Blockchain logging for process mining: a systematic review

Leyla Moctar M'Baba, Mohamed Sellami, Walid Gaaloul, Mohamedade Farouk Nanne

Blockchain logging for process mining: a systematic review

Leyla Moctar M'Baba
Télécom SudParis, SAMOVAR
Institut Polytechnique de Paris
University of Nouakchott Al Aasriya
leyla.moctar_mbaba@telecom-sudparis.eu

Mohamed Sellami
Télécom SudParis, SAMOVAR
Institut Polytechnique de Paris
mohamed.sellami@telecom-sudparis.eu

Walid Gaaloul
Télécom SudParis, SAMOVAR
Institut Polytechnique de Paris
walid.gaaloul@telecom-sudparis.eu

Mohamedade Farouk NANNE
University of Nouakchott Al Aasriya
mohamedade@gmail.com

Abstract

Considerable progress was forcasted for collaborative business processes with the rise of blockchain programmable platforms. One of the saliant promises was auditable traces of business process execution, but practically it has posed challenges specially with regard to blockchain logs' structure who turned out to be inadequate for process mining techniques. Approaches to answer this issue have started to emerge in the literature; some focusing on the creation process of event logs, and others dealing with their retrieval from the blockchain. This work outlines the generic steps required to solve these challenges and analyzes findings in these approaches with a consideration for efficiency and future research directions.

1. Introduction

Blockchain has proved to be a promising vector for the materialization of collaborative inter-organizational business processes by means of consensus enforced trust among organizations, smart contract enabled compliance by design and auditable immutable execution history. While the blockchain based execution of collaborative business processes has been made possible in the literature through various approaches [1, 2], the use of process mining techniques on data generated by this execution has not been largely explored. Even less so, process mining is not widely considered for blockchain applications in general although it allows for user behavior analysis and security audits [3]. This reluctant use of process mining techniques on blockchain data could be explained by the difficulty to extract process data from blockchain data. This difficulty is caused by challenges such as (*i*) the format of blockchain data which is unfit for process mining techniques [4]. Indeed, data related to a process instance or an activity can be found in blockchain transaction data or smart contract events in the form of hash-codes. Therefore, knowledge of the process and reverse-engineering are required to extract event logs from all this data, as hash-codes can not be reverted into the original data [5]. Another reason, is (*ii*) the arbitrary structure of transactions data payloads [6] and heterogeneity of event logs, i.e., event logs resulting from different smart contracts, and even sometimes from the same smart contract, will not have the same structure [7]. Furthermore, (*iii*) the clustering of transactions in a block is independent from process instances, i.e., one block can contain transactions resulting from the execution of multiple unrelated process instances. There is also the difficulty of aggregating the event logs of one process that is executed over multiple smart contracts. The events of this process will not only be fragmented over multiple blocks but multiple smart contract event logs. Therefore manual effort and domain knowledge are necessary to aggregate process data according to process instances.

To tackle these issues several approaches are emerging; some focusing on data extraction from the blockchain and its transformation into process mining suitable formats as an important pre-processing step, and others on structuring event logs prior to storing them on the blockchain. However a map and clear assessment of these approaches, in order to establish their common ground and critically analyse them to identify limitations and define research directions that will fast track this promising research area, has yet to be provided in the literature. To fill this gap, we conducted a systematic review where we studied, assessed and compared these approaches with regard

URI: https://hdl.handle.net/10125/80091
978-0-9981331-5-7

to their common goal, i.e., achieving process mining exploitable data, to determine how well they have served it and to point out the weaknesses and overlooked aspects to be explored in future work. We categorize the studied approaches according to the methods used to identify both process instances and activities, the format used to store blockchain event data, and the process mining technique they serve best. We also contribute by identifying the necessary steps to adapt blockchain data to process mining requirements.

The structure of the paper is as follows. Section 2 provides definitions of concepts used in the paper. Section 3 presents the methodology used to conduct this review. In Section 4 we present in detail the selected works for this review. We compare these works and discuss the results of this comparison in Section 5. Concluding remarks and future directions are presented in Section 6.

2. Background

2.1. Blockchain

Blockchain is a distributed ledger of transactions; duplicated across a network of nodes to prevent single points of failure, immutable to assure no forging is possible, and decentralized to dismiss the need for a central authority to achieve trust. Decentralized trust is reached through a consensus protocol which guarantees that only validated transactions are added to the blockchain. Most blockchains come with a built-in scripting language varying from basic registry operations [8] to Turing completeness [9]. The latter allows for the implementation of executable programs on the blockchain known as smart contracts. The general structure of a blockchain consists of blocks linked together cryptographically and containing transactions. Each transaction contains data, e.g., a transfer of asset/cryptocurrency or a call to a smart contract function. A smart contract function is represented by its signature, i.e., the name of the function and its parameters.

The studied works in this review relies on two blockchain platforms, namely Ethereum and Hyperledger Fabric. Ethereum [9] is the first platform to introduce the concept of smart contracts to blockchain and its cryptocurrency is called Ether. In the Ethereum context smart contracts are programs written in dedicated languages, the most frequently used being Solidity, and they run in a virtual machine known as the Ethereum Virtual Machine (EVM). An account in Ethereum can be externally owned or refer to a contract, each account is identified by an address. When called, an Ethereum smart contract function can trigger events, which are stored alongside the blockchain as logs. These logs serve as meta-data related to the execution and can serve as event notifications.

