Quantifying L2 Rollup Usage through K-means Clustering

Jack Liston
Laguna Beach High School

INTRODUCTION

Cryptocurrency faces a fundamental dilemma: balancing decentralization/security with scalability. One of the most promising technologies attempting to rectify this are rollups. Rollups are essentially a second layer (L2), built on top of the main chain, that maintains the security of the primary chain, while allowing for additional capacity.

In this project, we look at three different rollups on the Ethereum chain, Abitrum, Base, and zkSync Era. Both Abitrum and Base take Optimistic approaches (fraudulent transactions must be challenged) while zkSync uses mathematical proofs to very the validity of every single transaction. Interestingly, Base is influenced heavily by Coinbase and thus benefits from the Coinbase ethereum.

This project ultimately sets out to understand, to some extent, the progress and utilization of rollups in crypto and discern its potential to exist as a safe and powerful abstraction to add safe scaling capacity.

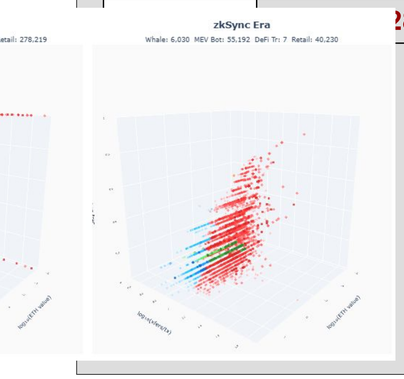
RESEARCH METHODOLOGIES

Rollups, as a relatively new technology, that exist solely on chain are not exactly accessible to data collection directly. Though each rollup publicizes their transitions publicly, to some extent, the collection of this data requires an active Ethereum node using RPC. (Note: there aren't any extensive, downloadable public data sets)

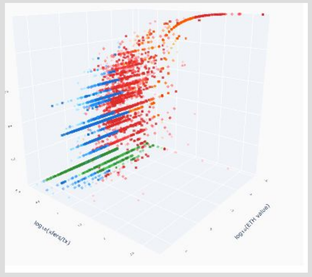
Using Alchemy's API, a crypto infrastructure platform for developers, I queried all transactions, and related data, on a specific 2 hour period in 2024, 2025, and 2026. This approach was necessary to limit the sheer size of the data, still taking nearly a full 24 hour period of just API querying. Just these 18 hours (3*2*3) of data is already millions of CSV rows and over a gigabyte of data. Importantly, these data sets are not uniform due to the differences in available data across the rollups. After gathering this data, inaccurate/non-valid rows were necessarily culled using a script. Finally, multiple different approaches were taken to analyze the gathered data. After many were ineffective/ clearly not productive, I focused primarily on a k-means clustering approach as well as, to a lesser extent, currency composition and TPS.

DATA AND FINDINGS

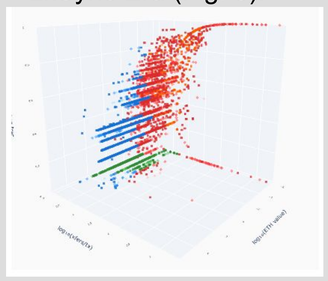
	Number of Rows	% of data set
zkSync Era	**536,325**	**9.2%**
Arbitrum	**1,217,201**	**20.8%**
	28	**70.0%**



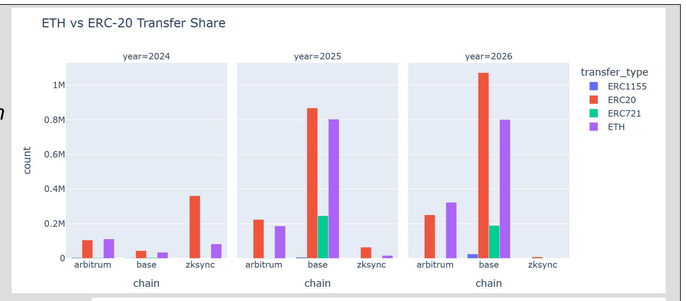
Base (Fig. 2):



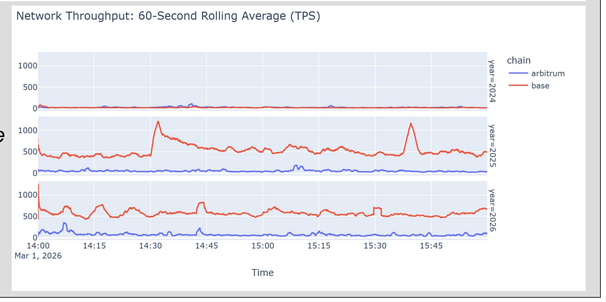
zkSync Era (Fig. 3):



Evolution of currency/token composition by year (Fig. 4)

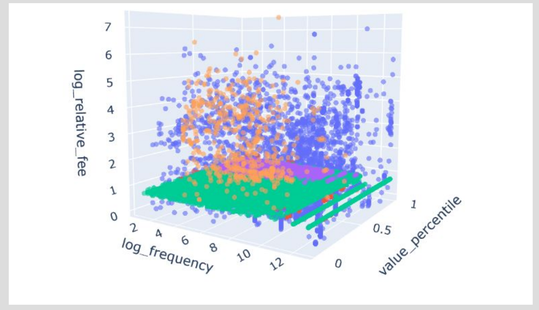


60 second rolling average (TPS) (Fig. 5)



CONCLUSIONS, IMPLICATIONS, AND NEXT STEPS

1. Rollups are certainly desmonstrating their ability to allow greater throughput while not sacrificing the decentralization/security predicated on the base chain.
2. However, the decline of zkSync Era coupled with the near-hyperbolic rise of Base likely does reveal the motives behind the adoption of these rollups: convenience and cost-efficiency, not just further throughput.
3. Further research would benefit from better raw data. While the overarching methodology is sound, the discrepancies in data across different rollups greatly inhibited the data analysis. Pruning is of course useful here but was unfortunately insufficient in identifying the complex nuances of each and every transaction.



A clear failure in clustering.

DISCUSSION, ANALYSIS, AND EVALUATION

Though not exactly new, our data corroborated many widely accepted ideas empirically, like how Base has grown into a dominant position. Moreover, our data also corroborated the narrative of ZK-sync's decline in terms of volume as it's architecture has dramatically shifted, like its move to Zk-stack (an environment that allows for more customizable rollups).

The clustering has revealed that the overall composition of transactions are somewhat similar throughout all of the rollups, when standardized to account for the total number of transactions. Although, there are obviously are differences, like zkSync's high number of bot transactions in 2024, presumably farming an airdrop during that time.

Because of the varying nature of data availability across different rollups, the data sets are fundamentally different. Most annoyingly, on zk-sync, many transactions cause multiple sub-actions/calls. For example, a single transaction could have multiple transfers if interacting with special contracts. Notably, a simple pruning mechanism fails to work well here because of the complete vastness of contract reactions and specialized mechanisms (like its MsgValueSimulator system contract that accounts for the way zkSync Era treats Ethereum at the EVM level: like a token). Hence, its exclusion from figure 5.

ACKNOWLEDGEMENTS / REFERENCES

Special thanks to Mr. Shen for helping make this project possible.

QR CODE TO FULL PAPER: