

Event Logs of Ethereum-Based Applications

A Collection of Resources for Process Mining on Blockchain Data

H.M.N. Dilum Bandara^{1,4}, Hendrik Bockrath², Richard Hobeck²,
Christopher Klinkmüller¹, Luise Pufahl², Martin Rebesky², Wil van der Aalst³ and
Ingo Weber²

¹Data61, CSIRO, Sydney, Australia

²Chair of Software and Business Engineering, Technische Universität Berlin, Germany

³RWTH Aachen University, Germany

⁴The authors are ordered alphabetically by family name.

Abstract

Process mining has become an established set of tools and methods for analyzing process data, while blockchain is emerging as a platform for decentralized applications and inter-organizational processes. Approaches and tools have been developed for analyzing blockchain data with process mining methods, including the tools created by us: BlockXES, ELF, and BLF. Recently, we have shown that process mining on blockchain data is valuable, among others for understanding user behavior and for security audits. With this resources paper, we make four different data sets available in XES format, stemming from four different blockchain applications: Augur, Forsage, CryptoKitties, and ChickenHunt. We describe the method of extraction, data sets, and conduct preliminary analyses to demonstrate feasibility. This publication aims to help researchers and practitioners to understand the application domain, and enables future process mining research on the data sets.

Keywords

Ethereum Logging Framework, Event logs, Process Mining, Blockchain

1. Introduction

Process mining [1] has established as a set of tools and methods for analyzing process data. Blockchain [2] is emerging as a platform for decentralized applications and inter-organizational processes. Approaches and tools have been developed for analyzing blockchain data with process mining methods, including the tools created by us: BlockXES [3], ELF [4], and BLF [5]. Although challenging [6], we recently showed that process mining on blockchain data is valuable, among others to understand user behavior and for security audits [7].


With this resource paper, we publish a collection of event logs from blockchain-based *decentralized applications* (DApps). The event logs are available in XES format and currently cover

Proceedings of the Demonstration & Resources Track, Best BPM Dissertation Award, and Doctoral Consortium at BPM 2021 co-located with the 19th International Conference on Business Process Management, BPM 2021, Rome, Italy, September 6-10, 2021

✉ dilum.bandara@data61.csiro.au (H.M.N. D. Bandara); {firstname}. {lastname}@tu-berlin.de (H. Bockrath); {firstname}. {lastname}@tu-berlin.de (R. Hobeck); christopher.klinkmueller@data61.csiro.au (C. Klinkmüller); {firstname}. {lastname}@tu-berlin.de (L. Pufahl); {firstname}. {lastname}@tu-berlin.de (M. Rebesky); wvdaalst@pads.rwth-aachen.de (W. van der Aalst); {firstname}. {lastname}@tu-berlin.de (I. Weber)



© 2021 Copyright for this paper by its authors. Use permitted under Creative Commons License Attribution 4.0 International (CC BY 4.0).

 CEUR Workshop Proceedings (CEUR-WS.org)

four DApps:

1. Augur, a prediction and betting marketplace;
2. Forsage, an investment application, which turns out to be a Ponzi scheme;
3. CryptoKitties, a game where virtual cats can be bred and traded as assets;
4. ChickenHunt, a game for collecting chickens and upgrading the avatar.

All four applications and their data are available on the public Ethereum blockchain. Still, extracting the data is non-trivial [6], and with this publication, we also release the artifacts for the data extraction. In particular, we use the open-source *Ethereum Logging Framework* (ELF) [4], which takes a manifest as input. Manifests define which on-chain data to extract, and how to transform and format it, e.g., as CSV or XES files. They can hence be used for various purposes. For example, users can query log entry data from a given smart contract address over a range of blocks. For each of the four DApps, a manifest was crafted and used with ELF to extract data from a full Ethereum archival node. The collection of event logs is made available via a website¹. For each data set it includes the ELF manifest, the XES event log, links to the DApp source code and website, a description of the XES log content, and preliminary analysis results.

In the following, we describe the data sets and conduct preliminary analyses to demonstrate feasibility. This publication aims to help researchers and practitioners to understand the application domain, and enables future process mining research on the data sets, e.g., for analysis and evaluation purposes.

2. Description of the data sets

All data sets are made available as event logs in XES format. The events we extracted from the DApps were encoded in the blocks of the public Ethereum blockchain. Data extraction for each DApp started with the first block after its deployment and ends with block 12,243,999 (one block before the Berlin Hard Fork)². Note that we extracted data from Augur at an earlier point for our case study in [7]. Hence, the respective log only covers data until block 10,336,628. While the logs have a varying number of attributes depending on the events generated by the corresponding DApp, each log has a common set of attributes, namely *Case ID*, *Activity*, *Complete Timestamp*, and *lifecycle:transition*. The additional attributes are described on the website accompanying this paper (see Footnote 1). The timestamps of the events correspond to the timestamps of the block they were extracted from. Additionally, the logs contain DApp or Ethereum-specific attributes, e.g., *gasPaid* or *receivingContract* in Augur. Table 1 presents key figures of the data sets.

