# Frontrunning

No Author Given

Concordia University

**Abstract.**

## 1 Introduction

## 2 Preliminaries

### 2.1 What is Front running?

*Front running* is analogous to any course of action during which a person benefits from prior access to inside or confidential market information to upcoming transactions and trades. This problem can occur in both financial and non-financial systems, however, it is more noticeable within the trading and exchange systems as it was originated back in stock markets. In the old times, all the trades were executed on papers, so in case of receiving a large order from a client, the broker might say this loud and other people around the table could be informed. Therefore, a malicious trader would now run in front of that order and put his own transaction in between (before the trade was executed) and profit from the price increase of the stock. In other words, a group of market participants obtain non-public market information which allows them to front-run other users trades by putting their orders a head of those trades and benefit from advance knowledge of pending client orders. The two significant factors which cause the front-running practice to happen within the financial markets are (i) imperfect competition and (ii) liquidity uncertainty. **should I explain more?!** Any sort of front-running activity within the financial markets is considered as an unethical and illegal practice as it is unfairly beneficial for a few market participants who have the privilege of acting on this information and taking advantages at the expense of the investor.

### 2.2 Front Running on the Blockchains

Blockchain technology has received an exceptional amount of attention since it was first introduced as the underlying technology of the cryptocurrency Bitcoin in Satoshi Nakamoto's (pseudonymous) 2008 white paper [14]. Many decentralized applications that are nowadays built on top of this technology represent the significance of the blockchain as it completely eliminates the single points of failures within any systems. However, blockchains have some inherent characteristic that leave them vulnerable to *front running* behaviour. Although this data structure is known to be publicly visible to every network participants, information is layered and some of them are accessed only by insiders. Any network participants that runs a full node on the blockchain is able to obtain those information and as mentioned in Section 2.1, this leaves the application vulnerable to front running. Decentralized exchanges are a group of blockchain financial applications where front running can be executed. Bancor [9] is an example of such systems in which front runners benefit from potential price increase of the market stocks (details are provided in Section ...). Decentralized namespaces, as blockchain non-financial applications, are another example. Ghazal [13] is such system in which upon seeing a transaction from Alice to register `alice.com`, front runner can go ahead of her and register this domain name. He can later sell `alice.com` to Alice for a higher price and benefit from the price increase.

## 3 Related Work

As the traditional frontrunning was originated to trading financial instruments most of the literature are focused on financial aspects of the markets.

### 3.1 Historical Context / Classic Frontrunning

Frontrunning was first appeared on the Chicago Board Options Exchange ( *CBoE*) [12], and was identified by *Securities Exchange Commission* in 1977 as the following:

> The practice of effecting an options transaction based upon non-public information regarding an impending block transaction[1] in the underlying stock, in order to obtain a profit when the options market adjusts to the price at which the block trades. [16]

On the years after there were ongoing discussions between self-regulatory exchanges (*e.g., CBoE*) and *SEC* to regulate, detect and define laws and regulations to deal with frontrunning [12], with *SEC* stating:

> It seems evident that such behaviour on the part of persons with knowledge of imminent transactions which will likely affect the price of the derivative security constitutes an unfair use of such knowledge. [2]

As defining what exactly is considered illegal front-running required more knowledge of how these new markets behave, *CBoE* and other exchanges (and brokers) issued educational circulars for their members asserting that frontrunning violates existing rules, with some examples of what is considered frontrunning. However difference of opinion regarding the unfairness o front running

[1] Block in the stock market is large number of shares, 10 000 or more, to sell which will heavily changes the price

[2] Securities Exchange Act Release No. 14156, November 19, 1977, (Letter from George A. Fitzsimmons, Secretary, Securities and Exchange Commission to Joseph W. Sullivan, President CBoE).

activities, insufficient exchange rules and lack of a precise definition in this area resulted in no action by self-regulator organizations citesec1978optionsmarket.

Further reading on the early details on the history and challenges of detecting and regulating frontrunning can be found in [12] .

Initially the frontrunning policies only applied to certain option markets, later on in 2002, the rule was refined to include the same prohibitions to security futures [4], which then in 2012 with the new amendment, FINRA Rule 5270, the frontrunning rule was extended to cover trading in an option, derivative, or other financial instruments overlying a security with some exceptions [5, 17].