Hyperpeledger Fabric [10] is a permissioned blockchain, i.e., nodes need permission to join the network. It has a modular and extensible architecture and no cryptocurrency. The transactions serve solely to capture changes in business objects. Information about business objects is stored in a ledger comprised of a blockchain and a state database. The state database is a key value database which allows for quick access to the current state of the objects, i.e., the current values of their attributes, and the blockchain stores the transaction history that led to this state. Smart contracts in Hyperledger, also called chaincodes, are stored on dedicated nodes and their access is subject to privacy policies. As a consequence, the source code of the contract is not part of the blockchain. The only stored information related to the execution of the chaincode is the set of information it queries from the blockchain, called *read set*, and the set of new information it stores in the blockchain, called the *write set*.

2.2. Event data

Business processes (BP) are flows of activities manipulating data and performed with an organization's resources in the scope of an internal strategy or to serve a customer's need. The management and automated execution of these BPs can be achieved through IT tools such as Business Process Management Systems (BPMSs) [11]. Each complete execution of a BP is known as a process instance or a case. It encompasses grouped correlated activities representing a possible play out of a BP model.

The execution of each BP activity results in data known as event data which informs on the many aspects of the activity such as the process instance to which it belongs, its name, the time of execution, the resource responsible for the execution and other attributes. Event data is the basis for process mining (Section 2.3) and so, the availability and the quality of data is a key for applying process mining techniques.

2.3. Process mining

Process mining is a discipline at the intersection of data science and process science concerned with the improvement of processes through analysis of event data provided by BPMSs or process aware information systems. Process mining techniques encompass: process discovery which infers BP models from event data, performance evaluation which detects delays

in activities execution, conformance checking which compares the event data to the original model to find deviation and process enhancement which consists of using information found in event logs to improve or extend a process model [12, 13].

3. Methodology

The purpose of a literature review is to summarise the state of the art in a subject field with the intent to identify specific future research questions [14]. The quality of the synthesis a review contributes to the literature is determined by the rigour of the review protocol. Systematic reviews are literature reviews oriented by research questions. They assess and critically analyse all resources related to these questions. The beginning of a systematic review is the definition of research questions, which will guide the search process. For our review we defined the following research questions:

RQ1 What are the approaches attempting to apply process mining to blockchain data ? This question aims at gathering the work that propose a solution to the issue of blockchain data's incompatibility with process mining techniques. Our focus is on the approach, therefore we do not exclude papers that do not have POCs.

RQ2 How are these approaches pre-processing the blockchain data ? The aim of this question is to determine the steps used by the previously gathered approaches to address the issue.

RQ3 What process mining techniques are they using on the processed data ? The objective of this question is to determine the efficiency of these approaches to solve the issue.

The next step for a systematic review is the literature analysis. For this review the analysis process was conducted in line with the method proposed in [15], which devises literature analysis into five steps: 1) Defining the scope of the review (characteristics of the review); 2) Conceptualizing the topic (identifying key concepts of the topic); 3) Performing the literature search (gathering sources); 4) Analyzing and synthesizing the literature (arranging, discussing, and synthesising the sources); 5) Developing a research agenda for future studies (insightful questions for future research).

For the first step we used the scheme developed by [16] cited in [15] as a good mean to determine the scope of a review. The scope is determined by identifying the focus of the review, its goal, the perspective of the analysis, the sources covered, the organisation of the review and the audience for which it is best suited. By applying this scheme to our work we determined that, since our goal is to gather and assess approaches dealing with the issue of blockchain data's incompatibility with process mining, our focus for this review is on: results, methods and theories. The goal is to synthesize the literature, critically analyze it and identify central issues. In terms of perspective, we adopt a neutral position. The coverage is pivotal since we only include papers related to both Process mining and blockchain. The presentation of the approaches is both conceptual and methodological. The target audience of this paper is mainly specialized scholars but it is also suited for general scholars and practitioners interested in the subject. The scheme for our work is summarized in Table 1.

Table 1. Taxonomy of the Literature Review

Characteristics	categories			
Focus	Results	Methods	Theories	Applications
Goal	Synthesis	Critical Analysis	Identification of Central Issues	
Perspective	Neutral representation		Espousal of position	
Coverage	Exhaustive	Exhaustive Selective	Representative	Pivotal
Organization	Historical	Conceptual	Methodological	
Audience	Specialized Scholars	General Scholars	Practitioners	General Public

For the second step, we conceptualized the topic, i.e., the concepts related to our research questions, as suggested by [15], through a concept mapping as show in Figure 1.

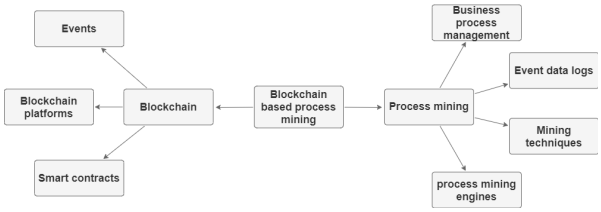

Figure 1. Concept map of the topic.

In line with [15], the search process of the third step was divided into four steps: 1) Identifying scholar databases which provide access to leading journals; 2) Querying the journals/databases on the basis of a keyword search; 3) Backward and/or forward search through the resulting articles; 4) Evaluating the relevance of the results (analysing their titles, abstracts or even full texts).

Four our search process we used the following search query derived from our concept map (Figure 1): "Process mining" OR "Process Discovery" OR "analyzing business process management" OR "BPM analysis" AND Blockchain OR "Distributed ledger" AND "Data extraction" OR Logging OR "Event data" OR "Event logs". We then queried the Springer, IEEE explorer, Scopus, ScienceDirect, ACM Digital Library, Researchgate and Google Scholar databases.

The considered period is from January 2014 (when blockchains with smart contracts started emerging) to March 2021 (time of writing). The queries returned more than 2000 results which were brought down to 30 after title and duplicates filtering. Further filtering was performed through reading of the abstracts and parts of the papers. The inclusion criteria we used were : i) Approaches extracting blockchain data for process mining purposes ii) Approaches transforming blokchain logs into process mining suitable data iii) Approaches storing process event data in blockchain. We also used the following exclusion criteria : i) Approaches extracting blockchain data for data analysis techniques other than process mining ii) Approaches using blockchain to store event data not resulting from a process execution iii) papers not written in English.

This resulted in 9 papers, on which we used backward and forward searches to finally identify 13 papers. The papers were then analyzed, as part of the fourth step, to derive a generic list of steps to reach process mining usable data. Finally, we used the result of this analysis as a basis for the comparison of the papers, which we discussed and used in the fifth step for deriving future research agendas. The low number of papers considered may constitute a threat to the validity of the review. This is due to the novelty of the topic, but the suggested research agendas could contribute to its expansion and a larger review could then be made.

4. Blockchain data for process mining

In this section we present existing approaches related to the usage of blockchain data as a source for process mining. We categorize them according to the source of the event data: the ones resulting from the execution of existing blockchain-based BPMSs, the ones generated by the execution of regular smart contracts, and the ones originating from information systems outside of the blockchain.

4.1. Blockchain-based BPMSs data

In [5], Di Ciccio et al. present an approach allowing the traceability of collaborative BPs executed on the Ethereum blockchain. This approach relies on Caterpillar [1] a blockchain-based BPMS that translates a BP model into a group of smart contracts grouped in two factories. A process factory which generates for each process instance a smart contract where the activity execution logic is encoded. A worklist factory which generates for each process smart contract a worklist smart contract. This worklist smart contract manages the activities' workflow and triggers their execution. Since the worklist smart contract acts as a portal for each activity execution, the authors consider its address as the id of the process instance. Therefore to identify a process instance they fetch the blocks and filter their transactions to retrieve those sent to the address of the worklist smart contract. Caterpillar being designed so as each transaction corresponds to an activity. Therefore, the attributes of the activity, i.e., its identifier and parameters, are obtained by taking each transaction's data field and reverse hashing it with the hash-codes of the signature of the smart contract functions to find the function corresponding to the activity.

In [6], Mühlberger et al. used a similar approach to the previous one. They also relied on caterpillar to identify process instances and activities on Ethereum. In addition, in this work they focus on log files production. First, the signatures of all the process instance smart contract functions are collected and their hashes are computed. Then blocks are collected from an Ethereum client through remote procedure calls (RPC)[1] and the transactions of each block are retrieved and filtered by the signatures hashes, previously obtained. The transactions are then grouped by process instance, i.e., the smart contract address found in the worklist factory. The hash values of the transaction are decoded and the data is extracted and mapped to event and traces attributes which are exported as an XES log [17]. Additional data like the block timestamp can be added to the collected transaction data before the XES log generation.

4.2. Regular smart contract data

The approach of [18] is designed for the blockchain Hyperledger and does not rely on an existing blockchain-based BPMS. In this approach each block is considered as a process instance and its transactions as possible activities. First, blocks are fetched by registering a client on the blockchain to return ledger blocks as JSON objects. Then, read/writes are extracted by iterating through the blocks. By comparing the object states (last and current), changes are detected and corresponding events are transcribed with predefined attributes (ex., event name, event level, and event definition). The events are fed as input to the system. The resulting data is exported as comma-separated values (CSV) files and used to discover process models through Disco [19] and to detect non-conforming behavior of the smart contract with the help of ProM [20].

Since they are concerned mainly with conformance checking, the authors in [21] scan the Ethereum blockchain to get auditable contracts. The criteria

[1] https://www.rfc-editor.org/info/rfc5531

of selection of smart contract candidates is that a contract needs to have more than one hundred related transactions recorded on the blockchain and at least one user linked to multiple interaction on different methods of the smart contract. The list of the smart contract's transactions are then collected using the JSON format[2]. The process instance is considered by the authors to be the sum of all transactions linking one user to the smart contract and each transaction is considered an activity. Therefore, transactions are grouped according to the sender and timestamp thereby creating a trace for each user that interacted with the smart contract. The identified event data is formatted in XES and fed to three process mining algorithms for process discovery: the Heuristics miner, the Inductive miner and the Split miner. This is done using the Apromore process mining tool[3] for the split miner algorithm to get behavioral model and using ProM [20] [4] for the Heuristic and Inductive miner algorithms. The results of this process mining is three models representing different and possible scenarios, on which conformance checking is later applied.

Authors in [22] present a framework consisting of a module for extracting blockchain logs and a module for generating smart contract events. They both rely on a file, named manifest, which contains a description of how the event data has been or is going to be logged in the blockchain. The file can be accessed by all parties in a cross-organization BP without access to the source code of the smart contracts. Among the provided data in the manifest, there is the address of the smart contracts whose data is to be collected, along with the signatures of Solidity events responsible for generating the data. The file aslo contain mapping rules to structure the data into event data according to the process instance and activities. The correctness of the manifest is checked by a validator module provided with the framework. During the extraction, transaction logs are retrieved from the blockchain and filtered using the manifest to obtain the specified smart contracts' logged data. This log data is transformed into event data, according to mapping rules, and exported in XES format. Regarding the generation of logs, the signatures specified in the manifest are fetched and corresponding Solidity events are generated. To reduce the cost of event emission by the smart contract, optimization patterns are included in the framework, such as a value dictionary for mapping parameter values to bits. The framework was tested on CryptoKitties [5], an Ethereum Dapp [6] for breeding and exchanging cats. Event data was extracted from the Dapp's blockchain logs and fed as XES files to the process mining tools ProM [20] and Disco [19] for analysis.

The work in [23] builds upon [22] to provide a user configurable logging framework with extraction capabilities which are not restricted to process data. Similarly to the framework proposed in [22], this one relies also on rules defined in a manifest for the extraction process and for log generation. Unlike the framework in [22], this framework does not rely exclusively on smart contract events to create event logs, but also includes data about the transaction that triggered the events. The extraction is done through iterating over each block, each transaction in the selected block and each log entry in this transaction to access more parameters, e.g. block mining difficulty, transaction gas price or transferred value. This allows for applying filters to obtain fine-grained queries, e.g., applying filters to retrieve transactions from a block based on sender's account addresses. State filters can also be applied providing access the smart contract variables, i.e., the smart contract state. Another type of filters can also be applied with user-defined variables, e.g., to filter transactions based on consumed gas. The target formats vary according to the extraction purpose and different formats are supported: XES, CSV and textual (TXT). Cost reduction is enabled by both the data source choice, i.e., using solidity events parameters and transaction data to create event logs, and compression functionalities, such as value dictionaries and bit mappings. The framework was tested through three case studies : Ethereum network statistics generation, monitoring a prediction market dapp named Augur for contract updates and a study of the Cryptokitties dapp for process model visualization and conformance checking.

In [24] the authors propose a process mining approach for decentralized applications with an emphasis on the control flow, i.e., ordering of activities, and organizational, i.e., resources/actors performing activities, perspectives of a process. Their approach is applied on the Ethereum platform and uses the Solidity events API along with transaction data to collect event data. By considering the control flow perspective the authors are aiming to examine the way Ethereum is used and how that usage changes over time. To do so, they identify a process instance or case as the sum of smart contract events present in one transaction's log since it is ordered as a sequence and grouped under one transaction hash. To provide meaningful event data for the analysis of the organizational perspective, the authors define a list of activity labels according

[2] https://www.rfc-editor.org/info/rfc7159
[3] http://apromore.unicam.it/
[4] http://www.promtools.org/doku.php
[5] https://www.cryptokitties.co/
[6] A Dapp is decentralized application running on the blockchain

to the actors involved, such as utc (user to contract transaction) or ctc (contract to contract call). These labels vary according to the process mining objective. An activity is assigned all the attributes of the log entry, i.e., Solidity event present in the transaction log, it is mapped from. They tested their approach on three years of transactions on Ethereum and derived a process model that gives insight into the different usages of the platform and their complexity over time.

In [25] the authors aim at including the transaction time [7] in event logs generated by a process executed on a blockchain. The motivation behind adding transaction time to event data is to consider temporal properties of this type of processes and how their execution affects the blockchain. They model their framework' functionalities as a process workflow encompassing a sequence of workflows, namely: Extraction workflow, Decoding workflow, Process mining workflow and Transition system analysis workflow. The Extraction workflow collects pending and confirmed transactions, then merges and sorts them according to the timestamp. This step takes as input a manifest where the time elapsed during which pending transactions will be collected and the address of the smart contract linked to them are specified. During the Decoding workflow the event logs of the merged and sorted transactions are decoded into event signatures using the smart contract ABI, provided as input. Then, the transactions and event logs are linked and converted to XES format and stored in one file containing log objects. Each log object contains trace objects, i.e., events representing atomic process activities and describing the execution of one process instance. The Process mining workflow applies the Flexible Heuristic algorithm [26] to generate a process model from the event logs. The trace fitness of the generated model is evaluated through alignment-based conformance checking with the PNetReplayer package[8]. Finally, Transition system analysis workflow augments the discovered model with time information, where a graph of the states of the process is constructed and for each state, remaining, elapsed and sojourn time are calculated.

