EX: Technology Preview

5HT AUG 25 2019

TOC

1. EXCHANGE USER INTERFACE

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

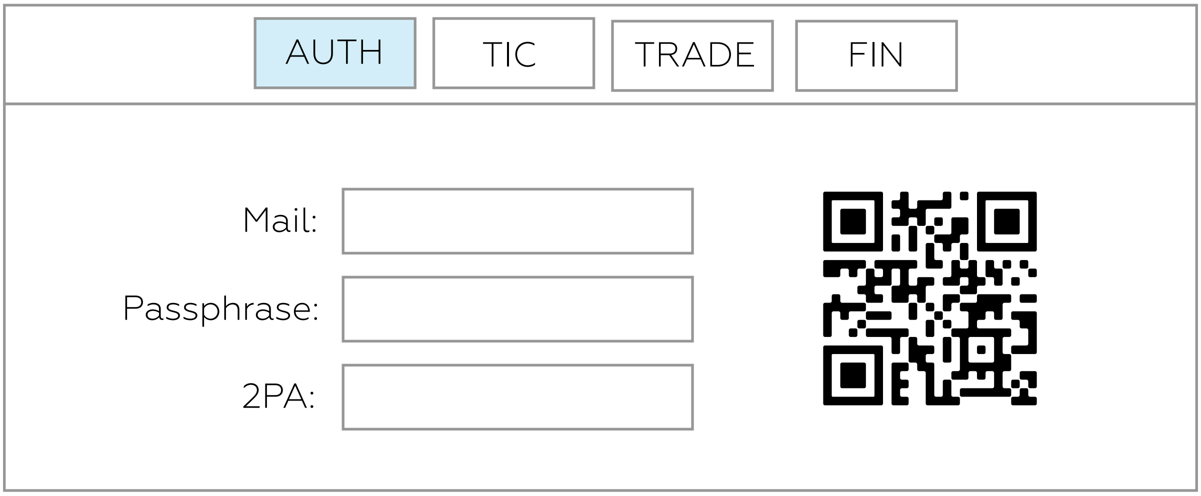
NOTES: Usually Exchanges provide Google Auth secured own token issue machanism based on HashiCorp Vault or other key stores. While we see most powerful security based on ECC/RSA Public Key Cryptography:

— https://github.com/synrc/ca

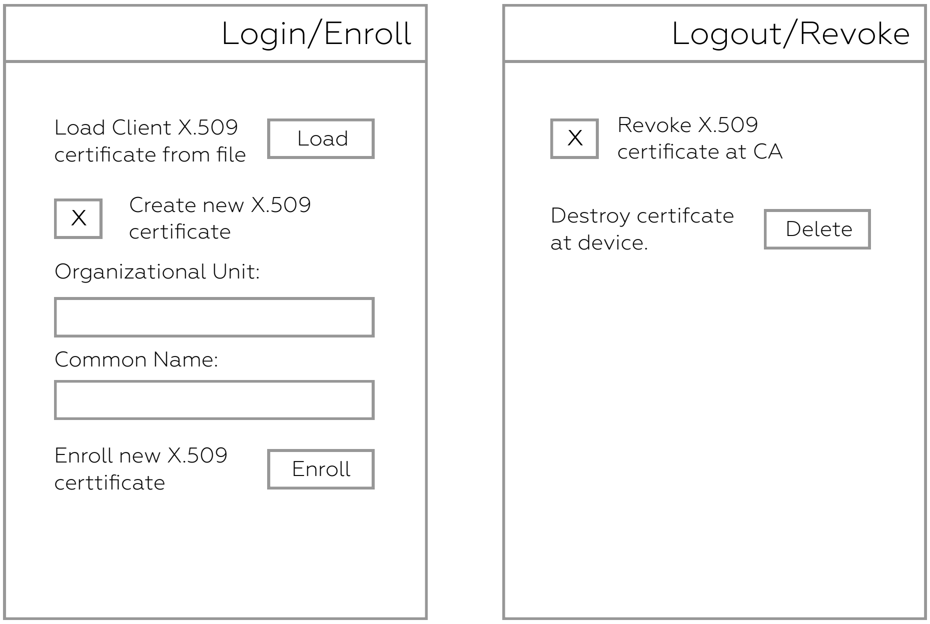
EX/UI JS CSS

5HT AUG 25 2019

1.1. GOOGLE AUTHENTICATOR



1.2. PUBLIC KEY INFRASTRUCTURE



Load Client X.509 certificate from file

Create new X.509 certificate

Enroll new X.509 certtificate

Revoke X.509 certificate at CA

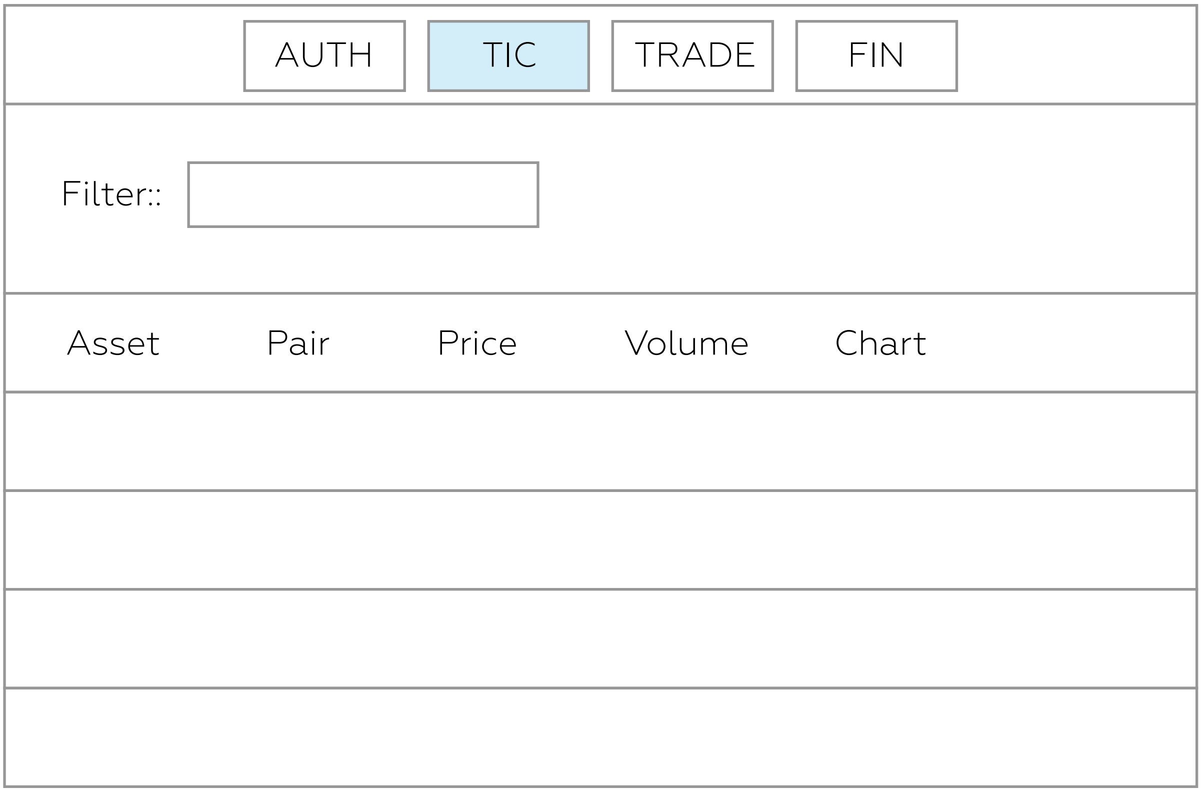
Destroy certifcate at device.

EX/UI JS CSS

5HT AUG 25 2019

NOTES: During the MVP development, for Order Book testing purposes it is good to support client exnchange connectivity for external existed exchanges. MARKET page is perfect aggregator of tickers for arbitrage network.