### 3.2 Recent Literature on Blockchain frontrunning

[11] [3] [7]

## 4 (On) Blockchain Front-running

### 4.1 Who Can Front-run?

In general, all the network participants have the ability to front run specific transactions that are sent to the network. However, miners can include any transactions they like into the block they attempt to mine. Thus, they possess special power in terms of front running as opposed to other users of the network. In the following, we discuss and compare the two groups of blockchain potential front runners.

**Miners** As mentioned above, since the Blockchain miners are the only parties who can decide on the order of transactions within a block they mine, they can easily intercept and reorder the pending transactions sitting in the mempool and profit from a guaranteed price-rise. For example in an Ethereum based application, a miner learns about the pending *buy* transaction of 1000 units of a token, i.e. TKN, presumably if this transaction goes through, it causes the price the purchased token to increase. So the dishonest miner can step in front of this transaction and place his own buy order ahead of it. He would simply create his *buy* of 1000 TKN and include it within a block and now he mines the previous *buy* transaction of 1000 TKN. Doing so, he would receive a better rate than other network users, and can sell the assets he has received and gain a price advantage at the expense of others. Similarly, a dishonest miner can sell his tokens in front, if he sees a pending *sell* transaction. In an open source blockchain application, there is no rules on how transactions must be ordered and miners are free to send transactions in the order they prefer. However, miners can only front run other transactions (by reordering them) within the block they happen to mine. They could do this also by not broadcasting their own transactions to the network, this makes the miners to be less noticeable to the network when front running. Another example here is in the case of decentral exchanges and order cancellation. Every exchange should have the functionality to cancel the orders, especially for a volatile market. In this case, when user decides that her order is not profitable anymore, she would send a cancelation transaction. Here the miner sees the cancelation transaction and puts his own order before the cancellation transaction to fill the order, potentially profiting from the order. They can also include the canceled transaction in their block to collect the full gas limit used by that transaction [10].

**Blockchain users/nodes** Any regular (non-miner or miner) user can also front-run other transactions in the network. For regular users to front run others on the blockchain, they need to be fully/well connected to other nodes on the network. Doing so, they are able to listen to the network and monitor all transactions that are broadcast. On the Ethereum blockchain users have to pay for the computations in small amount of Ether called *gas* [1]. The price that users pay for transactions (a.k.a. transaction fees) can increase or decrease how quickly miners will execute them and include them within the block they mine. This is because the Ethereum miners consume resources to process the transactions and so receive the transaction fees after creating the blocks. Once seeing two identical transactions with different transaction fees, miners act to maximize their profit and are free to mine the transaction which offers the highest fee. Therefore, any regular users who run a full-node Ethereum client can modify the order of pending transactions by paying with a higher gas price*i.e.,* by monitoring the network, upon seeing a pending *buy* transaction which will further increase the price of the asset, a font-runner user can pay a higher gas price and hope to have his transaction a head of the other one mined. By doing so, he achieves a better rate from the other network users. Note that in this case, blockchain front runners are more visible to the network as they broadcast their transactions to all the network participants.

### 4.2 Historical evidence

As blockchain records are immutably recorded, there is enough historical data to analyze for possible front running detection. For examples here we research some of the events of such attacks happening in the Ethereum blockchain applications.

**Status ICO** ICO, Initial Coin Offering, is one of the blockchain applications, specially blockchains such as Ethereum with smart contract capability. The common practice is to deploy a smart contract on the blockchain indicating the details of the ICO such as the trade ratio, when it starts and ends, and more details on how it will be capped. In June 2017, Status.im started their ICO and within 3 hours they reached the dynamic ceiling in place that triggered the end of the ICO, summing in 300,000 ether in funds, estimated at more than 200 million dollars at the time of their ICO. [15]. The idea behind Dynamic Ceilings is to make it more costly for larger contributors, in the form of transaction fees, which have to split their contributions to different addresses, with minimal impact to smaller contributors [18]. On the time of the ICO there were reports of Ethereum network being unusable and transactions were not confirming. Further study showed that some mining pools (: define mining pool) could have been manipulating the network for their own profit.

