

# EX

5HT AUG 25 2019

## TOC

- UI JS CSS
- ARCH ERLANG ELIXIR
- HEX.PM

## 1. EXCHANGE USER INTERFACE

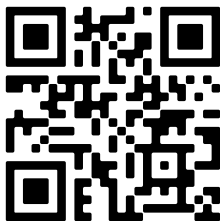
- AUTH (Google vs PKI)
- Market Assets/Tickers
- Market Data History
- Depth of Market Order Book
- Order Maker
- Personal Wallet

NOTES: Usually Exchanges provide Google Auth secured own token issue mechanism based on HashiCorp Vault or other key stores. While we see most powerful security based on ECC/RSA Public Key Cryptography.

# EX/UI JS CSS

5HT AUG 25 2019

## 1.1. GOOGLE AUTHENTICATOR

<div>AUTH</div>		<div>TIC</div>	<div>TRADE</div>	<div>FIN</div>
Mail:	<input type="text"/>			
Passphrase:	<input type="text"/>			
2PA:	<input type="text"/>			

## 1.2. PUBLIC KEY INFRASTRUCTURE

Login/Enroll	Logout/Revoke
<div>Load Client X.509 certificate from file <div>Load</div></div> <div><div>X</div> Create new X.509 certificate</div> <div>Organizational Unit: <input type="text"/></div> <div>Common Name: <input type="text"/></div> <div>Enroll new X.509 certificate <div>Enroll</div></div>	<div><div>X</div> Revoke X.509 certificate at CA</div> <div>Destroy certificate at device. <div>Delete</div></div>

# EX/UI JS CSS

5HT AUG 25 2019

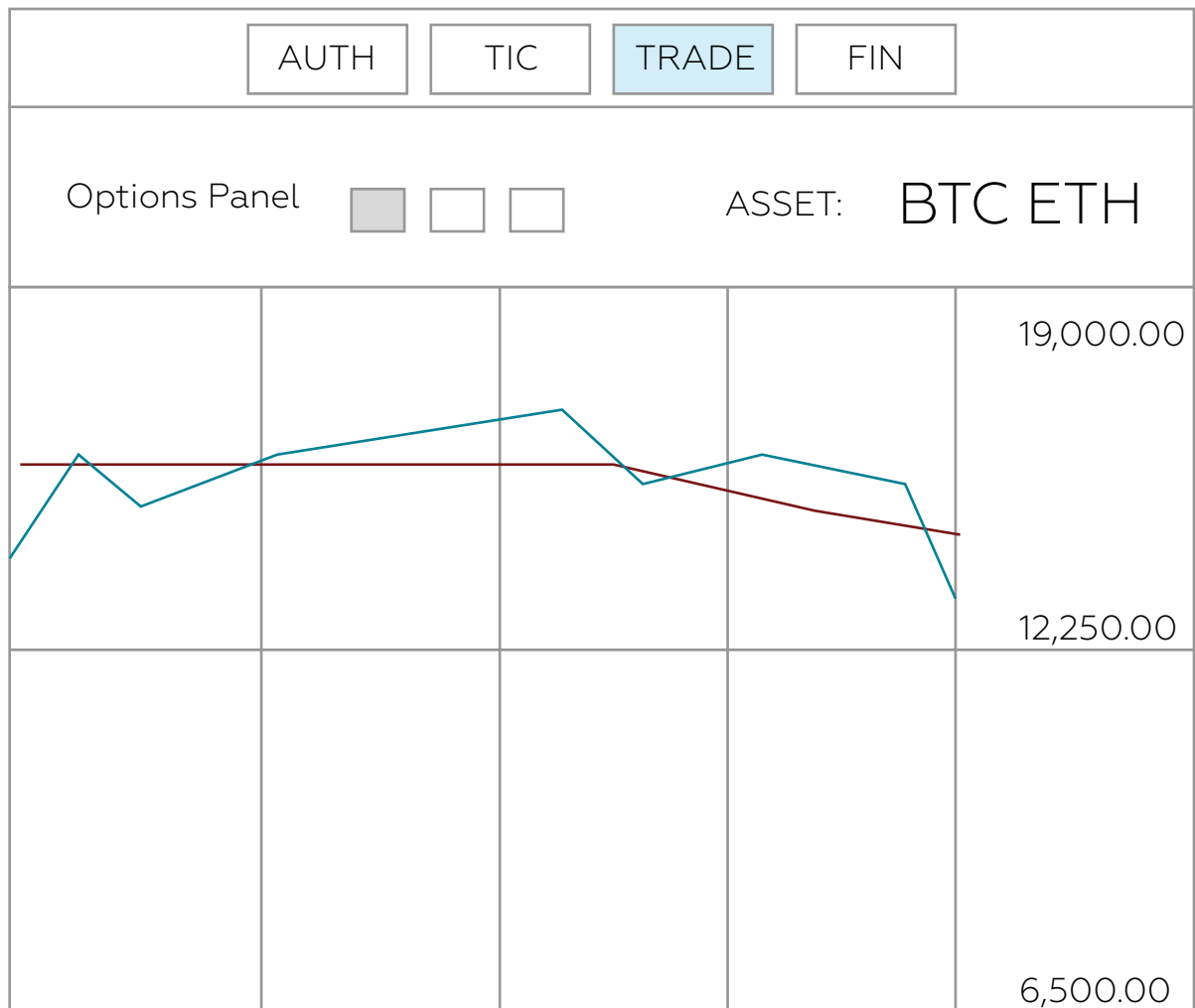
NOTES: Usually Exchanges provide Google Auth secured own token issue mechanism based on HashiCorp Vault or other key stores. While we see most powerful security based on ECC/RSA Public Key Cryptography.

## 1.3. MARKET

	AUTH	TIC	TRADE	FIN	
Filter::	<input type="text"/>				
Asset	Pair	Price	Volume	Chart	

NOTES: List of all tickers/assets on the given exchanges (or all tokens from peer-exchanges in case of arbitrage support).

## 1.4. MARKET DATA HISTORY and PREDICTION

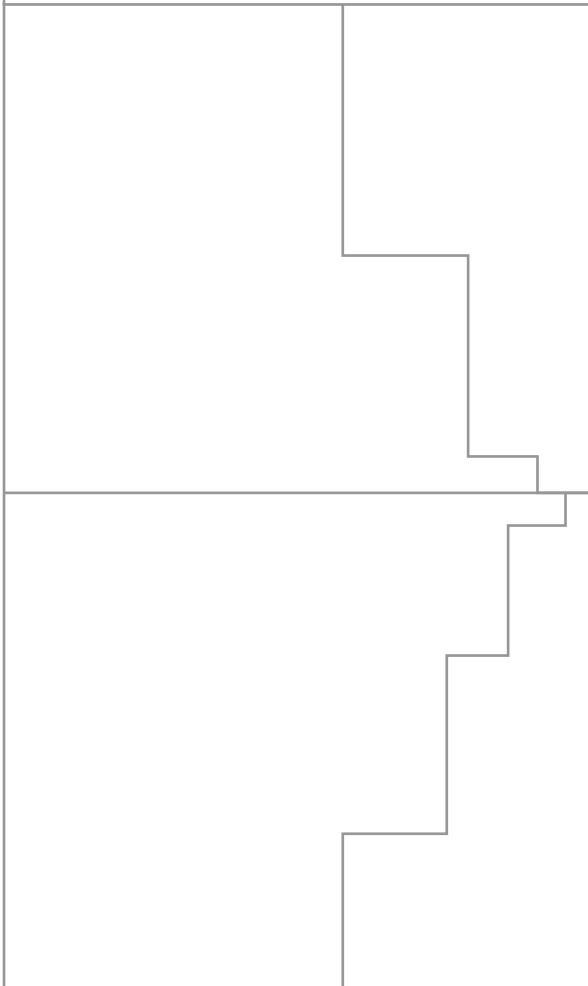


NOTES: Market Data History Control Element that is in conjunction with DOM and Order Placer forms the Trading Interface that in modern exchanges always placed at the same trading dashboard page.