4.3. External BPMSs data

Although the focus of [27] is to validate event data for process mining, the controlling of log creation present in their approach constitutes an answer to the problem of blockchain event data's inadequacy for process mining. They use the Hyperledger blockchain to process and store event data for process mining as objects, making their future querying easier. The stored event data originated from the execution of business processes by a BPMS external to the blockchain. They developed an application to send event data to a smart contract in time with their generation to be validated and stored in the blockchain. For logging into blockchain, their application needs to provide only the caseID and activityName because the timestamp is appended to the event data object by the smart contract when its validation and storing functions are invoked. The validation consist of making sure no duplicate events are stored and that an event is valid when its mandatory attributes are not empty.

Similarly the approach in [28] tackles the problem by controlling how the logs are created and stored into the blockchain. Their focus is on science processes, similar to business processes in that they can be modeled as self-contained tasks with specific data dependency and execution logic. They execute the processes with the Pegasus workflow tool, then write the resulting log events into a private Ethereum blockchain to be later crawled and analyzed through process mining. To do so they extended the Pegasus tool to enable it to send log events through Aladdin[9] to a Smart Contract written to keep a record of these events. In another fold of their study, the authors investigated the conversion of blockchain data by crawling blocks and transferring their content into a relational database to allow for querying with traditional SQL languages. The motivation behind this transfer is the recurrent observation in the literature about the inefficiency, and in some cases impossible, querying of blockchain for data science purposes. In the last fold they applied machine learning algorithms, namely support vector machines and K-means clustering, on the public Ethereum transaction data to detect suspicious behavior of blockchain accounts.

Authors of [29] aim at solving two problems of the educational process, reporting and course completion rate, by using blockchain. A solution to the first problem is presented as a grades report based on the interactions of the students on the blockchain where time parameters and identifiers are automatically registered. To solve the second problem they suggest to use adaptive learning. To do so they model the characteristics of students, based on their behavior during the educational process, so as to choose the proper learning path for every student. To model the student behavior, identification and timestamps are mandatory and are both provided by the blockchain. The authors' approach is to use blockchain to store all the required data to make creating

[7] The transaction time is the time at which a transaction is included in the blockchain

[8] https://svn.win.tue.nl/trac/prom/browser/Documentation/ReplayerPackageDocumentation

[9] an API for interactions with Ethereum smart contracts over HTTP

reports or student models easier. They collect logs of students' activities during their study of a course from a Moodle platform based system. Then, they register the events of this learning process in the blockchain as smart contracts. The collected logs contain heterogeneous data of learning events, they are analyzed to determine the types of events and their parameters along with their frequency. The results of this analysis are used to create Solidity smart contracts responsible for storing the events. The contracts contain descriptions of the events and a set of parameters for every event type. They also contain handler methods for each event, i.e., when called they initiate events and record them into the blockchain with the timestamp and the student's address. To create a student model or an educational report the data is exported into text file through the geth [10] console. The resulting log is a list of events from which a student's learning path can be traced.

In [30] the aim is to develop a system for information sharing between government and businesses, in which the latter's sensitive information are protected. The purpose of this system is for government organizations to ensure public safety and security and for businesses to improve supply chains. The case study used to identify the requirements of the system is that of goods control by customs. The system consists of a blockchain that stores events and rules for information sharing. These rules are used by businesses to ensure the confidentiality of the information according to their company's policy on access control. The authors designed their own events sharing blockchain, where each new event is added to the ledger as a new block. The consensus is achieved through the confirmation of the event's correctness by two parties and the verification of the confirmation by the nodes of the network. The mandatory data elements of an event are the event type, e.g. container stuffed, the time at which the event happened, and the ID of the object, e.g. container number, to which the event happened. Additional data elements like a description of the goods which the container was stuffed with and the weight of the container can also be added. The event data can be encrypted, therefore meta-data is added to inform on the nature of the encrypted data and thus allow parties to assess their interest in them before requesting access.

In [31], authors investigate a blockchain based solution to media breaks in large business processes. These breaks lead to incomplete event logs, thus, process mining will be performed only on sub-processes. They suppose that using blockchain will allow the integration of all the process participants in order to have an end to end process for which process mining results can be optimal. To verify their hypothesis, they develop an artifact for the case study of a multinational company in the commercial vehicle industry with several production sites and suppliers. Their solution consists of integrating blockchain in the existing IT-systems, to serve as a connector between the processes. In the case of a delivery, the data record is written by the supplier's IT system in the blockchain and forwarded by a smart contract to the receiving company's IT system. The event data is, thus, recorded on the blockchain through smart contracts.

[10] https://geth.ethereum.org/

5. Comparison and discussion

In the previously presented approaches we identified two families dealing differently with the problem of blockchain logs' inadequacy for process mining. They intervene on different phases of the blockchain data life cycle [11] to create event data logs. Depending on which phase they intervene on, we identify two categories of works :

- The one treating the problem at the source, preventing it from happening, and handling themselves the logging of event data into the blockchain as objects. We will name this family 'Pre-blockchain'
- The one transforming data found in blockchain into event data logs. This one will be named 'Post-blockchain'

The previously cited works are categorized according to these families in Table 2.

