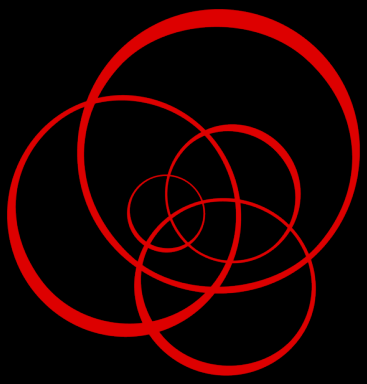


Modulus®

Exchange Solution – Technical Info

Introduction

The Modulus Exchange Solution is a turnkey solution for block-chain based token trading and broker businesses. The solution has been purposefully designed with a semi-monolithic architecture with a small number of micro-services. The block-chain components are executed in a secure server-side environment. The matching, risk and trade surveillance engines have all been developed in Golang. The core business logic for block-chain manipulation has been developed in the C# programming language for rapid development and ease of customization.

Following best security practices, the solution discourages storage of private keys, mnemonic phrases, or secret keys anywhere within the file storage, blob objects, databases, or configuration files on the servers.

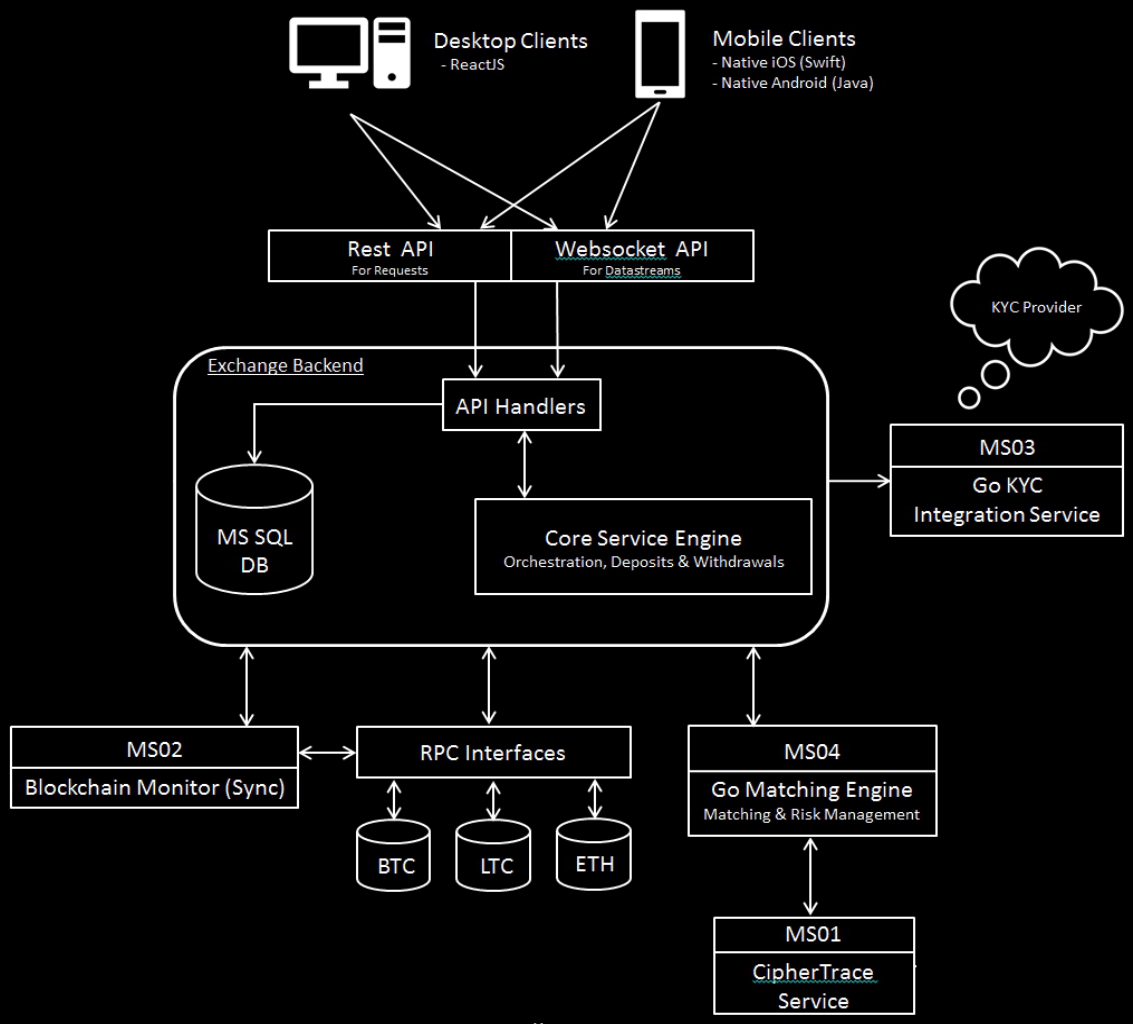
All private keys are generated client-side and remain with the client. Clients are never requested to share their private keys with anyone from Modulus.

