Non-Custodial Blockchain ATS: Architecture of a Regulatory-Compliant Securities Exchange

Zach Kelling

Lux Industries

2025

Abstract

We present the architecture of a non-custodial Alternative Trading System (ATS) built on Lux Network, designed to trade digital securities in full compliance with SEC Regulation ATS, broker-dealer requirements under the Securities Exchange Act of 1934, and transfer agent obligations under Section 17A. The system is implemented as a single Go binary running on Hanzo Base, comprising 32 collections, 64 API routes, and 19 scheduled tasks. Non-custodial operation is achieved through MPC threshold signing (`luxfi/mpc`) with a 2-of-3 threshold requirement for any asset transfer—the platform never holds private keys. This work descends directly from Arca's ArCoin infrastructure (2018), which pioneered interest-bearing digital securities backed by US Treasury bills using ERC-1400/ERC-1404 security token contracts. We describe the complete system: order matching, settlement on Lux EVM with T+0 atomic finality, KYC/AML integration via Simplici, equities routing through Alpaca OmniSub, and regulatory reporting. We compare the architecture against existing platforms (tZERO, Securitize Markets, INX) and demonstrate that non-custodial MPC custody eliminates the single largest attack surface in securities infrastructure while preserving regulatory compliance.

This paper provides the formal architecture for LP-001 (digital securities), LP-005 (regulated ATS stack), and LP-007 (CLOB matching engine) of the Lux Improvement Proposals (https://github.com/luxfi/lps).

1 Introduction

The United States securities market operates under a regulatory framework designed for centralized intermediaries. Broker-dealers hold customer securities in street name. Transfer agents maintain issuer registries on private databases. Alternative Trading Systems match orders through proprietary systems that custody assets during settlement. Every intermediary holds keys, creating attack surfaces that have produced losses exceeding $10B in the traditional finance sector over the past decade through insider theft, operational failures, and cyberattacks.

The blockchain industry has attempted to address these structural risks by building "decentralized exchanges," but DEX architectures designed for fungible utility tokens are fundamentally incompatible with the regulatory requirements of securities trading. Securities require transfer restrictions, accreditation verification, jurisdictional compliance, forced transfers for regulatory actions, and auditable cap table management. No existing DEX satisfies these requirements.

1.1 From Arca to Lux

This work originates in the Arca platform (ar.ca), where the author served as CTO in 2018. Arca built several foundational components for digital securities infrastructure:

- **ArCoin**: An interest-bearing digital security backed by US Treasury bills, implemented as an ERC-1400 token with automated NAV calculation and distribution.
- **st-contracts**: A security token protocol implementing ERC-1400 (Security Token Standard) and ERC-1404 (Simple Restricted Token Standard) with on-chain transfer restrictions.
- **exchange-sdk**: A trading SDK for digital securities with order book management, matching engine, and settlement coordination.
- **sec-wallet**: An institutional-grade wallet for securities custody with multi-signature authorization and audit logging.
- **Arca Funds**: The first crypto hedge fund structure with tokenized fund interests represented as security tokens.
- **miners/binance-recorder**: Market data infrastructure for recording and replaying exchange data feeds.

The central lesson from Arca was that custodial architectures—even well-engineered ones—create irreducible counterparty risk. When a platform holds private keys on behalf of investors,

every operational failure, insider threat, and regulatory seizure becomes a potential loss event. The path forward required eliminating custodial risk at the protocol level.

This paper describes that path: a non-custodial ATS built on Lux Network that satisfies every requirement of SEC Regulation ATS, broker-dealer registration, and transfer agent registration while never holding a single private key.

1.2 Contributions

1. A complete ATS architecture that is non-custodial by construction, using MPC threshold signatures for all asset movements.
2. A single-binary deployment model on Hanzo Base that packages order matching, settlement, compliance, and reporting into 32 collections and 64 routes.
3. Atomic T+0 settlement on Lux EVM, eliminating the T+1/T+2 settlement risk present in traditional securities infrastructure.
4. A regulatory compliance framework that maps SEC/FINRA requirements to on-chain enforcement mechanisms.
5. Integration architecture for hybrid traditional/digital securities via Alpaca OmniSub.

