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 logs record specific occurrences or outputs generated during the execution of contract 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 new tokens. In the second, the `to` address is the zero address and the contract burns existing tokens.

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*. Our aim is to identify instances where a token is tokenised by another token. 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 follow 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 using the pattern described above. Table I shows a sample of the data. The first (resp., last) two rows are the earliest (resp., latest) two occurrences of tokenising meta-events in the 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 [**?**] to a contract in a one-to-one exchange for newly minted SWT [**?**] in January 2017. The third row indicates a transaction that withdrew 5183 BONE [**?**] from a contract 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 filter the tokenising meta-events to include only those meta-events 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 previous paragraph 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 criterion. We will refer to the unfiltered and filtered tokenising meta-events in Sect. IV-C.

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: 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

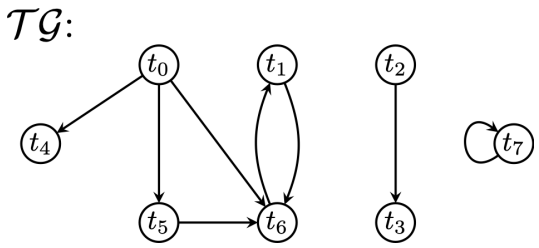

Fig. 1. A token graph, $\mathcal{TG}$, with eight vertices, $t_0, t_1, \ldots, t_7$, and eight edges. Each edge corresponds to a token tokenising another token. For example, the token represented by t_5 tokenises the token represented by t_0. The graph contains an undirected cycle (t_0, t_5, t_6), a directed cycle (t_1, t_6), and a loop (t_6).

TABLE II

EACH EDGE IN A TOKEN GRAPH REPRESENTS A SET OF TOKENISING META-EVENTS. THE TOP FIVE MOST SIGNIFICANT EDGES IN TERMS OF THE NUMBER OF TOKENISING META-EVENTS THEY CONTAIN ARE SHOWN. THE RESULTS ARE THE SAME FOR BOTH THE UNFILTERED AND FILTERED TOKEN GRAPHS, I.E., THE SAME FIVE EDGES ARE PRESENT IN BOTH.

Source Token	Target Token	# Tokenising Meta-Events
SHIB (0x95ad61)	xSHIB (0xb4a812)	402 186
BONE (0x981303)	tBONE (0xf7a038)	203 734
SUSHI (0x6b3595)	xSUSHI (0x879824)	120 221
LEASH (0x27c70c)	xLEASH (0xa57d31)	75 180
USDC (0xa0b869)	aUSDC (0xbcca60)	69 373

C. The Token Graphs

We can construct a directed graph from the tokenising meta-events as follows. Each vertex corresponds to a distinct token. Each directed edge from a source vertex to a target vertex corresponds to a set of tokenising meta-events that deposits the source token and mints the target token, and/or withdraws the source token and burns the target token. In the terminology of ERC 4626, the source token is the *asset* and the target token is the *share*. Figure 1 shows an example token graph.

In the unfiltered case, the directed graph has 23 687 vertices (distinct tokens) and 23 549 edges representing pairs of tokens where the second tokenises the first with either deposit & mint or withdraw & burn actions. In the filtered case, the directed graph has 8424 vertices that are incident with at least one edge and 7536 edges representing pairs of tokens where the second tokenises the first with both deposit & mint and withdraw & burn actions. We will refer to the unfiltered and filtered token graphs in the remainder of the paper.

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 an event of its choosing. There is no guarantee that, say, an ERC-20 `Transfer` event accurately reflects an actual transfer [**?**]. Additionally, the first special case highlighted in Sect. IV-A is stipulated by ERC-20[1] but the second is not. 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 in the past, but, at the time the data was gathered, the entry no longer existed. It is also possible that a token 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 or inaccurate. However, as a high-level measure of token popularity, the impact of the mismatch should be minimal.

Thirdly, tokenising meta-events are a heurstic for identifying instances where one token is tokenised by another. False positives create edges in the token graph where the token corresponding to the source vertex is not tokenised by the token corresponding to the target vertex; false negatives are pairs of tokens where one is tokenised by the other but there is no corresponding edge in the token graph. We will discuss both cases in Sect. V and Sect. VI.