3. Preliminary analysis

For the preliminary analysis, we focus on the event log of ChickenHunt. For Augur, an extensive case study has been published recently [7]. Preliminary analyses of Forsage and CryptoKitties

¹<https://ingo-weber.github.io/dapp-data/>

²<https://blog.ethereum.org/2021/03/08/ethereum-berlin-upgrade-announcement/>

Table 1
Overview over the data sets.

DApp data set	Augur	Forsage	CryptoKitties	ChickenHunt
Start date	2018-07-10	2020-01-31	2017-11-23	2018-06-25
Start block	5,937,093	9,391,531	4,605,167	5,851,533
Last date	2020-11-10	2021-04-15	2021-04-15	2021-02-16
Last block	10,336,628	12,243,749	12,243,893	11,866,129
Events	23,021	13,368,052	18,059,296	138,889
Cases	2897	1,055,931	1,997,604	715
Activities	11	12	12	17

can be found on the accompanying website (see Footnote 1).

ChickenHunt is an incremental game that is deployed as a DApp on Ethereum. The game’s goal is to collect chickens through farming and attacking other players. Players also have the option to upgrade the attack (“Upgrade Hunter”), defense (“Upgrade Depot”), and collection capabilities (“Upgrade Pet”) of their avatars. The player pays the gas costs for the Ethereum transactions. The game concept includes two types of incentives for playing. *Shareholder*: through certain transactions, players can become shareholders of the game; and *financial reward*: players can sacrifice collected chickens for Ether.

We loaded the event log into several process mining tools to analyze the players’ behavior,

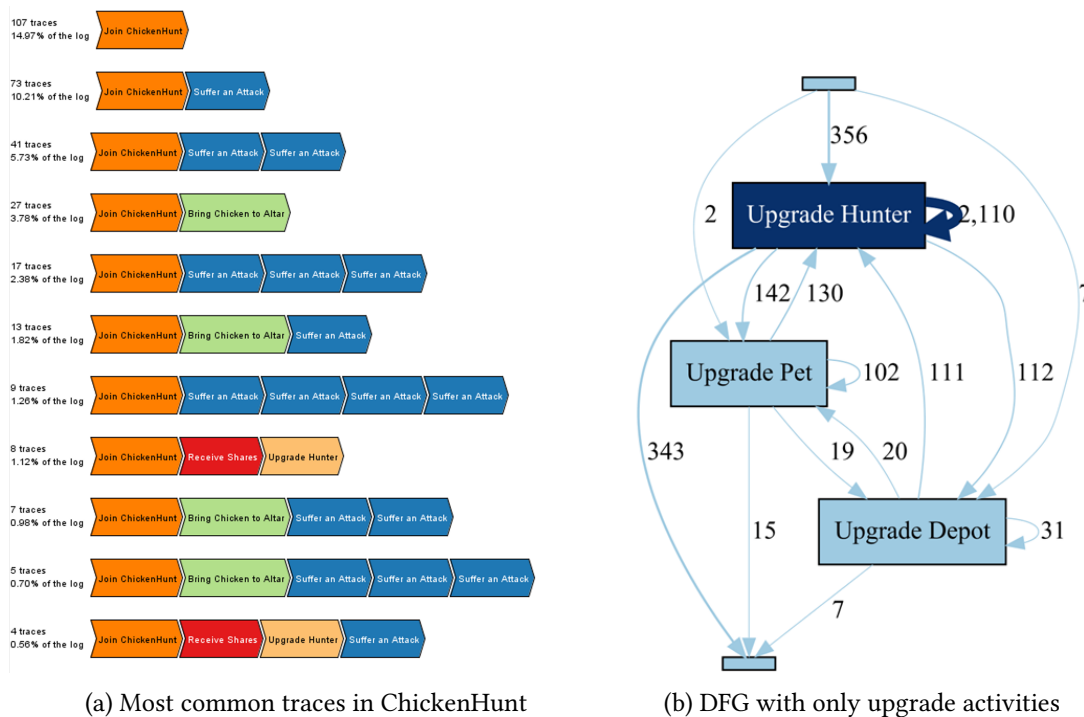


Figure 1: Initial process mining results from the *ChickenHunt* log.

