Klinkmüller, Christopher; Bandara, H. M. N. Dilum; van der Aalst, Wil; Hobeck, Richard; Weber, Ingo

Article — Published Version

On the Suitability of Process Mining for Enhancing Transparency of Blockchain Applications

Business & Information Systems Engineering

Suggested Citation: Klinkmüller, Christopher; Bandara, H. M. N. Dilum; van der Aalst, Wil; Hobeck, Richard; Weber, Ingo (2024) : On the Suitability of Process Mining for Enhancing Transparency of Blockchain Applications, Business & Information Systems Engineering, ISSN 1867-0202, Springer Fachmedien Wiesbaden, Wiesbaden, Vol. 67, Iss. 6, pp. 777-796, https://doi.org/10.1007/s12599-024-00903-5

Bus Inf Syst Eng 67(6):777–796 (2025)
https://doi.org/10.1007/s12599-024-00903-5

RESEARCH PAPER

On the Suitability of Process Mining for Enhancing Transparency of Blockchain Applications

Richard Hobeck · Christopher Klinkmüller · H. M. N. Dilum Bandara · Ingo Weber · Wil van der Aalst

Received: 18 December 2022 / Accepted: 28 June 2024 / Published online: 25 October 2024

Abstract Blockchain technology is known for its transparency properties due to its publicly available, immutable data. Yet, as data availability does not inherently ensure transparency, further analytical methods may be required for human interpretation of data traces. Process mining has emerged as a popular toolbox for understanding processes and how they are executed in practice. The paper studies process mining as a method to enhance the transparency of blockchain data. To this end, two popular Ethereum applications were analyzed using process mining: the prediction and betting marketplace *Augur* and the network marketing platform *Forsage*. Observations from the process-mining analyses are used to discuss if process mining can serve as a method to establish transparency of a blockchain. For both applications, new insights are generated for usage scenarios such as application redesign, security analysis, user behavior analysis, and revealing blind spots in Augur's and Forsage's documentation. The paper concludes that there is evidence that process mining can serve as a method to enhance transparency in blockchains at the cost of technical setup and knowledge acquisition.

Keywords Blockchain · Process mining · Transparency · Process discovery · Conformance checking · Process enhancement · Ethereum

1 Introduction

A blockchain can be characterized as a distributed, append-only data store for transactions (Xu et al. 2019). Second-generation blockchains allow for deploying and executing user-defined programs called *smart contracts*. On this basis, blockchain has emerged as a technology that enables the automation of cross-organizational processes on a neutral platform (Mendling et al. 2018; Weber et al. 2016), and more generally the design, development, and operation of *decentralized applications* (DApps) (Xu et al. 2019). In theory, public blockchains offer transparency properties (Xu et al. 2019, p.19f). That is, public blockchains constitute a distributed computing environment in which users can host, execute, and store applications and data. A core argument for transparency properties of public blockchains is the accessibility of this environment, including access to the deployed programs, stored data, and, if made available, the source code of applications. Following Leonardi and Treem (2020), we argue that information availability alone is insufficient to establish transparency. Additional analytical steps are required to present the data in easily accessible formats and thereby achieve transparency. Hence, examining the behavior of blockchain applications

Accepted after 3 revisions by Daniel Beverungen.

R. Hobeck (✉)
Chair of Service-centric Networking, Technische Universitaet Berlin, Berlin, Germany
e-mail: richard.hobeck@tu-berlin.de

C. Klinkmüller
BPMotion, Sydney, Australia

H. M. N. D. Bandara
Data61, CSIRO, Sydney, Australia

I. Weber (✉)
School of CIT & Fraunhofer-Gesellschaft, Technical University of Munich, Munich, Germany
e-mail: ingo.weber@tum.de

W. van der Aalst
RWTH Aachen University, Aachen, Germany

and users requires effort to turn the information from the blockchain into clear insights.

Process mining (van der Aalst 2016) provides a set of tools to extract knowledge from data, e.g., through the discovery of process models from data without prior information about the process (IEEE Task Force on Process Mining 2011). Process mining has become popular as a toolbox for understanding processes and how they are executed in practice. For example, many case studies ranging from healthcare (Andrews et al. 2018; Mans et al. 2009; Rovani et al. 2015; Suriadi et al. 2014), finance (De Weerdt et al. 2013; Jans et al. 2011), manufacturing (Rozinat et al. 2009), and public services (van der Aalst et al. 2007; Leemans et al. 2019) to software development (Lemos et al. 2011) have applied process mining to analyze processes from different perspectives such as control flow, conformance, drifts, and performance (Reinkemeyer 2020). Nevertheless, process mining on blockchain data has turned out to be a challenging task (Di Ciccio et al. 2018). Hence, recently researchers have created techniques to extract authoritative data from blockchains (Klinkmüller et al. 2019, 2020).

In this article, we complement those propositions by studying the utility of process mining on blockchain data in the context of real-world use cases. To this end, we view process mining from a methodological perspective (van Eck et al. 2015; Klinkmüller et al. 2019). While process mining provides valuable tools for data-driven analysis of all types of processes, we do not only want to understand the utility of these tools but also the feasibility of applying them. Here, we focus on DApp transparency and specifically on two prominent research goals from the blockchain domain (see Sect. 2.3):

G1: Code Validation & Verification: Determine to what extent process mining can contribute to making DApps more transparent by supporting the validation and verification of their source code.

G2: User Behavior Analysis: Determine to what extent process mining can contribute to making DApps more transparent by supporting the analysis of their users' behavior.

To this end, we conduct two in-depth DApp analyses with process mining on popular Ethereum applications: *Augur*[1] and *Forsage*.[2] Augur is a prediction and betting marketplace, where users (1) raise questions about future events, (2) predict and bet on answers, and (3) settle the bets after the event occurred and the answer is known. Forsage is a network marketing (aka multi-level marketing) platform that was very popular until being declared a Ponzi scheme by market regulators in several countries[3],[4]. Note that we presented the Augur analysis in a previous paper (Hobeck et al. 2021) in which we outlined our general experience and findings from using process mining to explore this application in an open-ended analysis. By contrast, in this paper, we examine the utility of process mining for specific application scenarios and synthesize our initial insights with those of the second DApp analysis. In this way, we provide evidence that process mining can enhance the transparency of blockchains and, as a result, generate valuable insights for code validation & verification and user behavior analysis.

The remainder of the paper is structured as follows. First, we motivate our research goals by (1) summarizing blockchain research challenges, (2) outlining why process mining can help address some of them in principle, and (3) reviewing related work. In Sect. 3, we present our methodology, including justifications for choosing Augur and Forsage. Section 4 outlines the data extraction and preprocessing procedures underlying both DApp analyses. The paper's focal point are the DApp analyses for Augur (Sect. 5) and Forsage (Sect. 6). In each section, we describe the respective application and data analyses, covering insights from data exploration, process discovery, conformance checking, and performance analysis. In Sect. 7, we then discuss and synthesize the results to assess the contribution of process mining in enhancing the transparency in a blockchain environment in alignment with our research goals. Finally, we conclude in Sect. 8.

2 Motivation

This section motivates our work. First, we introduce the paper's understanding of transparency in Sect. 2.1 and summarize the process-mining discipline in Sect. 2.2. We then introduce basic blockchain concepts and discuss open research challenges in Sect. 2.3, highlighting code validation & verification and user behavior analysis as challenges that could benefit from process mining. Lastly, we review work related to process mining in the context of blockchain in Sect. 2.4.

2.1 Software Transparency

According to Leite and Cappelli (2010), transparency in software concerns the disclosure of information, i.e., execution data and software functions that transform input to

[1] https://augur.net/, accessed 12 June 2022.

[2] https://forsage.io/, accessed 12 June 2022.

[3] https://www.sec.gov.ph/cdo-2020/forsage-and-forsage-philippines/, accessed 28 June 2022.

[4] https://csimt.gov/2021/04/07/us-state-issues-cease-and-desist-order-against-dapp-forsage/, accessed 28 Aug 2023.

output data. They note that although disclosed, software functions can remain obfuscated if the software code is presented in a way that is hard to read. Leonardi and Treem (2020) have a similar standpoint with respect to transparency and data availability. They argue that efforts to achieve greater transparency by disclosing more data can obfuscate the visibility of information in that data. They call that effect the *transparency paradox* and argue that it takes additional analytical steps to retrieve information from the data. In this paper, we use the transparency notion described by Leite and Cappelli (2010) and follow the argument of Leonardi and Treem (2020) that information availability requires appropriate analytical steps on the disclosed data.

2.2 Process Mining

Process mining (van der Aalst 2016) is increasingly used to monitor and improve operational processes. It offers a rich tool set for analyzing event data, i.e., data that contain information about events occurring during the execution of processes (van der Aalst 2016). Each event must at least have three attributes: (1) a *case identifier* of the case that the event belongs to, (2) an *activity name* that represents the activity whose execution led to the event, and (3) a *timestamp* at which the event occurred. Additional attributes may refer to locations, resources, costs, transaction data, and on the Ethereum blockchain, the consumed gas (i.e., a measure for the computational effort of an operation). A *trace* is the sequence of all events belonging to the same case and sorted by timestamps. Focusing on the activity names only, traces can be transformed into *variants*, i.e., unique sequences of activities. Lastly, an *event log* is a collection of events stored in a format like XES (Extensible Event Stream) (Acampora et al. 2017).

Figure 1 illustrates an overview of process-mining categories (van der Aalst 2016). Note that the categories are shown as rectangles. First, *process discovery* algorithms infer process models, thus visualizing the control flow of the process that generated the data. Second, *conformance checking* analyzes the degree to which the behavior of individual traces and/or entire event logs adheres to a normative process model that describes the expected control flow. Third, *performance analysis* can be used to obtain insights into the key performance indicators like cycle time or staff productivity, and to detect drivers for those indicators. Fourth, *comparative process mining and drift detection* enable the investigation of process behavior under different conditions, allowing analysts to, e.g., inspect behavioral differences between user groups or changes over time. Fifth, *exploration* enables analysts to understand data characteristics using visual analytics and to detect outliers.

Additionally, process-mining research addresses methodological aspects, see e.g., van Eck et al. (2015) and Klinkmüller et al. (2019). Such research aims to understand and provide guidelines for conducting process-mining projects, including aspects like planning, data extraction, data preparation, and insight validation. Sect. 3 presents the process-mining methodology applied in the context of this work.

