vSTRC Protocol Whitepaper

Bitcoin-Backed Yield-Bearing Tokens with Self-Tuning Peg Stability

Version 1.0 — February 2026

Author: Nikhil Ranjan (@niklabh)

Abstract

vSTRC (Vaulted STRC) is a decentralized finance protocol that creates a Bitcoin-backed, yield-bearing token pegged to $100 USD. Inspired by MicroStrategy's STRC perpetual preferred stock instrument, vSTRC brings the concept on-chain with full transparency, composability, and automated peg maintenance. The protocol employs a novel **Self-Tuning Variable Dividend Rate (VDR)** mechanism that dynamically adjusts yield to maintain price stability, creating a negative feedback loop between market price deviation and dividend distribution. This paper describes the economic model, smart contract architecture, security mechanisms, and governance framework.

---

1. Introduction

1.1 Background

In March 2025, MicroStrategy (now "Strategy") announced STRC, a perpetual preferred stock offering a 10% annual cumulative dividend backed by the company's substantial Bitcoin treasury. STRC was designed to appeal to income-oriented investors while maintaining Bitcoin exposure through the corporate treasury.

The key properties of STRC are:

- **Fixed income**: 10% cumulative annual dividend
- **Bitcoin-backed**: Collateralized by corporate BTC holdings (~500,000+ BTC)
- **Liquidation preference**: $100 per share
- **No maturity**: Perpetual instrument

However, STRC exists within the traditional financial system, subject to:

- Trading hours limitations
- Custodial intermediaries
- Limited composability
- Opaque treasury management
- Geographic restrictions

1.2 Vision

vSTRC reimagines STRC as a fully decentralized, permissionless, and transparent DeFi primitive. By encoding the economic logic into smart contracts on Ethereum, we achieve:

1. **24/7 trading** on decentralized exchanges
2. **Transparent treasury** — all BTC holdings are verifiable on-chain
3. **Composability** — vSTRC can be used in other DeFi protocols (lending, LP, etc.)
4. **Automated peg maintenance** — no human intervention needed for yield adjustments
5. **Programmable governance** — parameter changes require on-chain execution