Table 2. Categorization of the existing works

Pre-blockchain	Post-blockchain
[28] [27] [29] [30] [31]	[5] [6] [18] [21] [22] [23] [24] [25]

In our reading of these works we identified the generic steps followed by each group of approaches to solve the problem. The approaches of the post-blockchain family will be compared according to the method they propose for each step in the preprocessing of blockchain data for process mining, illustrated in figure 2. Whereas, the approaches of the other family will be compared according the steps illustrated in figure 3.

5.1. Post-blockchain : Transforming blockchain data

For the first step, i.e., collecting data, the methods used vary according to the blockchain used and consist

[11] We define the blockchain data life cycle as: the *entrance*, i.e., when data is sent to the blockchain, *the storage*, i.e., when the received data is stored, in general via smart contracts, into the blockchain, and the *retrival*, i.e., when the data is queried or fetched from the blockchain.

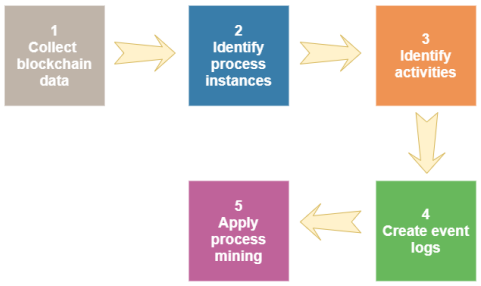

Figure 2. The steps for blockchain data preprocessing for process mining.

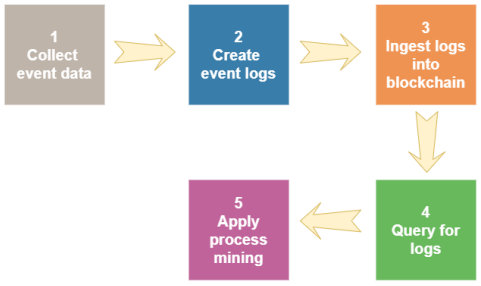

Figure 3. The steps for ingesting event data into blockchain for process mining.

of practices and tools put in place by the community behind the blockchain. Most of the approaches used Ethereum which has a very active community and several tools [12] [13] for querying it are already in place. Approaches that used Hyperledger wrote their own script for collecting the data with the help of Hyperledger dedicated clients.

When it comes to process instance identification two types of approaches emerge, the ones :

Relying on existing BPMSs where process instances are identified as the smart contracts representing them and thus can easily be tracked with the smart contracts addresses [6] [5]

Using blockchain data which is not structured to be queried according to process instance and therefore delimiting process instances, i.e., when does it start and when does it end, becomes a matter of decision making and are manually assigned by authors or users [18] [21] [22] [23] [24] [25]

For the activities identification phase, three types of approaches present themselves, the ones:

Relying on existing BPMSs where the functions of business process smart contract are designed to produce activities as blockchain transactions and therefore a transaction can be directly be identified as an activity [6] [5]

Relying on smart contract events where the activity properties were already logged as smart contract events and identifying them becomes a matter of decoding the data in these events[22]

Using blockchain data where transactions or smart contract events are not by design activities. Therefore the identification requires either a predefined list, a rule for combining various blockchain data into an activity or a mapping from objects changes to activities [18] [21] [23] [24] [25]

The logs creation in most of the approaches consists of structuring the gathered event data in XES or CSV log files. These files then serve as input to the last step, i.e., process mining, where almost all the approaches apply process discovery techniques and some of them also use the discovered processes for conformance checking.

Table 3 summaries the comparison of the works found in this family according to the method used in each preprocessing step.

Table 3. Comparison of the post-blockchain family

	Step 1		Step 2		Step 3				Step 4		Step 5	
	Tools	Script	Smart contract address	Manually ssigned	Transactions	Predefined list	Smart contract events	object changes	XES	CSV	Discovery	Conformance
[21]		x	x		x					x		x
[18]	x			x		x		x		x	x	x
[6]	x		x		x				x		x	x
[5]	x		x		x							
[22]	x			x			x		x		x	x
[23]	x			x	x		x		x	x	x	x
[24]	x			x		x	x				x	
[25]	x		x		x		x		x		x	x

5.2. Pre-blockchain : Ingesting event data into blockchain

The data collecting method in this family of approaches depends on the source, if the latter is a BPMS, the tool to do so is already in place; and if the data comes from different information systems, a script for aggregating the data is used.

Similarly, for the creation of event logs all the approaches either use their own script or an existing tool.

The ingestion of the created logs is done in most cases by sending them to a smart contract, in charge of storing them in the blockchain. This is done through a communication middleware or tools provided by

[12] https://geth.ethereum.org/
[13] https://etherscan.io/

the blockchain platform community or by a dedicated application implemented by the authors. One approach [30] used an events storing oriented blockchain where blocks are events and therefore created logs are ingested simply through transaction submission.

The querying of the event logs is said to be done either directly by sending a transaction to the smart contract or by extending the BPMS with this functionality. No retrieval of the ingestion data for testing purposes was done in the studied works, as a consequence no process mining techniques were applied.