V. ANALYSIS

In this section we examine the macro-topological structure of the unfiltered and filtered token graphs including degree distributions, connected component structure, and cyclic structure. We also examine the micro-topological structure of some individual tokens. We visualise the composition of tokens and identify their direct and transitive dependencies.

We note that each edge in a token graph represents a set of tokenising meta-events. Before examining the structure of the graphs, we can identify the most significant

[1] "A token contract which creates new tokens SHOULD trigger a Transfer event with the `_from` address set to `0x0` when tokens are created." [**?**]

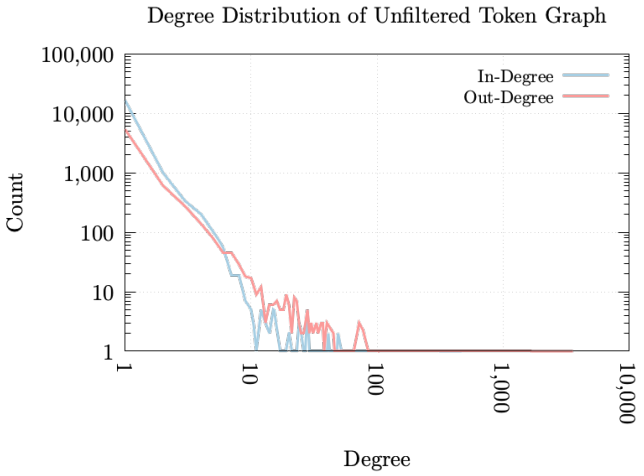

Fig. 2. The in- and out-degree distributions of the unfiltered token graph show an inverse relationship between the degree of a vertex and the number of vertices with that degree in the graph. There are a small number of vertices with high degree and a large number of vertices with low degree.

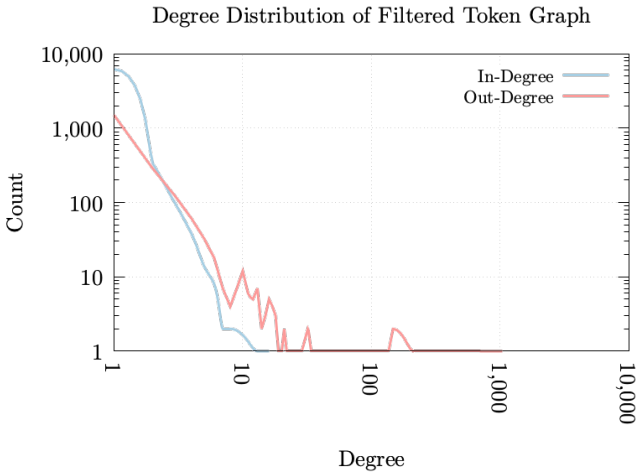

Fig. 3. The in- and out-degree distributions of the filtered token graph show a similar inverse relationship as in Fig. 2.

edges in terms of the number of tokenising meta-events they contain. Table II shows the results for both the unfiltered and filtered token graphs. Three of the five edges represent the staking of memecoins (SHIB → xSHIB, BONE → tBONE, and LEASH → xLEASH), one represents the staking of a governance token for a decentralised exchange (SUSHI → xSUSHI), and one represents the supply of a stablecoin to a decentralised lending market (USDC → aUSDC). Of course, we could also measure the significance of a edge based on, say, the volume of tokens transacted, the present USD value of the locked tokens, etc.