In essence, process mining provides means to analyze different process perspectives, including control flow, conformance, drifts, and performance (Reinkemeyer 2020). Therefore, we argue that those analyses can reveal meaningful insights from blockchain transaction execution data. That is, we expect insights into the blockchain application behavior to assist software developers with *code validation & verification* in general, and specifically with ensuring that applications correctly implement process requirements. Considering that blockchain events are the results of transactions initiated by users, we also expect that insights into the application behavior reveal insights into *user behaviors*. Note that blockchains record the user address that triggered a specific transaction, enabling us to enrich extracted event logs with this information. Next, we outline our rationale for investigating process mining for blockchain analysis, and in particular for code validation & verification and user behavior analysis.

2.3 Blockchain Research Challenges

A *blockchain* is an append-only store of transactions, distributed across a peer-to-peer network and structured as a linked list of blocks (Xu et al. 2019). Second-generation blockchains also provide a neutral execution infrastructure for running user-defined programs, called *smart contracts*. Applications that operate autonomously through smart contracts and run on top of a blockchain are called *decentralized applications* (DApps). Blockchain provides immutability, transparency, and data integrity to DApps. Similar to traditional enterprise information systems, many DApps support operational processes. For example, there are decentralized autonomous organizations (DAOs) for which blockchain applications define a set of transparent processes and rules that allow the organizations' members to control the organizations without requiring centralized leadership (Prusty 2017). Similarly, cross-organizational processes, such as supply chains, can be facilitated by blockchain technology that enforces business rules and exchanges business information (Mendling et al. 2018; Weber et al. 2016).

There are various reviews and research agendas that map out challenges and directions for blockchain research. Table 1 categorizes challenges identified in those publications. Overall, there are six categories. First, *application*

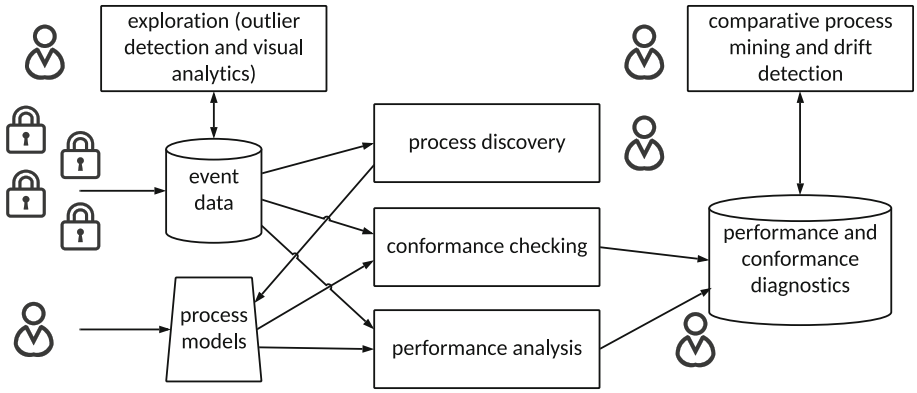

Fig. 1 Process-mining categories for event data analysis

Table 1 Categories of challenges from related work (× – generally mention category; CV – mention challenges related to code validation & verification; UB – mention user behavior analysis)

	Application design	Data privacy	Socio-technical	Scalability	Interoperability	Consensus mechanisms
Risius and Spohrer (2017)	CV	×	UB	×	×	×
Zheng et al. (2018)		×		×		×
Casino et al. (2019)	CV	×	×	×	×	×
Rossi et al. (2019)	CV	×	UB	×	×	×
Zheng et al. (2020)	CV	×			×	
Vacca et al. (2021)	CV			×	×	
Sharma et al. (2023)	CV	×	×			×

design concerns the development of DApps, given the limitations of a blockchain environment. While this category subsumes aspects like programming language features, application optimization, and immutability of smart contracts, a key area of concern is code validation & verification. Second, *data privacy* covers ethical and legal data privacy concerns, as well as data confidentiality challenges. Third, the *socio-technical* category focuses on interactions between users and blockchain technology. In this regard, a central challenge is to understand user behavior and how it is influenced by technological features. Fourth, *scalability* revolves around issues concerning latency, and limited transaction and storage volumes of blockchains. Fifth, *blockchain interoperability* includes integration with existing systems and off-chain components. Lastly, the development of *consensus mechanisms* focuses on protocols for blockchain nodes to agree on the blockchain state in a cryptographically secure and environmentally sustainable way.

In this work, we study process mining as a means to enhance blockchain transparency. Considering the analytical nature of process mining, we believe that process mining can help users in filtering, analyzing, and visualizing blockchain data so that it is easier to understand (Leonardi and Treem 2020, p. 1611). Hence, our interest specifically lies in exploring process mining in the context of challenges that can benefit from the analysis of blockchain data.

First, this includes smart contract *code validation & verification* challenges from the application design category. Here, testing and evaluating DApps after deployment is most relevant to our work, as it is of interest to users who want to monitor if the DApp and any updates comply with expected behavior (Casino et al. 2019; Rossi et al. 2019; Zheng et al. 2020), e.g., to detect fraudulent schemes (Risius and Spohrer 2017; Casino et al. 2019; Zheng et al. 2020). Note that while the deployed code of individual smart contracts is immutable, DApp behavior can be altered at run-time by dynamically changing parameters and the binding of smart contracts. In this regard, efforts by external users are frequently exacerbated by the unavailability or readability of smart contract code (Zheng et al. 2020; Sharma et al. 2023). We argue that analyzing event data generated during smart contract execution using process mining can enhance transparency and assist users in smart contract code validation & verification (G1). In particular, we believe that process-discovery and conformance-checking capabilities support the detection of deviations from the expected behavior. In practical terms, users can access the public blockchain data of a DApp but

require methods to check if the DApp implements services as advertised or not, and here process mining could be the method of choice.

Second, there is consensus that from a socio-technical perspective, blockchain research must become multi-disciplinary and consider the perspectives of all involved stakeholders. While there is a general call for theories that explain how technological blockchain features influence user behavior (Rossi et al. 2019; Risius and Spohrer 2017), we focus on a particular challenge for individual stakeholders. That is, many DApps, such as those implemented for DAOs, can be viewed as process coordination or orchestration mechanisms that allow individuals to transact. This creates the need for DApp users to decide whether they want to trust other users. However, insights into the motivations and behaviors of other users are not readily available. Similar to code validation & verification, mechanisms that can help make user behavior more transparent are thus required (Rossi et al. 2019). Here, we consider process mining to examine user behavior as a potential means for overcoming this challenge (G2). For instance, while data might capture the users that are involved in a case, their general behavior and potentially their motivation only become apparent when analyzing their actions across cases using, e.g., comparative process mining.

2.4 Related Work

Researchers have explored process mining on blockchain data before. First, research has focused on extracting event logs in the XES format (Mühlberger et al. 2019; Klinkmüller et al. 2019, 2020; Koschmider and Duchmann 2021) or a specialized, object-centric log format (Moctar et al. 2023; Hobeck and Weber 2023). The corresponding publications focus on outlining technical implementation details and present illustrative applications of the approaches, in part relying on real-world applications.

Second, building on data extraction capabilities, researchers have suggested technical concepts that use process mining for auditing and monitoring. Di Ciccio et al. (2020) propose an approach for DApp monitoring but do not apply their approach to blockchain data. Corradini et al. (2019) introduce a methodology for DApp auditing that is based on trace clustering and process discovery. While they evaluate their methodology on a small real-world application, they solely focus on measuring the quality of the discovered models and do not provide insights into how their methodology helps to contribute to transparency. Müller and Ruppel (2019) demonstrate an approach that relies on process mining to monitor the blockchain network, but not a single process or DApp.

Finally, only two publications report insights from applying process mining on real-world DApp data. Hobeck et al. (2021) generate value-adding findings by applying process mining to data extracted from a DApp deployed on Ethereum. Lamghari (2023) applies process mining to suggest ideas for extending an Ethereum-based game. Both publications interpret the ability to derive insights as a general indicator of the suitability of process mining.

Considering these publications, the contribution of this paper is twofold. First, it studies process mining as a means to enhance the transparency for DApp users in the context of code validation & verification and user behavior analysis, aspects that have not been addressed by prior work. Second, we aim to understand the utility of process mining from a methodological perspective. That is, we do not only consider the application of a technique to data, but cover all steps required in a process-mining project including e.g., familiarization with a DApp, data extraction, and iterative investigation.

3 Research Methodology

In this paper, we adopt the research methodology depicted in Fig. 2. As a first step, we defined our research goals. To this end, we identified current challenges commonly discussed in the blockchain literature and argued how process mining could in principle contribute to solving them (see Sect. 2). Consequently, our goal is to provide deeper insights into the actual suitability of process mining for code validation & verification and user behavior analysis in real-world settings. As such, this paper can be classified as process-mining research on the individual level (i.e., examining specific tasks like code verification & verification and user behavior analysis) and the ecosystem level (i.e., focused on inter-organizational settings such as decentralized blockchain applications) (vom Brocke et al. 2021).

To investigate the research goals, we then defined our research approach. Considering that the utility of process mining in the context of blockchain transparency has not been studied yet (see Sect. 2.4), we chose an exploratory approach suited to examine such novel phenomena (Recker 2021). We decided to analyze two DApps using process mining, in particular focusing on software validation & verification and user behavior analysis. In both analyses, we followed best practices for process-mining projects (see below and Fig. 3) and conducted all steps required to infer insights from blockchain data. These steps include DApp familiarization, data extraction, and insight evaluation. The goal is to share and discuss observations from these analyses. The observations do not only refer to the insights that we obtained. They also cover methodological aspects that

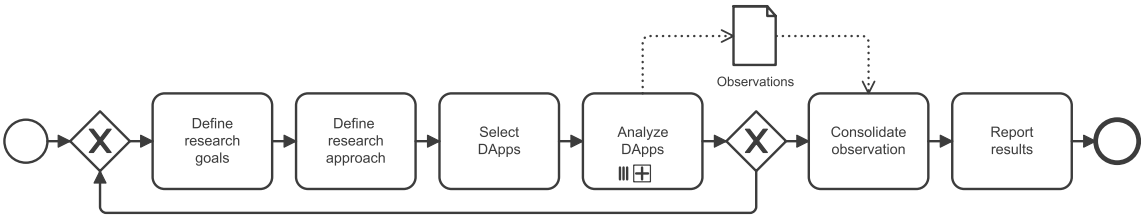

Fig. 2 Overview of the research approach

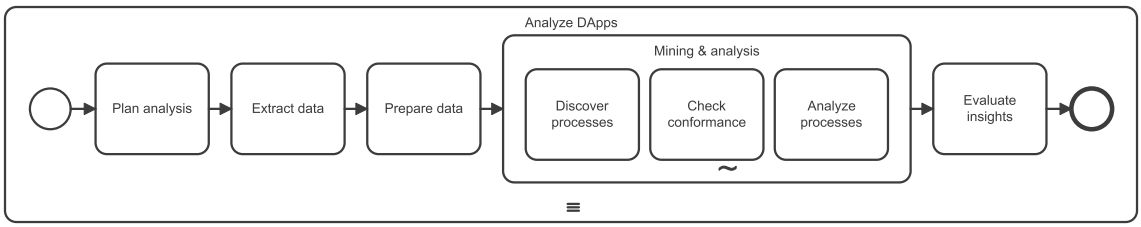

Fig. 3 The process-mining methodology used for the DApp analyses (adapted from van Eck et al. (2015))

must be considered when applying process mining. By relying on such observations, we go beyond a demonstration of the potential value of applying process mining. Instead, we more broadly discuss contextual factors that impact, e.g., the reliability of insights or the effort to generate them, and thus the suitability of process mining for establishing blockchain transparency. Note that we decided to rely on our own observations due to lacking initiatives and experts that applied process mining on blockchain data. The implications of this decision for the validity of our findings are discussed in Sect. 7.3.

