustment factor single-step change $\leq 5\%$

FeedValidation (Oracle Validation Library):

- `readAggregator(feed, maxStaleness)`: Chainlink feed reading and normalization
- `normalizeDecimals(value, fromDecimals, toDecimals)`: Precision conversion
- `checkFreshness(timestamp, maxAge)`: Data freshness verification

OracleMath (Oracle Math Library):

- `normalizeAmount(amount, decimals, targetDecimals)`: General precision normalization
- `deviationWithin(value1, value2, maxBps)`: Deviation percentage calculation and determination

CollateralMath (Collateral Ratio Calculation Library):

- `calculateCR(btcAmount, btcPrice, btdSupply, iusdPrice)`: Collateral ratio calculation
- `redeemAmounts(btdAmount, cr, btcPrice, btbPrice, brsPrice)`: Redemption asset portfolio calculation

MintLogic / RedeemLogic (Minting/Redemption Logic Libraries):

- Encapsulate complete minting and redemption calculation logic
- Separate business logic from state management
- Facilitate unit testing and auditing

InterestMath / RewardMath (Interest/Reward Calculation Libraries):

- `calculateInterest(principal, rate, timeDelta)`: Interest calculation
- `calculateReward(staked, totalStaked, rewardPerSecond, timeDelta)`: Mining reward calculation
- Precise accumulation algorithm based on block time

PrecisionMath (Precision Handling Library):

- Safe multiplication and division at 18-digit precision
- Overflow protection and rounding error control
- Supports conversion between assets of different precisions

A.1.3 Contract Layer Responsibilities

The contract layer is only responsible for:

- **Minimal State Management**: Store necessary addresses, balances, accumulated values, etc.
- **Permission Control**: Verify caller identity and operation permissions
- **Event Logging**: Record state changes and key operations
- **Library Function Calls**: Delegate computation to pure function libraries

A.1.4 Layering Advantages

- **Testability**: Pure function libraries can be independently unit tested without contract deployment
- **Auditability**: Algorithm logic is clear, easy to manually review and formally verify
- **Upgradability**: During contract upgrades, library function logic remains unchanged, reducing risk
- **Gas Efficiency**: Inline functions avoid cross-contract calls, immutable parameters save 99.86% reading cost
- **Composability**: Library functions can be reused across different contracts, maintaining consistency

B System Parameter List

This appendix summarizes all core parameters in the BRS system, categorized by mutability into three types: static constants (constant), immutables (immutable), and governable variables (governable).

B.1 Static Constants (Constant)

Static constants are determined at compile time, written into contract bytecode, and permanently immutable. Primarily stored in the `Constants.sol` library.

B.1.1 Precision Constants

Parameter Name	Value	Description
PRECISION_18	10^{18}	18-digit standard precision
PRECISION_8	10^{8}	8-digit precision (BTC)
PRECISION_6	10^{6}	6-digit precision (USDC/USDT)
SCALE_WBTC_TO_NORM	10^{10}	WBTC→standard precision scale
SCALE_USDC_TO_NORM	10^{12}	USDC→standard precision scale
BPS_BASE	10000	Basis point base (100.00%)
PERCENT_BASE	100	Percentage base

B.1.2 Time Constants

Parameter Name	Value	Description
SECONDS_PER_YEAR	31536000	Seconds per year (365 days)
SECONDS_PER_DAY	86400	Seconds per day
SECONDS_PER_WEEK	604800	Seconds per week
ERA_PERIOD	126144000	BRS halving cycle (4 years)

B.1.3 Inflation Parameter Constants

Parameter Name	Value	Description
ANNUAL_INFLATION_RATE	2×10^{16}	Annual inflation rate 2%
MONTHLY_GROWTH_FACTOR	1001615813019201 74	$1.02^{1/12}$

B.1.4 Supply and Limit Constants

Parameter Name	Value	Description
BRS_MAX_SUPPLY	2.1×10^{27}	BRS maximum supply (2.1 billion)
MIN_BTC_AMOUNT	1	Minimum BTC operation amount
MIN_ETH_AMOUNT	10^{10}	Minimum ETH operation amount
MIN_USD_VALUE	10^{15}	Minimum USD value
MIN_STABLECOIN_18_AMOUNT	10^{15}	18-digit stablecoin minimum amount
MIN_STABLECOIN_6_AMOUNT	1000	6-digit stablecoin minimum amount
MAX_WBTC_AMOUNT	10^{12}	WBTC single transaction maximum
MAX_ETH_AMOUNT	10^{23}	ETH single transaction maximum
MAX_STABLECOIN_18_AMOUNT	10^{27}	18-digit stablecoin maximum amount
MAX_STABLECOIN_6_AMOUNT	10^{15}	6-digit stablecoin maximum amount
MAX_USD_VALUE	10^{27}	Single transaction maximum USD value