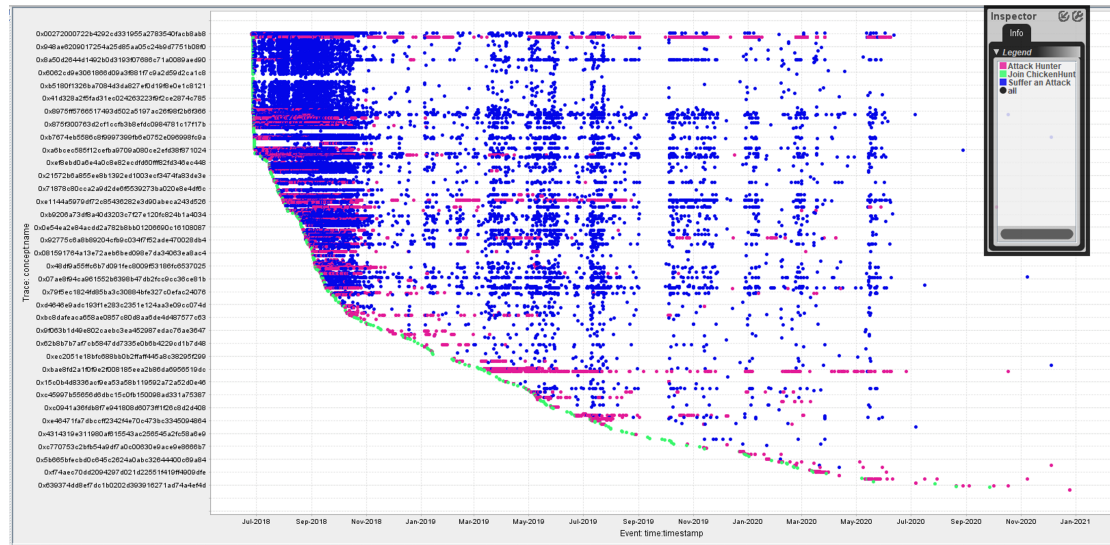


Figure 2: Dotted chart: ChickenHunt attack behavior.

but here we focus on results obtained with ProM. In Fig. 1a, the most common behavior of players is shown: 107 players out of the 715 cases join chicken hunt and never did anything else. Several frequent traces show players joining, and then being attacked (one or more times) without doing anything else. Some players follow a similar pattern, but first they succeed with bringing chickens to the altar. These insights could help understand why players stop early, and be used by the developers when working on improvements to promote the user base to grow.

Players who actively play the game have highly varied individual traces: 402 distinct traces exist for the 715 cases. In Fig. 1b, we analyzed the order and frequency of the different types of upgrades. Upgrading the hunter avatar is the most popular choice, and by far the most frequent first and last upgrade. In other words, active players may also upgrade their pet and their depot, but they typically come back to upgrade their hunter further. These insights, too, appear to be of value for the providers of such a game.

Next, we direct our attention to attack behavior. The dotted chart in Fig. 2 shows only the events from joining, attacking, and suffering from an attack. It can be observed that only a few players attack others, but a large number of players are suffering from attacks. Additionally, the attacks appear to happen in synchronized waves, as indicated by the vertical patterns in the dotted chart. The reasons behind those waves may well be connected to the gas prices (and accordingly the fees) per transaction on Ethereum³: from a visual comparison of the timelines, higher gas prices on Ethereum may well correlate with periods without attacks on ChickenHunt. Presumably, the attackers stole chickens from ordinary users, brought them to the altar, and received Ether in return, all of which entailing transactions with associated fees. If the returns in Ether are not high enough, the fees may well render this operation a financial loss.

³<https://etherscan.io/chart/gasprice>

4. Conclusion

With this paper, we provide a collection of four event logs extracted from blockchain applications, with detailed descriptions and preliminary analyses. The collection is publicly available (see Footnote 1). Currently, it comprises a set of four event logs that were extracted with the tool ELF from DApps deployed and executed on the public Ethereum blockchain. In the paper, we included an analysis based on the ChickenHunt event log, which serves as evidence that insights can be discovered from these logs with standard process mining techniques. For the other logs, analyses are available via the website. The data can be analyzed in much more detail by applying additional process mining methods, and presumably holds blockchain-specific and independent insights which we invite the community to explore.

We plan to amend the collection with additional data sets. In addition, we invite other researchers to contribute their data sets via the openly accessible GitHub repository⁴.

References

- [1] W. M. P. van der Aalst, *Process mining: Data science in action*, Springer-Verlag, Berlin, 2016.
- [2] X. Xu, I. Weber, M. Staples, *Architecture for Blockchain Applications*, Springer, 2019.
- [3] C. Klinkmüller, A. Ponomarev, A. B. Tran, I. Weber, W. M. P. van der Aalst, Mining blockchain processes: Extracting process mining data from blockchain applications, in: *BPM (Blockchain Forum)*, 2019, pp. 71–86.
- [4] C. Klinkmüller, I. Weber, A. Ponomarev, A. B. Tran, W. Aalst, Efficient Logging for Blockchain Applications, *Computing Research Repository (CoRR)* in arXiv abs/2001.10281 (2020). URL: <https://arxiv.org/abs/2001.10281>.
- [5] P. Beck, H. Bockrath, T. Knoche, M. Digtar, T. Petrich, D. Romanchenko, R. Hobeck, L. Pufahl, C. Klinkmüller, I. Weber, A blockchain logging framework for mining blockchain data, in: *BPM (Demos & Resources Forum)*, 2021.
- [6] C. Di Ciccio, et al., Blockchain-based traceability of inter-organisational business processes, in: *BMSD*, 2018.
- [7] R. Hobeck, C. Klinkmüller, H. M. N. D. Bandara, I. Weber, W. van der Aalst, Process mining on blockchain data: A case study of augur, in: *BPM*, 2021.

⁴<https://github.com/ingo-weber/dapp-data>