After defining the research approach, we selected the DApps, choosing Augur and Forsage as suitable, independent DApps for the following reasons. First, both DApps operate on public Ethereum, the most popular platform for decentralized applications (Wu et al. 2021; Qasse et al. 2020). Hence, time-stamped event data is publicly accessible for both DApps. Second, both DApps were among the most popular Ethereum DApps at times[5,6], resulting in the availability of substantial volume of data to analyze. Third, Augur and Forsage were designed so that log entries are tracked and stored by a central logging contract to reflect major events during contract execution, which enables meaningful analyses and simplifies data extraction. Fourth, information on both DApps is widely available, such as in Augur's white paper (Peterson et al. 2018) and Forsage's marketing material. Fifth, because we wanted to study DApps that were not developed with a process-centric focus, we reviewed the publicly available source code and documentation for both DApps. We did not find any indication that they were developed in a process-driven fashion or that their execution was administered by any business process management system, e.g., as proposed by López-Pintado et al. (2017). Hence, confirmatory evidence regarding our research goals implies that process mining can be a useful tool for blockchain applications, independent of whether the application is based on a process-centric design or not. Lastly, transparency is important for users of both DApps, as they invest cryptocurrency to receive some kind of return on investment. Hence, insights into source code validity and user behavior bear the potential to provide enhanced transparency into the DApps' trustworthiness.

We then analyzed both DApps following van Eck et al. (2015)'s widely adopted methodology for process-mining projects (see Fig. 3). We adopted this methodology to ensure that our analyses are aligned with common practices and hence are representative of how analysts generally conduct such projects. To that end, we collected data for both DApps. Foremost, we extracted log entries from public Ethereum to generate event data. Additionally, we collected supplementary data to reconstruct process models from, e.g., Augur's white paper (Peterson et al. 2018) and Forsage marketing material. These preparation steps are described in Sect. 4. As part of the data preparation for the process-mining analyses, we filtered the event data and created views, in part by altering the case notion or sampling the data (see also Sect. 4). In the case of Forsage, we enriched the event log with data on Ether transfers. During the mining and analyses, we applied process-discovery and conformance-checking algorithms, as well as other forms of process analysis. We also documented our observations. The analyses are described in detail in Sect. 5 and Sect. 6. We evaluated the results of our process-mining projects in

[5] Augur: https://dappradar.com/dapp/augur?range-ha=all, accessed 04 Aug 2023.

[6] Forsage: https://dappradar.com/dapp/forsage?range-ha=all, accessed 04 Aug 2023.

an interview with the lead architect of Augur and using literature on Forsage (Kell et al. 2021).

While we followed the same methodology for both DApps, the plans for the two analyses differed slightly. By analyzing Augur (Hobeck et al. 2021), we initially aimed to understand the general utility of process mining for blockchain applications. We hence conducted an open-ended analysis and among others obtained insights relevant to code validation & verification and user behavior analysis. Considering the relevance of these two topics for blockchain research (see Sect. 2.3), in this paper, we aim to refine the respective observations by conducting the Forsage analysis. The resulting iterative nature of our research approach is reflected in the loop in Fig. 2.

After that, we consolidated the observations from the DApp analyses and discuss the findings with respect to G1 and G2 in Sect. 7. Finally, we report the results of our research in this article.

4 Analysis Preparation

As depicted in Fig. 3, before extracting the data we had to plan the analysis. Besides selecting suitable DApps, this included becoming familiar with the inner workings of the DApps by examining the smart contract code and studying other materials, such as the documentation, Augur's white paper (Peterson et al. 2018), blog entries, posts on social media, and videos. For Augur, this step was exacerbated by the complexity of the source code which consisted of 95 Solidity source code files.[7] While there is a central logging contract that emits relevant execution information and serves as an entry point into understanding the source code, the large number of smart contracts and heavy use of dynamic binding impeded interpretation of the DApp's inner workings. With Forsage being a Ponzi scheme, we faced a different challenge. As also noted by Kell et al. (2021), Forsage's implementation is opaque and not all parts of the source code are available, making it hard to follow the program's logic in code and interpret the log entries. This was further exacerbated by the unavailability of developer documentation, forcing us to rely on second-hand documentation such as reports and blog posts.

Observation 1 Obtaining a foundational understanding of the two DApps and their inner workings was a critical step for data extraction and analysis, but was exacerbated by the unavailability and complexity of source code and documentation.

On a second-generation blockchain, such as Ethereum, log entries are frequently used by developers to communicate information related to the results of smart contract execution to off-chain components. We extracted the log entry data using the open-source Ethereum Logging Framework (ELF) (Klinkmüller et al. 2020). It allowed us to define declarative queries to extract, transform, and format data from Ethereum-based applications. ELF abstracts many technical details, such as establishing a connection to an Ethereum node or orchestrating API calls. However, developing ELF queries that extract high-quality data required us to refine and test the queries in multiple iterations. In each iteration, we (1) inspected sample query results, (2) compared them to other data sources, e.g., data from etherscan.io, and (3) evaluated basic data characteristics through explorative analysis.

Observation 2 While software such as ELF reduced technical implementation effort for data extraction, we still needed multiple iterations to diligently develop queries and ensure high data quality

For each DApp, ELF query execution resulted in one event log in the XES format, containing one event for each log entry recorded by the DApp.[8] For Augur, we extracted information related to 2,897 markets from 9 July 2018 to 10 November 2020. The former date marks the first execution of Augur v1.0, and the latter refers to the last event we extracted when running the ELF manifest on 16 November 2020. However, as outlined in Sect. 5, the launch of Augur v2.0 in July 2020 rendered Augur v1.0 "economically insecure" and unsurprisingly caused a decline in user interest, which already started after the announcement of Augur v2.0 in April 2020.

To account for this decrease, we removed 162 cases that were either created after v2.0 was announced on 2 April 2020 or were not finalized before its actual launch, resulting in 2,735 cases and 22,772 events. This log can be understood to cover the complete lifecycle of Augur v1.0. For Forsage, the log is ≈6 GB in size and contains 1,055,931 cases with 13,368,052 events between 31 January 2020 and 15 April 2021. The former date marks the first DApp execution, while the latter marks the day on which we extracted the log. During that time DApp activity was phasing out and at the time of writing Forsage does not permit new users to join its Ethereum version. Hence, our

[7] The source code for Augur v1 is available on: https://github.com/AugurProject/augur-core, accessed 25 June 2024.

[8] In the spirit of open science, we made artifacts created for data extraction and analysis available online via our blockchain data collection (Bandara et al. 2021): https://ingo-weber.github.io/dapp-data/index.html. For Augur and Forsage, this includes scripts for data extraction and high resolution figures. For Forsage, we also provide Jupyter notebooks for analysis. Note that we could not share scripts for analyses performed in UI-based tools such as ProM or Disco.

log captures events from the main period of activity of Forsage.

For Augur, we concluded that the extracted event log covers relevant process steps. In contrast, for an analysis of Forsage, we noticed a need for additional information. That is because, the event attributes did not provide a meaningful metric to observe payments to and from the smart contracts directly, which is central to understanding the economic workings and effects of Forsage activities. We hence used a second, complementing data source to deal with the challenges. We queried the Etherscan API[9] to retrieve regular and internal transactions to and from Forsage's central contract address. We also extracted the corresponding transaction fees for all user addresses active in the log, for the time frame of the analysis. Based on that, we created a balance sheet, allowing us to sum up each user's total income and spending through interactions with Forsage. The challenges of interpreting Forsage are best illustrated by a comparison of the total user income and spending which were approximately 728k ETH and 768k ETH, respectively. Of the remaining 40.1k ETH, 39.6k ETH can be attributed to transaction fees, i.e., user addresses paying the Ethereum network for including transactions and executing Forsage smart contract functions. At the time of writing, we cannot explain the whereabouts of the remaining 500 ETH (0.06% of the turnover) despite our efforts to that end.

Observation 3 The data available in blockchain log entries of Forsage did not cover all aspects relevant for the analysis. We were able to enhance the log-entry-based data structure with on-chain data on token movement to establish a database for our analysis

In general, the extraction and analysis of blockchain data impose specific demands on the computational infrastructure. First, to have access to historical data, we needed to set up an Ethereum archive node for which the current hardware requirements include up to 12TB SSD disk space[10]. Second, the size of the Forsage event log posed a problem when applying common process-mining tools and platforms (ProM, Disco, and Python) on an off-the-self notebook with a 1.80 GHz CPU and 16 GB RAM. In particular, the used software packages failed to load the event log or failed to allocate sufficient memory when applying process-discovery or conformance-checking algorithms (more details in Sect. 6). To cope with the event log's size and contrast behavior of different groups, we created subsets of the original log based on user success, as follows: *Group A – successful user addresses*: the 1000 addresses with the highest profits, *Group B – average user addresses*: 1000 addresses randomly sampled from user addresses with a balance between median and the 75%-percentile of the total, and *Group C – unsuccessful user addresses*: the 1000 addresses with the highest losses. For the analysis, event names were extended with the Forsage matrix upgrade level when applicable.

Observation 4 Data extraction and analysis required access to sufficient computing infrastructure. For data analysis, we had to create subsets of the Forsage event log due to limitations of available hardware and software.

Note that most data preparation, a crucial step in any process-mining project (see Fig. 3), was implemented as part of the ELF queries. However, depending on the specific analysis, we filtered the event data before applying process mining. Where relevant, details are outlined in Sect. 5 and 6.