Key features

- 100% secure proprietary code. No open source (commercial source code licenses are available).
- All business logic resides on the server side.
- 100% cold storage Integration for Ethereum and ERC20 tokens using 12 word mnemonic phrases. Keys never leave the computer.
- Official core / Qt clients as hot wallet for BTC, BCH, BTG, DOGE, DASH, LTC, USDT and other Bitcoin forked coins. Private keys never leave the computer and are protected by having a wallet.dat file encrypted with a user defined paraphrase.
- Extended public key based cold wallet integration for popular coins like BTC, BCH, LTC are available.
- Easy to list new coins and tokens. Bitcoin-forks and ERC20 Tokens listing takes less than a minute.
- Server side wrappers for interacting with block-chain nodes.
- 10M to ~100M orders per second transaction processing speed.
- Easy to scale using MSSQL replication & clustering.
- Load balancer friendly design. Cross device persistent session authentication and validation.
- Using MSSQL inbuilt SQL replication, real time data backups can be provisioned on real servers for disaster recovery.
- Built-in caching engine to cache order book, charting data and more.
- Built-in support for JS & CSS bundling.
- Extensive use of minified client side scripts.
- Code first entity framework (object-relationship mapping framework) for DB interaction instead of SQL queries & heavy dependency on stored procedures (helps prevent SQL injection attacks).
- Http header obfuscation.

- Anti XSS code practices.
- Code access policies to disallow client side scripts from unknown / unapproved sources.
- Rest API (available), FIX API (future release).
- HMAC protection for replay attacks.
- Supports Google 2FA authentication.
- Supports SSL / TLS.
- 100% cold storage integration for Ripple & Stellar Blockchain.
- Easy to customize colors and UI. Separate files for design & business logic.
- SignalR sockets for real-time data streaming.
- Messaging Bus (NATS) for order ingestion.
- Referral Module available (referral code, referral link, referral reporting in user and admin panel).
- Fee discounting module available. Trade volume based and pay-by-exchange token system.

System Architecture



View Full Size

Core Engine (semi-monolithic design)

Our exchange core engine is comprised of customer profile management, customer wallets, deposits, withdrawals, trade engine, order management, risk engine, trade surveillance, charting, transactions reports, cold wallet integration, hot wallet integration, administrative reports built right into one package/assembly using .NET framework and for IIS 7 and above.

List of micro-services:

1. Block-chain Monitoring Service (BMS): This service has the responsibility of monitoring various block-chains and listens for incoming payments. This service keeps a local copy of all addresses it has given to the DB service and intimates DB service soon after a transaction on a block-chain receives the desired number of confirmations. The prime function is to deliver real-time deposit notifications to the DB server and generate email notifications to clients.
2. CipherTrace™: This is an additional component of the service mentioned in #1. This service checks both incoming & outgoing transactions with a third party repository to see if the funds originate from legitimate sources or legal entities. If the transaction is found to be suspicious or originates from a stolen account, this module generates an admin alert and ensures such transactions do not get funded. An exchange owner can respond to such alerts by providing manual intervention.
3. Modulus-KYC: A hub is provided which connects with various KYC providers such as Trulioo, Jumio, IdentityMind, Sum & Substance, SynapseFi, ShuftiPro and others. With this module, you can pass identity documents to the backend service providers.
4. Risk-AML-Assessment: This module monitors for market manipulation and money laundering. These modules generate email and admin notifications for exchange the admin's perusal.
5. BOTS (additional as per requirement): Custom trading bots for market making, liquidity, and price-chart syncing can be provided.

Front End

React, Twitter bootstrap, CSS5 and MVC Razor. Native mobile apps.

Back End

MSSQL Sever 2016 Standard Edition

Block-chain Nodes

We use official block-chain nodes commonly known as core wallets or QT clients. Typically all Bitcoin & Bitcoin forked coins have QT clients for Windows, Linux and Macintosh. QT clients can be hosted anywhere and on any OS. These core wallets perform the following:

1. Synchronize with the block-chain and retain a copy of the entire block-chain on the server.

2. Maintain a wallet.dat file. The wallet is a non-hd wallet, therefore all addresses generated in this wallet will always have a distinct private key.

3. The wallet.dat file is encrypted with a passphrase which is set by the exchange admin. Encryption prevents theft of keys.

4. The wallet.dat file remains locked all the times. Whenever an exchange admin wishes to withdraw funds to cold wallets or process a customer's withdrawals, the wallet.dat file must be unlocked. Normally when this wallet is unlocked, it has a timeout after which the wallet locks itself again. Normally it takes less than a second to create, sign and broadcast a transaction, so our software unlocks the wallet for a very short duration.

5. This wallet paraphrase can be changed anytime and the admin must backup these .dat files regularly.

6. Admins can decide if they wish to use cold wallets for incoming funds instead of hot wallets. Outgoing transactions will always be processed using hot wallets and therefore funds must be moved to the hot wallet from the cold wallet if funds arrive originally in the cold wallet. We offer pubkey based cold wallet integration.

7. These core wallets have a daemon running in the background at the times to remain connected to several other nodes on the network and regularly listen to new transactions.

8. Because we use official wallets for each specific coin, unspent outputs from many transactions can be combined together into one output address.

9. These wallets have an RPC interface, which are enabled and configured to allow the exchange engine to communicate with the wallets. These RPC interfaces are configured to listen to requests from specific IPs, which ensures that only our application is able to communicate for wallet operations.

10. Ethereum is the only coin which does not offer combining multiple unspent inputs into one transaction. This is why full cold storage HD wallet integration is provided, where one mnemonic phrase is used to generate numerous addresses and a facility is provided in the admin panel for funds from multiple addresses to be combined to one address manually, as and when needed. Alternatively we also offer an auto-forwarded-contract address, which forwards incoming payments to primary address as soon as they arrive.

These auto-forwarded-contract addresses are deployed on the block-chain and a gas is spent when they are created. However, the HD-wallet address where manual collection is performed is gas-free and does not require any expense for their creation.