1.3 Related Work

The Lux MPC custody architecture is detailed in the companion paper on M-Chain [1]. The security token standard used for on-chain securities is formalized in the Lux Security Token Standard [2]. Smart order routing across multiple execution venues is described in the Lux Smart Order Routing paper [3]. The transfer agent subsystem is presented separately [4]. The Lux credit protocol, which enables margin lending against tokenized securities, is specified in [5].

2 Regulatory Framework

2.1 ATS Registration (Regulation ATS)

An Alternative Trading System under SEC Rule 300(a) is any organization that constitutes, maintains, or provides a marketplace for bringing together purchasers and sellers of securities. The Lux ATS registers on Form ATS-N with the SEC and operates under the following obligations:

1. **Fair access**: Any qualified investor meeting accreditation and KYC/AML requirements may trade.
2. **Order display**: Orders above applicable thresholds are displayed to all subscribers.
3. **Best execution**: The smart order router satisfies Regulation NMS best execution obligations [3].
4. **Record keeping**: All orders, executions, and cancellations are logged immutably on-chain.
5. **System integrity**: The platform maintains business continuity and disaster recovery per Regulation SCI.

2.2 Broker-Dealer Registration

The operating entity registers as a broker-dealer under Section 15 of the Securities Exchange Act of 1934. This registration carries obligations for:

- Net capital requirements (Rule 15c3-1)
- Customer protection (Rule 15c3-3)
- Anti-money laundering compliance (Bank Secrecy Act, USA PATRIOT Act)

- Books and records (Rules 17a-3, 17a-4)
- Supervision (FINRA Rules 3110, 3120)

The non-custodial architecture satisfies Rule 15c3-3 in a novel manner: because the platform never holds customer securities or funds (all assets remain in MPC-custodied wallets controlled by the investor), the customer protection rule's possession-or-control requirements are satisfied by the investor's own cryptographic control.

2.3 Transfer Agent Registration

The platform operates as a registered transfer agent under Section 17A of the Securities Exchange Act. Transfer agent obligations and their on-chain implementations are described in the companion paper [4].

3 System Architecture

3.1 Deployment Model

The ATS is implemented as a single Go binary (`liquidityio/ats`) running on Hanzo Base. Hanzo Base provides an embedded SQLite-compatible database with real-time subscriptions, authentication, and file storage. The binary packages all ATS functionality into a single deployable unit.

Table 1: ATS system components.

Component	Count	Description
Collections	32	Data models (orders, securities, accounts, ...)
API routes	64	REST endpoints (/v1/...)
Cron jobs	19	Scheduled tasks (settlement, compliance, ...)
Event hooks	47	Real-time triggers on collection changes
MPC signers	3	Threshold signature participants

3.2 Collection Schema

The 32 collections partition the system state into distinct concerns:

Table 2: Collection groups and their members.

Group	Collections
Identity	accounts, investors, accreditations, kyc_checks, aml_screens
Securities	securities, offerings, partitions, documents, corporate_actions
Trading	orders, executions, order_books, quotes, rfqs
Settlement	settlements, transfers, mpc_requests, mpc_signatures
Compliance	compliance_events, regulatory_reports, audit_logs, restrictions
Integration	alpaca_accounts, alpaca_orders, simplici_checks, provider_configs
System	configs, cron_logs, webhooks, notifications

3.3 API Surface

The 64 routes follow a consistent pattern under `/v1/`:

Listing 1: Route registration pattern.

```
func RegisterRoutes(app *base.App) {
    v1 := app.Group("/v1")

    // Trading
    v1.POST("/orders", createOrder)
    v1.GET("/orders/:id", getOrder)
    v1.DELETE("/orders/:id", cancelOrder)
    v1.GET("/orderbook/:security", getOrderBook)

    // Securities
    v1.GET("/securities", listSecurities)
    v1.GET("/securities/:id", getSecurityDetail)
    v1.POST("/securities/:id/transfer", initiateTransfer)

    // Settlement
    v1.POST("/settlements/execute", executeSettlement)
    v1.GET("/settlements/:id/status", getSettlementStatus)

    // Compliance
    v1.POST("/kyc/initiate", initiateKYC)
    v1.GET("/compliance/restrictions/:investor", getRestrictions)
}
```

All routes require Hanzo IAM JWT authentication. The `owner` claim scopes all queries to the authenticated organization.