A. Degree Distributions

The in- and out-degree distributions of the unfiltered and filtered token graphs show an inverse relationship between the degree of a vertex and the number of vertices with that degree in the graph (see Fig. 2 and Fig. 3). Table III shows the top five vertices in the unfiltered and filtered token graphs by in-degree and out-degree. The out-degree entries are easy to explain: they are the tokens that are deposited with contracts in order to mint many other types of tokens. They include stablecoins (USDC, DAI, USDT and sUSD), wrapped ether (WETH), and wrapped bitcoin (WBTC). The in-degree entries are more complex and have multiple explanations. For example, CHI [**?**] is a gas token created by 1inch, a decentralised exchange aggregator, that is burned to obtain a reduction in transaction fees; in some transactions the burning of CHI is combined with the withdrawal of another token. This is a false positive generated by our heuristic since CHI does not tokenise a token. The remaining in-degree entries in the unfiltered category are due to token swaps performed during a deposit. For example, aDAI [**?**] is a yield-bearing token issued by AAVE, a decentralised lending market, in exchange for the stablecoin DAI. However, in some transactions, other tokens are supplied and swapped to DAI. These are also false positives since only DAI is tokenised by aDAI. In the filtered category, the in-degree entries are more reliable. For example, the iDOL token is minted when a user deposits various forms of the SBT token (e.g., SBT09180200, SBT09250200, etc.) according to the Lien Protocol [**?**]. Similarly, the agEUR token is a stablecoin by the Angle Protocol [**?**] that accepts a variety of tokens as collateral.

TABLE III
THE TOP FIVE VERTICES (TOKENS) IN THE UNFILTERED AND FILTERED TOKEN GRAPHS BY IN-DEGREE AND OUT-DEGREE.

	Top Five by In-Degree		Top Five by Out-Degree	
	Token	Deg.	Token	Deg.
Unfiltered	CHI (0x000000^2)	471	USDC (0xa0b869)	3587
	USDP (0x145668)	117	DAI (0x6b1754)	1923
	aUSDC (0xbcca60)	84	USDT (0xdac17f)	1175
	aWETH (0x030ba8)	63	WETH (0xc02aaa)	951
	aDAI (0x028171)	54	sUSD (0x57ab1e)	548
Filtered	XDP2 (0xe68c1d)	16	USDC (0xa0b869)	1037
	XDP1 (0x134fc6)	15	DAI (0x6b1754)	752
	cyUSD (0x1d0914)	14	USDT (0xdac17f)	396
	iDOL (0x7591a3)	13	WETH (0xc02aaa)	281
	agEUR (0x1a7e4e)	8	WBTC (0x2260fa)	211

Figure 4 plots in-degrees against out-degrees in the unfiltered and filtered token graphs. The tokens whose corresponding vertices have both high in-degree and high out-degree are tokens that tokenise many other tokens, and are themselves tokenised by many other tokens. Examples include mUSD, a stablecoin issued by the mStable protocol [**?**] and the agEUR token. This makes intuitive sense as stablecoins can be minted from various forms of collateral, and stablecoins can be used as collateral to mint other tokens.

B. Connected Component Structure

The unfiltered graph has 4082 weakly connected components. A giant component contains 13 794 vertices (∼58%) and 17 711 edges (∼75%). 3336 components (∼82% of the total) contain two connected vertices. These vertices correspond to

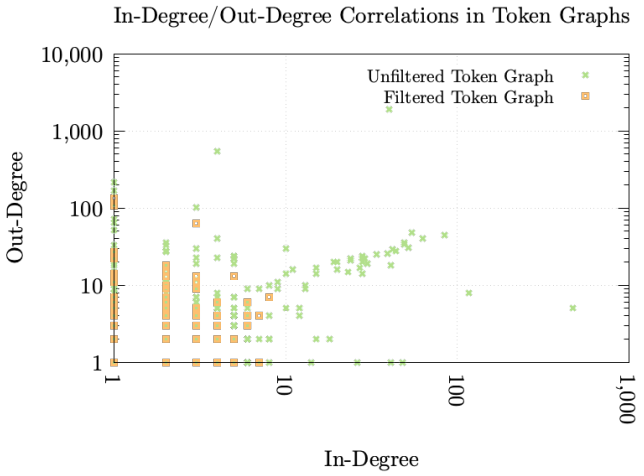

Fig. 4. In the unfiltered and filtered token graphs, we can identify vertices with both high in-degree and high out-degree.

pairs of tokens where at least one tokenises the other, but neither tokenises, or is tokenised by, a third. Interestingly, none of the tokens represented by vertices in these components have a non-zero market capitalisation according to CoinGecko or are traded in a liquidity pool according to DEX Screener. Their lack of popularity reflects their isolation in the token graph.

There are 418 components with more than two connected vertices other than the giant component.

C. Cyclic Structure

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.