Cashiva Standard Token (CST) WhitePaper

1. Introduction

Cashiva Standard Token (CST) is a family of tokens fully backed by cryptocurrency and issued on the Ethereum blockchain. Unlike traditional stablecoins, CST provides a fixed value in the selected fiat currency only at the moment of transaction, while the backing is always represented by liquid crypto assets (e.g., BTC, ETH, XMR).

Each CST token corresponds to a unique currency pair, and its symbol is constructed using the following scheme:

XXYYY, where:

- **XX** — fiat currency code (e.g., RUB, USD, KGS);
- **YYY** — cryptocurrency backing code (e.g., BTC, ETH, XMR).

Examples:

- **RUBTC** — Bitcoin-backed token denominated in Russian rubles;
- **USETH** — Ethereum-backed token denominated in US dollars;
- **KGXMR** — Monero-backed token denominated in Kyrgyz som.

Key Feature: CST fixes the reference value in fiat currency at the moment the transaction is confirmed, allowing accurate accounting of the transferred amount regardless of market volatility.

2. Problem Statement

2.1 Cryptocurrency Volatility

Cryptocurrencies are subject to high price volatility, making calculations in them unpredictable. When sending 0.1 BTC, one cannot precisely predict its value in rubles or dollars at the time of receipt. CST solves this by fixing the token price in fiat at the transaction moment, making calculations transparent even when the backing is cryptocurrency.

2.2 Why Not Use Fiat Stablecoins?

Fiat-backed stablecoins (USDT, USDC, and others) are widespread but have critical limitations:

- **Centralized custody:** These tokens are backed by fiat held in issuer-controlled accounts, making them vulnerable to freezes, sanctions, or recalls.
- **Trust dependency:** Users must rely on the issuer's reports and audits. There have been numerous cases of insufficient reserves and hidden risks.
- **Banking reliance:** Such tokens depend on banks and jurisdictions, exposing them to regulatory risks like asset freezes or delisting from exchanges.

2.3 CST — A Decentralized Alternative

CST solves these issues through its architecture:

- **Crypto-backed reserves:** No banking dependency. All assets exist on-chain and are verifiable.
- **Real-time transparency:** Any user can verify 100% backing of tokens at any time.
- **Fiat as a unit of account only:** CST does not hold fiat, only uses its value for pricing — providing calculation convenience without the systemic risks of fiat platforms.

3. CST Token Description

Each CST token is a wrapped token backed by a specific cryptocurrency and denominated in a selected fiat currency. This means users receive a token denominated in, for example, rubles or dollars, but actually storing an equivalent value in ETH, BTC, XMR, or other cryptocurrencies.

3.1 Token Symbol Scheme

Each token follows the naming convention: **XXYYY**, where:

- **XX** — fiat currency code (e.g., RUB, USD, KGS);
- **YYY** — crypto asset code (e.g., ETH, BTC, XMR).

3.2 Token Examples

- **RUBETH —** denominated in Russian rubles, backed by Ethereum;
- **USETH** — denominated in US dollars, backed by Ethereum;
- **KGXMR** — denominated in Kyrgyz som, backed by Monero;
- **RUBBTC** — denominated in Russian rubles, backed by Bitcoin.

3.3 Operating Principle

CST tokens are always backed 1:1 by their respective cryptocurrencies.

During each mint or burn operation, a reference fiat price is fetched from the oracle. This price does not influence the token amount minted but serves for accurate financial reporting, calculations, and value display.

The rate is fixed at transaction confirmation and recorded on-chain for transparency.

3.4 Comparison with Fiat Stablecoins

Unlike traditional stablecoins (USDT, USDC), CST:

- Does not depend on banks or hold fiat reserves;
- Is backed by cryptocurrency with full on-chain verifiability;
- Uses fiat solely as a unit of account, not as collateral.

3.5 Status and Launch Plan

At launch, the following tokens are available: RUBETH, KGSETH, PRRUBETH.

Within three months, RUBBTC, KGXMR, and USDETH will be released.

New pairs will be added based on demand and community preferences.

4. Target Audience and Use Cases

4.1 Users Needing Price Stability and Transparency

- Individuals and businesses needing to fix amounts in fiat for transfers or payments;
- Online merchants accepting cryptocurrency;
- Users from countries with high inflation rates.

4.2 Crypto Traders and Investors

- For temporary value stabilization;
- For calculations without dependence on the backing asset's exchange rate.

4.3 DeFi Platforms and Developers

- For use in lending, trading, derivatives, and other applications.

5. Technical Architecture

CST is implemented as ERC-20 standard smart contracts on the Ethereum network. Each contract corresponds to a specific currency pair with hardcoded collateral and unit of measurement settings.

5.1 CST Smart Contract

Collateral is stored:

- Directly in the contract address if the backing cryptocurrency is on the same network (e.g., ETH on Ethereum);
- Via a custodian if the backing is on another network (e.g., BTC, XMR).

Each operation (wrap, burn, transfer) fetches a price from the oracle and records the reference value in fiat currency.

Key contract methods:

- **wrap()** — deposits crypto collateral and mints CST tokens;
- **burn()** — burns CST and returns the backing cryptocurrency;
- **transfer()** — transfers CST tokens between users;
- **transferFrom()** — delegated transfer;
- **setOracle()** — update oracle address;
- **setWrapFeeParams()** — configure fee parameters;
- **setValidTimePeriod()** — set oracle data freshness threshold.

Rate fixing is implemented via the oracle contract, with the fixed reference value recorded in blockchain events.

The wrap and burn methods include Reentrancy Guard protection to prevent attacks similar to the DAO incident.