1.3. MARKET



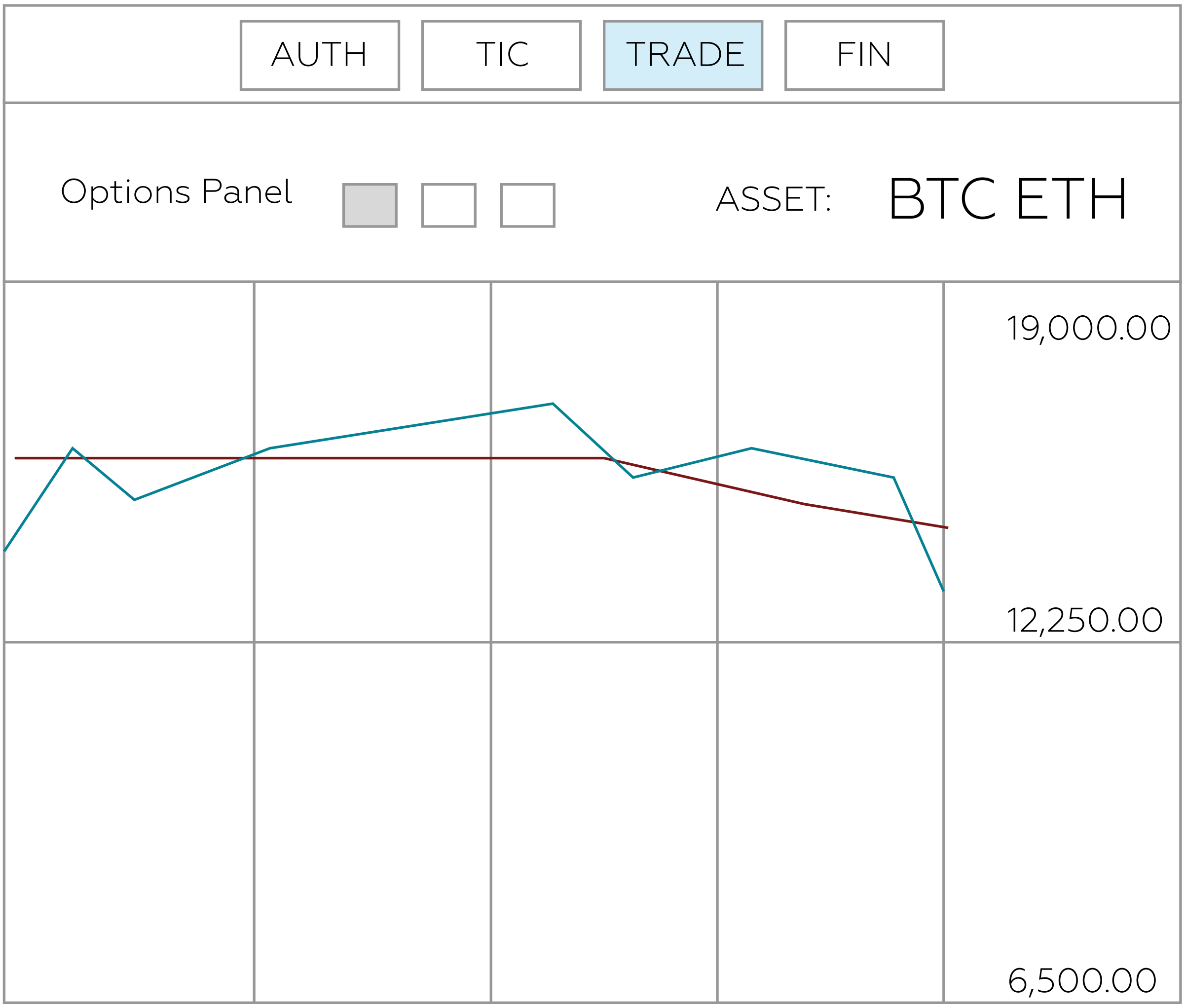
Asset	Pair	Price	Volume	Chart

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

EX/UI JS CSS

5HT AUG 25 2019

1.4. MARKET DATA HISTORY and PREDICTION



