Cryptocurrency trading bot

Live class for Eat the blocks Pro

Dhruvin Parikh, June 2021

application.

George Brown College

Blockchain Professor *Toronto, Canada* *August 2019 - Present*

Instructor and Curiculumn developer for Blockchain Development Program.

Peer Tutor *Toronto, Canada* *May 2019 - August 2019*

Help students and assisted professors.

York University

Blockchain Professor *Toronto, Canada* *June 2020 - Present*

Instructor and Curiculumn developer for Blockchain Development Program.

BlockX Labs

Software Developer *Toronto,Canada* *May 2018 - August 2020*

Contributed software engineering expertise in development of products

About The Instructor

Skills

Programming: React, html/JS/CSS, node.js/Express, Ethereum, Solidity, Keil, C

Software Development: git, linux/windows, agile development, docker, CI/CD

Engineering: PCB design using Altium

Education

Humber College

Information Technology Solutions *Toronto, Canada* *January 2017 - April 2018*

Opted for enterprise development profile

Course work in enterprise software development using

Gujarat Technological University

Bachelors in Electronics And Communication Engineering *India* *August 2012 - May 2015*

Focussed on embedded systems and its applications.

Maharaja Sayajirao University Of Baroda

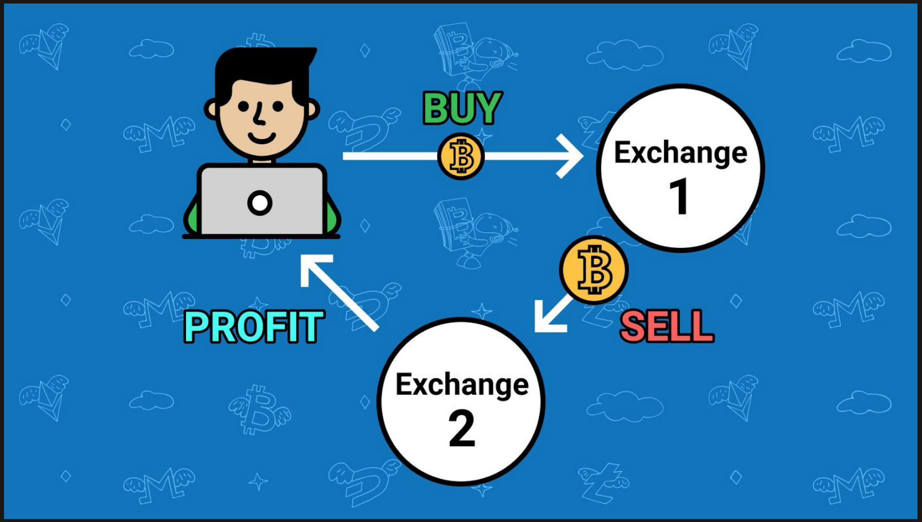
Diploma in Electronics And Communication Engineering *India* *August 2009 - May 2012*

Focussed on embedded systems and its applications.

Plan

- Basic terminology
- Reasons to trade using a bot
- Developing your own trading bot
- Architecture and API level access to exchanges
- Demo

Arbitrage



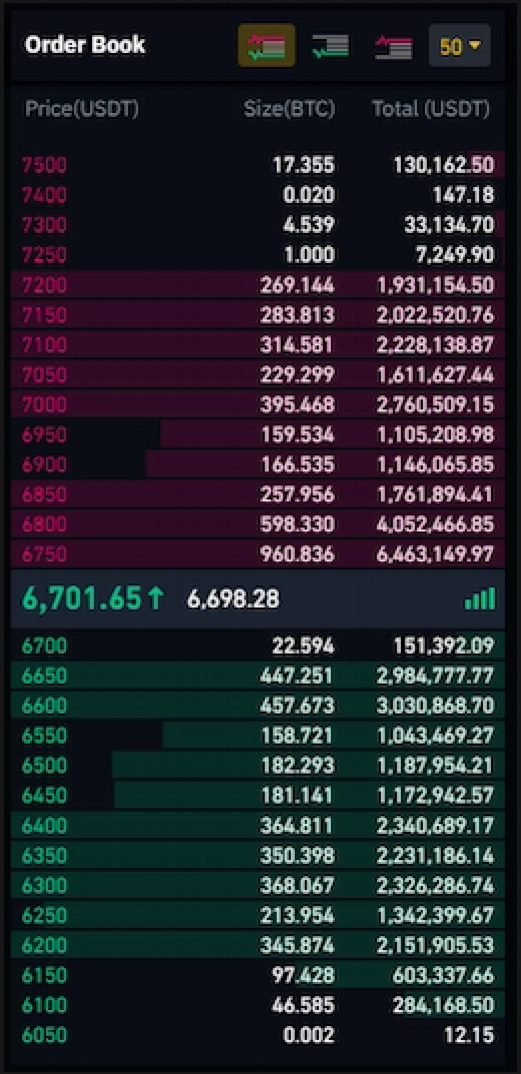
- purchase and sale of an asset in order to profit from a difference in the asset’s price between marketplaces