B.1.5 Oracle Security Constants

Parameter Name	Value	Description
MAX_DEVIATION_CEILING	500 bps	Price deviation ceiling (5%)
MIN_DEVIATION_FLOOR	50 bps	Price deviation floor (0.5%)
DEVIATION_UPDATE_COOLDOWN	1 day	Deviation adjustment cooldown
PYTH_MAX_STALENESS	60 seconds	Pyth price maximum staleness
PYTH_MAX_CONF_RATIO	100	Pyth confidence threshold (1%)
TWAP_PERIOD	30 minutes	TWAP observation period

B.1.6 Interest Rate Interval Constants (InterestPool)

Parameter Name	Value	Description
LOWER_PRICE_THRESHOLD	0.99 BTD	BTB price lower threshold
UPPER_PRICE_THRESHOLD	1.01 BTD	BTB price upper threshold
CR_MIN	30%	CR minimum threshold
CR_LOW	90%	CR low threshold
CR_HIGH	120%	CR high threshold
CR_MAX	200%	CR maximum threshold
BTD_BASE_RATE	400 bps	BTD base rate (4%)
BTD_MAX_RATE	2000 bps	BTD maximum rate (20%)
BTD_MIN_RATE	0 bps	BTD minimum rate (0%)

B.1.7 Governance Constants

Parameter Name	Value	Description
Voting Delay	1 day	Proposal creation to voting start
Voting Period	1 week	Voting duration
Proposal Threshold	250,000 BRS	Tokens required to create proposal
Quorum	4%	Required voting participation ratio
Timelock Delay	2 days	Proposal passage to execution
SHARE_BASE	100	Fund allocation base

B.1.8 IUSD Manual Override Security Constants

Parameter Name	Value	Description
MIN_OVERRIDE_INTERVAL	7 days	Manual override minimum interval
MAX_MANUAL_DEVIATION	5%	Manual adjustment maximum deviation

B.2 Immutables (Immutable)

Immutables are set through the constructor at contract deployment and never changeable afterward. Storage cost is only 3 gas (vs 2100 gas for storage).

B.2.1 Core Token Addresses (ConfigCore)

Parameter Name	Description
WBTC	Wrapped BTC token address
BTD	Stablecoin BTD token address
BTB	Bond token BTB token address
BRS	Governance token BRS token address
WETH	Wrapped ETH token address
USDC	USDC stablecoin address
USDT	USDT stablecoin address

B.2.2 Core Contract Addresses (ConfigCore)

Parameter Name	Description
TREASURY	Treasury contract address
MINTER	Minting and redemption contract address
PRICE_ORACLE	Price oracle contract address
IDEAL_USD_MANAGER	IUSD manager address
INTEREST_POOL	Interest pool contract address
STAKING_ROUTER	Staking router contract address
FARMING_POOL	Mining pool contract address
ST_BTD	stBTD vault address
ST_BTB	stBTB vault address
VESTING_VAULT	Vesting vault address
GOVERNOR	Governance contract address
TWAP_ORACLE	TWAP oracle address

B.2.3 Price Oracle Data Source Addresses (ConfigCore)

Parameter Name	Description
CHAINLINK_BTC_USD	Chainlink BTC/USD price feed
CHAINLINK_WBTC_BTC	Chainlink WBTC/BTC price feed
PYTH_WBTC	Pyth WBTC price feed
REDSTONE_WBTC	Redstone WBTC price feed

B.2.4 Uniswap V2 Trading Pair Addresses (ConfigCore)

Parameter Name	Description
POOL_WBTC_USDC	WBTC-USDC trading pair (for WBTC price)
POOL_BTD_USDC	BTD-USDC trading pair (for BTD price)
POOL_BTB_BTD	BTB-BTD trading pair (for BTB price)
POOL_BRS_BTD	BRS-BTD trading pair (for BRS price)

B.2.5 Oracle Configuration Parameters (PriceOracle)