**Bancor** Bancor is an Ethereum-based application that allows users to exchange their tokens without any counter-party risk. This protocol aims to solve the cryptocurrency liquidity issue by introducing *Smart Tokens* [9]– ERC20 compatible tokens with a built-in liquidity mechanism that are always available to users. Smart Tokens can be bought and sold through the users smart contract at an automatically calculated price which displays supply and demand. Doing so, Bancor protocol provides continuous liquidity for digital assets without relying on an order book as there is no requirement to match sellers and buyers.
**Front-running Bancor** Recently, researchers have shown that Bancor is vulnerable to front running attacks. Implemented on the Ethereum blockchain, when Bancor transactions are broadcast to the network, they sit in a pending transaction pool known as *mempool* waiting for the miners to mine them. Since Bancor handles all the trades and exchanges on the Ethereum blockchain (unlike other existing decentralized exchanges), these transactions are all visible to the public for 16s (the average Ethereum blocktime) before being included within a block. This leaves this blockchain-based decentralized exchange vulnerable to the blockchain race condition attack as attackers are given enough time to front-run other transactions and gain favourable profits [8]. As mentioned in Section 4.1 and same as every other Ethereum based application, Bancor transactions can be front run by miners and other regular (non-miner) users. There is an implementation of a Bancor front running attack in the Python programming language which represent how a non-miner user can front run other network participants [2].

**Domain Name Registration** Although frontrunning attacks have been more showcased in the context of decentralized exchanges and trading systems, they are are not yet limited to the financial markets. Frontrunning can occur within other blockchain based applications such as naming systems. Blockchain-based namespaces have been introduced to eliminate the role of central parties *i.e.,* domain name system (DNS) which introduces single point of failures in the entire web. One such system is Ghazal, an Ethereum-based naming and PKI system [13]. Ghazal users rely on the Ethereum blockchain to register their `.ghazal` domain names and bind certificates to those names. In Ghazal model, a user would register domain name by executing the *registerdomain* function from the Ghazal smart contract with the domain name in plain text as the function argument. As mentioned before, These transactions will sit in the mempool so that it would be mined by Ethereum miners and included in the block. During this period in which the transaction is not yet confirmed, frontrunning attack can occur by (i) dishonest miners and/or (ii) regular non-miner user. In the first case, a miner would intercept the *registerdomain* transaction and register that name ahead of the user. A regular non-miner node in the Ethereum network can frontrun other user's *registerdomain* at a good profit by paying higher transaction fees. In both cases the adversarial party could sell the domain name to the users for higher price.

## 5 How to stop frontrunning?

In the traditional markets, frontrunning is considered unethical and also illegal. In the blockchain space, the designers of the decentral applications cannot rely on the justice system for unethical behaviour, and they must assume that each participant in the network acts rationally in their own self interest, the application will be operating in an adversarial environment [19]. A few decentral exchange projects such as *EtherDelta* and *0xProject*, have proposed solutions for frontrunning, which is off-chain orderbooks. The methods discussed within these projects prevent blockchain network participants from front running the orders as the orders are private and not known to the network and will not be broadcasted. However they are introducing third parties, *e.g.,* relayers in 0xproject, to be managing the orders with the promise not to reorder or frontrun other orders.

*Traditional Frontrunning Prevention Methods.* regulations/enforcement/broker education/sealed order

*Blockchain Frontrunning mitigation/prevention.* The traditional methods of preventing frontrunning are based on regulation and restrictions applied to the brokers and actors within the markets, such methods do not apply to blockchains, no enforcement methods, ...

There are two main approaches to prevent front running, one to design a blockchain that is frontrun-resistent , and the other to design the application logics in a way that front running is not possible.

### 5.1 Frontrun-resistent Blockchain

What does this mean to have a frontrun-resistent blockchain? There are technical difficulties to achieve such solutions as there are unknown factors within such network designs (corner/edge cases).

- **Privacy Preserving Blockchians**: Shielded transactions in ZCash do not reveal the sender, receiver, the amount and the data included in the transactions, hence they cannot be seen by network participants to be frontrunned. Although this limites the functionality of the blockchain/smartcontracts/-dapps
- textbfLoopring/Dual Athoring: Talk about this possible solution

### 5.2 Application design to prevent frontrunning

Applications, in this case Dapps, could be designed in a way to prevent frontrunning.

