

# Lux for Coinbridge.US

A sovereign, post-quantum, non-custodial stack for regulated securities tokenization

Lux Network

## Abstract

This document describes what Lux proposes to deliver to Coinbridge.US as our first white-label customer: a complete, regulated digital-securities platform that Coinbridge.US operates under its *own* brand, on its *own* sovereign Layer-1 blockchain, with keys it never has to custody. The platform folds a core bank, banking-as-a-service, payment rails, a general-securities tokenization engine, an alternative trading system, broker-dealer and transfer-agent functions, and a non-custodial custody service into a single binary whose every capability can be turned on or off per deployment. The chain underneath it carries its own EVM, its own Uniswap-compatible DEX, and an FHE coprocessor for confidential compute. Signing is post-quantum throughout. A live demonstration runs on two isolated networks — testnet and devnet — reachable today at `*.testnet.lux.financial` and `*.devnet.lux.financial`.

## 1 The proposition

Coinbridge.US needs to issue, trade, settle, and custody tokenized securities under U.S. and offshore regulatory regimes, with the operational independence of running its own infrastructure rather than renting a slot on someone else's. Lux delivers exactly that as a **white-label reference stack**: a single deployable that Coinbridge.US brands as its own, configures to the capabilities it is licensed for, and runs on a sovereign chain it controls.

Three properties define the offer:

1. **Sovereign.** Coinbridge.US runs its own Layer-1 — its own validator set, its own EVM chain id, its own DEX and FHE coprocessor. The platform does not depend on any shared mainnet to function. Any customer can launch its own L1 the same way.
2. **Non-custodial.** Private keys are never assembled in one place. Custody is threshold MPC with an FHE policy layer; Coinbridge.US — and Lux — can be structurally incapable of unilaterally moving a client's assets.
3. **Post-quantum.** Service identity and signing use lattice-based ML-DSA today, so records signed now remain verifiable after a cryptographically-relevant quantum computer exists.

The remainder of this document walks the capabilities and the live demo.

## 2 The sovereign Layer-1

Each deployment settles on its own sovereign L1. Following the sovereign-L1 model, the primary network id *equals* the EVM chain id — one identifier names the chain everywhere (wallets, tools, staking-key derivation), eliminating the ambiguity of separate network and chain ids. The reference demo runs two isolated sovereign networks:

| Environment | Network id (= EVM chain id) | Demo hosts                           |
|-------------|-----------------------------|--------------------------------------|
| testnet     | 5                           | <code>*.testnet.lux.financial</code> |
| devnet      | 6                           | <code>*.devnet.lux.financial</code>  |

These ids are deliberately distinct from any production chain id. Each network carries:

- a full **EVM** with the complete precompile set, so standard Solidity tooling, audited token templates, and existing contracts work unmodified;
- a **Uniswap-compatible DEX** for on-chain secondary liquidity in tokenized instruments, white-labeled into Coinbridge.US's swap surface;
- an **FHE coprocessor** for confidential on-chain compute — encrypted balances, private order matching, and policy checks that never expose the underlying values.

A block explorer is published per network so Coinbridge.US and its auditors can inspect every issuance, transfer, and settlement.

### 3 General-securities tokenization

The tokenization engine treats a security as an ERC-20-shaped on-chain instrument wrapped in a regulatory record. It supports a general, extensible set of asset classes:

| Class       | Examples                                    |
|-------------|---------------------------------------------|
| equity      | common / preferred shares, SAFE conversions |
| debt        | notes, bonds, revenue-based instruments     |
| fund        | LP interests, tokenized funds               |
| note        | convertible and promissory notes            |
| warrant     | options and warrants                        |
| real_estate | fractional property, REIT units             |
| commodity   | metals, carbon, tokenized reserves          |
| stablecoin  | fiat-referenced settlement tokens           |

Each issuance records the issuer, the asset class, decimals and supply (in base units), the jurisdiction, and whether the instrument is transfer-restricted (e.g. Reg D / Reg S lock-ups). Off-chain the issuance is a signed registry entry; on a connected chain it deploys the contract and anchors the registry id, so the registry and the chain agree by construction. Symbols are unique per issuer.