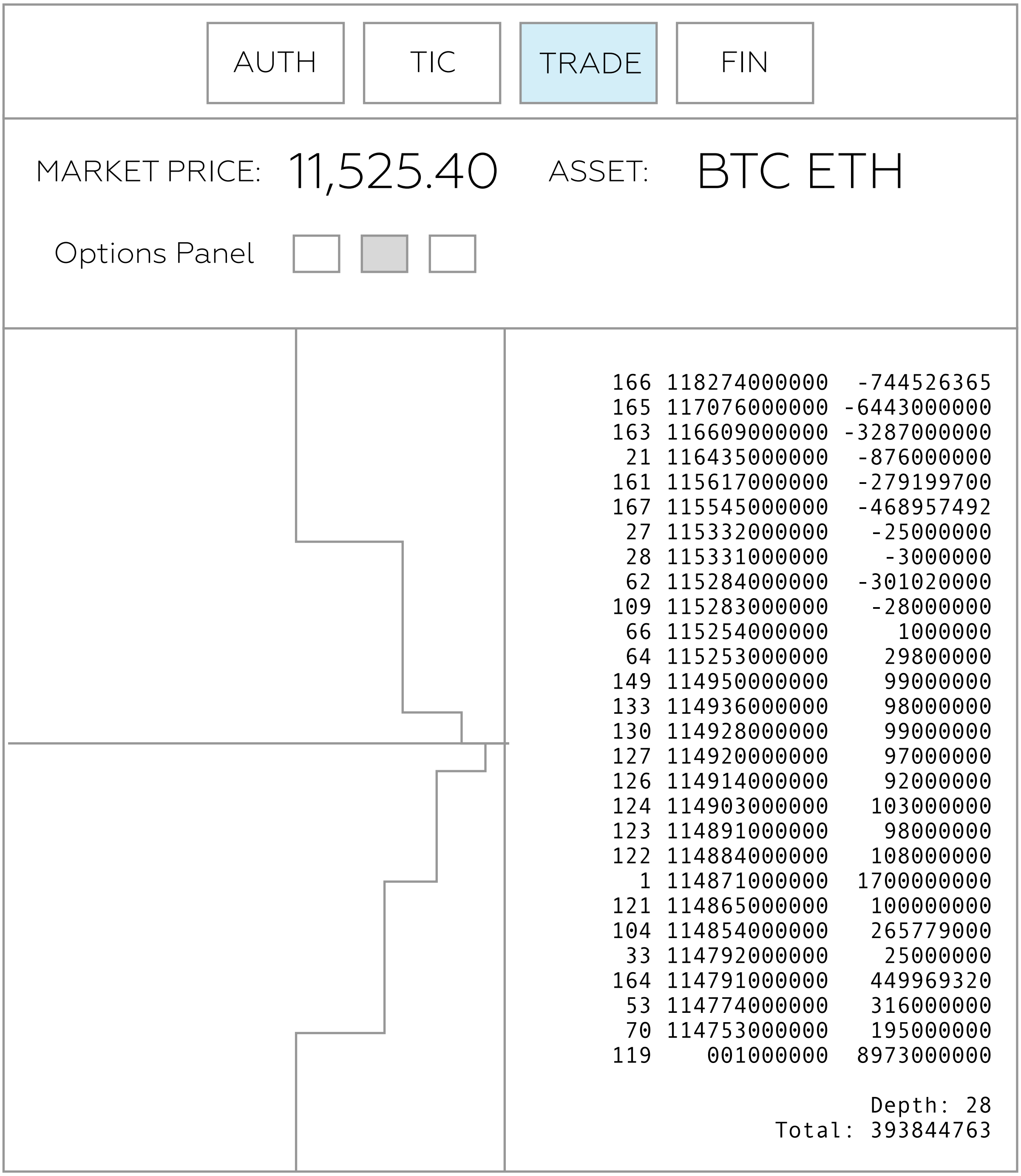
NOTES: Market Data History Contorol Element (1) that is in conjunction with DOM (2) and Order Placer (3) forms the Trading Interface that in modern exchanges always placed at the same trading dashboard page.