5 Process-Mining Analysis of Augur

Augur's white paper (Peterson et al. 2018) characterizes the mechanics of a prediction and betting market: "individuals can speculate on the outcomes of future events; those who forecast the outcome correctly win money, and those who forecast incorrectly lose money." As a betting market organized on Ethereum, the developers claim that Augur bypasses the disadvantages of traditional betting markets, such as trusted market operators and limited participation (Peterson et al. 2018). Currently, two versions of Augur are available in parallel: Augur v1.0 (launched 9 July 2018) and Augur v2.0 (details announced April 2020[11], launched 28 July 2020[12]). To gain user trust, the Augur developers open-sourced the smart contracts and deployed both versions without any option to update or stop them – as giving themselves the privilege to do either might result in the loss of users' cryptocurrency, so omitting that possibility strengthens trustworthiness. Hence, the new version is deployed in parallel to the old version, as such *not comprising an update* in any traditional sense. However, once the new version was deployed and users migrated to it, the old version became "economically insecure" according to the developer team, and therefore should no longer be used. Because prediction markets are long-running and hence extended observation time frames are crucial for their analysis, we nevertheless focused on

[9] https://docs.etherscan.io/api-endpoints/accounts, accessed 1 Jun 2022.

[10] https://ethereum.org/en/developers/docs/nodes-and-clients/archive-nodes/, accessed: 12 Dec 2023.

[11] https://twitter.com/AugurProject/status/1245715269042888706, accessed 14 Mar 2021.

[12] https://www.augur.net/blog/augur-v2-launch/, accessed 14 Mar 2021.

Augur v1.0 and considered the data from its launch until its use was no longer recommended in July 2020.

Augur markets follow a four-stage procedure for which the Augur smart contracts specify 35 different types of events. During *market creation*, a market creator instantiates the market, specifies the market question that revolves around a market event, and appoints a designated reporter to report on the market. In the *trading phase*, traders can place bets on the outcome of the market by buying shares for that outcome. The *reporting phase* begins when the market event occurred, giving the designated reporter three days to provide the first tentative outcome; otherwise, reporting opens to public reporters. Disputants can challenge outcomes by crowd-sourcing a dispute bond. If this bond crosses a preset threshold, the crowd-funded outcome becomes the new tentative outcome. The market finalizes when a tentative outcome has not been successfully disputed for seven days. In principle, disputes could also be resolved by creating parallel instantiations of Augur for the different outcomes (aka forks). This, however, is considered "very disruptive" and has not been triggered yet (Peterson et al. 2018). In the *settlement* phase, market creators receive the market creation fee. Designated reporters receive a fee, if their report represents the final outcome. Lastly, traders settle their positions.

The process model in Fig. 4 provides a detailed view of the market process. We derive this based on the white paper (Peterson et al. 2018) and, where the white paper was not detailed or precise enough, we enriched it with analysis insights. We used this model for conformance checking in Sect. 5.2.

5.1 Exploring the Event Data and Process Discovery

The Augur event log (see Sect. 4) has 2735 cases (each case refers to a market), 22,772 events, and 11 unique activities.

Moreover, there are 414 variants, where 35 variants have at least ten corresponding cases and describe 2203 cases. This implies that 80% of the cases are described by less than 8.5% of variants. 319 cases have a unique sequence of activities. Fig. 5 shows a Directly-Follows Graph (DFG) for the whole log obtained using the Directly Follows visual Miner (Leemans et al. 2019)[8]. The discovered model represents the market process implemented by Augur with strong resemblance to the to-be model depicted in Fig. 4.

Observation 5 Process discovery helped to visualize Augur's smart contract execution and thus to conduct a first coarse-grained validation of the DApp's implementation.

Figure 6 shows a *dotted chart* (van der Aalst 2016) in which each dot represents an event (i.e., 22,772 dots). The x-axis refers to the event time, the y-axis corresponds to the cases (i.e., markets) sorted by the time of the first event, and a dot's color refers to the activity name (e.g., blue represents the creation of a market). The chart shows that many markets were created in the first month after Augur's launch (July/August 2018). After that, there was a steady flow of new cases, until the arrival rate decreased after May 2019. The vertical patterns indicate *batching*, i.e., shorter periods where the same activity occurs in many cases. Some of these batching patterns are highlighted in Fig. 6. For example, on 11 February 2020, activity *claim trading proceeds* is executed 63 times for 53 cases in less than one hour. Similarly, there are bursts of activities *claim trading proceeds* and *redeem as initial reporter* on 7 July 2019. Horizontal patterns indicate a sequence of events for the same case in quick succession. For example, for the market "Ethereum Price at end of March 2019", we witnessed *redeem dispute crowdsourcer* and *claim trading proceeds* four and 132 times, respectively, within a few weeks. Thereby, Fig. 6 also shows that certain market events are subject to seasonality, e.g., termination of markets towards the end of a year, which is an important calendrical and psychological milestone for the occurrence of future events.

Observation 6 Process exploration helped plot an overview of DApp execution data, showing user activity gradients throughout Augur's life cycle, incl. seasonal peak activity intervals.

5.2 Conformance Checking and Unusual Cases

To quantify the differences between expected and observed behavior, we applied *conformance checking*, and in particular alignment-based conformance checking (van der Aalst 2016; Carmona et al. 2018). That is, for each trace in the event log, we searched for a path through the closest model. As a reference point, we created the normative process model in Fig. 4 mapping activities in the event log to the description in the Augur white paper (Peterson et al. 2018).

Observation 7 Conformance checking required information about a normative process that we had to transfer into a process model format suitable for the conformance-checking algorithm.

ProM's diagnostics show that the reference model in Fig. 4 explains 2511 of the 2735 cases, i.e., 224 cases have at least one deviation. There are 21,647 synchronous moves (i.e., events in the log that fit the model) and 1125

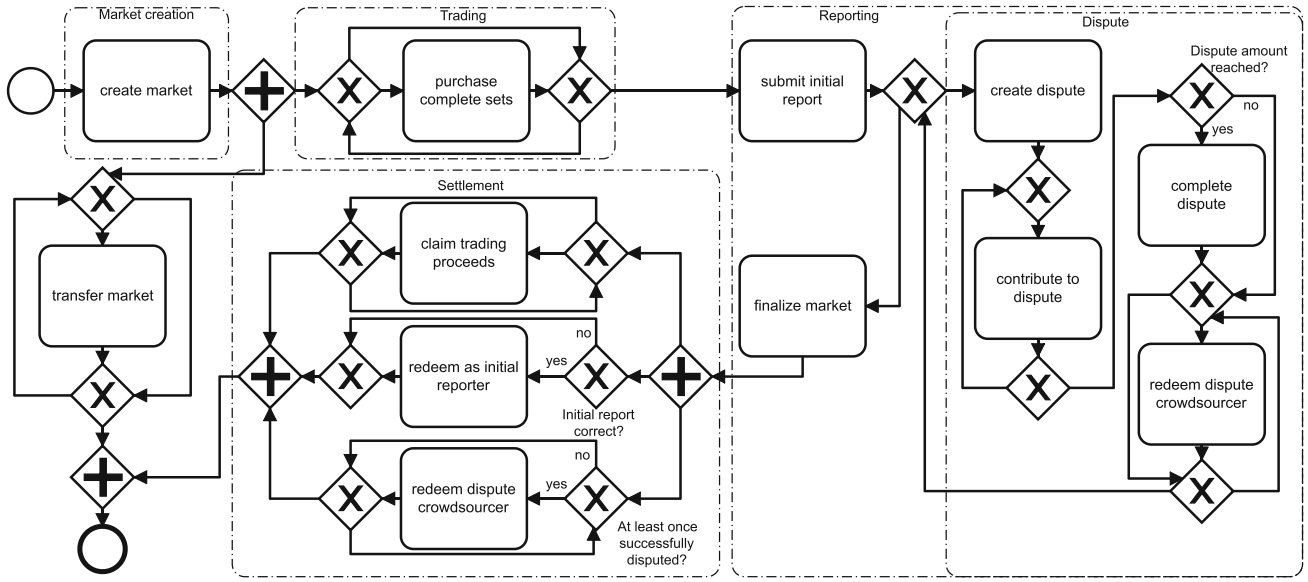

Fig. 4 Augur normative BPMN model

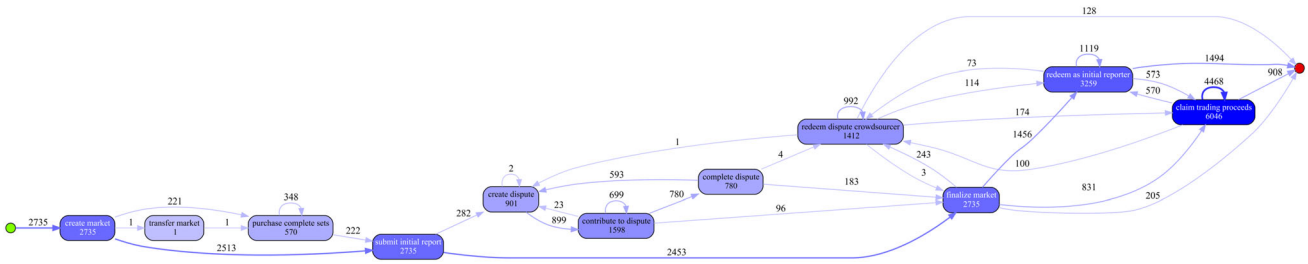

Fig. 5 Directly-follows graph (DFG) for the whole event log without filtering [8]

Fig. 6 Dotted chart showing all 22,772 events. The vertical patterns indicate batching[8]

moves on the log (i.e., events in the log that do not fit the model). However, there are no moves on the model (i.e., missing events). 223 of the 224 deviating cases have multiple *redeem as initial reporter* events, and 1119 of the 1125 log-only moves fall into this category. For a random sample of 20 of these 1119 occurrences, we inspected the underlying blockchain transactions, and observed the following pattern in all cases: the first *redeem as initial*

reporter event resulted in a payout, the second did not; the first and second transactions always came from the same account; and the pairs were close together (between 0 and 47 blocks, the large majority with less than ten blocks). The logging on Augur could be made more precise here, and differentiate successful, legitimate transactions from others.

Observation 8 Blockchain account pseudonymity allowed us to relate a behavioral pattern to a user, although personal information about the individual was not available.

There are multiple possible explanations for the phenomenon of the repeated events, including: (1 the reporter was impatient; (2) the reporter used an automated tool with a time-out before retry, but the tool did not implement *retry* correctly (as per Weber et al. (2017)); or (3) the reporter tried to cheat or hack the system. Given that these attempts were unsuccessful, the reporter had to pay fees, and the same reporter accounts showed the same behavior repeatedly, we find (2) the most plausible of the three scenarios. The automated functionalities could be converted into professionally implemented DApp features to avoid transaction fees.

Observation 9 Using conformance checking, we found that in most instances Augur's execution could be explained with a normative process description. We also found hints towards user-implemented automation protocols interacting with Augur's application interface.

Like in process discovery, it is possible to focus on selected parts of the process in conformance checking. Figure 7 shows conformance-checking results for the dispute subprocess. There is only a single deviating case (see the upper part of Fig. 7) where two instances of two subsequent occurrences of *create dispute* without any contributions in between. The four transactions (two pairs of two) were sent from the same blockchain account, and each pair was included in the ledger in direct succession in the same block. The four transactions initialized four different dispute rounds, although, at any time, only one of those was active. By initializing future dispute rounds, the user "pre-staked" tokens for these future rounds. This was a *bug* in Augur v1.0, but it turned out to be useful and was made a feature in v2.0, as we established in discussions with Augur's chief architect (see Sect. 7.1).

Observation 10 Using conformance checking, we found a deviation from the expected DApp behavior that turned out to be a bug in Augur's smart contract code.

5.3 Performance Analysis

A key capability of process mining is to enrich process models with timing information, e.g., waiting times and service times. We identified the following unusual cases that slowed down market resolutions: In our data, we observed 13 cases with nine or more *complete dispute* events, all of which were created in 2018. The market for the question "Will the weather be good for the Bastille day military parade in Paris tomorrow?" yielded the highest number of contributions to disputes (98) and was created on 13 July 2018. The ambiguity of the question led to a debate in the Augur community about the wording of market questions and application forking. After 15 rounds of dispute, the market was resolved as invalid. On 10 July 2018, four identical markets were created within 17 s asking, "Will Bitcoin go below $6000". One of the markets resolved as invalid after five weeks, while the other three went through dispute rounds until mid-September 2018, before also resolving as invalid, all on the same day. This market question went through the highest number of dispute rounds (20). Eventually, the users learned to pose less ambiguous market questions, leaving less wiggle room for interpretation and reducing the potential for disputes. We ascribe the initial high number of disputes to less mature user behavior showing in ambiguous market questions in the early phase after the DApp's launch.

Observation 11 With the help of performance analysis, we observed user strategy adjustments and maturing effects

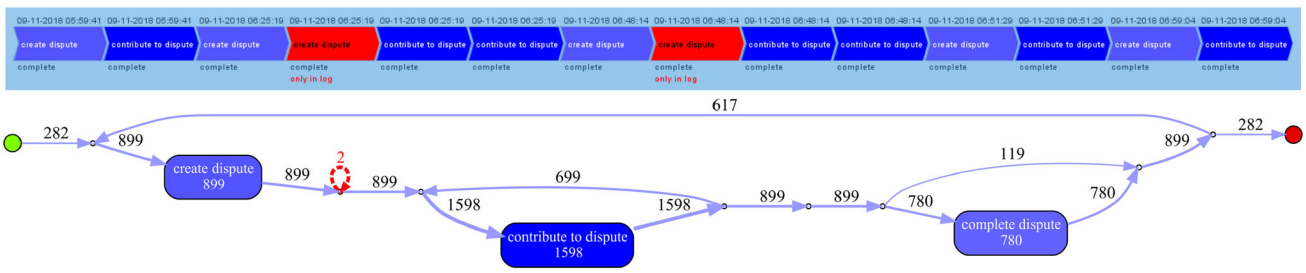

Fig. 7 Conformance-checking results for the dispute phase including the activities *create dispute*, *contribute to dispute*, and *complete dispute*[8]. Only one case is non-fitting

in the community behavior, streamlining market processes over time.

6 Process-Mining Analysis of Forsage

We conducted a second process-mining project based on Forsage data to refine the observations from the previous process-mining project with a focus on code validation & verification and user behavior analysis. The code validation & verification focused on testing Forsage marketing claims.

Forsage is an investment scheme implemented with smart contracts on Ethereum. Forsage's self-description characterizes the DApp as a "[d]ecentralized networking platform based on smart contracts that [..] opens the limitless possibilities of the new economic financial system." Forsage was unmasked as a Ponzi scheme by financial market regulators[3,4] and researchers (Kell et al. 2021), despite the organizers' claims to the contrary.

Forsage's smart contracts specify 12 activities that describe a user journey. A new user registers by transferring 0.05 Ether (ETH) to the Forsage main contract. By registering, a user's Ethereum address gets contingents of investment slots assigned in structures called *matrix*. The investment slots can be filled by other users' investments through referrals. A newly recruited user's fees are immediately passed on to existing members. Once all slots in a matrix are filled, another matrix (i.e., contingent of slots) is issued automatically for the cost of the last slot taken (reinvestment). Matrices are available at 12 different levels and can be unlocked by transferring ETH to the Forsage smart contract. Once unlocked, in addition to the lower-level matrix slots, users receive higher-level matrix slots in the same fashion (per upgrade and reinvestment). Each matrix level is twice as expensive as the previous level ranging from 0.025−51.200 ETH. Users are incentivized to use the upgrade mechanism: user *a* might not receive investments in their matrix slots if a downstream user *b* upgrades to a level higher than *a*'s level. Vice versa, a user can receive extra income if an upstream user lags behind their level.

Forsage has two kinds of matrices: x3 Matrix (named "Matrix 1" in the contract) and x4 Matrix (named "Matrix 2" in the contract). The difference between the two is the number of slots and different distribution of new investments to upstream users and a higher chance for passive income without direct referrals in the x4 Matrix. The normative process is flexible, as depicted in Fig. 8, which we created based on Forsage documentation[13,14].

The Forsage marketing material makes several claims about the DApp such as (i) Forsage's mechanics are transparent[15] and processes are executed according to the documentation; (ii)) high investments right after registration lead to higher profit[16]; (iii) equal chance for profits independent from the registration date; and (iv) active users have a higher chance to succeed.[17]

As explained in Sect. 4, we coped with the large size of Forsage's event log by creating three subsets of the original log based on user success: *Group A – successful user addresses*: the 1000 addresses with the highest profits, *Group B – average user addresses*: 1000 addresses randomly sampled from user addresses with a balance between median and the 75%-percentile of the total, and *Group C – unsuccessful user addresses*: the 1000 addresses with the highest losses[8]. In the remainder of the section, we examine the data on Forsage with process mining and discuss whether these marketing claims hold up. Note that we distinguish between Forsage *users* (people who interact with the application) and Forsage *user addresses* (pseudonymous accounts that users use to interact with Forsage) to take into account that one user can have multiple accounts.

Observation 12 Blockchain account addresses provide a notion to assign events to actors on the blockchain. The pseudonymity of the accounts, however, did not permit relating user accounts to individuals interacting with the application.

6.1 Exploring the Event Data and Process Discovery

The distribution of events across the user groups differs widely, although all groups covered 1000 traces. Group A emitted 351,488 events in 940 case variants, Group B emitted 4463 events in 40 variants, and Group C emitted 55,176 events in 703 variants. The most successful user addresses (Group A) gathered by far the highest number of events with relative consistency, partially validating the fourth marketing claim "Active users have a higher chance to succeed." The highest matrix level reached in Group B

[13] https://lk.forsage.io/guide/, accessed 8 Jun 2022.

[14] https://web.archive.org/web/20220613022844/https://community.forsage.io/, accessed 28 Aug 2023.

[15] https://web.archive.org/web/20210226151101/https://community.forsage.io/en/tutorials/what-is-forsage-and-how-does-it-work, accessed 21 Jun 2022.

[16] https://web.archive.org/web/20220402044348/https://www.youtube.com/watch?v=OjT0I6ofM-0 at 9 min 21 s, accessed 25 Jan 2024.

[17] https://web.archive.org/web/20200604020759/https://www.youtube.com/watch?v=m0NzYwFfGH4 at 5 min 34 s, accessed 25 Jan 2024.

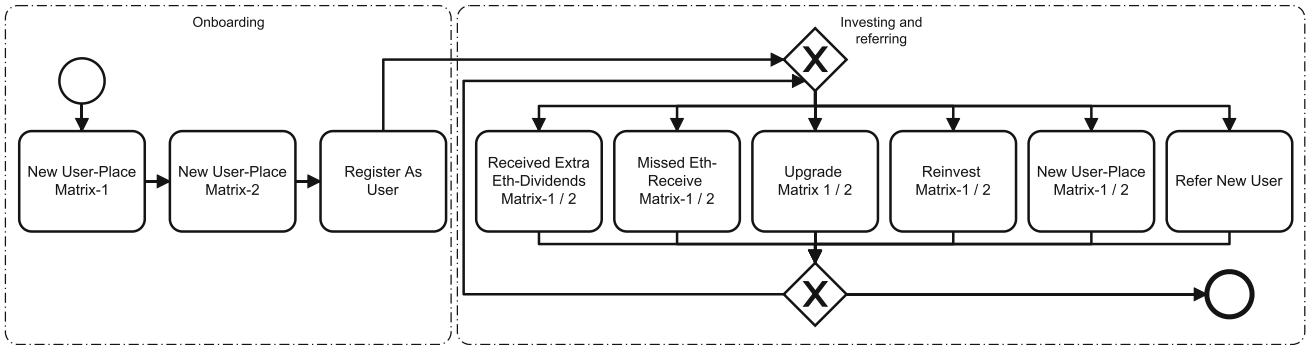

Fig. 8 Forsage normative BPMN model ("Matrix-1/2" indicates activity exists for both matrices)

is level 6 (entailing 1.6 ETH in cost). In Groups A and C, a comparable number of cases upgraded at least one matrix to level 12 (Group A: 381 and Group C: 307), at a cost of at least 102.4 ETH each. Also, in both groups, the upgrades to level 12 mostly concerned the x4 Matrix (Group A: 339 cases and Group C: 255) and less on the x3 Matrix (Group A: 141 and Group C: 77) – these upgrades often happened within a few hours after the registration. The core difference between the two groups is that Group A managed to attract users to sign up in their matrices, while Group C did not. Thus, the second marketing claim "High investments right after registration lead to higher profit" was falsified.

Figure 9 plots the total earnings or losses (x-axis) of user addresses relative to their registration date. Of the 1,055,931 traces, 11.3% (119,382 addresses) had a positive total and 88.7% (936,549) had a negative total. Dark vertical lines show groups of addresses with similar totals that registered on different dates. The most successful addresses (Group A) registered between the Forsage deployment date and December 2020. The least successful addresses (Group C) show a higher concentration of registrations around August 2020. Additionally, users registering after January 2021 almost exclusively lost money, even with high investments. Hence, the registration date seems to influence user success, falsifying the third marketing claim "Equal chance for profits independent from the registration date." Users experiencing potentially high losses to the benefit of others albeit complying with the suggested behavior is typical for fraudulent schemes (Bartoletti et al. 2019).

Observation 13 With process exploration and process discovery, we respectively confirmed or falsified claims about outcomes of usage scenarios of Forsage, based on event order and frequency. We found indicators for a fraudulent scheme using time-sensitive profit/loss analysis.

