Frostbyte White Paper

Core Banking System: Interactive COBOL-Emulating CLI

Author: Frostbyte Research | *Date: November 22, 2025*

This white paper outlines the core interactive banking system, a Python CLI replicating COBOL main.cob programs. It provides account management with strict financial precision matching PIC 9(6)V99, validators, and logging for production banking operations.

Introduction

The foundational interactive banking module emulates COBOL's menu-driven transaction processing. It enables real-time account operations-balance inquiry, credit, and debit-while strictly preserving legacy precision and business rules. This allows for a "look and feel" that is familiar to mainframe operators while running on modern infrastructure.

Architecture

Core Components

- run_banking_system.py: The entry point that configures logging and initializes the Account, TransactionService, and CLI modules.
- CommandLineInterface (ui/cli.py): Manages the menu loop (View Balance, Credit, Debit, Exit) and handles input validation.
- Account Model: A Python class that enforces PIC 9(6)V99 Decimal precision, ensuring no floating-point errors occur.
- TransactionService: Encapsulates the business logic, separating it from the UI and data storage.
- Validators: Strict validation logic that mirrors COBOL PIC clauses (e.g., numeric checks, length constraints).

Data Flow

The flow is linear and deterministic: CLI Input -> Validate -> TransactionService -> Account Update -> Log.

Key Features

- Interactive Menu: A modern UX that replaces legacy COBOL PERFORM loops with clean, event-driven handlers.
- Financial Precision: Uses Python's Decimal library to enforce COBOL-grade accuracy, preventing "penny drift."
- Input Validation: rigorous checks for monetary inputs (must be positive, max 2 decimal places).
- Error Handling: Graceful failure modes with detailed logging for troubleshooting.
- Configurable Logging: Centralized configuration for audit trails and system monitoring.

COBOL Emulation Details

- Precision: Preserves PIC 9(6)V99 behavior using Decimal.
- Logic: Encapsulates the business logic found in legacy operations.cob files.
- UX: Replaces the main.cob structure with a maintainable Python CLI.
- Validation: Ensures strict compliance with COBOL PIC definitions.

Conclusion

This CLI core serves as a production-ready foundation. It mirrors the interactive capabilities of COBOL banking systems, proving that modern languages can uphold the rigorous standards of legacy financial software. It forms the basis for further extensions into batch processing, APIs, and CICS integration.