Price slippage

expected trade price - trade execution price

- Causes
  - High volatility in market for short period
  - Current trade volume exceeds the existing bid/ask spread.

Orderbook


Order Book

50

Price(USDT)	Size(BTC)	Total (USDT)
7500	17.355	130,162.50
7400	0.020	147.18
7300	4.539	33,134.70
7250	1.000	7,249.90
7200	269.144	1,931,154.50
7150	283.813	2,022,520.76
7100	314.581	2,228,138.87
7050	229.299	1,611,627.44
7000	395.468	2,760,509.15
6950	159.534	1,105,208.98
6900	166.535	1,146,065.85
6850	257.956	1,761,894.41
6800	598.330	4,052,466.85
6750	960.836	6,463,149.97
6,701.65 ↑	6,698.28	
6700	22.594	151,392.09
6650	447.251	2,984,777.77
6600	457.673	3,030,868.70
6550	158.721	1,043,469.27
6500	182.293	1,187,954.21
6450	181.141	1,172,942.57
6400	364.811	2,340,689.17
6350	350.398	2,231,186.14
6300	368.067	2,326,286.74
6250	213.954	1,342,399.67
6200	345.874	2,151,905.53
6150	97.428	603,337.66
6100	46.585	284,168.50
6050	0.002	12.15

- Collection of bid-and-ask orders.
- Orders are matched and executed only when a bid and ask price are the same.

Programmable trading

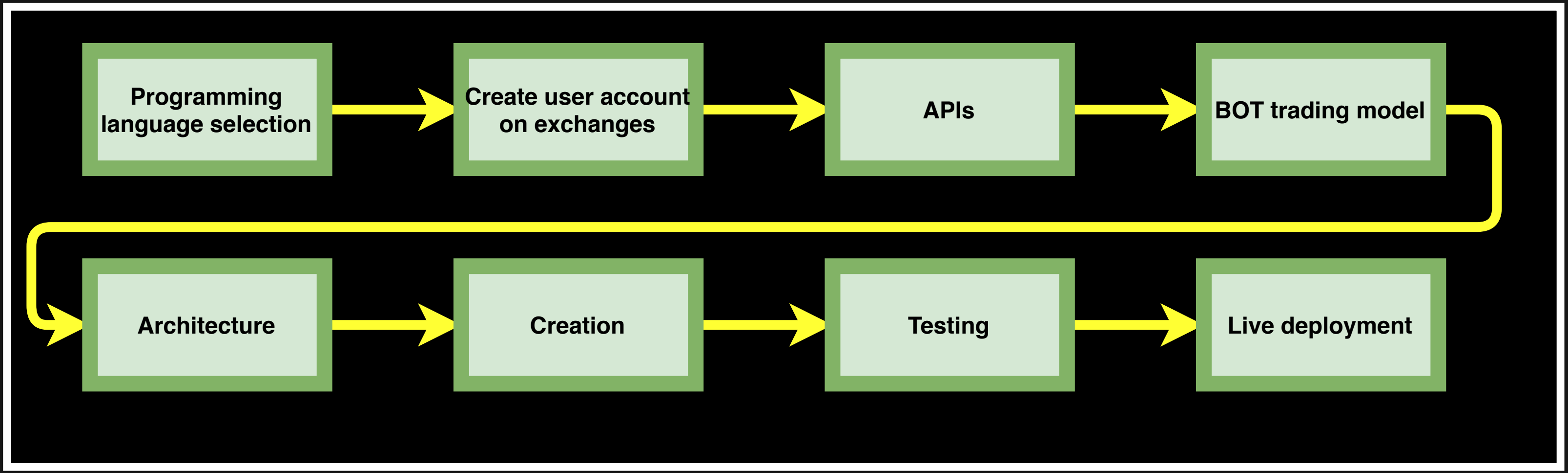
- Possible through a software program that uses API's to interact with financial exchanges
- This software program actively monitor exchanges around the clock and react with whatever predetermined criteria they have been programmed with.