EX/UI JS CSS

5HT AUG 25 2019

1.5. DEPTH OF MARKET for TICKER

Order Book Real-time Visualization



MARKET PRICE: 11,525.40 ASSET: BTC ETH

Options Panel □ ▨ □

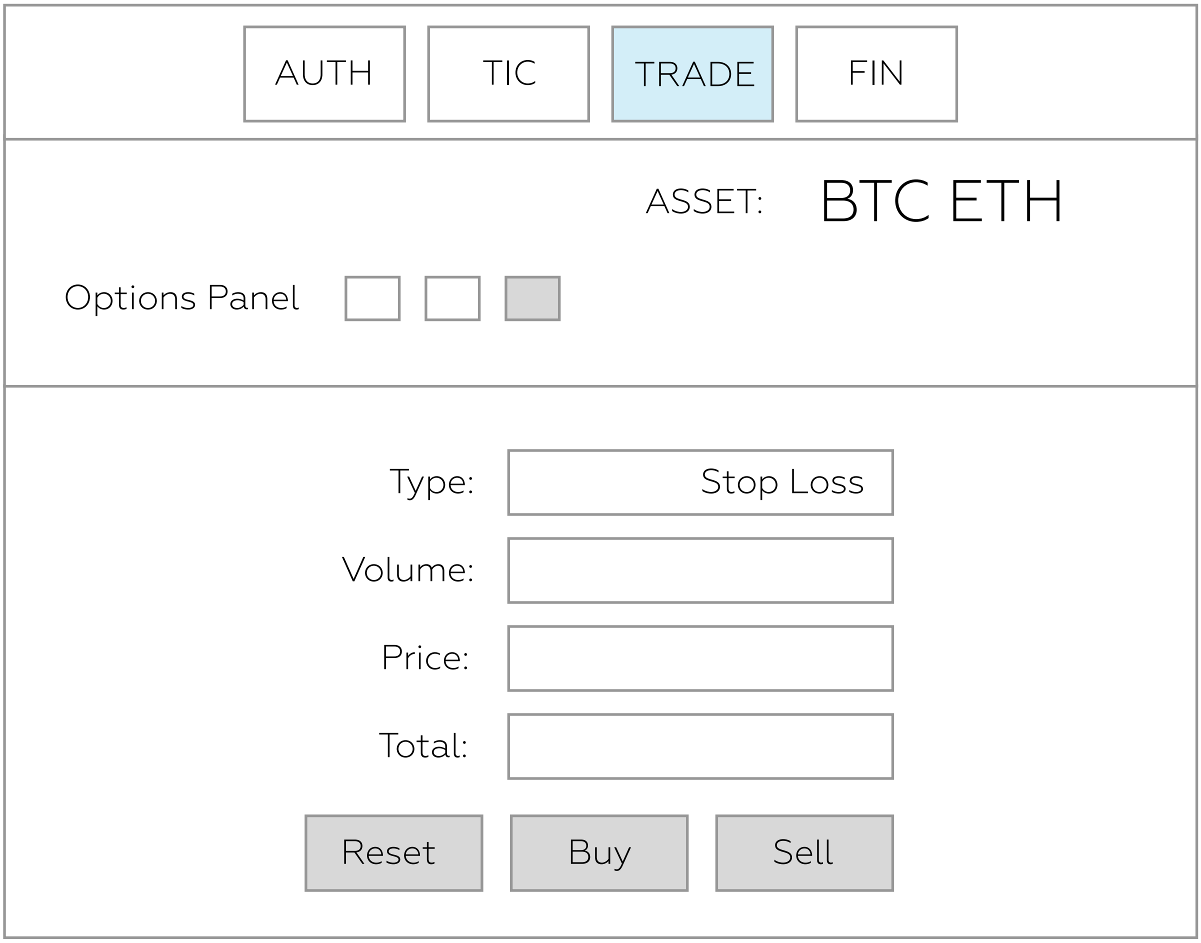
```
166 118274000000  -744526365
165 117076000000 -6443000000
163 116609000000 -3287000000
 21 116435000000  -876000000
161 115617000000  -279199700
167 115545000000  -468957492
 27 115332000000   -25000000
 28 115331000000    -3000000
 62 115284000000  -301020000
109 115283000000   -28000000
 66 115254000000     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



ASSET: BTC ETH

Type: Stop Loss

Volume:

Price:

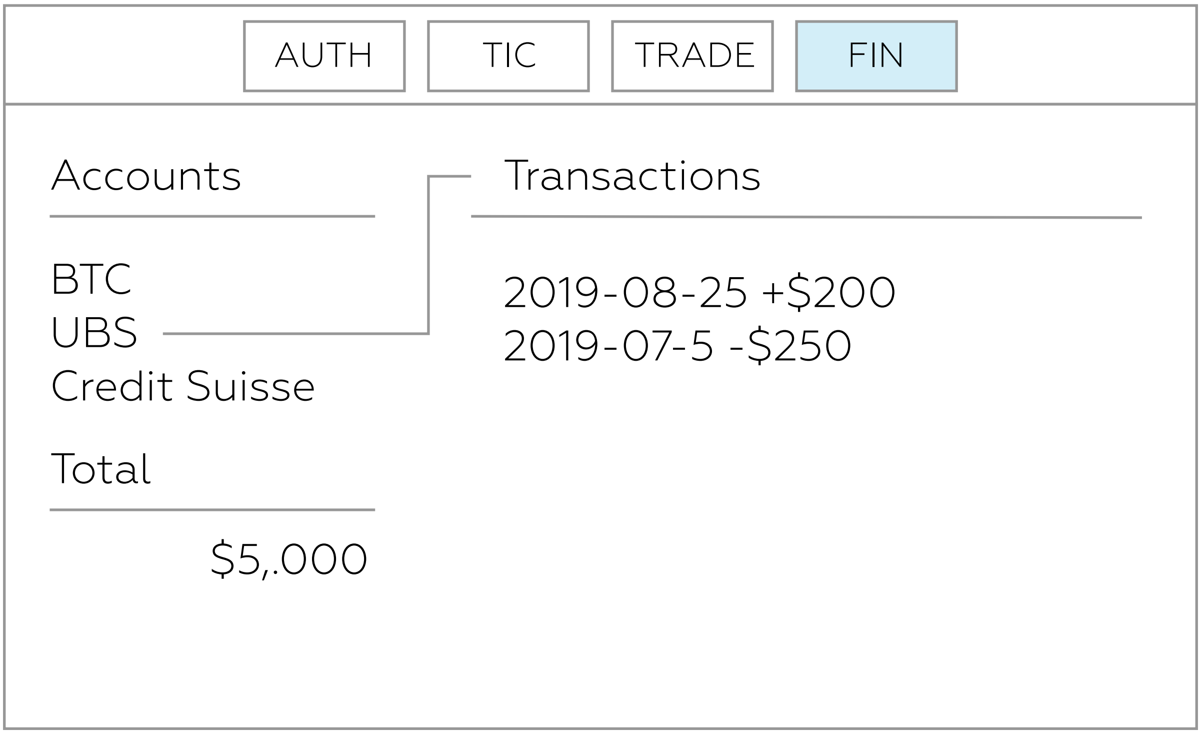
Total:

NOTES: Order Maker is a direct interface to Matching Engine. It creates rules in BTree that are being processed immediately with the next trading tick from the top of the book. We’ve measured all data-structures (hashtables, skip-lists, trees) and skip-lists are known to best for order books. Also we can test TRADE maching engine agains existing exchanges (ghost mode).

EX/UI JS CSS

5HT AUG 25 2019

1.7. WALLET



Accounts

BTC
UBS
Credit Suisse

Total

$5,.000

Transactions

2019-08-25 +$200
2019-07-5 -$250

NOTES: Wallet is a personal mini-bank with accounts, that in potential could be both FIAT and CRYPTO. We have FIN enterprise module that can be easily adapted to exchange wallet. Wallet is just a list of accounts and its transactions. Also we need here PAY module for various payment providers.

— https://github.com/synrc/bank