The flow is live in the demo today. Issuing an instrument is a single call — the platform validates the request, scales the amount to base units, persists the regulatory record, and (with a chain connected) deploys and mints. The same procedure is reachable as an HTTP endpoint and as a binary RPC the admin console uses directly.

### 4 Non-custodial custody (MPC + TFHE)

Custody is built on threshold MPC: a private key is split into shares held by independent parties and *never* reconstructed — signatures are produced by a distributed protocol, so no single host (and no single operator) ever holds a spendable key. An FHE/TFHE policy layer evaluates spending policy over encrypted state, so authorization checks — limits, allow-lists, dual control — run without decrypting balances or beneficiary data.

For Coinbridge.US this means client assets are protected by cryptography, not merely by access control, and the custody posture is demonstrable to a regulator: the system is *structurally* non-custodial.

## 5 Regulated trading and transfer

**ATS** An alternative trading system with a high-throughput central-limit order book for tokenized instruments, multi-tenant from day one, settling on the sovereign L1.

**Broker-Dealer** Order intake, suitability, and execution surfaces for the regulated broker-dealer function.

**Transfer Agent** The authoritative cap table: holders, positions, restrictions, corporate actions — kept in lock-step with on-chain balances.

These compose with tokenization and custody as modules in the same binary, so an instrument issued by the tokenization engine is immediately tradeable on the ATS and tracked by the transfer agent without integration glue.

## 6 Banking: BaaS and payments

The core bank ledger fronts a banking-as-a-service backend. The reference integration targets **SF Private Bank LTD** (International Banking Licence L17601/SFPB, Union of Comoros Offshore Finance Authority — FSA Offshore; LEI 254900Y05KODH3HSN321) as the BaaS provider; a customer swaps in its own provider through one configuration block. A payment service provider module supplies fiat on/off ramps and settlement rails. Credentials are resolved from KMS at boot — never inlined.

## 7 White-label and sovereignty

Everything a customer sees is data, not code. Brand name, legal name, domain, logo, colors, and the regulatory licence citation come from one white-label configuration resolved per request by hostname. The product ships brand-neutral; Coinbridge.US supplies its identity and the platform renders as Coinbridge.US's own. The capability set is gated the same way: Coinbridge.US enables exactly the modules it is licensed to operate, and the rest are never mounted.

Because the chain is sovereign, Coinbridge.US owns the network — and any future customer launches its own L1 by the same procedure. There is no shared multi-tenant chain to trust.

## 8 The live demonstration

The demo is deployed on the two isolated sovereign networks above. On each of `*.testnet.lux.financial` and `*.devnet.lux.financial`, Coinbridge.US will see:

- `admin.{testnet,devnet}.lux.financial` — the unified, brand-configurable console listing every composed capability;
- `ats. /bd. /ta. /baas.` — the regulated stack surfaces, each on its subdomain;
- the **tokenization** flow end-to-end: issue a tokenized security, see it on the cap table and in the explorer;
- the **swap** surface: a Uniswap-compatible DEX trade of a tokenized instrument;
- the **explorer**: every issuance, transfer, and settlement on-chain;
- post-quantum signing and non-custodial custody operating underneath all of it.

The networks are demo-only (testnet/devnet); production for a licensed customer settles on that customer's own mainnet L1.

## 9 Architecture

The stack is one Go binary. Each capability — bank, BaaS, PSP, tokenization, ATS, broker-dealer, transfer-agent, custody — is a module that mounts its own HTTP routes and binary RPC procedures onto a shared host; the host gates which modules are enabled per customer. A module never owns a process, so the identical code runs as one folded binary or as independent services. Persistence is an encrypted embedded store; transport is a single binary wire protocol that browsers reach over WebSocket, so the console and back-end services share one procedure registry. The build is pure Go — no native toolchain — and post-quantum signing is on by default.

## 10 Engagement

We propose a staged engagement: (1) Coinbridge.US explores the live demo on the testnet and devnet sovereign L1s; (2) we configure Coinbridge.US’s brand, jurisdiction, licence citation, and enabled capabilities; (3) we stand up Coinbridge.US’s own sovereign mainnet L1 under Coinbridge.US’s validators and brand. The platform Coinbridge.US evaluates is the platform Coinbridge.US ships — the only differences are configuration and the network it settles on.