# Blockchain Smart Contracts vs. SWIFT: Settlement & Transfer Comparison

## Question

> **How does the blockchain smart contract as a means to settle and transfer equal or different from the SWIFT transfer and settlement?**

## Fundamental Differences

| Aspect | Blockchain Smart Contracts | SWIFT System |
|---|---|---|
| **Core Technology** | Distributed ledger technology | Centralized messaging network |
| **Settlement Mechanism** | Direct asset transfer on-chain | Messaging that triggers separate settlement |
| **Trust Model** | Trustless (code-based execution) | Trusted intermediaries required |
| **Settlement Finality** | Minutes to hours (blockchain-dependent) | 1-3 business days (correspondent banking) |
| **Available 24/7** | Yes | No (follows banking hours) |

| | | |
|---|---|---|
| **Transaction Visibility** | Public (on public blockchains) | Private between participants |

## Key Differentiating Features

### 1. Settlement Process

**Blockchain Smart Contracts:**

- **Direct Settlement**: Assets move directly between parties on the blockchain
- **Atomic Execution**: All-or-nothing execution guaranteed by code
- **Self-Execution**: Contract automatically executes when conditions are met
- **No Intermediaries**: Direct peer-to-peer settlement
- **Programmable Logic**: Complex settlement conditions can be encoded

**SWIFT System:**

- **Messaging Only**: SWIFT only communicates instructions, doesn't move money
- **Multi-Step Process**: Separate clearing and settlement phases
- **Correspondent Banking**: Requires a network of intermediary banks
- **Manual Intervention**: Often requires human approval steps
- **Standardized Messages**: Uses predefined message formats (MT103, MT202, etc.)

### 2. Asset Type Handling

**Blockchain Smart Contracts:**

- **Native Digital Assets**: Direct transfer of cryptocurrencies
- **Tokenized Assets**: Can handle tokenized versions of any asset
- **Programmable Money**: Can include complex transfer logic

- **Multi-Asset Capability**: Can settle multiple assets in one transaction

**SWIFT System:**

- **Traditional Currencies Only**: Primarily fiat currencies
- **Bank Ledger Entries**: Represents bookkeeping entries, not direct asset transfers
- **Separate Asset Classes**: Different message types for different transactions
- **Separate Processes**: Securities settlement requires different systems

## 3. Security Model

**Blockchain Smart Contracts:**

- **Cryptographic Security**: Based on public-key cryptography
- **Immutable Records**: Transaction history cannot be altered
- **Code-Based Execution**: Relies on correctness of smart contract code
- **Consensus Mechanism**: Validated by network consensus
- **Security Risks**: Smart contract vulnerabilities, private key management

**SWIFT System:**

- **Network Security**: Closed proprietary network
- **Institutional Security**: Based on trusted relationships between banks
- **Layered Authentication**: Multiple security checks and balances
- **Human Oversight**: Manual verification at multiple points
- **Security Risks**: Fraudulent message insertion, insider threats

### 4. Compliance and Regulation

**Blockchain Smart Contracts:**

- **Regulatory Uncertainty**: Evolving regulatory frameworks
- **Compliance Challenges**: KYC/AML integration more complex
- **Jurisdictional Questions**: Unclear which laws apply
- **Programmable Compliance**: Can encode some compliance rules in contracts
- **Sanction Screening**: More challenging to implement effectively

**SWIFT System:**

- **Established Regulatory Framework**: Well-defined compliance requirements
- **Integrated Compliance Tools**: Built-in SWIFT compliance utilities
- **Clear Jurisdiction**: Based on established banking regulations
- **Standardized Procedures**: Well-defined sanction screening processes
- **Regulatory Reporting**: Built-in capabilities for required reporting

## Implementation in NVC Banking Platform

### 1. Blockchain Settlement Implementation (Current)

The NVC platform currently uses:

- **Settlement Smart Contract**: Handles the transfer of assets between parties
- **MultiSig Wallet**: Requires multiple approvals for high-value transactions
- **NVC Token (NVCT)**: Digital asset used for settlement
- **Public Network**: Ethereum blockchain (Sepolia testnet)