The dotted chart for Group A is depicted in Fig. 10, with the level indication removed from the event names. The figure shows two clusters of highly active user addresses: one cluster with high activity from January to August 2020 and one cluster with high activity from June 2020 to February 2021. Towards the end of June 2020, short traces with many upgrades and quick but short-lasting high profits occurred. Two blank vertical lines in mid and end of August 2020 indicate reduced user activity. Also, Group C addresses were faster at upgrading their matrices than Group A addresses (also see Sect. 6.3) and often registered in the most active phase of Forsage. Because Group C users concentrated on passive income upgrading mostly their x4 Matrix, we conclude that during the high time of Forsage, there were attempts to overtake the highest levels in sub-pyramids that formed over time. Some users registered accounts in supposedly active user groups and upgraded quickly to the highest level, putting themselves at the top of a sub-pyramid, and receiving passive income through other users' upgrades and referrals. While in some cases that was a successful strategy, in other cases, the supposedly active group did not generate the income hoped for. Users with that strategy can be found in both Groups A and C.

Observation 14 With process exploration and process discovery, we analyzed user engagement with Forsage and user strategies across user groups, including strategy adjustments over time.

6.2 Conformance Checking and Unusual Cases

The Forsage process is designed flexibly (see Fig. 8), meaning that many events can be executed at any time. A conformance-checking approach well-fitted for flexible processes is rule-checking (Letia and Goron 2015). Although flexible, the process described in the Forsage marketing material follows precedence rules which we extracted and checked for.

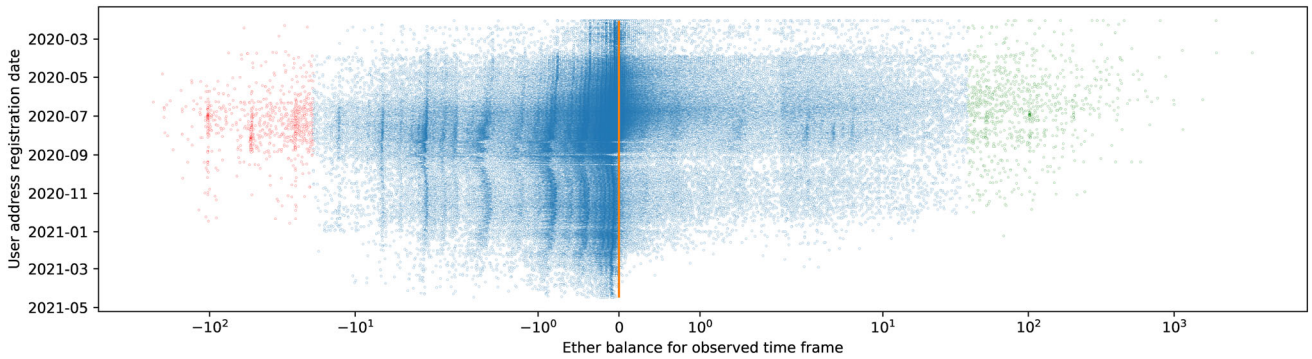

Fig. 9 Total per user address ordered by registration date (Group A: green, Group C: red, else: blue)[8]

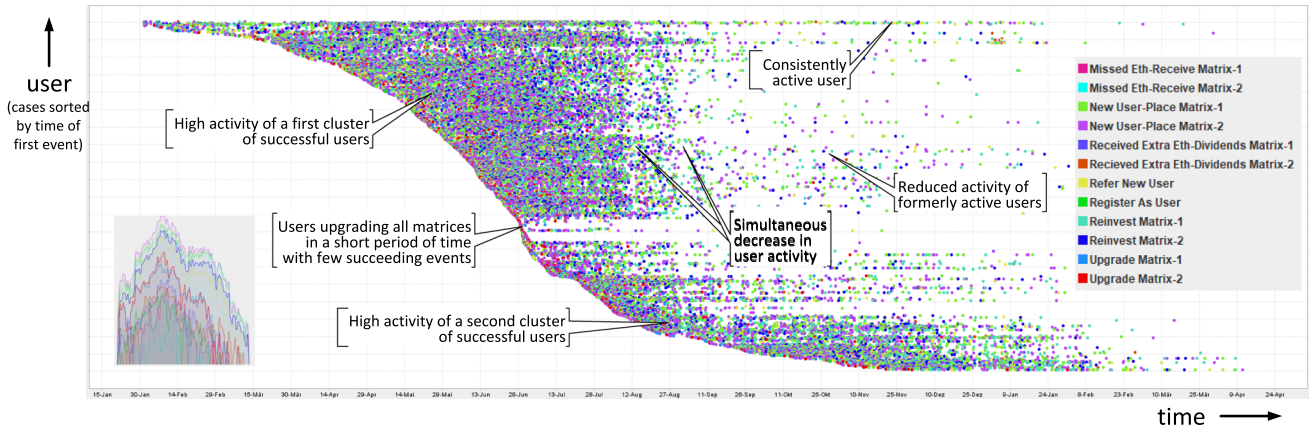

Fig. 10 Dotted chart of the Forsage top user addresses[8]

Observation 15 Conformance checking required information about a normative process, which we captured in precedence rules that served as input for the conformance-checking algorithm.

The Forsage documentation claims that, once a certain spot (place 6) in an x4 Matrix is filled, a new x4 Matrix is issued. We restructured the log assigning matrix sign-ups to investment receivers (i.e., referrers) and added the log attribute *place* to the event name. We then used a precedence rule to check the claim (i.e., the event "New User-Place Matrix-2: Place 6" must be directly followed by the event "Reinvest Matrix-2"), which did *not* hold for 82.2% of the successful, 0.2% of the average, and 27.4% of the unsuccessful user addresses[8]. Investigating the issue, it turned out that reinvestment only took place once *all* spots of an x4 Matrix are filled; and place 6 is not always the last spot taken. This contradicts the documentation. The first marketing claim "Forsage mechanics are transparent" can therefore be refuted.

Observation 16 Using conformance checking, we tested the mechanics of Forsage and were able to point out deviations between the DApp's communicated design specifications and the execution of the smart contract code.

6.3 Performance Analysis

Enhancing the analysis with the temporal perspective shows that Group B accounts have longer reaction times between similar events compared to the other two groups. While it took Group B accounts a median time of 11.7 days (6 times) between the activities "New User-Place Matrix 1" on levels 2 and 3, Group A accounts had a median time of 11.4 h (9959 times). Group C accounts had even shorter leap times of 4.3 min (1,157 times) between the same activities. The median time between "New User-Place Matrix 1" events on levels 2 and 3 was 49.6 h in Group A (14,422 times), 58.5 days in Group B (1 time), and 10.9 days in Group C (594 times). Considering all activities, events followed each other within minutes in Group A, within days to weeks in Group B, and within hours in Group C. Concerning the fourth marketing claim "Active users have a higher chance to succeed," the performance analysis showed that Group A user addresses may not have been the fastest in terms of the median time between filling matrices at low levels, but the shortest median time between events generally.

Observation 17 Performance analysis delivered clues to confirm claims about temporal on-chain activity-outcome relations of Forsage.

We also found social media posts with Forsage referral links and could associate them with some of the Group A accounts.[18] This indicates that accounts with active recruiting efforts may have generated higher income, supporting the fourth claim. However, searching for the most successful accounts or their referral links did not always turn up results.

Observation 18 Blockchain account pseudonymity does not permit linking individuals to blockchain accounts, hampering the analysis of factors for user success based on off-chain components.

7 Discussion

We started this paper by posing two research goals regarding the contributions of process mining to information transparency in a blockchain environment. In particular, we set a thematic focus on the support process mining can offer to address two widely recognized challenges of the blockchain community: *code validation & verification* and *user behavior analysis*. In the preceding sections, we applied process mining to blockchain applications and highlighted our observations. These observations are summarized in Table 2. In this section, we discuss these observations and put them in context of the two research goals. Subsequently, we discuss threats to validity.

7.1 Code Validation and Verification

G1: Code Validation & Verification: Determine to what extent process mining can contribute to making DApps more transparent by supporting the validation and verification of their source code.

In Sect. 2.3, we argued that process mining can contribute to solving *code validation & verification* challenges in the blockchain domain. In the two DApp analyses, we observed that applying process mining tackles these challenges to some degree. Process discovery helped with the coarse-grained validation of a DApp-based service by providing a control-flow visualization (O5). Process discovery also offered more fine-grained insights when the frequency and order of events were the subject of analysis, helping to validate and invalidate claims about expected behavior (O13). In this way, it delivered clues to identify a fraudulent scheme implemented in smart contracts (O13), a specific challenge highlighted in the blockchain literature (Risius and Spohrer 2017; Casino et al. 2019; Zheng et al. 2020). Conformance checking yielded detailed comparisons between the expected behavior of smart contracts and their actual execution. It showed conforming behavior (O9) and deviations from design specifications (O16), validating and invalidating different functionalities of the smart contract code. Conformance checking also helped to detect a software bug in smart contract code (O10) that can be addressed in new or updated versions of the DApp before deployment, a blockchain challenge highlighted by Rossi et al. (2019); Zheng et al. (2020). Using performance analysis, temporal aspects of code validation & verification can be checked (O17).

To validate the correctness of our findings and assess the usefulness of the insights generated by our analyses, we interviewed Paul Gebheim, the chief architect of Augur. Given that we only interviewed one person, we classify results from this interview as *anecdotal evidence*; however, given his position, we believe this evidence is valuable. We asked him to check our assumptions – all of which he confirmed – and presented intermediate results from our analyses to him. From his perspective, using process mining for analyzing DApps generally, and Augur, in particular, provides value in three ways. First, it helps to verify the design mechanisms and check for unintended behavior and bugs in the (immutable) code; immutability poses a challenge from a BPM perspective (Mendling et al. 2018) and software engineering in general (Weber and Staples 2021). Second, process mining provides a clear view of how an application is used, which is also helpful for designing updated versions of an application. Third, it has great potential for technical and economic security analysis, e.g., an auditor could create a model and conformance-check it against actual user behavior. Also, even though a smart contract typically implements a fixed set of rules, analyses of process variability may reveal valuable insights that could help evolve future versions of the smart contract, e.g., to align them better with changed user expectations.

Based on these findings, we infer that process mining can contribute to code validation & verification efforts intended to strengthen DApp transparency. Like other means of visualization and analysis (van Wijk 2005), the utility of process mining for code validation & verification depends on whether users can expect the value of the insights to exceed the cost to generate them. In this regard, we encountered a few issues that might impact the cost or value of process-mining insights.

Familiarization with the DApp and its implementation is a crucial step for preparing data extraction and analysis. Costs to execute this step can be affected by the complexity

[18] https://www.youtube.com/c/SergeyMaslakovprofitbiz, accessed 22 Jun 2022.