5.2 Oracle — oracle.cashiva.com

- Operated by LTD Cashiva Community (Kyrgyzstan).
- Data sources: major centralized and decentralized exchanges (e.g., Binance, Coinbase, Uniswap), and central bank official websites.
- Rates are considered stale if not updated for more than 1 hour.
- Users can manually trigger rate updates when transferring tokens.
- If the rate is stale or the oracle is unavailable, the transaction is automatically rejected.

6. Token Mechanics

CST operation is based on a simple and transparent process of minting and burning tokens, where all information about operations and values is recorded on the blockchain. Cryptocurrency backing, automatic rate retrieval, and reserve transparency make the process secure and verifiable.

6.1 Minting

- User sends cryptocurrency to the contract or custodian address (depending on the backing blockchain).
- Contract calls the oracle to get the current exchange rate relative to the selected fiat currency.
- Contract mints CST tokens in an amount equivalent to the deposited cryptocurrency (minus a fee).
- Simultaneously, the reference price (in fiat) obtained from the oracle is recorded on the blockchain and can be used for accounting, calculations, and external value display.

6.2 Burning

- User sends CST tokens back to the contract.
- Contract fetches the current rate from the oracle and fixes it at the transaction moment.
- CST is burned, and the user receives back the backing cryptocurrency minus a fee.
- The recorded rate is preserved as a reference in blockchain events.

6.3 Price Fixing

- The rate is always fixed at the moment of transaction confirmation: minting, burning, or transfer.
- If the rate hasn't been updated for more than one hour or the oracle is unavailable, the transaction is canceled.
- If necessary, users can initiate an immediate rate update.

6.4 Transparency

- All information about the volume of issued tokens, reserve size, fixed rates, and transactions is available on the public blockchain.
- This allows any participant to verify 100% collateralization and operation history at any time.

7. Security and Audit

The security of CST architecture is crucial for user trust and stable protocol operation. It is ensured through a combination of technical, organizational, and procedural measures aimed at protecting funds, preventing errors, and ensuring constant verifiability.

7.1 Audit

- Before launch, CST contracts undergo mandatory verification by LTD Cashiva Community (oracle owner).
- Each new contract version release or parameter change requires a mandatory audit.
- Regular independent third-party audits by reputable members of the crypto community are planned.
- Audit results will be publicly published, including verified contract hashes, vulnerability lists, and recommendations.

7.2 Technical Protection

- **Reentrancy Guard** — prevents repeated calls in wrap and burn methods, protecting against attacks similar to the DAO incident.
- **Rate freshness check** — if the oracle rate is stale (older than 1 hour), the operation is automatically canceled.
- **Oracle availability verification** — when the data source is unavailable, the transaction is blocked to prevent calculation errors.
- **Restricted owner rights** — despite the ability to change parameters, the contract's basic properties (fiat and backing cryptocurrency) are locked from changes after deployment.

7.3 Reserve and Operations Transparency

- The balance of collateral and issued tokens is always accessible on the blockchain.
- Contracts do not use external databases, only verifiable on-chain state.
- Any user can verify in real-time that the number of tokens in circulation does not exceed the cryptocurrency backing amount.

8. Economics and Fees

The CST economic model is based on simplicity, predictability, and transparency. It excludes hidden income mechanisms, fully relying on fees that are charged only for key user actions — minting and burning tokens.

8.1 Fee Structure

- Fees are charged for mint (issuance) and burn (redemption) of CST tokens.
- The default rate is 1% of the amount, but it can be individually set for each contract and differ from the base rate.
- The fee is calculated as a percentage of the cryptocurrency amount sent to or returned to the user.
- Fees are deducted in the base backing cryptocurrency (e.g., ETH for RUBETH).

Examples:

- A user sends 1 ETH to the RUBETH contract — receives 0.99 RUBETH.
- When redeeming 1 RUBETH — receives 0.99 ETH back.

8.2 Fee Usage

Collected fees are directed to:

- Infrastructure support: hosting, maintenance, smart contract and oracle development.
- Audit and security: covering expenses for regular technical and external audits.
- Development and scaling: implementing new currency pairs, launching tokens on other networks.
- Marketing and integrations: connecting to DeFi protocols, exchange listings, attracting partners.

8.3 Fee Transparency

- All fees go to the contract and can only be withdrawn by the owner via public transaction.
- Information about the current fee balance is always available on the blockchain.
- The contract ensures that access to CST backing is impossible — only to fees.

9. Project Governance

Effective governance of the CST project is built on role separation, strict parameter fixation, and controlled update capability. Contracts are designed to prevent owner abuse and maintain maximum transparency.

9.1 Governance Structure

- **Token issuer:** WMT Prime Corp (Panama). The company deploys CST contracts, sets parameters, and manages fees.
- **Oracle operator:** LLC Cashiva Community (Kyrgyzstan). Maintains oracle infrastructure and is responsible for the accuracy and timeliness of exchange rate data.

This separation reduces the risks associated with centralized control.

9.2 Contract Owner Powers

The contract owner can change technical parameters:

- Oracle address;
- Fee size;
- Acceptable rate staleness interval.

The owner can withdraw only accumulated fees. Access to backing reserves is impossible at the contract logic level.

Changing the basic CST parameters (fiat and backing cryptocurrency) is impossible after contract deployment.

9.3 Future Decentralization

- Within 6 months after launch, multisig control for key operations will be implemented.
- Multisig participant addresses and roles will be published in advance, with blockchain verification capability.
- Eventually, transition to a DAO model with voting for changes is possible if supported by the community.

Project governance is built on a combination of strict access discipline, immutability of critical parameters, and transparency of all owner actions in a public environment.