A comparison of these works, according to the method used in each step for data ingestion into blockchain, is grouped in Table 4

Table 4. Comparison of the pre-blockchain family

	Step 1		Step 2		Step 3				Step 4		Step 5	
	Tools	Script	Tools	Script	Dedicated application	Middleware	Smart contract	Transaction submission	Dedicated extension	Smart contract call	Discovery	Conformance checking
[28]	x		x			x	x		x			
[27]		x		x	x		x			x		
[29]	x		x				x			x		
[30]	x		x					x				
[31]	x		x				x					

5.3. Discussion

The set of pre-blockchain approaches have a claim for solving the problem of blockchain logs' inadequacy for process mining by addressing it at the first stage of blockchain data life cycle. This type of approaches has the benefit of alleviating the burden that is the search of event data, i.e., process instances and activities/events, in the logs of blockchain. Since it is all sorted out at the *entrance*, easier retrieval is possible. However, we identified an issue with some of these approaches, which is the systematic outer blockchain origin of their event data. This prevents leveraging the full potential of blockchain for establishing trust in the used data, which in turn affects the reliability of the process mining results for this data.

In the post-blockchain approaches the problem is solved by transforming the data in blockchain logs, and the trust of the event data is not an issue, since the execution happens on the blockchain. Nevertheless, we found that a possible downside to the approaches relying on blockchain data, is the fact that they arbitrarily delimit the process instances and sometimes activities as well. This can lead to questioning the completeness of the event logs generated by these approaches and therefore questioning the accuracy of the process models discovered from them. The works relying on existing BPMSs are not at risk for this issue and as a result we think they serve well a conformance checking purpose, but the overall performance of their process instances search mechanism could be furthermore investigated.

We believe that a combination of these two ways to solve the problem could be more efficient. The control of event logs structure in blockchain by making them objects could be used alongside the execution on the blockchain to ensure both trust and easier retrieval of event data. In other words, we think that the log creation should be addressed at design time by defining more detailed smart contract events with meta-data that will help the search process.

Thus, the querying of event data from the blockchain will be made easier by defining an architecture design for a blockchain based BPMS and blockchain applications with event logs creation in mind.

We also notice that every approach studied is designed for a specific context, i.e., suitable for only one blockchain platform and for only one blockchain-based BPMS, therefore more efforts to achieve blockchain agnosticism and genericity need to be made and it constitutes a research gap in our opinion.

6. Conclusion and Future work

This review of the works in the literature concerned with the fact that blockchain transactional data structure is unfit for process mining showed the existence of two categories of approaches dealing with the issue on different stage of the blockchain data life cycle. Through their examination, we identified the steps for solving the problem followed by each family, and the approaches were compared with respect to those steps. We brought to light potential limitations of each group of approaches, and we provided a suggestion combining parts on both sides. This suggestion will be the basis for our future work on blockchain-based and process mining friendly BPMSs.

References

[1] O. López-Pintado *et al.*, "CATERPILLAR: A Business Process Execution Engine on the Ethereum Blockchain," *CoRR*, vol. abs/1808.03517, 2018.

[2] A. B. Tran *et al.*, "Lorikeet: A model-driven engineering tool for blockchain-based business process execution and asset management," in *Dissertation Award, Demonstration, and Industrial Track at BPM 2018 co-located with 16th International Conference on Business Process Management (BPM 2018), Sydney, Australia*, vol. 2196 of *CEUR Workshop Proceedings*, pp. 56–60, CEUR-WS.org, Sept 2018.

[3] R. Hobeck *et al.*, "Process Mining on Blockchain Data: A Case Study of Augur," in *Business Process Management - 19th International Conference, BPM 2021, Rome, Italy*, pp. 306–323, Springer, sept 2021.

[4] J. Mendling *et al.*, "Blockchains for business process management - challenges and opportunities," *ACM Trans. Manag. Inf. Syst.*, vol. 9, no. 1, pp. 4:1–4:16, 2018.

[5] C. D. Ciccio *et al.*, "Blockchain-based traceability of inter-organisational business processes," in *Business Modeling and Software Design - 8th International Symposium, BMSD 2018, Vienna, Austria*, vol. 319 of *Lecture Notes in Business Information Processing*, pp. 56–68, Springer, July 2018.

[6] R. Mühlberger *et al.*, "Extracting event logs for process mining from data stored on the blockchain," in *Business Process Management Workshops - BPM 2019 International Workshops, Vienna, Austria*, vol. 362 of *Lecture Notes in Business Information Processing*, pp. 690–703, Springer, Sept 2019.

[7] N. Wirawan *et al.*, "Distributed architecture for extracting process mining data from blockchain applications," Nov 2019.

[8] S. Nakamoto, "Bitcoin: A Peer-to-Peer Electronic Cash System," *Cryptography Mailing list*, Mar. 2009.

[9] V. Buterin, "Ethereum: A next-generation smart contract and decentralized application platform," 2014.