Table 2 Observations (O) from the process-mining projects on Augur and Forsage

Augur	Forsage
O1: Obtaining a foundational understanding of the two DApps and their inner workings was a critical step for data extraction and analysis, but was exacerbated by the unavailability and complexity of source code and documentation	
O2: While software such as ELF reduced technical implementation effort for data extraction, we still needed multiple iterations to diligently develop queries and ensure high data quality	
O4: Data extraction and analysis required access to sufficient computing infrastructure. For data analysis, we had to create subsets of the Forsage event log due to limitations of available hardware and software	
O5: Process discovery helped to visualize Augur's smart contract execution and thus to conduct a first coarse-grained validation of the DApp's implementation O6: Process-exploration helped plot an overview of DApp execution data, showing user activity gradients throughout Augur's life cycle, incl. seasonal peak activity intervals O7: Conformance checking required information about a normative process that we had to transfer into a process model format suitable for the conformance-checking algorithm O8: Blockchain account pseudonymity allowed us to relate a behavioral pattern to a user, although personal information about the individual was not available O9: Using conformance checking, we found that in most instances Augur's execution could be explained with a normative process description. We also found hints towards user-implemented automation protocols interacting with Augur's application interface. O10: Using conformance checking, we found a deviation from the expected DApp behavior that turned out to be a bug in Augur's smart contract code O11: With the help of performance analysis, we observed user strategy adjustments and maturing effects in the community behavior, streamlining market processes over time	O3: The data available in blockchain log entries of Forsage did not cover all aspects relevant for the analysis. We were able to enhance the log-entry-based data structure with on-chain data on token movement to establish a database for our analysis O12: Blockchain account addresses provide a notion to assign events to actors on the blockchain. The pseudonymity of the accounts, however, did not permit relating user accounts to individuals interacting with the application O13: With process exploration and process discovery, we respectively confirmed or falsified claims about outcomes of usage scenarios of Forsage, based on event order and frequency. We found indicators for a fraudulent scheme using time-sensitive profit/loss analysis O14: With process exploration and process discovery, we analyzed user engagement with Forsage and user strategies across user groups, including strategy adjustments over time O15: Conformance checking required information about a normative process, which we captured in precedence rules that served as input for the conformance-checking algorithm O16: Using conformance checking, we tested the mechanics of Forsage and were able to point out deviations between the DApp's communicated design specifications and the execution of the smart contract code O17: Performance analysis delivered clues to confirm claims about temporal on-chain activity-outcome relations of Forsage O18: Blockchain account pseudonymity does not permit linking individuals to blockchain accounts, hampering the analysis of factors for user success based on off-chain components

and unavailability of information and code (O1). The costs might further be affected when the log entries emitted by the DApp do not sufficiently cover relevant process steps (O3). In the case of the Ponzi scheme Forsage, this was presumably due to the developers trying to obscure their intentions. Also, emitting log entries (on Ethereum) generally incurs transaction fees (Wood et al. 2014) that are ultimately paid by DApp users and might hence be avoided to not jeopardize DApp usage. Moreover, ensuring data quality required multiple iterations of developing and testing extraction scripts (O2). Testing of the data extraction is a common step in any process-mining project. However, in the context of blockchain applications the complexity (and hence the costs) of data extraction is higher when log entries are insufficient and users need to resort to other on-chain data sources (e.g., blocks, transactions, and transaction replay), or off-chain databases (O3). Lastly, costs are potentially incurred by hardware and capacity requirements that must be satisfied to synchronize and access the data of a blockchain node or to perform data analysis (O4).

Regarding the value generated by process mining, there are a few constraints. In general, process mining focuses on behavioral aspects and might hence not be appropriate when users want to validate & verify aspects beyond DApp behavior, such as safety or costs.

Similarly, we did not apply process mining as a design time or pre-deployment test for software vulnerabilities. That is, we relied on event logs that consist of blockchain events that were emitted at runtime, i.e., when code segments were invoked by users of the deployed DApps. Consequently, we could only analyze code sections and behavior when they were executed during runtime. Code that was not executed remained hidden from our analysis. Nevertheless, our DApp analyses show that process mining can serve as a tool to detect bugs and performance issues for blockchain applications post-deployment (based on actual code execution). Note that users could, in principle, also apply process mining for static analysis before code deployment – however, this is outside the scope of this article.

Lastly, some process-mining analyses (in particular conformance checking) hinge on the documentation and

descriptions of the DApps' to-be processes (O7 and O15), which may not be exhaustive or available in full. For behaviors that are not documented and/or that users might not be aware of, they will thus not be able to validate & verify them based on the actual code execution.

7.2 User Behavior Analysis

G2: User Behavior Analysis: Determine to what extent process mining can contribute to making DApps more transparent by supporting the analysis of their users' behavior.

As pointed out in Sect. 2.3, the blockchain literature recognizes user behavior analysis as a research challenge. Our observations from the DApp analyses imply that process mining can contribute to understanding user behavior. The visualizations from process exploration provided insights about user behavior on a macro level, including waxing and waning user activity in the application's lifetime and peak activity phases (O6). Gaining more fine-grained insights into user behavior required combining several findings from process exploration and process discovery. They included comparisons of behavior between user groups and information on user strategies adjusting to a changing application environment (O14). Additional behavioral adjustments over time, e.g., maturing effects in user behavior, can also be observed with performance analysis (O11). Performance analysis also helped to examine behavior across user groups in more depth, with information on user reaction time (O17). A different aspect of user behavior became evident from conformance checking, where results suggest that users automated part of their interaction with a DApp and thereby, by proxy, their interaction with other users (O9).

Given the insights presented above, we infer that process mining can indeed contribute to DApp transparency by supporting the analysis of user behavior. Similar to code validation & verification, considerations related to the cost and value associated with applying process mining might limit the utility of process mining in a specific context. That is, familiarization can be exacerbated by the complexity and unavailability of information and code (O1), data extraction requires thorough testing (O2), and compute & storage capacity requirements must be met (O4). Regarding the degree to which log entries provide relevant data (O3), we note that essential user activity might not be organized on-chain, but off-chain, e.g., user recruiting on social media platforms in Forsage. Account pseudonymity allowed assigning events to accounts in the first place (O8, O12). However, in part due to the account pseudonymity, establishing data connectivity between off- and on-chain data is challenging (O12, O18). In fact, during our analyses, we learned about off-chain activity but could not reliably relate such data with process participants in our event data on a larger scale (O18), apart from a few exceptions where pseudonyms were disclosed.

The pseudonymity of blockchains does not only impact analysis costs but also limits the value of applying process mining to analyze user behavior. In our analyses (O11, O17), we interpreted results under the assumption that every Ethereum account address represents a unique user. The pseudonymity of accounts on Ethereum does not guarantee that assumption, so that multiple accounts may belong to the same user(s) (O12). For example, for Forsage, our loss-profit calculations may not represent some user's net gains through Forsage. We also cannot distinguish multiple users copying each other's behavior, or a single user applying the same (possibly automated) behavior multiple times. Finally, one (or multiple) accounts might be controlled by a group or team of users. We addressed the issue by differentiating *users* from *user addresses* in our writing (see Sect. 6), but the limitations on the possible insights remain.

7.3 Threats to Validity

There are several threats to the validity of our findings (Wohlin et al. 2012, p. 68). For our analyses, we took on the role of *conductors* of process-mining activities and *observers* examining the process-mining activities. Our taking on the role of conductors does not affect the external validity of our analyses. On the one hand, we were not involved in the development of the DApps and hence took on the role of DApp users whose goal is to make these DApps more transparent. On the other hand, we followed a commonly applied methodology for applying process mining (see Sect. 3). However, we might have introduced a confirmation bias to our findings. The bias is mitigated to a certain degree as we initially conducted the Augur analysis open-ended, before finalizing this paper's research goals.

Furthermore, our analyses results are constrained by a few threats to internal validity. We might have introduced a bias in our conformance-checking approach for Augur. As a basis for conformance checking, we used the entire normative process model (see Fig. 4) and thus the overall control flow without checking the gate conditions for individual cases. That might have led to overly generalized results, ignoring non-conforming cases. Similarly, we might not have included all possible combinations of rules in the rule-based conformance-checking approach and hence might have missed non-conforming cases in Forsage. There is also a chance that Group B user addresses were not representative due to the random sampling. We might have introduced a form of selection bias by the choice of conformance-checking techniques, applying an alignment-

based technique to Augur's well-structured market process and rule checking to Forsage's flexible investment process. Given the nature of the processes, these choices are sensible but might have influenced the quality of the analysis results.

An external threat concerns our selection of DApps. We intended to strengthen the generalizability by conducting two DApp analyses but might have introduced a bias by our choice of use cases – different DApps might not have had deviations between their specifications and actual code execution. Similarly, another DApp's log entries might not have covered relevant sections of the code execution leading to less insights with respect to the research goals. Another external threat to the study may be that the data we performed our analysis on were incomplete or its quality was corrupted. We did, however, take precautions in reducing these threats by validating intermediate results and findings with Augur's user interface and their chief architect and cross-checking our results for Forsage with findings by Kell et al. (2021).

8 Conclusion and Future Work

Information transparency is attributed to blockchain environments. Achieving such transparency in practice, however, depends on the availability of information and adequate data processing. We suggested that process mining could provide a data processing toolkit fit to contribute to blockchain transparency by addressing two blockchain research challenges: (1) code validation & verification and (2) user behavior analysis. We conducted two process-mining analyses for data extracted from the blockchain applications Augur and Forsage. To this end, we used ELF to extract data over essentially the entire lifecycle of Augur v1.0 and Forsage in its two-matrix setup. We used process-mining methods and tools to explore the data, discover process models, and conduct conformance-checking and performance analyses. We were able to show deviations from the expected code execution, we detected a bug in Augur's smart contract code, and we falsified three of the four main claims about the system from Forsage's promotional material. Process mining was also helpful for examining user behavior, including individual user activities and user strategies and their adjustments over time. Finally, we interviewed the chief architect of Augur to validate our insights and understand their usefulness, and cross-checked our Forsage results with academic publications. In summary, we identified patterns and motives in the blockchain data using process mining. Therefore, we conclude that there is strong support that process mining contributes to establishing transparency in blockchain environments in terms of (1) code validation & verification and (2) user behavior analysis.

A critical direction for future research is the development of methods and tools that help users with DApp familiarization, data extraction, and insight validation – three areas that incurred costs or limited the utility of process-mining insights in our analyses. Moreover, the data basis can be extended beyond log entries and token transactions, e.g., to comprise replayed transaction traces in order to ensure completeness of the execution data. Also, our use cases relied on historic data generated partially years before the extraction. Real-time monitoring could provide timely insights into blockchain operations enabling swift responses to irregularities. Integrating process mining into the development lifecycle of smart contracts could provide pre-deployment and post-deployment monitoring, ensuring that smart contracts perform as intended. This integration could also facilitate continuous improvement of blockchain applications, aligning them with user expectations and requirements.