3.4 Cron Schedule

The 19 scheduled tasks maintain system invariants:

Table 3: Cron job schedule.

Job	Schedule	Function
settle_pending	every 30s	Execute pending atomic settlements on Lux EVM
match_orders	every 5s	Run matching engine against open order books
sync_alpaca	every 1m	Synchronize state with Alpaca OmniSub
kyc_poll	every 5m	Poll Simplici for KYC/AML status updates
nav_calculate	daily 00:00 UTC	Calculate NAV for fund securities
compliance_scan	every 15m	Check all positions against restriction rules
dividend_distribute	daily 06:00 UTC	Distribute dividends for eligible securities
report_generate	daily 23:00 UTC	Generate regulatory reports (OATS, CAT)
mpc_health	every 1m	Verify MPC signer liveness
lockup_expire	every 1h	Release securities from expired lockup periods
accreditation_check	daily 02:00 UTC	Verify investor accreditation currency
market_data	every 10s	Record market data snapshots
fee_sweep	daily 04:00 UTC	Sweep collected trading fees
position_reconcile	every 30m	Reconcile on-chain state with ATS records
audit_export	weekly Mon 01:00	Export audit trail for compliance review
corporate_action	every 1h	Process pending corporate actions
margin_call	every 5m	Evaluate margin positions per [5]
signer_rotate	monthly 1st 00:00	Initiate MPC key rotation ceremony
backup_state	every 6h	Backup ATS state to encrypted off-site storage

4 Non-Custodial MPC Custody

4.1 Threshold Signing Architecture

The central innovation of this ATS is non-custodial operation through MPC threshold signing. The `luxfi/mpc` library implements a 2-of-3 threshold scheme based on the GG20 protocol for ECDSA and FROST for EdDSA signatures.

Three key shares are distributed:

1. **Investor share**: Held on the investor's device, never transmitted to the platform.
2. **Platform share**: Held by the ATS in an HSM-backed enclave, used only upon authenticated instruction.
3. **Recovery share**: Held by a regulated custodian (e.g., a qualified custodian under the Investment Advisers Act), used only for key recovery.

Any 2 of these 3 shares can produce a valid signature. The platform alone cannot move

assets. The investor alone cannot bypass compliance checks. Both must cooperate for a transfer to execute.

Definition 1 (Non-Custodial ATS)**.** *An ATS is* non-custodial *if for every asset transfer τ, the platform's key share alone is insufficient to produce a valid signature σ authorizing τ. Formally, for threshold $t = 2$ and platform share index i_p, the signing function $\mathsf{Sign}(\{s_{i_p}\}, m) = \perp$ for any message m.*

4.2 Transfer Flow

A securities transfer proceeds through the following steps:

Algorithm 1 Non-custodial securities transfer.

Require: Order o, buyer B, seller S, security token T
Ensure: Atomic settlement on Lux EVM

1: ATS verifies B passes KYC/AML and accreditation checks
2: ATS verifies S holds sufficient T in partition p
3: ATS verifies transfer satisfies all restriction rules on T
4: ATS constructs settlement transaction tx
5: ATS requests MPC partial signature from S (investor share)
6: S reviews tx details on device, provides partial signature σ_S
7: ATS provides platform partial signature σ_P
8: $\sigma \leftarrow \mathsf{Combine}(\sigma_S, \sigma_P)$ ▷ Valid 2-of-3 threshold signature
9: ATS submits signed tx to Lux EVM
10: Lux EVM executes atomic transfer: T from $S \to B$, payment from $B \to S$
11: Settlement confirmed in same block (T+0)

4.3 Security Properties

Theorem 1 (Platform Non-Access)**.** *Under the 2-of-3 threshold scheme, the platform cannot unilaterally transfer any investor asset, even if the platform's HSM is fully compromised.*