[10] E. Androulaki *et al.*, "Hyperledger fabric: a distributed operating system for permissioned blockchains," in *Thirteenth EuroSys Conference, EuroSys 2018, Porto, Portugal*, pp. 30:1–30:15, ACM, Apr 2018.

[11] M. Dumas *et al.*, *Fundamentals of Business Process Management, Second Edition*. Springer, 2018.

[12] W. van der Aalst, "Process mining: Overview and opportunities," *ACM Trans. Manage. Inf. Syst.*, vol. 3, no. 2, pp. 7:1–7:17. Place: New York, NY, USA Publisher: Association for Computing Machinery.

[13] W. M. P. van der Aalst, *Process Mining - Data Science in Action, Second Edition*. Springer, 2016.

[14] J. Rowley and F. Slack, "Conducting a literature review," *Management Research News*, vol. 27, June 2004.

[15] J. vom Brocke *et al.*, "Reconstructing the giant: On the importance of rigour in documenting the literature search process," in *17th European Conference on Information Systems, ECIS 2009, Verona, Italy*, pp. 2206–2217, 2009.

[16] H. M. Cooper, "Organizing knowledge syntheses: A taxonomy of literature reviews," *Knowledge in Society*, vol. 1, p. 104, Mar. 1988.

[17] "IEEE standard for extensible event stream (XES) for achieving interoperability in event logs and event streams," *IEEE Std 1849-2016*, pp. 1–50, 2016.

[18] F. Duchmann and A. Koschmider, "Validation of smart contracts using process mining," in *11th Central European Workshop on Services and their Composition, Bayreuth, Germany*, vol. 2339 of *CEUR Workshop Proceedings*, pp. 13–16, CEUR-WS.org, Feb 2019.

[19] C. W. Günther *et al.*, "Disco: Discover your processes," in *Demonstration Track of the 10th International Conference on Business Process Management (BPM 2012), Tallinn, Estonia*, vol. 940 of *CEUR Workshop Proceedings*, pp. 40–44, CEUR-WS.org, Sept 2012.

[20] B. F. van Dongen *et al.*, "The prom framework: A new era in process mining tool support," in *Applications and Theory of Petri Nets 2005, 26th International Conference, ICATPN 2005, Miami, USA*, vol. 3536 of *Lecture Notes in Computer Science*, pp. 444–454, Springer, June 2005.

[21] F. Corradini *et al.*, "Enabling auditing of smart contracts through process mining," in *From Software Engineering to Formal Methods and Tools, and Back*, vol. 11865 of *Lecture Notes in Computer Science*, pp. 467–480, Springer, 2019.

[22] C. Klinkmüller *et al.*, "Mining blockchain processes: Extracting process mining data from blockchain applications," in *Business Process Management: Blockchain and Central and Eastern Europe Forum - BPM 2019 Blockchain and CEE Forum, Vienna, Austria*, vol. 361 of *Lecture Notes in Business Information Processing*, pp. 71–86, Springer, Sept 2019.

[23] C. Klinkmüller *et al.*, "Efficient logging for blockchain applications," *CoRR*, vol. abs/2001.10281, 2020.

[24] M. Müller and P. Ruppel, "Process mining for decentralized applications," in *IEEE International Conference on Decentralized Applications and Infrastructures, DAPPCON 2019, Newark, CA, USA*, pp. 164–169, IEEE, Apr 2019.

[25] N. Y. Wirawan *et al.*, "Incorporating Transaction Lifecycle Information in Blockchain Process Discovery," in *Blockchain Technology for IoT Applications*, pp. 155–172, Singapore: Springer Singapore, 2021.

[26] A. J. M. M. Weijters *et al.*, "Flexible Heuristics Miner (FHM)," in *IEEE Symposium on Computational Intelligence and Data Mining, CIDM 2011, Paris, France*, pp. 310–317, IEEE, Apr 2011.

[27] B. Ekici *et al.*, "Data Cleaning for Process Mining with Smart Contract," in *2019 4th International Conference on Computer Science and Engineering (UBMK)*, pp. 1–6, Sept. 2019.

[28] E. Brinckman *et al.*, "Techniques and applications for crawling, ingesting and analyzing blockchain data," in *2019 International Conference on Information and Communication Technology Convergence, ICTC 2019, Jeju Island, Korea (South)*, pp. 717–722, IEEE, Oct 2019.

[29] D. Zimina and D. Mouromtsev, "Applying blockchain technology for improvement of the educational process in terms of data processing," in *11th Majorov International Conference on Software Engineering and Computer Systems (MICSECS 2019), Saint Petersburg*, vol. 2590 of *CEUR Workshop Proceedings*, CEUR-WS.org, Dec 2019.

[30] S. van Engelenburg *et al.*, "Design of a software architecture supporting business-to-government information sharing to improve public safety and security - combining business rules, events and blockchain technology," *J. Intell. Inf. Syst.*, vol. 52, no. 3, pp. 595–618, 2019.

[31] S. Tönnissen and F. Teuteberg, "Using blockchain technology for cross-organizational process mining - concept and case study," in *Business Information Systems - 22nd International Conference, BIS 2019, Seville, Spain*, vol. 354 of *Lecture Notes in Business Information Processing*, pp. 121–131, Springer, June 2019.