Token Composition: A Graph-Based Exploration of Ethereum Event Logs

Martin Harrigan
Dept. of Computing, Carlow Campus
South East Technological University
Rep. of Ireland
martin.harrigan@setu.ie

Thomas Lloyd
Dept. of Computing, Carlow Campus
South East Technological University
Rep. of Ireland
tomlloyd1992@gmail.com

Daire O'Broin
Dept. of Computing, Carlow Campus
South East Technological University
Rep. of Ireland
daire.obroin@setu.ie

***Abstract*—Abstract goes here.** ***Keywords*—blockchain, tokens, tokenisation, decentralised finance**

I. Introduction

Introduction goes here.

II. Related Work

Related work goes here.
[1]

III. Token Composition

Token composition goes here.

IV. Data

A. Ethereum Event Logs and Meta-Events

The Ethereum event log records specific occurrences or outputs generated during the execution of constract code. They enable off-chain applications to react to on-chain events. A popular event is the `Transfer` event emitted by ERC 20 tokens [**?**]:

```
event Transfer(address indexed from,
               address indexed to,
               uint256 value);
```

Listing 1. The ERC 20 `Transfer` event specifies three parameters: `from`, `to`, and `value`.

The event has two special cases. In the first, the `from` address is the zero address (`0x0`) and the contract mints a token. In the second, the `to` address is the zero address and the contract burns a token.

A single transaction can emit multiple events. We define a *meta-event* to be a sequence of events that match some pattern and are emitted by a single transaction. We define a *tokenising meta-event* to be a meta-event where the pattern is two `Transfer` events: one must indicate a transfer of tokens to the contract and another must indicate a new token being minted (*deposit & mint*), or one must indicate a token being burned and another must indicate a transfer of an existing token from the contract (*withdraw & burn*. In the terminology of ERC 4626, a tokenising meta-event corresponds to either a `Deposit` event or a `Withdraw` event. However, our tokenising meta-event does not require the contract to implement ERC 4626.

We extracted all `Transfer` events from Ethereum mainnet from block height 0 to 16 685 101 (February 2023) inclusive using Geth's `eth_getLogs` RPC method [**?**]. From the `Transfer` events, we identified 4 032 033 tokenising meta-events. Table [**?**] shows a sample of the data. The first (resp., last) two rows are the earliest (resp., latest) two occurrences of tokenising meta-events in that data that perform a deposit & mint, and a withdraw & burn.

For example, the first row indicates a transaction that deposited a dust amount of ARC [**?**] in a one-to-one exchange for newly minted SWT [**?**] in January 2017. The third row indicates a transaction that withdrew 5183 BONE [**?**] in exchange for burning 5160 `tBONE` in February 2023. We are not concerned with the individual utility or value of the tokens (or lack thereof); we are only interested in the fact that Token $\mathcal{X}$ can be deposited with a contract to mint Token $\mathcal{Y}$, and/or Token $\mathcal{Y}$ can be burned by a contract to withdraw Token $\mathcal{X}$.

We can filter the tokenising meta-events to include only those that involve two tokens, Token $\mathcal{X}$ and Token $\mathcal{Y}$, such that there is at least one instance of Token $\mathcal{X}$ being deposited with a contract to mint Token $\mathcal{Y}$, and at least one instance of Token $\mathcal{Y}$ being burned by a contract to withdraw Token $\mathcal{X}$. In other words, the "and/or" conjunction in the above is replaced by "and". This excludes *one-way token upgrades* where Token $\mathcal{X}$ can be deposited with a contract to mint Token $\mathcal{Y}$ but Token $\mathcal{X}$ cannot be withdrawn from the contract, and *one-way token burns* where Token $\mathcal{Y}$ can be burned by a contract to withdraw Token $\mathcal{X}$ but Token $\mathcal{X}$ cannot be deposited with the contract to mint Token $\mathcal{Y}$. Of the 4 032 033 tokenising meta-events, 3 461 723 meet the additional criteria. We will refer to the unfiltered and filtered tokenising meta-events in the subsequent sections.

B. Off-Chain Data

CoinGecko [**?**] is a cryptocurrency data platform that aggregates fundamental analysis of tokens including market price, exchange volume, and market capitalisation.

DEX Screener [**?**] stores, parses, and analyses blockchain data to produce a token screener, charts, and analytics. They cover many blockchains, decentralised exchanges, and tokens.

979-8-3503-4647-3/23/$31.00 ©2023 IEEE

TABLE I
EACH TOKENISING META-EVENT CONTAINS THE ADDRESS OF THE SOURCE TOKEN, THE ADDRESS OF THE TARGET TOKEN, ONE OF TWO POSSIBLE PAIRS OF ACTIONS (DEPOSIT & MINT OR WITHDRAW & BURN), THE AMOUNT OF THE SOURCE TOKEN THAT WAS DEPOSITED OR WITHDRAWN, THE AMOUNT OF THE TARGET TOKEN THAT WAS MINTED OR BURNED, AND A TRANSACTION HASH. THE TABLE INCLUDES FOUR SAMPLE ENTRIES FROM THE FULL SET OF 4 032 033 TOKENISING META-EVENTS: THEY ARE THE EARLIEST AND LATEST TOKENISING META-EVENTS THAT HAVE DEPOSIT & MINT AND WITHDRAW & BURN ACTIONS.

Source Token	Target Token	Actions	Source Amount	Target Amount	Tx Hash
`ARC` (`0xac709f`)	`SWT` (`0xb12a3c`)	deposit & mint	*dust*	*dust*	`0x549a12`
`DGZ` (`0x84178d`)	`preDGZ` (`0x18aa6e`)	withdraw & burn	1371	150	`0x2da232`
`BONE` (`0x981303`)	`tBONE` (`0xf7a038`)	withdraw & burn	5183	5160	`0x5dbe32`
`WETH` (`0xc02aaa`)	`aWETH` (`0x030ba8`)	deposit & mint	25	25	`0xb4281a`

C. The Directed Graph

It involves 8424 distinct tokens.

We constructed a directed graph from the tokenising meta-events as follows. Each vertex corresponds to a distinct token. Each directed edge from a source to a target corresponds to a set of tokenising meta-events that either deposits the source token and mints the target token, or withdraws the source token and burns the target token. In the terminology of ERC 4626, the source is the *asset* or and the target is the *share*. The graph has 8424 vertices and

D. Data Limitations

Our input data, namely, Ethereum event logs, CoinGecko market data, and DEX Screener liquidity pool data, have limitations. Firstly, Ethereum event logs are unauthenticated. A contract can emit any event with parameters set to any values of its choosing. There is no guarantee that, say, an ERC-20 `Transfer` event accurately reflects an actual transfer [**?**]. However, event logs are generally accurate and malicious contracts can be easily excluded. For an aggregated analysis, such as ours, the impact should be minimal. Secondly, the data from CoinGecko and DEX Screener are snapshots that were gathered in April 2024 whereas the Ethereum event logs have a temporal component. It is possible that a token had an entry on CoinGecko with a non-zero market capitalisation in the past, but, at the time the data was gathered, the entry no longer existed. It is also possible that a token was deposited to a liquidity pool that was tracked by DEX Screener in the past, but, at the time the data was gathered, it was no longer being tracked. It is also possible that CoinGecko's and DEX Screener's coverage is incomplete. However, as a high-level measure of token popularity, the impact of the mismatch should be minimal.

V. ANALYSIS

Analysis goes here.

VI. CONCLUSION

Conclusion goes here.

REFERENCES

[1] T. Lloyd, D. O'Broin, and M. Harrigan, "Emergent outcomes of the veToken model," in *The IEEE International Conference on Omni-Layer Intelligent Systems (COINS)*, 2023.