— https://github.com/synrc/plm
— https://github.com/synrc/pay

EX/ARCH ERLANG ELIXIR

5HT AUG 25 2019

2.EXHANGE ARCHITECTURE

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

INTRODUCTION

Erlang and Elixir has a known track of trading applications expertise. We rely on proven telecommunication practices for developing real-time applictions.

2.1. NETWORK ROUTER

As of network router we use N2O that is known to be the best in Erlang world for WebSocket applications. It can serve C50K at lowest MacBook Air and is 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 (Cryptomessanger)

and many more...

2.2. MATCHING ENGINE

Order maching in Trade Engine is implemted 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 hard to parallelize. Trade maching 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

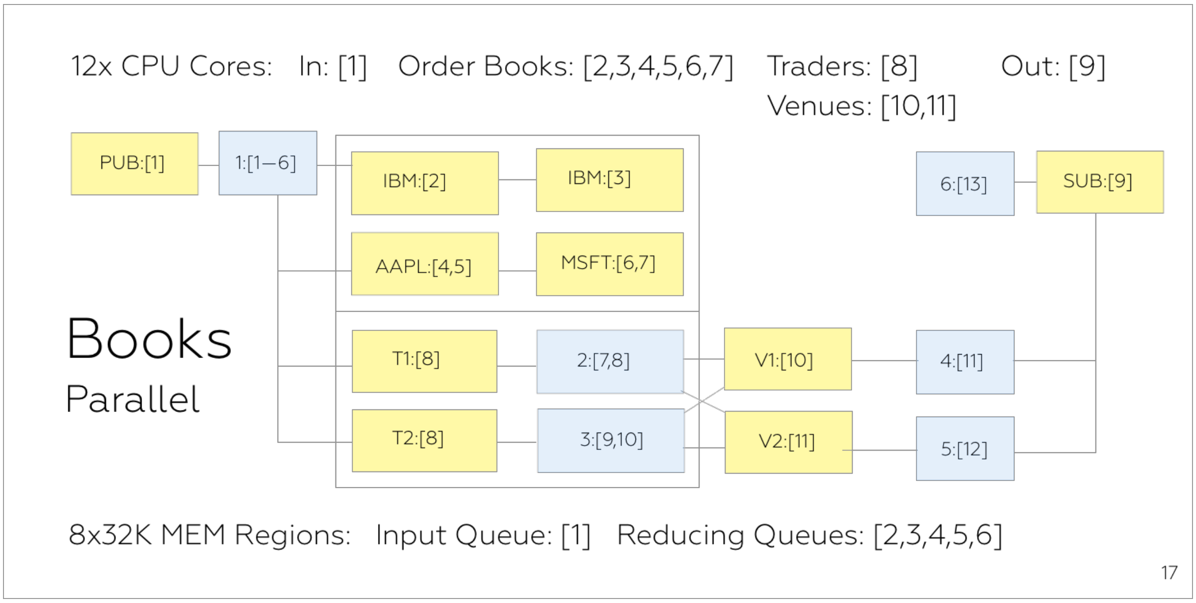
2.3. ORDER BOOK ASSEMBLY

Order Book algorithm is implemented as part of Market Data Unification Module or Crypto Liquitity 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 maching trades). You can see our TIC implementation here:

— https://github.com/enterprizing/tic

2.4. ORDER BOOK SHARING AMONG TRADERS

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



12x CPU Cores: In: [1] Order Books: [2,3,4,5,6,7] Traders: [8] Venues: [10,11] Out: [9]

PUB:[1]
1:[1—6]
IBM:[2]
IBM:[3]
AAPL:[4,5]
MSFT:[6,7]
6:[13]
SUB:[9]
T1:[8]
2:[7,8]
V1:[10]
4:[11]
T2:[8]
3:[9,10]
V2:[11]
5:[12]

Books
Parallel

8x32K MEM Regions: Input Queue: [1] Reducing Queues: [2,3,4,5,6]