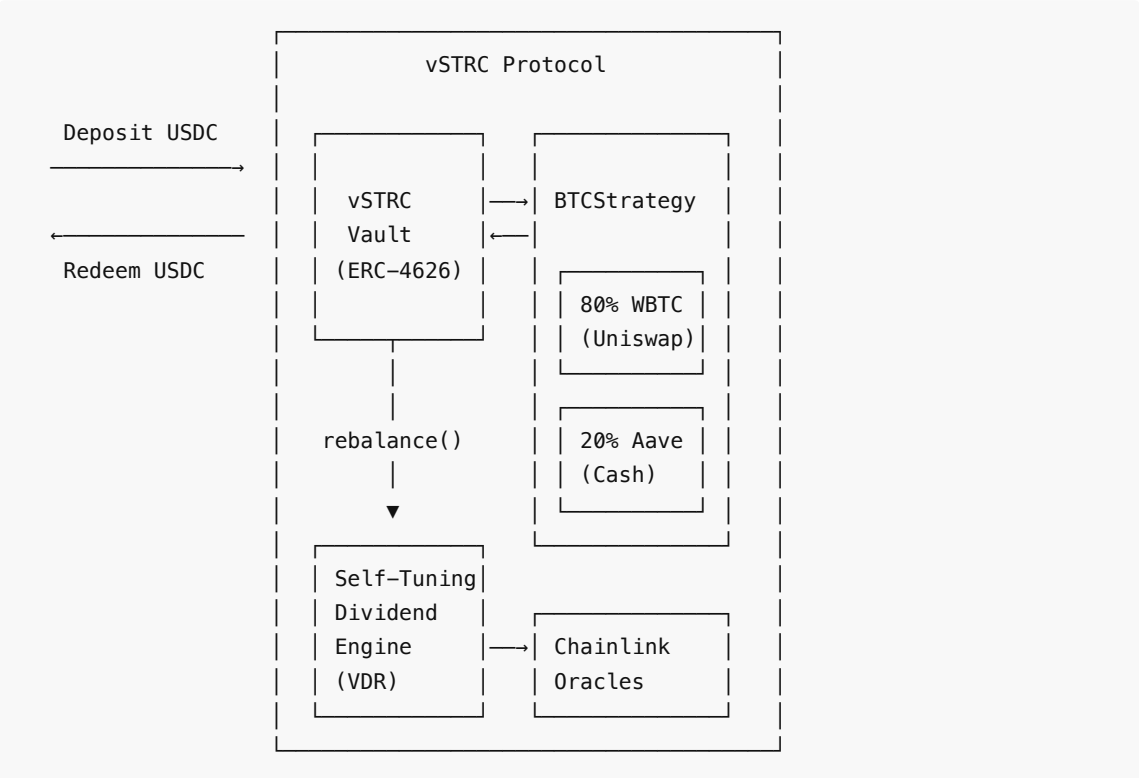
1.3 Key Innovation: Self-Tuning VDR

Unlike STRC's fixed 10% dividend, vSTRC employs a **variable dividend rate** that automatically adjusts based on the token's secondary market price relative to the $100 target. This creates a stabilizing feedback loop without requiring active market-making or reserve manipulation.

---

2. Protocol Architecture

2.1 System Overview



2.2 Token Standard

vSTRC implements the **ERC-4626 Tokenized Vault** standard, which provides:

- **Standardized deposit/withdraw** interface
- **Share/asset accounting** — shares represent proportional ownership of vault assets
- **Preview functions** — users can estimate outputs before transacting
- **Composability** — any protocol supporting ERC-4626 can integrate vSTRC

The **asset** is USDC (a stablecoin), and the **share** (vSTRC) represents the user's stake in the Bitcoin treasury.

2.3 Capital Deployment

When a user deposits USDC, the vault deploys capital to the BTCStrategy contract:

Allocation	Destination	Purpose
80%	WBTC via Uniswap V3	Bitcoin treasury (core backing)

20%	USDC in Aave V3	Cash reserve for redemptions and dividends

The 80/20 split is configurable by governance. The rationale:

- **80% BTC** maximizes Bitcoin exposure (the core value proposition)
- **20% cash** provides liquidity for redemptions without selling BTC and generates additional yield via Aave lending rates

---

3. Self-Tuning Variable Dividend Rate

3.1 The Peg Problem

A $100-pegged yield token faces a fundamental challenge: market supply and demand cause price deviations. Traditional solutions include:

1. **Algorithmic minting/burning** (fragile, prone to death spirals)
2. **Market-making reserves** (expensive, requires active management)
3. **Over-collateralization** (capital-inefficient)

vSTRC takes a different approach: **yield-based price stabilization**.

3.2 The VDR Formula

The Variable Dividend Rate is calculated as:

$$VDR = R_{base} + K \times \frac{P_{target} - P_{market}}{P_{target}}$$

Where:

- (R_{base}) = Base yield rate (default: 8% annually)
- (K) = Sensitivity coefficient (default: 20%)
- (P_{target}) = Target peg price ($100)
- (P_{market}) = Current market price of vSTRC

The rate is bounded:

$$R_{min} \leq VDR \leq R_{max}$$ $$1% \leq VDR \leq 25%$$

3.3 Stability Mechanism

The VDR creates a **negative feedback loop**:

When price < $100 (under peg):

1. VDR increases → Higher yield offered
2. Higher yield attracts rational investors to buy vSTRC
3. Buy pressure increases → Price rises toward $100
4. As price approaches $100, VDR decreases back to base rate

When price > $100 (over peg):

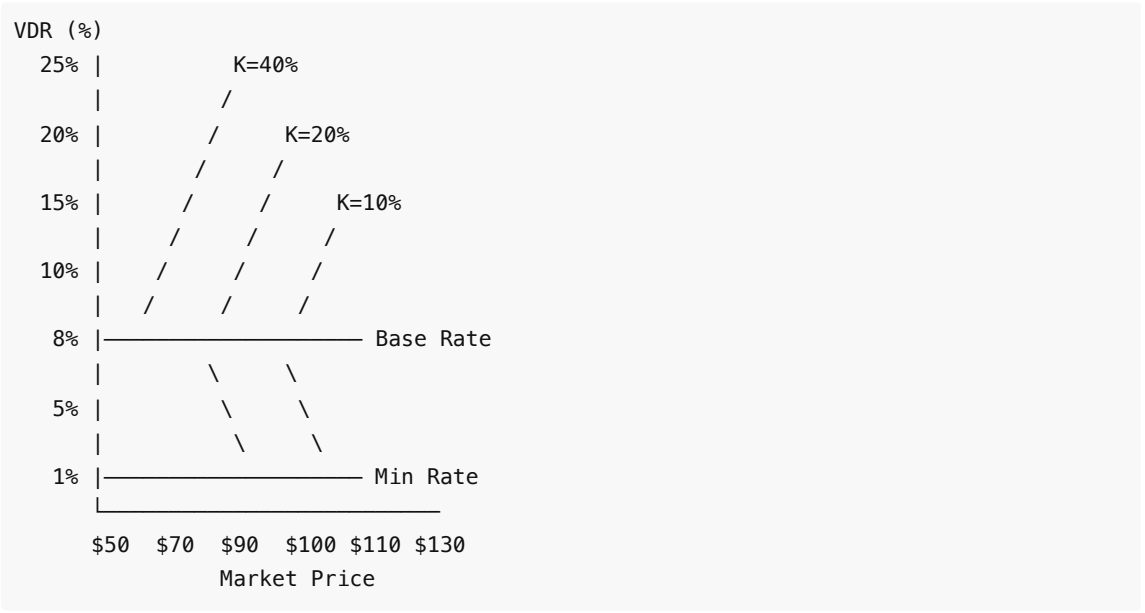
1. VDR decreases → Lower yield offered
2. Lower yield reduces attractiveness → Some holders sell
3. Sell pressure increases → Price falls toward $100
4. As price approaches $100, VDR increases back to base rate

3.4 Sensitivity Analysis

The sensitivity coefficient (K) determines how aggressively the rate responds to price deviations.

K Value	Response	Use Case
10%	Conservative	Stable markets, high confidence
20%	Moderate (default)	Normal operation
40%	Aggressive	Volatile markets, rapid correction needed

Response curves for different K values:



3.5 Epoch System

The VDR is recalculated once per **epoch** (default: 7 days). This provides:

1. **Gas efficiency** — one transaction per week instead of per-block
2. **Stability** — prevents rate oscillation from short-term price noise
3. **Predictability** — users know when the next adjustment occurs
4. **Keeper compatibility** — standard cron job or Chainlink Automation

Each epoch, the keeper calls `rebalanceYield()`, which:

1. Reads the current vSTRC market price from the oracle
2. Computes the new VDR using the formula
3. If price is below target: increases cash outflow to fund higher dividends
4. If price is above target: reduces distributions, accumulates more BTC
5. Emits events for off-chain tracking

3.6 Dividend Accounting

The epoch dividend is computed as:

$$D_{epoch} = \frac{A_{total} \times VDR \times T_{epoch}}{10000 \times T_{year}}$$

Where:

- (A_{total}) = Total assets under management
- (T_{epoch}) = Epoch duration in seconds
- (T_{year}) = 365 days in seconds (31,536,000)

The dividend is reflected as **share price appreciation** through the ERC-4626 exchange rate, rather than direct token distributions. This is tax-efficient and gas-efficient.

---

4. Treasury Management

4.1 Bitcoin Acquisition

Capital is converted to WBTC through Uniswap V3 using the `exactInputSingle` function. Key protections:

- **Slippage control**: Maximum 1% (configurable) deviation from oracle price
- **MEV protection**: Deadline parameter prevents sandwich attacks across blocks
- **Price feeds**: Chainlink BTC/USD oracle validates expected output

4.2 Cash Reserve Management

The 20% cash reserve is deposited into Aave V3:

- **Yield generation**: Earns Aave supply APY on USDC (typically 3-8%)
- **Instant liquidity**: Can withdraw at any time for redemptions
- **No lock-up**: Aave V3 does not have minimum deposit periods

4.3 Rebalancing

The strategy can rebalance between BTC and cash positions:

Trigger	Action
Need higher dividends	Sell BTC → Increase cash reserve
Excessive cash	Buy BTC → Increase treasury
Redemption pressure	Withdraw from Aave → Serve redemptions
BTC price appreciation	Treasury grows naturally