Proof. The platform holds exactly one key share s_P. A valid signature requires $t = 2$ shares. Compromising s_P yields only one share. The adversary must additionally compromise either the investor's device (share s_I) or the recovery custodian (share s_R). By assumption, these are independent security domains. The platform alone is insufficient. □

Theorem 2 (Regulatory Compliance)**.** *The MPC architecture preserves the platform's ability to enforce transfer restrictions and comply with regulatory orders, despite being non-custodial.*

Proof. The platform's share is required for any 2-of-3 signature involving the platform. The platform will only provide its partial signature when the transfer satisfies all compliance checks (KYC/AML, accreditation, transfer restrictions, court orders). If a regulatory order prohibits a transfer, the platform withholds its share, making the transfer impossible through the normal (investor + platform) signing path. The recovery custodian share exists solely for key recovery under legally defined circumstances and cannot be used for trading. □

5 Order Matching

5.1 Matching Engine

The matching engine implements price-time priority for continuous order matching:

Listing 2: Matching engine core.

```
type MatchEngine struct {
    books map[SecurityID]*OrderBook
    mu    sync.RWMutex
}

type OrderBook struct {
    Bids *PriorityQueue // max-heap by price, FIFO by time
    Asks *PriorityQueue // min-heap by price, FIFO by time
}

func (e *MatchEngine) Match(secID SecurityID) []Execution {
    e.mu.Lock()
    defer e.mu.Unlock()

    book := e.books[secID]
    var executions []Execution

    for book.Bids.Len() > 0 && book.Asks.Len() > 0 {
        bestBid := book.Bids.Peek()
        bestAsk := book.Asks.Peek()

        if bestBid.Price.LessThan(bestAsk.Price) {
            break
        }

        qty := min(bestBid.Remaining, bestAsk.Remaining)
        price := bestAsk.Price // price-time priority: aggressor pays

        executions = append(executions, Execution{
            BuyOrder:  bestBid.ID,
            SellOrder: bestAsk.ID,
            Quantity:  qty,
            Price:     price,
            Timestamp: time.Now().UTC(),
        })

        bestBid.Remaining -= qty
        bestAsk.Remaining -= qty

        if bestBid.Remaining == 0 { book.Bids.Pop() }
        if bestAsk.Remaining == 0 { book.Asks.Pop() }
    }
    return executions
}
```

5.2 Order Types

The ATS supports the following order types, consistent with standard equity market conventions:

- **Limit**: Execute at specified price or better.
- **Market**: Execute immediately at best available price.
- **IOC** (Immediate-or-Cancel): Fill what is available, cancel remainder.
- **GTC** (Good-Till-Cancelled): Remain on book until filled or cancelled.
- **GTD** (Good-Till-Date): Remain on book until specified date.
- **RFQ** (Request-for-Quote): Solicit quotes from registered market makers.

6 Settlement

6.1 Atomic T+0 Settlement

Settlement occurs atomically on Lux EVM. A settlement transaction bundles the securities transfer and the payment transfer into a single atomic operation:

Listing 3: Atomic settlement contract.

```
contract AtomicSettlement {
    function settle(
        address securityToken,
        address paymentToken,
        address seller,
        address buyer,
        uint256 securityAmount,
        uint256 paymentAmount,
        bytes calldata mpcSignature
    ) external {
        require(
            verifyMPCSignature(
                keccak256(abi.encodePacked(
                    securityToken, paymentToken,
                    seller, buyer,
                    securityAmount, paymentAmount
                )),
                mpcSignature
            ),
            "invalid MPC signature"
        );

        IERC1400(securityToken).transferByPartition(
            DEFAULT_PARTITION,
            seller,
            buyer,
            securityAmount,
            ""
        );

        IERC20(paymentToken).transferFrom(
            buyer, seller, paymentAmount
        );

        emit Settled(
            securityToken, seller, buyer,
            securityAmount, paymentAmount,
            block.timestamp
        );
    }
}
```

6.2 Settlement Finality

Lux consensus provides probabilistic finality within 1–2 seconds. For securities settlement, the ATS waits for deterministic finality (block confirmation) before updating the settlement status. The settlement timeline comparison:

Table 4: Settlement time comparison.