Trading bot strategies

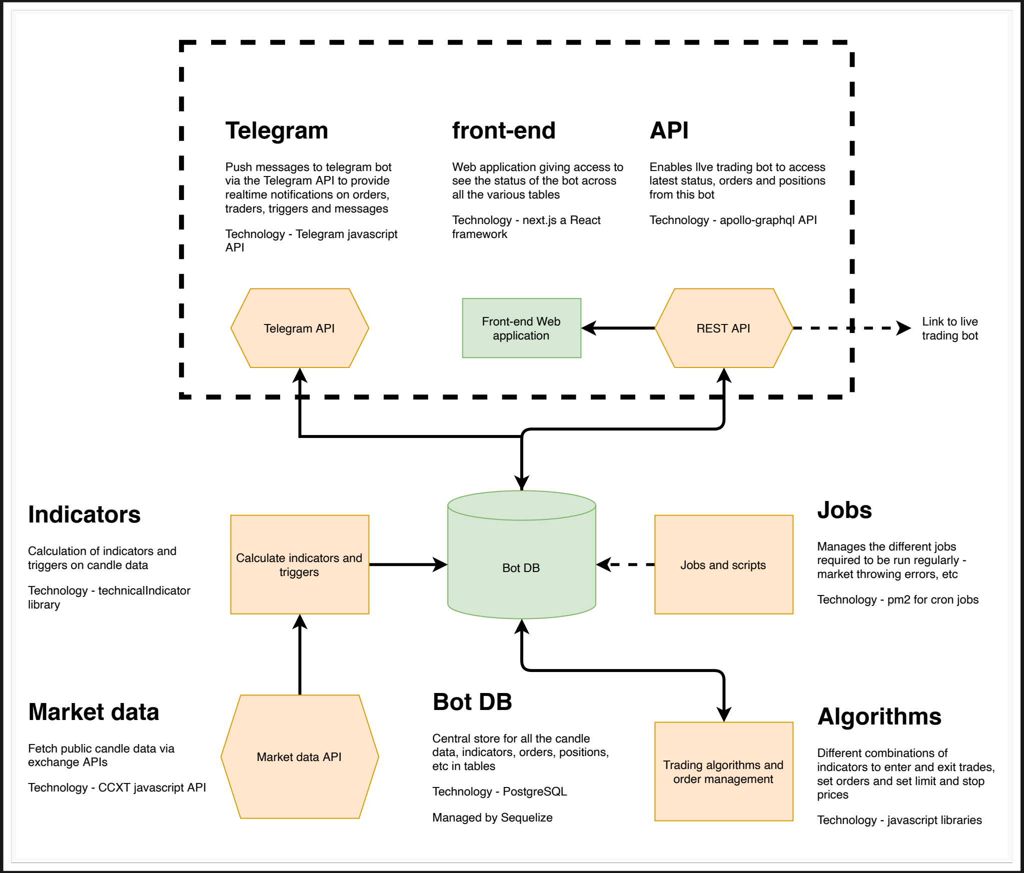
- Trend following strategy
- Arbitrage
- Market making