4.4 Collateralization

The protocol tracks a **Collateral Ratio**:

$$CR = \frac{V_{BTC} + V_{cash}}{L_{total}}$$

Where:

- (V_{BTC}) = BTC treasury value in USDC terms (via Chainlink)
- (V_{cash}) = Cash reserve value (Aave aToken balance)
- (L_{total}) = Total liabilities (totalSupply × targetPrice)

CR Range	Status	Action
> 1.5x	Over-collateralized	Normal operation

1.0x – 1.5x	Adequately collateralized	Monitor closely
0.8x – 1.0x	Under-collateralized	Reduce dividends, pause minting
< 0.8x	Critical	Circuit breaker activates

---

5. Security Framework

5.1 Circuit Breaker

The protocol includes an automatic circuit breaker that triggers when BTC drops more than 20% within a 1-hour window.

Trigger conditions:

```
IF (previousBTCPrice − currentBTCPrice) / previousBTCPrice ≥ 20%
   AND timeSincePreviousCheck ≤ 1 hour
THEN:
   − Halt all new deployments to strategy
   − Prevent new minting
   − Require manager manual reset after assessment
```

Reset procedure:

1. Manager assesses market conditions
2. Verifies oracle data integrity
3. Calls `resetCircuitBreaker()` with updated price checkpoint
4. Protocol resumes normal operation

5.2 Access Control Matrix

Function	MANAGER	KEEPER	VAULT	PUBLIC
Set Strategy	✓			
Update Params	✓			
Pause/Unpause	✓			
Emergency Withdraw	✓			
Reset Circuit Breaker	✓			
Rebalance Yield		✓		
Deploy Capital			✓	
Withdraw Capital			✓	
Deposit/Redeem				✓

5.3 Oracle Security

- **Primary**: Chainlink decentralized price feeds (BTC/USD, USDC/USD)
- **Staleness checks**: BTC feed must update within 1 hour, USDC within 24 hours

- **Fallback**: NAV-based pricing (totalAssets / totalSupply)
- **vSTRC price**: Configurable oracle (Chainlink, TWAP, or manual feed)

5.4 Additional Safeguards

1. **Reentrancy guards** — All external calls protected by OpenZeppelin's `ReentrancyGuard`
2. **Safe token transfers** — `SafeERC20` library for all token operations
3. **Integer overflow** — Solidity 0.8.x built-in checks
4. **Deposit limits** — Configurable min/max per transaction and total
5. **ERC20Permit** — Gasless approvals to prevent front-running

---

6. Governance

6.1 Roles

The protocol uses OpenZeppelin's `AccessControl` for fine-grained permission management:

- **DEFAULT_ADMIN_ROLE**: Can grant/revoke any role. Initially held by deployer; intended to be transferred to a multisig or DAO.
- **MANAGER_ROLE**: Operational control. The "Michael Saylor" role — manages strategy parameters, allocation, and emergency actions.
- **KEEPER_ROLE**: Automated role for calling `rebalanceYield()` each epoch. Can be assigned to a Chainlink Automation or Gelato keeper.

6.2 Configurable Parameters

Parameter	Default	Range	Description
`baseRateBps`	800 (8%)	0–10000	Base yield rate
`sensitivityBps`	2000 (20%)	0–10000	VDR sensitivity coefficient K
`minRateBps`	100 (1%)	0–baseRate	Minimum yield floor
`maxRateBps`	2500 (25%)	baseRate–10000	Maximum yield ceiling
`targetPrice`	100e6 ($100)	> 0	Target peg price
`epochDuration`	604800 (7d)	> 0	Epoch length in seconds
`btcAllocationBps`	8000 (80%)	0–10000	% of deposits to BTC
`cashAllocationBps`	2000 (20%)	0–10000	% of deposits to cash
`maxSlippageBps`	100 (1%)	0–1000	Max swap slippage
`circuitBreakerThresholdBps`	2000 (20%)	0–10000	BTC drop threshold

6.3 Future: DAO Governance

The protocol is designed for progressive decentralization:

1. **Phase 1** (Launch): Manager is a multisig (e.g., Safe)
2. **Phase 2** (Maturity): Governor contract with vSTRC-based voting
3. **Phase 3** (Full Decentralization): Timelock + Governor + Community

---

7. Risk Analysis

7.1 Identified Risks

Risk	Severity	Mitigation
BTC price crash (>50%)	High	Circuit breaker, cash reserve, CR monitoring
Oracle manipulation	High	Chainlink decentralized feeds, staleness checks
Smart contract exploit	Critical	Audits, formal verification, bug bounty
Uniswap liquidity drought	Medium	Slippage protection, multiple DEX routes
Aave insolvency	Low	Monitor Aave health, diversify across protocols
Governance attack	Medium	Timelock, multisig, role separation
Bank run (mass redemption)	Medium	Cash reserve buffer, redemption queuing

7.2 Worst-Case Scenarios

Scenario: BTC drops 50% in 24 hours

1. Circuit breaker triggers at -20%, halting minting
2. Existing redemptions served from 20% cash reserve
3. If cash depleted, BTC sold at market (at a loss)
4. CR drops below 1.0 → protocol enters recovery mode
5. VDR increases to maximum (25%) to attract fresh capital
6. If BTC recovers, CR naturally improves

Scenario: vSTRC price crashes to $50

1. VDR maxes out at 25%
2. High yield attracts arbitrageurs: buy vSTRC at $50, earn 25%
3. If fundamental backing (CR) remains > 1.0, rational investors arbitrage
4. Price gradually recovers toward $100

---

8. Comparison with Real-World STRC

Feature	STRC (MicroStrategy)	vSTRC (DeFi)
Dividend rate	Fixed 10%	Variable 1-25% (self-tuning)
Backing	Corporate BTC treasury	On-chain WBTC treasury
Transparency	Quarterly filings	Real-time on-chain
Trading hours	NYSE hours only	24/7/365
Settlement	T+1	Instant
Composability	None	Full DeFi integration
Custody	Centralized	Non-custodial smart contracts

Governance	Board of directors	On-chain roles + future DAO
Peg mechanism	Market-driven	Automated self-tuning VDR
Minimum investment	$100 (1 share)	$1 USDC

---

9. Technical Specifications

9.1 Contract Addresses (Sepolia Testnet)

Deployed addresses are saved to `deployment-sepolia.json` after running the deployment script.

9.2 Standards Compliance

- **ERC-4626**: Tokenized vault standard
- **ERC-20**: Fungible token standard
- **ERC-2612**: Permit (gasless approvals)

9.3 Compiler Settings

- Solidity: 0.8.20
- Optimizer: 200 runs
- EVM Target: Shanghai
- Via-IR: Enabled

---

10. Conclusion

vSTRC demonstrates that the concept of a Bitcoin-backed, yield-bearing preferred instrument can be successfully ported to DeFi with meaningful improvements. The self-tuning VDR mechanism provides an elegant solution to the peg stability challenge, leveraging the predictable behavior of rational market participants seeking yield.

By combining the ERC-4626 vault standard with Chainlink oracles, Uniswap V3 for execution, and Aave V3 for cash management, vSTRC creates a transparent, composable, and automated alternative to traditional structured products.

The protocol is designed for progressive decentralization, starting with multisig governance and evolving toward community-driven decision-making as the system matures and proves its resilience.

Appendix A: Contract Inheritance

```
vSTRC
 ├── ERC4626 (OpenZeppelin)
 │    └── ERC20
 ├── ERC20Permit
 ├── AccessControl
 ├── Pausable
 └── ReentrancyGuard

BTCStrategy
 ├── IStrategy
```

```
 ├── AccessControl
 └── ReentrancyGuard

SelfTuningMath (Library)
 └── Pure functions only
```

Appendix B: Gas Estimates

Operation	Estimated Gas
deposit()	~180,000
redeem()	~150,000
rebalanceYield()	~250,000
strategy.deploy()	~350,000
strategy.withdraw()	~300,000

References

1. MicroStrategy STRC Prospectus, SEC Filing, March 2025
2. EIP-4626: Tokenized Vault Standard, ethereum.org
3. Chainlink Price Feeds Documentation
4. Uniswap V3 Core Whitepaper
5. Aave V3 Technical Paper
