GSCX — Tech Stack

Node.js LTS | REST + WebSocket | Blockchain

Full-system technology overview · GSCX API + Custodian Blockchain Engine

SYSTEM 1 — GSCX API

Exchange & Member Platform

Stateless REST API serving the member portal SPA. Handles authentication, role-based access, bonds, products, and transactions.

Runtime	Node.js (no transpile)
Framework	Express 4.21.2
Database	MariaDB 10.x+
Auth	TOTP · RFC 6238
Ports	3000 (HTTP) · 3443 (HTTPS)
Endpoints	22 REST under /api/*

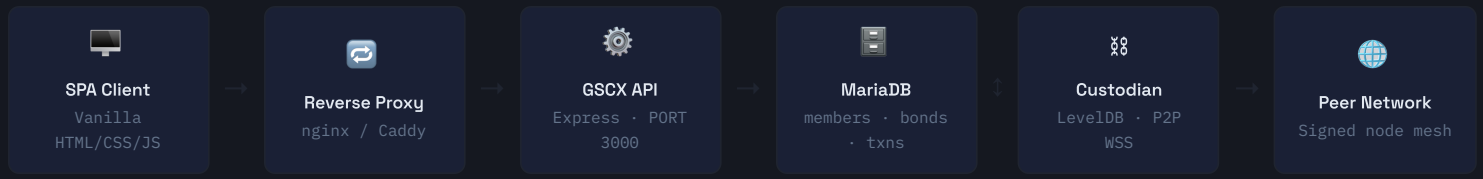
SYSTEM 2 — CUSTODIAN ENGINE

Blockchain & Settlement Layer

Custom blockchain engine with P2P networking, a native order book, wallet management, and optional fiat rails via Stripe.

Runtime	Node.js ES Modules
Framework	Express + WebSocket (ws)
Ledger DB	LevelDB (level pkg)
Crypto	secp256k1 · Ed25519 · BIP39
Transport	WSS peer mesh
Fiat Rails	Stripe SDK (optional)

ARCHITECTURE FLOW



LANGUAGES & RUNTIME

JavaScript (ES2020+) | Node.js LTS | ES Modules | Vanilla HTML | CSS Custom Props | No build step

FRAMEWORKS & LIBRARIES

Express 4.21.2 | cors 2.8.5 | morgan 1.10.0 | dotenv 16.4.7 | body-parser | ws (WebSocket) | qrcode 1.5.4 | uuid | elliptic

DATA PERSISTENCE

MariaDB 10.x+ | LevelDB | mariadb driver 3.4.0 | Connection pooling | JSON columns | File-based asset vault

Append-only audit chain

SECURITY & CRYPTOGRAPHY

RFC 6238 TOTP
HMAC-SHA1
SHA-256
secp256k1
Ed25519
BIP39 Mnemonics
48-char hex sessions
Parameterized SQL
TLS / Let's Encrypt

INFRASTRUCTURE & DEPLOYMENT

Docker Compose
PM2
systemd
nginx / Caddy
Let's Encrypt TLS
Health endpoints
Startup validation

NETWORKING & P2P

WebSocket (WSS)
Signed node handshakes
Host allowlisting
Trusted peer pinning
Key fingerprinting
Tx propagation

TRADING & MARKETPLACE

Custom order book
Matching engine
Price oracle
Market depth analytics
Trade records
Stripe SDK (fiat on/off ramp)

ACCESS CONTROL

8 member role categories
RBAC middleware
Two-step login
TOTP-gated privileged ops
In-process session Map
Redis-ready (HA)

FRONTEND

Single-file SPA shell
Dark theme
CSS custom properties
Space Grotesk
IBM Plex Mono
QR code (CDN)
Responsive layout

WHY THIS CHANGES EVERYTHING

One platform. Eight parties. Zero intermediary friction.

Cross-border trade has always required a chain of disconnected systems — exporters emailing invoices, banks manually verifying documents, insurers issuing paper certificates, logistics providers sending PDFs. GSCX collapses that entire chain into a single cryptographically secured exchange where every party operates in real time on the same ledger, with the same information, under the same rules.

THE EIGHT PARTIES — ALL IN ONE SYSTEM

ADMIN

Platform operators with full system access and governance controls.

EXPORTER

Sellers listing products for international trade, matched directly to buyers.

IMPORTER

Buyers purchasing across borders, with order matching handled on-chain.

BANK

Financial institutions providing trade finance and settlement backing.

INSURANCE

Risk and underwriting providers covering shipment and counterparty exposure.

TRANSPORT

Logistics operators tracking and confirming physical delivery of goods.

REMITTER

Payment entities executing fund transfers between trading counterparties.

INSPECTOR

Compliance and quality officials verifying goods before settlement releases.