17

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
— https://github.com/synrc/rest

2.6. REAL TIME APPEND LOG

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

— https://github.com/synrc/kvs

2.7. NITROGEN WEB FRAMEWORK

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

— https://github.com/synrc/nitro

3. HEX.PM libraries for EX application

The structural MVP already resides in the following repositories:

— https://hex.pm/packages/tic
— https://hex.pm/packages/trade
— https://hex.pm/packages/fix
— https://hex.pm/packages/ex

and underlying libraries:

— 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

3.1. N2O SPACE

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

— n2o.dev — Erlang User Manuals
— n2o.systems — Erlang Business Apps
— n2o.cloud — N2O Cloud

EX/ROADMAP

5HT AUG 25 2019

4. FEATURES ROADMAP

4.1. Auth Page [Elixir]
4.2. List of Assets (Market) [Elixir]
4.3. History of Market Data [JS/CSS]
4.4. Order Book for Asset [JS/CSS]
4.5. Liquidity Integrator [Erlang]
4.6. Order Book Assembler [Erlang]
4.7. Trading Matching Engine [Erlang]
4.8. Order Maker [Elixir]
4.9. Wallet [Elixir]
4.10. Payment Providers [Elixir]

Alsmost any task could be done in parallel with proper and coordinated supervision:

[JS/CSS] tasks (4.3, 4.3) are the consumers of real-time feeds provided by 4.5 and 4.6. 4.8 is the consumer of 4.7. 4.9 is the consumer of 4.10. 4.2 is a consumer of 4.5. So the project perfectly fits teams consisting of pairs.