System	Settlement Time	Counterparty Risk
Traditional equities (DTCC)	T+1	Full during settlement period
tZERO	T+0 (claimed)	Custodial during settlement
Securitize Markets	T+1 to T+3	Custodial during settlement
INX	T+0 to T+2	Custodial during settlement
Lux ATS	**T+0 atomic**	**None (non-custodial)**

7 Integration Layer

7.1 Alpaca OmniSub

For traditional equities and hybrid securities, the ATS integrates with Alpaca via OmniSub for brokerage services:

- Account creation and management via Alpaca Broker API
- Order routing for listed equities (NYSE, NASDAQ)
- Position synchronization between on-chain and off-chain holdings
- Fractional share support for tokenized equity wrappers

7.2 Simplici KYC/AML

Investor identity verification uses Simplici for:

- Document verification (government-issued ID, proof of address)
- Biometric authentication (liveness detection, face matching)
- AML screening (OFAC, PEP, adverse media)
- Accredited investor verification (income, net worth, professional certifications)
- Ongoing monitoring (transaction monitoring, periodic re-verification)

KYC status is stored in the `kyc_checks` collection and linked to the investor's I-Chain DID, enabling portable identity across the Lux ecosystem [4].

7.3 I-Chain Identity

Investor identity on the Lux ATS is anchored to the I-Chain decentralized identity system. Each investor has a DID (Decentralized Identifier) that links their KYC/AML verification status, accreditation credentials, and jurisdictional permissions. Transfer restriction checks reference the investor's DID claims rather than centralized database records, enabling:

- Portable identity across multiple securities platforms on Lux
- Zero-knowledge proofs of accreditation without revealing underlying documents
- Revocable credentials for expired accreditations or regulatory sanctions

8 Platform Comparison

Table 5: Comparison with existing digital securities platforms.

Feature	**Lux ATS**	**tZERO**	**Securitize**	**INX**
Non-custodial	Yes	No	No	No
MPC threshold signing	2-of-3	N/A	N/A	N/A
Atomic settlement	T+0	T+0*	T+1–3	T+0–2
Post-quantum ready	Yes	No	No	No
On-chain cap table	Yes	Partial	Off-chain	Partial
Open-source protocol	Yes	No	No	No
Multi-jurisdiction	Yes	US only	US + EU	US + EU
ERC-1400 compliant	Yes	Proprietary	ERC-3643	Proprietary
DID-based identity	Yes	No	No	No
Integrated order router	Yes	Limited	No	No

*tZERO claims T+0 but requires custodial intermediary during settlement.

9 Security Analysis

9.1 Threat Model

We consider the following adversaries:

1. **External attacker**: Attempts to steal assets by compromising platform infrastructure.

2. **Malicious insider**: Platform employee attempts unauthorized transfers.
3. **Compromised HSM**: Platform's HSM is fully compromised, exposing the platform key share.
4. **Regulatory seizure**: Government orders asset freeze or forfeiture.
5. **Quantum adversary**: Future attacker with access to a cryptographically relevant quantum computer.

9.2 Mitigation

Table 6: Threat mitigation through non-custodial MPC.

Threat	Mitigation
External attacker	Platform key share alone insufficient; attacker must also compromise investor device
Malicious insider	Same as external attacker; no single insider has access to 2 shares
Compromised HSM	Investor retains independent share; recovery custodian can facilitate key rotation
Regulatory seizure	Platform can freeze by withholding its share; no need to seize keys
Quantum adversary	Lux post-quantum precompiles enable migration to PQ-safe threshold schemes [1]

9.3 Operational Security

- MPC key shares are generated in a distributed ceremony; no single party ever sees the full private key.
- The platform's share is stored in an HSM with FIPS 140-2 Level 3 certification.
- Key rotation occurs monthly via the `signer_rotate` cron, using proactive secret sharing to rotate shares without changing the public key.
- All MPC signing sessions are logged immutably in the `mpc_requests` and `mpc_signatures` collections.