# EX/UI JS CSS

5HT AUG 25 2019

## 1.5. DEPTH OF MARKET for TICKER

AUTH			TIC			TRADE			FIN		
MARKET PRICE: 11,525.40						ASSET: BTC ETH					
Options Panel			<input type="checkbox"/>	<input checked="" type="checkbox"/>	<input type="checkbox"/>						
						166	118274000000	-744526365			
						165	117076000000	-6443000000			
						163	116609000000	-3287000000			
						21	116435000000	-876000000			
						161	115617000000	-279199700			
						167	115545000000	-468957492			
						27	115332000000	-25000000			
						28	115331000000	-3000000			
						62	115284000000	-301020000			
						109	115283000000	-28000000			
						66	115254000000	1000000			
						64	115253000000	29800000			
						149	114950000000	99000000			
						133	114936000000	98000000			
						130	114928000000	99000000			
						127	114920000000	97000000			
						126	114914000000	92000000			
						124	114903000000	103000000			
						123	114891000000	98000000			
						122	114884000000	108000000			
						1	114871000000	1700000000			
						121	114865000000	100000000			
						104	114854000000	265779000			
						33	114792000000	25000000			
						164	114791000000	449969320			
						53	114774000000	316000000			
						70	114753000000	195000000			
						119	001000000	8973000000			
									Depth: 28		
									Total: 393844763		

# EX/UI JS CSS

5HT AUG 25 2019

## 1.6. MAKE ORDER

AUTH		TIC		TRADE		FIN	
				ASSET: BTC ETH			
Options Panel		<input type="checkbox"/>	<input type="checkbox"/>	<input checked="" type="checkbox"/>			
Type:		<input type="text" value="Stop Loss"/>					
Volume:		<input type="text"/>					
Price:		<input type="text"/>					
Total:		<input type="text"/>					
Reset		Buy		Sell			

# EX/UI JS CSS

5HT AUG 25 2019

## 1.7. WALLET

<div>AUTH</div> <div>TIC</div> <div>TRADE</div> <div>FIN</div>	
<div>Accounts</div> <div>BTC</div> <div>UBS</div> <div>Credit Suisse</div> <div>Total</div> <div>\$5,000</div>	<div>Transactions</div> <div>2019-08-25 +\$200</div> <div>2019-07-5 -\$250</div>

# EX/ARCH ERLANG ELIXIR

5HT AUG 25 2019

## 2.EXCHANGE ARCHITECTURE

- High Performance Network Router
- WebSocket real time endpoints
- Matching Engine
- L3 Order Book Assembling
- Formatters agnostic schema

## INTRODUCTION

Erlang and Elixir has a know track of FIX and crypto expertise. We rely on proven telecommunication practices for developing real-time applications.

## 2.1. NETWORK ROUTER

As of network router we use N2O that is know to be the king in Erlang world for WebSocket applications. It can serve C50K at lowest MacBook Air and is know to be scalable. Some projects that are using N2O:

- <https://explorer.axel.network> (Blockchain Explorer)
- <https://pb.ua/depozit> (scaled to 30M clients)
- <https://n2o.tech/jp/dseg.htm> (Trading Appliances)
- <https://nynja.io> (Cryptomessenger)

and many more...



# EX/ARCH ERLANG ELIXIR

5HT AUG 25 2019

## 2.2. MATCHING ENGINE

Order matching in Trade Engine is implemented using in-memory BTree structures (ETS tables) which is known to be not only fastest indexing mechanism in Erlang but also powerful enough to implement waltzdb test-suite for RETE algorithms that is known to be not parallelized. Trade matching engine is also a backend for TRADE EX/UI pages. You can see RETE implementation of waltzdb here:

- <https://github.com/enterprizing/rete>
- <https://github.com/enterprizing/trade>

## 2.3. ORDER BOOK ASSEMBLY

Order Book algorithm is implemented as part of Market Data Unification Module or Crypto Liquidity Integrator TIC. For now it unifies Bitmex, GDAX and OKCoin exchanges. For EX we are going to implement even faster version as Order Book assembling is essential subroutine for all exchanges (L3 for matching trades). You can see TIC implementation here:

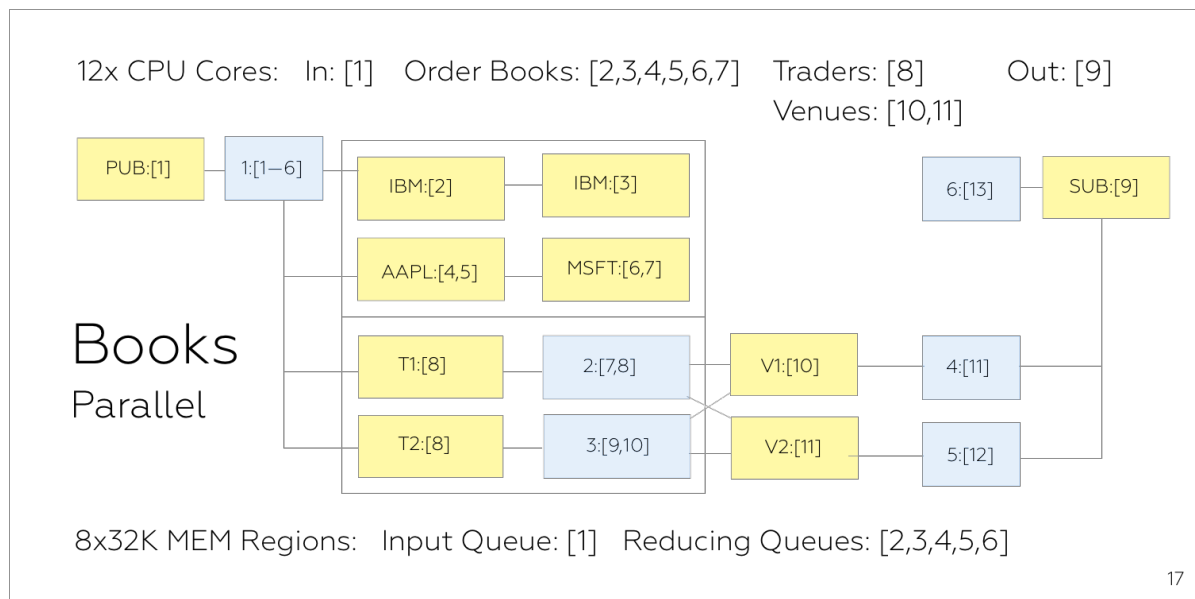
- <https://github.com/enterprizing/tic>

# EX/ARCH ERLANG ELIXIR

5HT AUG 25 2019

## 2.4. ORDER BOOK SHARING AMONG TRADERS

The crucial for exchange is the fast ticker order book sharing between traders sessions.



# EX/ARCH ERLANG ELIXIR

5HT AUG 25 2019

## 2.5. FORMATTER AGNOSTIC

Thanks to BERT module we can expose JSON/XML and any formatter atomatically:

— <https://github.com/synrc/bert>

## 2.6. REAL TIME APPEND LOG

As for database storage for the sake of availability we use RocksSB library and its stream interface for Erlang and Elixir KVS:

— <https://github.com/synrc/kvs>

## 2.7. NITROGEN WEB FRAMEWORK

NITRO is a boosted version of Nitrogen Web Framework created by Rusty Klohouse. It allows to manipulate DOM directly from server and is the best and fastest Erlang/Elixir web framework for pure real-time WebSocket applications.

— <https://github.com/synrc/nitro>

## 3. HEX.PM libraries for EX application

The future MVP will be placed into the following repositories:

- <https://hex.pm/packages/bert>
- <https://hex.pm/packages/kvs>
- <https://hex.pm/packages/n2o>
- <https://hex.pm/packages/nitro>
- <https://hex.pm/packages/rocksdb>
- <https://hex.pm/packages/tic>
- <https://hex.pm/packages/trade>
- <https://hex.pm/packages/fix>
- <https://hex.pm/packages/ex>

### 3.1. N2O SPACE

For more information about N2O family (libraries and applications) consider following sources:

- [n2o.dev](https://n2o.dev) — Erlang User Manuals
- [n2o.systems](https://n2o.systems) — Erlang Business Apps
- [o7.network](https://o7.network) — Elixir Business Apps
- [n2o.space](https://n2o.space) — Erlang and Elixir for Business
- [n2o.tech](https://n2o.tech) — N2O Type Specification
- [n2o.cloud](https://n2o.cloud) — N2O Cloud