- **Commitment scheme**: Also known as *Commit and Reveal*, is a cryptographic primitive that enables one to commit to a value (*e.g.,* statement, document, ... while keeping it a secret and on a later time reveal the committed value [6]. Commitment scheme is a good method to prevent information leakage from the sensitive transactions. This could be done by simply broadcasting the hash value of the committed data and later on revealing the values that would be hashed to the committed hash. In the case of decentral exchanges, user can send a commit transaction which will be cryptic to network participants but will act as a placeholder in the queue for the user, after the transaction is mined, user sends reveal transaction revealing the order details which will be executed in a fair order. Another use of such commitment scheme, is in Decentral DNS, such as Ghazal or ENS. User will send a commit transaction similar to a sealed bid, once the transaction is confirmed and the grace period is over, user sends a reveal transaction revealing the bid and also the details of the requested domain.
- **Submarine sends**: describe. General solution similar to commit and reveal. This is to solve the participatory factor of the commit and reveal solution.
- **Application logic specific solution**: Depending on the use case of the Dapp or the application, it could be designed in a way to deincentives some actors to frontrun the transactions or prevent them from doing so. As an example on desiging a decentral orderbook, it could be said that a miner could front run orders for financial gain on price improvement, however if the fees of the orders are sent to the miner of the block, it deincentiveses the miner to front run the orders as they already gain enough financial benefit from including the correct order of transactions. Another example for decentral orderbook design could be using call market design instead of time-sensitive orderbooks. In such design the arrival time of the order does not matter as they are executed in batches.

## 6 Concluding Remarks

## References

1. Account types, gas, and transactions ? ethereum homestead 0.1 documentation. `http://ethdocs.org/en/latest/contracts-and-transactions/account-types-gas-and-transactions.html#what-is-gas`. (Accessed on 06/14/2018).
2. New tab. `https://hackernoon.com/front-running-bancor-in-150-lines-of-python-with-ethereum-api-d5e2bfd0d798`. (Accessed on 08/13/2018).
3. R. T. Aune, A. Krellenstein, M. OHara, and O. Slama. Footprints on a blockchain: Trading and information leakage in distributed ledgers. *The Journal of Trading*, 12(3):5–13, 2017.
4. F. I. R. Authority. *IM-2110-3. Front Running Policy.* 2002.
5. F. I. R. Authority. *5270. Front Running of Block Transactions.* 2012.
6. G. Brassard, D. Chaum, and C. Crépeau. Minimum disclosure proofs of knowledge. *Journal of Computer and System Sciences*, 37(2):156–189, 1988.

7. L. Breidenbach, I. Cornell Tech, P. Daian, F. Tramer, and A. Juels. Enter the hydra: Towards principled bug bounties and exploit-resistant smart contracts. In *27th {USENIX} Security Symposium ({USENIX} Security 18)*. {USENIX} Association, 2018.
8. P. D. Emin Gun Sirer. Bancor is flawed. `http://hackingdistributed.com/2017/06/19/bancor-is-flawed/`, 2017. (Accessed on 06/14/2018).
9. E. Hertzog, G. Benartzi, and G. Benartzi. Bancor protocol. 2017.
10. P. D. A. J. Y. L. X. Z. Iddo Bentov, Lorenz Breidenbach. The cost of decentralization in 0x and etherdelta. `http://hackingdistributed.com/2017/08/13/cost-of-decent/`, 2017. (Accessed on 08/14/2018).
11. K. Malinova and A. Park. Market design with blockchain technology. 2017.
12. J. W. Markham. Front-running-insider trading under the commodity exchange act. *Cath. UL Rev.*, 38:69, 1988.
13. S. Moosavi and J. Clark. Ghazal: toward truly authoritative web certificates using ethereum.
14. S. Nakamoto. Bitcoin: A peer-to-peer electronic cash system. 2008.
15. C. Petty. A look at the status.im ico token distribution. `https://medium.com/the-bitcoin-podcast-blog/a-look-at-the-status-im-ico-token-distribution-f5bcf7f00907`, 2017. Accessed: 2018-06-10.
16. SEC. 96th cong, 1st sess, report of the special study of the options markets to the securities and exchange comission. *Comm. Print 1978*, pages 183–189, 1978.
17. SEC. Notice of filing of proposed rule change to adopt finra rule 5270 (front running of block transactions) in the consolidated finra rulebook. *Release No. 34-67079; File No. SR-FINRA-2012-025*, 2012.
18. S. Team. The status network, a strategy towards mass adoption of ethereum. `https://status.im/whitepaper.pdf`, 2017. Accessed: 2018-06-10.
19. W. Warren. Front-running, griefing and the perils of virtual settlement. `https://blog.0xproject.com/front-running-griefing-and-the-perils-of-virtual-settlement-part-1-8554ab283e97`, 2017. (Accessed on 08/14/2018).