Acknowledgements We are very grateful for the input of Paul Gebheim, chief architect at the Augur Project. We would also like to thank Martin Rebesky and Hendrik Bockrath for writing the first versions of the ELF manifests to extract Augur and Forsage event logs.

Funding Open Access funding enabled and organized by Projekt DEAL.

References

Acampora G, Vitiello A, Stefano B, van der Aalst W, Günther C, Verbeek E (2017) IEEE 1849: the XES standard. IEEE Comput Intell Mag 12(2):4–8

Andrews R, Suriadi S, Wynn M, ter Hofstede AHM, Rothwell S (2018) Improving patient flows at St. Andrew's War Memorial Hospital's emergency department through process mining. In: Business process management cases: digital innovation and business transformation in practice, pp 311–333

Bandara HD, Bockrath H, Hobeck R, Klinkmüller C, Pufahl L, Rebesky M, van der Aalst W, Weber I (2021) Event logs of ethereum-based applications. In: BPM'21: international conference on business process management, Rome, Italy

Bartoletti M, Carta S, Cimoli T, Saia R (2019) Dissecting ponzi schemes on Ethereum: identification, analysis, and impact. arXiv:1703.03779. Accessed 22 Jun 2022

Carmona J, Dongen B, Solti A, Weidlich M (2018) Conformance checking: relating processes and models. Springer, Heidelberg

Casino F, Dasaklis TK, Patsakis C (2019) A systematic literature review of blockchain-based applications: current status, classification and open issues. Telemat Inform 36:55–81

Corradini F, Marcantoni F, Morichetta A, Polini A, Re B, Sampaolo M (2019) Enabling auditing of smart contracts through process mining. In: From software engineering to formal methods and tools, and back, pp 467–480

De Weerdt J, Schupp A, Vanderloock A, Baesens B (2013) Process mining for the multi-faceted analysis of business processes—a case study in a financial services organization. Comput Ind 64(1):57–67

Di Ciccio C et al (2018) Blockchain-based traceability of inter-organisational business processes. In: Business modeling and software design, pp 56–68

Di Ciccio C, Meroni G, Plebani P (2020) Business process monitoring on blockchains: potentials and challenges. In: Enterprise, business-process and information systems modeling, pp 36–51

Hobeck R, Weber I (2023) Towards object-centric process mining for blockchain applications. In: Köpke J, López-Pintado O, Plattfaut R, Rehse JR, Gdowska K, Gonzalez-Lopez F, Munoz-Gama J, Smit K, van der Werf JMEM (eds) Business process management: blockchain, robotic process automation and educators forum. Springer, Cham, pp 51–65

Hobeck R, Klinkmüller C, Bandara HD, Weber I, van der Aalst W (2021) Process mining on blockchain data: a case study of Augur. In: BPM'21: International conference on business process management, Italy, Rome, pp 306–323

IEEE Task Force on Process Mining (2011) Process mining manifesto. In: Bpm workshops, LNBIP, vol 99. Springer, Heidelberg

Jans M, van der Werf JM, Lybaert N, Vanhoof K (2011) A business process mining application for internal transaction fraud mitigation. Expert Syst Appl 38(10):13351–13359

Kell T, Yousaf H, Allen S, Meiklejohn S, Juels A (2021) Forsage: anatomy of a smart-contract pyramid scheme. CoRR abs/2105.04380. Accessed 22 Jun 2022

Klinkmüller C, Müller R, Weber I (2019) Mining process mining practices: an exploratory characterization of information needs in process analytics. In: Hildebrandt T, van Dongen BF, Röglinger M, Mendling J (eds) Business process management. Springer, Cham, pp 322–337

Klinkmüller C, Ponomarev A, Tran AB, Weber I, van der Aalst WMP (2019) Mining blockchain processes: extracting process mining data from blockchain applications. In: BPM blockchain forum, pp 71–86

Klinkmüller C, Weber I, Ponomarev A, Tran AB, van der Aalst W (2020) Efficient logging for blockchain applications. CoRR abs/2001.10281. Accessed 21 Mar 2021

Koschmider A, Duchmann F (2021) Extraction of meaningful events for process mining from blockchain. Springer, Cham, pp 13–29

Lamghari Z (2023) Towards the process mining applicability in the chickenhunt blockchain game. Int J Comput Digital Syst 13(1):1–1

Leemans SJ, Poppe E, Wynn MT (2019) Directly follows-based process mining: exploration and a case study. In: 2019 international conference on process mining (ICPM), pp 25–32. https://doi.org/10.1109/ICPM.2019.00015

Leite JCSP, Cappelli C (2010) Software transparency. Bus Inf Syst Eng 2(3):127–139. https://doi.org/10.1007/s12599-010-0102-z

Lemos AM, Sabino CC, Lima RMF, Oliveira CAL (2011) Using process mining in software development process management: a case study. In: 2011 IEEE international conference on systems, man, and cybernetics, pp 1181–1186

Leonardi PM, Treem JW (2020) Behavioral visibility: a new paradigm for organization studies in the age of digitization, digitalization, and datafication. Organ Stud 41(12):1601–1625

Letia IA, Goron A (2015) Model checking as support for inspecting compliance to rules in flexible processes. J Vis Lang Comput 28:100–121

López-Pintado O, García-Bañuelos L, Dumas M, Weber I (2017) Caterpillar: a blockchain-based business process management system. BPM (Demos) 172

Mans R, Schonenberg MH, Song M, van der Aalst WMP, Bakker P (2009) Application of process mining in healthcare: a case study in a Dutch hospital. Biomed Eng Syst Technol 25:425–438

Mendling J et al (2018) Blockchains for business process management–challenges and opportunities. ACM Transact Manag Inf Syst (TMIS) 9(1):4:1-4:16

Moctar M'Baba L, Assy N, Sellami M, Gaaloul W, Farouk Nanne M (2023) Process mining for artifact-centric blockchain applications. Sim Model Pract Theor 127(102):779. https://doi.org/10.1016/j.simpat.2023.102779

Mühlberger R, Bachhofner S, Di Ciccio C, García-Bañuelos L, López-Pintado O (2019) Extracting event logs for process mining from data stored on the blockchain. In: Business process management workshops, pp 690–703

Müller M, Ruppel P (2019) Process mining for decentralized applications. In: IEEE international conference on decentralized applications and infrastructures, pp 164–169

Peterson J, Krug J, Zoltu M, Williams AK, Alexander S (2018) Augur: a decentralized oracle and prediction market platform. Technical report, Forecast Foundation. https://github.com/AugurProject/whitepaper/blob/master/v1/english/whitepaper.pdf. Accessed 05 Jan 2021

Prusty N (2017) Building blockchain projects. Packt, Birmingham

Qasse IA, Spillner J, Talib MA, Nasir Q (2020) A study on Đapps characteristics. In: 2020 IEEE international conference on decentralized applications and infrastructures (DAPPS), pp 88–93

Recker J (2021) Scientific research in information systems: a beginner's guide. Springer, Heidelberg

Reinkemeyer L (2020) Process mining in action: principles, use cases and outlook. Springer, Heidelberg

Risius M, Spohrer K (2017) A blockchain research framework: what we (don't) know, where we go from here, and how we will get there. Bus Inf Syst Eng 59:385–409

Rossi M, Mueller-Bloch C, Thatcher JB, Beck R (2019) Blockchain research in information systems: current trends and an inclusive future research agenda. J Assoc Inf Syst 20(9):14

Rovani M, Maggi FM, Leoni M, van der Aalst WMP (2015) Declarative process mining in healthcare. Expert Syst Appl 42(23):9236–9251

Rozinat A, de Jong ISM, Günther CW, van der Aalst WMP (2009) Process mining applied to the test process of wafer scanners in ASML. IEEE Trans Syst Man Cybern Part C 39(4):474–479

Sharma P, Jindal R, Borah MD (2023) A review of smart contract-based platforms, applications, and challenges. Cluster Comput 26(1):395–421

Suriadi S, Mans RS, Wynn MT, Partington A, Karnon J (2014) Measuring patient flow variations: a cross-organisational process mining approach. In: Asia pacific business process management, pp 43–58

Vacca A, Di Sorbo A, Visaggio CA, Canfora G (2021) A systematic literature review of blockchain and smart contract development: techniques, tools, and open challenges. J Syst Softw 174(110):891

van der Aalst WMP et al (2007) Business process mining: an industrial application. Inf Syst 32(5):713–732

van der Aalst WMP (2016) Process mining: data science in action. Springer, Heidelberg

van Eck ML, Lu X, Leemans SJJ, van der Aalst WMP (2015) PM2: a process mining project methodology. In: Zdravkovic J, Kirikova M, Johannesson P (eds) Advanced information systems engineering. Springer International, Cham, pp 297–313

van Wijk J (2005) The value of visualization. In: Vis 05. IEEE visualization, pp 79–86. https://doi.org/10.1109/VISUAL.2005.1532781

vom Brocke J, Jans M, Mendling J, Reijers HA (2021) A five-level framework for research on process mining. Bus Inf Syst Eng 1–8

Weber I, Staples M (2021) Programmable money: next-generation conditional payments using blockchain—keynote paper. In: International conference on cloud computing and services science (CLOSER)

Weber I, Xu X, Riveret R, Governatori G, Ponomarev A, Mendling J (2016) Untrusted business process monitoring and execution using blockchain. In: International conference on business process management, Rio de Janeiro, Brazil

Weber I et al (2017) On availability for blockchain-based systems. In: IEEE international symposium on reliable distributed systems (SRDS), pp 64–73

Wohlin C, Runeson P, Höst M, Ohlsson MC, Regnell B, Wesslén A (2012) Experimentation in software engineering. Springer, Heidelberg

Wood G et al (2014) Ethereum: a secure decentralised generalised transaction ledger. Ethereum project yellow paper 151:1–32

Wu K, Ma Y, Huang G, Liu X (2021) A first look at blockchain-based decentralized applications. Softw Pract Exp 51(10):2033–2050

Xu X, Weber I, Staples M (2019) Architecture for blockchain applications. Springer, Heidelberg

Zheng Z, Xie S, Dai HN, Chen X, Wang H (2018) Blockchain challenges and opportunities: a survey. Int J Web Grid Serv 14(4):352–375

Zheng Z, Xie S, Dai HN, Chen W, Chen X, Weng J, Imran M (2020) An overview on smart contracts: challenges, advances and platforms. Futur Gener Comput Syst 105:475–491