

mUSD: Building on Mezo Bitcoin

A Developer's Guide to CDP Integration

Objective: High-level system overview + code integration patterns

What is mUSD?

- **Stablecoin** minted by creating loans against crypto collateral
- **System Purpose:**
 - Borrowing against collateral
 - Maintain mUSD peg as stablecoin
- **Key User Actions:**
 - Opening, adjusting, and closing troves
 - Liquidation
 - Redemption

Contracts Overview

Core Components

- **MUSD**: The stablecoin token contract
- **BorrowerOperations**: Basic borrower operations, interacts with TroveManager and asset pools
- **TroveManager**: Trove state + liquidations and redemptions logic
- **StabilityPool**: Handles liquidations

Supporting Infrastructure

- **Asset Pools**: Track collateral and debt

Liquidation Economics

- **Liquidator Incentives:** \$200 MUSD gas compensation + 0.5% collateral profit
- **Stability Pool Incentives:** ~9% BTC discount on liquidations
- **Risk Management:** 110% liquidation threshold safety buffer
- **Speed Matters:** Fast liquidations prevent bad debt

Redemption Arbitrage

- **Peg Maintenance:** Buy cheap mUSD, redeem for \$1 BTC
- **Market Pressure:** Redemptions hit lowest-CR troves first

Risk Management Deep Dive

- **Borrower Risks:** Liquidation (10% loss), redemption (BTC upside loss), bad debt, depegging
- **System Controls:** minNetDebt (1800 MUSD minimum), Recovery Mode
- **Stress Scenarios:** Large liquidation rebalancing

Interest Rate Mechanics

- **Simple vs Compound Interest:** MUSD uses simple linear interest
- **Rate Setting:** Set at trove creation based on global rate, kept for trove lifetime
- **Refinancing:** Costs percentage of borrowing to get new rate and capacity

PCV Economics

- **Bootstrap Loan:** Why chosen over token incentives
- **Fee Distribution with Active Loan:**
 - Flow: fees → PCV → split (60% debt repayment, 40% fee recipient)
- **Fee Distribution when Loan Repaid:** 100% to fee recipient or stability pool
- **distributeMUSD():** Manual governance process (weekly)
- **Fee Splits:** Governance controlled

Developer Deep Dive: Pending Rewards

Critical Concept for Integration

- **What:** Debt & collateral redistributed when Stability Pool insufficient
- **When Applied:** Next trove interaction (any borrower operation)
- **Code Impact:**
 - Wrong: `getTroveDebt()` (stored amounts only)
 - Right: `getEntireDebtAndColl()` (includes pending)

Developer Deep Dive: Hint Generation

Gas Optimization Essential

- **Why Important:** Troves in sorted list by CR, finding insertion point expensive
- **Solution:** Hints narrow search from $O(n)$ to $O(1)$ gas
- **Implementation:** HintHelpers with code examples
- **Freshness:** Always generate fresh hints before transactions

Developer Deep Dive: Integration Patterns & Best Practices

- Reading Trove Data correctly
- Event Monitoring patterns
- Error Handling: Common revert conditions

User Journey 1: Opening a Trove

Code Demo

- **User Action:** Deposit collateral, borrow mUSD
- **openTrove Function:**
 - debtAmount, assetAmount, collateralization ratio
 - ICR (Individual Collateralization Ratio), TCR (Total Collateralization Ratio)
 - upperHint and lowerHint for efficient trove placement
 - HintHelpers for hint generation
 - Gas compensation
 - Borrowing capacity

User Journey 2: Adjusting a Trove

Code Demo

- **User Action:** Adjust collateral, repay debt, increase borrowing
- **Key Functions:**
 - `adjustTrove` function
 - Convenience functions: `addColl`, `withdrawColl`, `repayMUSD`, `withdrawMUSD`
 - `TroveManager` functions for fetching data
 - Refinancing: Moving to new interest rates

User Journey 3: Closing a Trove

Code Demo

- **User Action:** Repay all debt, withdraw collateral, close trove
- **closeTrove Function:**
 - Collateral returned to user
 - Paid mUSD burned from balance
 - Gas compensation burned from gas pool

User Journey 4: Liquidation

Code Demo

- **User Experience:** Liquidation when undercollateralized
- **liquidate Function**
- **Three Liquidation Scenarios:**
 - Stability Pool absorption (normal)
 - Partial liquidation (insufficient SP)
 - Full redistribution (empty pool)
- **StabilityPool:**
 - Deposit to earn liquidation rewards proportional to stake

User Journey 5: Redemption

Code Demo

- **User Action:** Redeem mUSD for BTC collateral
- **Redemption Process:**
 - 1-1 mUSD exchange for collateral (minus fee)
 - Debt cancelled from active troves in ascending CR order
 - Collateral drawn from redeemed troves
- **Fee Structure:** 0.75% fee from collateral

Documentation Tour

- **README:** System overview, flow diagrams, economic model, liquidation scenarios
- **simpleInterest.md:** Interest calculation deep dive
- **Function Reference:** Contract interfaces
- **Event Reference:** Critical monitoring events
- **Test Files:** Integration examples

Risk Management Tools You Can Build

- Liquidation risk dashboards with borrower warnings
- Redemption risk calculators
- System health monitoring (TCR, Recovery Mode alerts)

Liquidation & Monitoring Tools You Can Build

- Liquidation bots with profit calculations
- Trove health monitoring dashboards
- Mobile liquidation risk alerts
- MEV-resistant liquidation strategies

DeFi Integrations You Can Build

- mUSD yield farming interfaces
- Automated trove rebalancing
- Arbitrage opportunity scanners

Testing and Development Environment

- Local setup: Running contracts locally
- Test Networks: Available deployments
- Useful development commands

Q&A and Resources

Questions & Discussion

Summary: mUSD CDP system on Mezo Bitcoin with developer-friendly integration patterns

Resources:

- Repository, documentation, Discord/community links

Key Visuals Needed

- Architecture diagrams
- Economic flow diagrams (from README)
- Liquidation scenario flows
- Code examples for each user journey
- Integration pattern examples
- Documentation screenshots