and many more. Reference

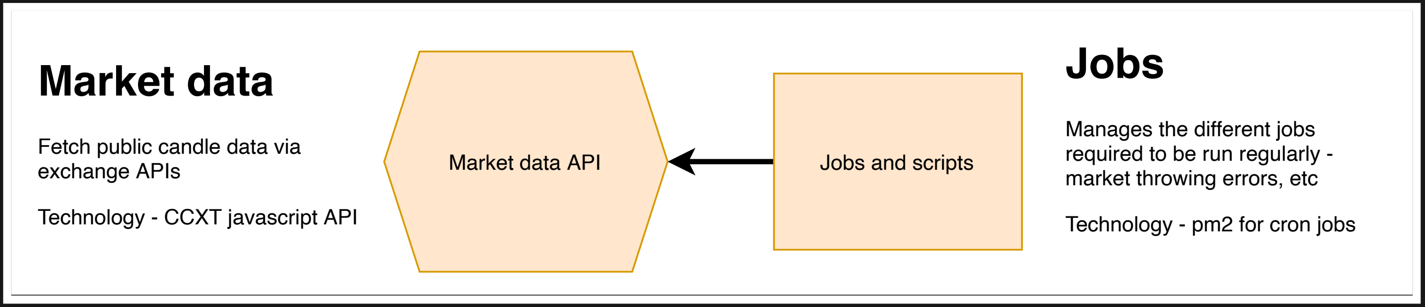
Building a cryptocurrency trading bot



Typical crypto trading bot architecture


Telegram
Push messages to telegram bot via the Telegram API to provide realtime notifications on orders, traders, triggers and messages
Technology - Telegram javascript API
front-end
Web application giving access to see the status of the bot across all the various tables
Technology - next.js a React framework
API
Enables llve trading bot to access latest status, orders and positions from this bot
Technology - apollo-graphql API
Telegram API
Front-end Web application
REST API
Link to live trading bot
Indicators
Calculation of indicators and triggers on candle data
Technology - technicalIndicator library
Calculate indicators and triggers
Bot DB
Jobs and scripts
Jobs
Manages the different jobs required to be run regularly - market throwing errors, etc
Technology - pm2 for cron jobs
Market data
Fetch public candle data via exchange APIs
Technology - CCXT javascript API
Market data API
Bot DB
Central store for all the candle data, indicators, orders, positions, etc in tables
Technology - PostgreSQL
Managed by Sequelize
Trading algorithms and order management
Algorithms
Different combinations of indicators to enter and exit trades, set orders and set limit and stop prices
Technology - javascript libraries

We'll focus on


Market data
Fetch public candle data via exchange APIs
Technology - CCXT javascript API
Market data API
Jobs and scripts
Jobs
Manages the different jobs required to be run regularly - market throwing errors, etc
Technology - pm2 for cron jobs

CCXT: Convenience crypto trading API

- Bridge to over 115 bitcoin/altcoin exchanges

```
# Node JS installation
$ npm install ccxt
```

Why CCXT?

- Not everyone trades on same crypto exchange
  - Binance, bitmex, coinbase, kraken, kucoin, etc
- Abstraction for multiple exchanges
- Multiple language support (Node JS, Python and PHP)

CCXT library structure

- Common set of methods

```
                             User
+--------------------------------------------------------------+
|                            CCXT                              |
+------------------------------+-------------------------------+
|            Public            |           Private             |
+==============================================================+
|                              .                               |
|                    The Unified CCXT API                      |
|                              .                               |
|       loadMarkets            .           fetchBalance        |
|       fetchMarkets           .            createOrder        |
|       fetchCurrencies        .            cancelOrder        |
|       fetchTicker            .             fetchOrder        |
|       fetchTickers           .            fetchOrders        |
|       fetchOrderBook         .        fetchOpenOrders        |
```

CCXT: Example Usage

```
const ccxt = require('ccxt');

// print all supported exchanges
console.log(ccxt.exchanges)

// instantiate kraken exchange object
const krakenExchange = new ccxt.kraken();

// print kraken id and market available on kraken
console.log(krakenExchange.id, await kraken.loadMarkets)

// instantiation with API key
const exchange = new ccxt.binance ({
            'apiKey': process.env.BINANCE_API_KEY,
            'secret': process.env.BINANCE_SECRET_KEY})
```

Demo

- A cli based automatic arbitrage crypto exchange
    - accepts the exchanges to
    - detects the arbitrage opportunity if profit is >6%
    - emits a message with all details of arbitrage among exchanges supplied

Backtesting

- Simulation to evaluate performance of trading strategy
- Requirements
  - Candlestick data
  - Tick-by-tick trade data
  - Order book snapshot data (Recommended)

Order book snapshot data

- Exact state of a market at the time of snapshot
- Orders available on an exchange at a particular time

Performance formulae

$$Performance = [(Vf - Vi) / Vi] * 100$$

- Where
  - **Vf** is the final value of the portfolio
  - **Vi** is the initial value of the portfolio
  - **Multiply by 100** to convert from a decimal to a percentage

References

- CCXT Github

End of Class