Key code in

```
blockchain.py
```

:

```
def settle_payment_via_contract(from_address, to_address,
amount_in_eth, private_key, transaction_id):
    """Settle a payment using the settlement smart
contract"""
    settlement_contract = get_settlement_contract()
    # Convert ETH amount to Wei (smallest Ethereum unit)
    amount_in_wei = Web3.toWei(amount_in_eth, 'ether')

    # Get transaction count to determine nonce
    web3 = get_web3()
    nonce = web3.eth.getTransactionCount(from_address)

    # Create contract transaction
    tx = settlement_contract.functions.createSettlement(
        to_address,
        str(transaction_id)
    ).buildTransaction({
        'chainId': int(web3.net.version),
        'gas': 2000000,
        'gasPrice': web3.eth.gasPrice,
        'nonce': nonce,
        'value': amount_in_wei
    })

    # Sign transaction with private key
    signed_tx = web3.eth.account.signTransaction(tx,
private_key)

    # Send transaction and get hash
    tx_hash =
web3.eth.sendRawTransaction(signed_tx.rawTransaction)
    return web3.toHex(tx_hash)
```

## 2. SWIFT Implementation (In Development)

The platform would implement SWIFT messaging through:

- **Message Generation**: Create standardized MT messages

- **Message Validation**: Ensure compliance with SWIFT standards
- **Secure Transmission**: Send messages through SWIFT network
- **Status Tracking**: Monitor message delivery and responses
- **Reconciliation**: Match SWIFT messages with internal transactions

Example implementation in

```
swift_integration.py
```

:

```
def create_mt103_message(transaction, sender_info,
receiver_info):
    """Create a MT103 (Single Customer Credit Transfer)
message"""

    # Format according to SWIFT MT103 standards
    message = {
        "Block1": {
            "ApplicationID": "F",
            "ServiceID": "01",
            "ReceivingLTAddress": receiver_info["bic"],
            "SessionNumber": generate_session_number(),
            "SequenceNumber": generate_sequence_number()
        },
        "Block2": {
            "MessageType": "103",
            "SenderInputTime": current_time_format(),
            "MIR": generate_mir(),
            "MessagePriority": "N"
        },
        "Block3": {
            "Tag108": "MT103",
        },
        "Block4": {
            "Tag20": transaction.transaction_id,  # Reference
            "Tag23B": "CRED",  # Credit instruction
            "Tag32A": format_date_amount(transaction.date,
transaction.amount, transaction.currency),
            "Tag50K": format_ordering_customer(sender_info),
            "Tag59": format_beneficiary(receiver_info),
```

```
            "Tag71A": "SHA",  # Details of charges (shared)
            "Tag71F": format_charges(transaction.fees) if
transaction.fees else None,
        },
        "Block5": {
            "MAC": calculate_mac(),
            "CHK": calculate_checksum()
        }
    }

    return format_swift_message(message)  # Convert to proper
SWIFT format
```

## Strategic Integration: Hybrid Approach for NVC

A hybrid approach leverages strengths of both systems:

1. **For Speed and Direct Settlement:**
    - Use blockchain for direct, immediate settlements
    - Ideal for platform-to-platform transfers
    - Perfect for NVC token transfers
2. **For Traditional Banking Integration:**
    - Use SWIFT for connecting to traditional banking system
    - Essential for large corporate and institutional clients
    - Required for cross-border fiat transactions
3. **Smart Routing System:**
    - Analyze each transaction for optimal settlement path
    - Consider speed requirements, cost, and counterparty capabilities
    - Automatically choose blockchain or SWIFT settlement
4. **Unified Customer Experience:**
    - Hide technical settlement details from users
    - Provide consistent interface regardless of settlement method

- Offer transparency options for those who want details

## Practical Considerations for Implementation

1. **Settlement Finality:**
    - Blockchain: Wait for sufficient confirmations (higher security)
    - SWIFT: Clear difference between message delivery and actual settlement
2. **Reconciliation Processes:**
    - Blockchain: On-chain verification sufficient
    - SWIFT: Need end-of-day reconciliation with correspondent banks
3. **Liquidity Management:**
    - Blockchain: Pre-fund wallets with sufficient crypto assets
    - SWIFT: Maintain nostro/vostro accounts with correspondent banks
4. **Cost Structures:**
    - Blockchain: Gas fees (variable) + platform fees (if any)
    - SWIFT: Message fees + correspondent banking fees (typically higher)

---