8

Distinct participant roles on a single regulated ledger

0

Third-party clearinghouses required for settlement

22

REST endpoints powering every trade lifecycle event

1 Every party sees the same truth

Today, an importer, their bank, and the insurer all hold separate records of the same trade. Disputes arise from discrepancies between those records. On GSCX, every party — exporter, importer, bank, insurance, transport, remitter, and inspector — reads from and writes to the same cryptographically linked ledger. There is no "version" to reconcile.

2 Settlement without a clearinghouse

Traditional cross-border payments route through correspondent banks, SWIFT messages, and central clearinghouses — each adding cost, delay, and a point of failure. The Custodian blockchain engine settles directly between parties using signed transactions on a shared ledger, with bonds as collateral and remitters handling fund movement. No middleman holds the float.

3 Compliance is structural, not procedural

The Inspector role exists at the protocol level — inspectors must sign off before certain transactions can proceed. This is not a checkbox in a workflow tool; it is enforced by the API's role-based access control. Compliance cannot be skipped because the system architecturally prevents it. Auditors get an append-only, hash-chained record they can verify independently.

4 Fiat and digital assets in one flow

Most blockchain trade platforms force participants to work entirely in crypto. GSCX bridges both worlds: Stripe handles fiat on/off ramps for parties who need to pay or receive in traditional currency, while the Custodian engine manages the on-chain asset leg. A bank can wire USD in; the exchange matches it to a token-settled trade on the other side.

5 Built to run anywhere, govern locally

6 No proprietary lock-in

Because GSCX is self-hosted — not dependent on any third-party cloud exchange — it can be deployed within a jurisdiction's regulatory boundary. A trade corridor between two countries can run its own node cluster, with host allowlisting ensuring only vetted participants join the network. Sovereignty over financial infrastructure stays with the operator.

The entire stack is open JavaScript on standard infrastructure — Node.js, MariaDB, LevelDB, Docker. There are no proprietary SDKs, no vendor-managed keys, no subscription to a SaaS exchange. Any competent engineering team can audit, fork, or extend it. That matters enormously for sovereign financial infrastructure where trust in the code must be verifiable.

What it is

GSCX is a full-stack financial exchange platform built entirely in JavaScript — no compiled languages, no heavyweight frameworks. It runs on Node.js and is intentionally lean: the frontend is a single HTML file, the backend is a straightforward Express API, and the data lives in a standard MariaDB relational database. The philosophy is simplicity you can actually read and maintain.

Member management & access control

The platform supports eight distinct member categories, each with different levels of access enforced at the API layer. Logging in requires two steps — a password followed by a time-based one-time code (TOTP) generated by an authenticator app. Sensitive operations are gated behind this second factor, meaning a stolen password alone is not enough to cause harm. Sessions are tracked server-side using cryptographically random tokens.

Blockchain & settlement

Beneath the exchange sits a custom-built blockchain engine called Custodian. Rather than relying on an existing chain, the team wrote their own: blocks are linked by SHA-256 hashes, transactions are signed with industry-standard elliptic curve keys (secp256k1 — the same curve Bitcoin uses), and wallets can be generated or recovered from a human-readable mnemonic phrase (BIP39). The ledger state is stored in LevelDB, a fast key-value store optimised for write-heavy workloads like recording transactions.

Peer-to-peer networking

Custodian nodes connect to each other over encrypted WebSocket connections (WSS). Each node has a signed identity, so peers can verify who they are talking to before accepting data. Nodes can be configured to only trust a known list of peers, reducing the attack surface. When a new transaction or block is created, it propagates across the mesh automatically.

Trading engine

The platform includes a fully custom order book and matching engine — the core of any exchange. When a buyer and seller agree on a price, the engine matches their orders, records the trade, and updates balances. A built-in price oracle monitors market conditions and can apply configurable thresholds to protect against runaway prices or manipulation. Market depth data (how many orders exist at each price level) is available for analytics.

Fiat on/off ramp

For users who want to move real money in or out, the system optionally integrates with Stripe — a mainstream payments processor. This means members can fund accounts with a credit card or bank transfer and withdraw back to fiat without the platform needing to build its own payment infrastructure.

Security posture

All database queries use parameterised statements, closing the door on SQL injection. Cryptography relies on Node's built-in `crypto`

Deployment & operations

The system is designed to run behind a standard reverse proxy (nginx or Caddy), which handles TLS termination and routes traffic

module rather than third-party libraries, reducing the dependency surface. TLS certificates are issued via Let's Encrypt (free, auto-renewing). An append-only audit chain with hash-linked records means tampering with historical data is detectable.

to the Node process. Process management is handled by PM2 or systemd — both will automatically restart the service if it crashes. For teams that prefer containers, Docker Compose configs are provided. Health and security endpoints let monitoring tools check system status without requiring full access.