Parameter Name	Description
pythWbtcPriceId	Pyth WBTC price ID (bytes32)
redstoneWbtcDataFeedId	Redstone WBTC data feed ID (bytes32)
redstoneWbtcDecimals	Redstone precision configuration
useTWAPDefault	TWAP default enable status

B.2.6 Mining Parameters (FarmingPool)

Parameter Name	Description
rewardToken	BRS reward token address
startTime	Mining start timestamp

B.2.7 Vesting Parameters (VestingVault)

Parameter Name	Type	Description
token	address	Vesting token (BRS)
foundationSchedule.beneficiary	address	Foundation beneficiary
foundationSchedule.cliff	uint64	Foundation lock period
foundationSchedule.duration	uint64	Foundation release cycle
foundationSchedule.total	uint256	Foundation total allocation
teamSchedule.beneficiary	address	Team beneficiary
teamSchedule.cliff	uint64	Team lock period
teamSchedule.duration	uint64	Team release cycle
teamSchedule.total	uint256	Team total allocation

B.3 Governable Variables (Governable)

Governable variables can be adjusted at runtime through governance mechanisms, but are strictly constrained: boundary checks, cooldown periods, whitelist restrictions, and event logs.

B.3.1 Fee Parameters (ConfigGov)

Parameter Name	Default	Range	Description
MINT_FEE_BP	100 bps	0–200 bps	Minting fee (1%)
REDEEM_FEE_BP	0 bps	0–200 bps	Redemption fee
INTEREST_FEE_BP	1000 bps	0–2000 bps	Interest fee (10%)

B.3.2 Price Protection Parameters (ConfigGov)

Parameter Name	Default	Range	Description
MIN_BTB_PRICE	0.5 BTD	0.3–0.7 BTD	BTB minimum price protection
MAX_BTB_RATE	2000 bps	1000–5000 bps	BTB maximum rate (20%)
PCE_MAX_DEVIATION	2%	0–10%	PCE maximum deviation rate

B.3.3 Address Parameters (ConfigGov)

Parameter Name	Description
PCE_FEED	Chainlink PCE oracle address

B.3.4 Oracle Runtime Parameters (PriceOracle)

Parameter Name	Default	Constraint	Description
maxDeviationBps	100 bps	Can only tighten	Price deviation tolerance
useTWAP	true	Boolean	TWAP mode switch
twapOracle	Address	Whitelist	TWAP oracle address

B.3.5 Interest Rate Parameters (InterestPool)

Parameter Name	Default	Description
btdPool.annualRateBps	400 bps	BTD annual rate, dynamically adjusted based on CR
btbPool.annualRateBps	400 bps	BTB annual rate, dynamically adjusted based on price
rateOracle	Address	Rate oracle address

B.3.6 Mining Allocation Parameters (FarmingPool)

Parameter Name	Description
fundAddrs[]	Fund address array (Treasury, Foundation, Team)
fundShares[]	Fund allocation ratios (base 100, sum $\leq$100)
poolInfo[].allocPoint	Allocation weight for each pool
totalAllocPoint	Total allocation weight

B.3.7 Governance Runtime Parameters

Parameter Name	Description
Voting Delay	Adjustable via governance proposal
Voting Period	Adjustable via governance proposal
Proposal Threshold	Adjustable via governance proposal
Quorum Ratio	Adjustable via governance proposal

B.3.8 IUSD Parameters (IdealUSDManager)

Parameter Name	Description
iusdValue	Current IUSD value (updatable via updateIUSD)
authorizedUpdaters[]	Authorized updaters whitelist
updaterWhitelistEnabled	Whitelist enable switch

B.3.9 Treasury Parameters (Treasury)

Parameter Name	Description
router	Uniswap V2 Router address (for buybacks)

B.4 Parameter Classification Statistics

Category	Count	Percentage
Static Constants (constant)	45	45%
Immutables (immutable)	35	35%
Governable Variables (governable)	20	20%
Total	**100**	**100%**

Design Notes:

Approximately 80% of system parameters are solidified at deployment (constant + immutable), with only 20% retaining governance capability. This "weak governance" design ensures:

- **Predictability**: Users and developers can rely on the system's core behavior not being changed by governance
- **Immutability**: Key economic parameters are as unmodifiable as Bitcoin
- **Minimal Trust**: Reduces trust assumptions about governors, lowers attack surface
- **Gas Optimization**: Immutable parameter reading costs only 3 gas, saving 99.86%
- **Emergency Flexibility**: Retains minimum necessary adjustment capability for extreme situations