ETH
Eidgenössische Technische Hochschule Zürich
Swiss Federal Institute of Technology Zurich

Distributed Computing

Prof. R. Wattenhofer

# No Frontrunning

Frontrunning is when a yet-to-be confirmed transaction ($A$) is overtaken to confirmation by a new transaction ($B$) such that $B$ takes advantage of whatever $A$ does when it gets confirmed. For example, $A$ might move the market price of an asset by a large value, and $B$ gets that same asset for cheap because it was able to sneak in before $A$ got confirmed. The initiator of $B$ can now sell the same asset after $A$ has been confirmed. The initiator of $B$ nets a profit because $A$ moved the market price of the asset in question.


In a typical market setting, unconfirmed transactions are not publicly visible and hence frontrunning requires insider knowledge. In a blockchain setting, unconfirmed transactions are publicly visible in the form of a "mempool" where all unconfirmed transactions wait before they are added to a block by a miner or a validator. Anyone who can watch the mempool can frontrun any transaction, as long as they can outbid the "victim transaction" and sneak in ahead to get confirmed first.

In this project we try to design Ethereum smart contracts that are immune to frontrunning. Can popular smart contracts like constant function market makers (Uniswap, Curve, Balancer, etc.) be tweaked so that they are resilient to frontrunning? Or do we need to design similar smart contracts entirely from scratch to achieve the no-frontrunning property?

**Requirements:** This project involves understanding Ethereum, popular smart contracts like Uniswap, and being able to understand and program in Solidity. Knowledge of applied cryptography "tricks" is a plus.

**Interested? Please contact us for more details!**
**Contact:** Tejaswi Nadahalli: tejaswin@ethz.ch, ETZ G97