10 Regulatory Reporting

The ATS generates the following regulatory reports:

- **OATS** (Order Audit Trail System): All order events with timestamps, required by FINRA.
- **CAT** (Consolidated Audit Trail): Customer and order event information, required by SEC Rule 613.
- **Form ATS-N**: Quarterly operational reports filed with the SEC.
- **FOCUS Reports**: Monthly and quarterly financial reports, required by FINRA Rule 4524.
- **SAR/CTR**: Suspicious Activity Reports and Currency Transaction Reports per FinCEN requirements.
- **Blue Sky**: State-level notice filings for securities offerings.

All reports are generated from on-chain data, ensuring that regulatory submissions exactly reflect the executed state of the system. The immutability of on-chain records eliminates the possibility of post-hoc report manipulation.

11 Performance

Table 7: ATS performance characteristics.

Metric	Value
Order submission latency (API)	< 5 ms
Matching engine throughput	10,000 orders/sec
MPC signing latency (2-of-3)	< 200 ms
Settlement finality (Lux EVM)	1–2 seconds
End-to-end trade lifecycle	< 3 seconds
Concurrent order books	1,000+
Binary size (single Go binary)	∼45 MB
Memory footprint (idle)	∼128 MB
Memory footprint (peak)	∼512 MB

12 Conclusion

We have presented the architecture of a non-custodial ATS built on Lux Network that eliminates custodial risk through MPC threshold signing while maintaining full regulatory compliance with SEC Regulation ATS, broker-dealer requirements, and transfer agent obligations. The system descends directly from the Arca securities infrastructure (2018) and represents the culmination of seven years of work on digital securities.

The key architectural decisions—single Go binary on Hanzo Base, 2-of-3 MPC threshold signing via `luxfi/mpc`, atomic T+0 settlement on Lux EVM, DID-based investor identity—together produce a system that is simultaneously the safest (non-custodial), most decentralized (no single custodian), and most regulatory-compliant (SEC-registered ATS/BD/TA) digital securities platform in operation.

The companion papers on the security token standard [2], smart order routing [3], and transfer agent architecture [4] describe the subsystems in detail. Together, these papers constitute a complete specification for regulatory-compliant, non-custodial securities infrastructure on Lux Network.

References

[1] Z. Kelling, V. Seesahai, "M-Chain: Decentralized Multi-Party Computation Custody with Quantum-Safe Threshold Signatures," Lux Partners, 2023.

[2] Z. Kelling, "Lux Security Token Standard: ERC-1400 on Post-Quantum EVM with MPC Custody," Lux Partners, 2025.

[3] Z. Kelling, "Smart Order Routing for Digital Securities: Multi-Provider Execution on Lux Network," Lux Partners, 2025.

[4] Z. Kelling, "Blockchain Transfer Agent: On-Chain Cap Table with MPC-Custodied Registry," Lux Partners, 2025.

[5] Z. Kelling, "Lux Credit Protocol Specification," Lux Partners, 2023.

[6] Securities and Exchange Commission, "Regulation of Exchanges and Alternative Trading Systems," Release No. 34-40760, 1998.

[7] Securities and Exchange Commission, "Customer Protection—Reserves and Custody of Securities," Rule 15c3-3, Securities Exchange Act of 1934.

[8] Financial Industry Regulatory Authority, "Consolidated Audit Trail (CAT)," FINRA Rule 6800 Series.

[9] A. Dossa, P. Ruiz, F. Vogelsteller, S. Gosselin, "ERC-1400: Security Token Standard," Ethereum Improvement Proposals, 2018.

[10] R. Bose, J. Cawley, "ERC-1404: Simple Restricted Token Standard," Ethereum Improvement Proposals, 2018.

[11] R. Gennaro, S. Goldfeder, "One Round Threshold ECDSA with Identifiable Abort," Cryptology ePrint Archive, 2020.

[12] C. Komlo, I. Goldberg, "FROST: Flexible Round-Optimized Schnorr Threshold Signatures," SAC 2020.