Preserving Data Secrecy in Inter-organizational Process Mining

Valerio Goretti[0000−0001−9714−4278], Davide Basile[0000−1111−2222−3333], Luca Barbaro[0000−0002−2975−5330], and Claudio Di Ciccio[0000−0001−5570−0475]

Sapienza University of Rome, Italy
name.surname@uniroma1.it

Abstract. Inter-organizational business processes involve multiple independent organizations collaborating to achieve mutual interests. Process mining techniques have the potential to allow these organizations to enhance operational efficiency, improve performance, and deepen the understanding of their business based on the recorded process event data. However, inter-organizational process mining faces substantial challenges, including topical secrecy concerns: The involved organizations may not be willing to expose their own data to run mining algorithms jointly with their counterparts or third parties. In this paper, we introduce a novel approach that unlocks process mining on multiple actors' process event data while safeguarding the secrecy and integrity of the original records in an inter-organizational business setting. To ensure that the data acquisition, merging and elaboration phases are secure and that the processed information is hidden from involved and external actors alike, our approach resorts to decentralized trusted applications running in Trusted Execution Environments (TEEs). We show the feasibility of our solution by showcasing its application to a healthcare scenario.

Keywords: Collaborative Business Processes · Trusted Execution Environment · Encryption · Decentralized Computing

1 Introduction

In today's business landscape, organizations constantly seek ways to enhance operational efficiency, increase performance, and gain valuable insights to improve their processes. Process mining offers techniques to discover, monitor, and improve business processes by extracting knowledge from chronological records known as *event logs*. Organizations record in these ledgers events referring to activities and interactions occurring within a business process. The vast majority of process mining contributions consider *intra-organizational* settings, in which business processes are executed inside individual organizations. However, organizations increasingly recognize the value of collaboration and synergy in achieving operational excellence. *Inter-organizational* business processes involve several independent organizations actively cooperating to achieve a shared objective. Despite the advantages in terms of transparency, performance optimization, and benchmarking that companies can gain from such practices, inter-organizational process mining raises challenges that make it still hardly applicable. The major issue concerns confidentiality. Companies are reluctant to outsource to their partners inside information that is required to execute process mining algorithms. Indeed, the sharing of sensitive operational data across organizational boundaries introduces concerns about data privacy, security, and compliance with regulations. *Trusted Execution Environments* (TEEs) can serve as fundamental enablers to balance the need for insights with the imperative to protect sensitive information in inter-organizational settings. TEEs offer secure contexts that guarantee code integrity and data confidentiality in external devices. *Trusted applications* are tamper-proof software objects running in these environments.

In this paper, we propose a novel approach for inter-organizational process mining that resorts to trusted applications to preserve the secrecy and integrity of shared data. To pursue this aim, we design a decentralized software architecture for a three-staged procedure: (i) the initial exchange of preliminary metadata (ii) the secure transmission of encrypted data amid multiple parties, (iii) the privacy-preserving merge of the shared information segments followed by the isolated and verifiable computation of process discovery algorithms on joined data. We evaluate our proof-of-concept implementation against synthetic and real-world-based data with a convergence test and memory effectiveness assessment.

The remainder of the paper is structured as follows: Section 2 provides an overview of related work inherent to the theme of inter-organizational process mining. In Section 3, we introduce a use case example that considers a healthcare scenario. The high-level architecture of our solution is presented in Section 4. Following on from this, we instantiate the addressed design principles in Section 5 focusing on the employed technologies, workflow, and implementation. In Section 6, we discuss our solution. Finally, we conclude and present directions for future work in Section 7.

2 Related Work

The theme of inter-organizational process mining is discussed in the literature from different perspectives. The work of Müller et al. [7] is the first contribution that considers TEEs in combination with process mining techniques. this work was one of the first to pay attention to data privacy and security within third-party systems that mine other activities' data. In order to preserve the information to be mined, this research proposes a conceptual architecture in which process mining algorithms are executed inside cloud service equipped with trusted execution environment. Inspired by this preliminary contribution, we design an approach where each organization can run process mining algorithms without involving external stakeholders. Unlike Müller et al. work in which an algorithm executed in the cloud sends the same result to all the organizations in the collaboration environment, in our architecture each organization is autonomous to choose when performing the mining operations. Elkoumy et al. [3,2] present a tool called Shareprom. Shareprom allows independent parties to perform mining

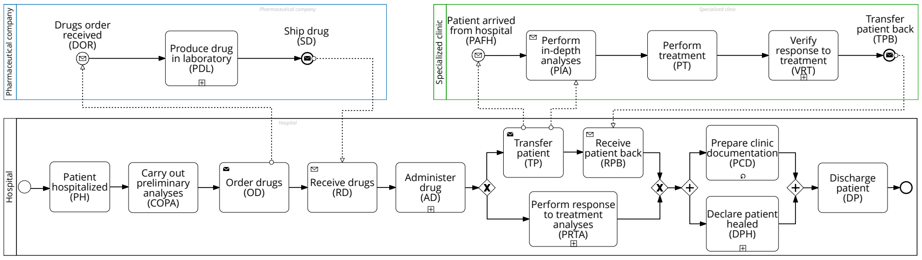

Pharmaceutical company
Drugs order received (DOR)
Produce drug in laboratory (PDL)
Ship drug (SD)
Specialized clinic
Patient arrived from hospital (PAFH)
Perform in-depth analyses (PIA)
Perform treatment (PT)
Verify response to treatment (VRT)
Transfer patient back (TPB)
Hospital
Patient hospitalized (PH)
Carry out preliminary analyses (COPA)
Order drugs (OD)
Receive drugs (RD)
Administer drug (AD)
Transfer patient (TP)
Receive patient back (RPB)
Perform response to treatment analyses (PRTA)
Prepare clinic documentation (PCD)
Declare patient healed (DPH)
Discharge patient (DP)

Fig. 1: A BPMN collaboration diagram of the healthcare scenario.

Table 1: Cases 312 and 711 recorded in the event logs of the Hospital, the Specialized clinic, and the Pharmaceutical company.

Hospital		
Case	Timestamp	Activity
312	2022-07-14T10:36	PH
312	2022-07-14T16:36	COPA
711	2022-07-14T17:21	PH
312	2022-07-14T17:36	OD
711	2022-07-14T23:21	COPA
711	2022-07-15T00:21	OD
711	2022-07-15T18:55	RD
312	2022-07-15T19:06	RD
711	2022-07-15T20:55	AD
312	2022-07-15T21:06	AD
312	2022-07-15T22:06	TP
711	2022-07-16T00:55	PRTA
711	2022-07-16T00:55	PCD
711	2022-07-16T02:55	DPH
711	2022-07-16T04:55	DP
312	2022-07-16T07:06	RPB
312	2022-07-16T09:06	DPH
312	2022-07-16T09:06	PCD
312	2022-07-16T11:06	DP

Pharmaceutical company		
Case	Timestamp	Activity
312	2022-07-15T09:06	DOR
711	2022-07-15T09:30	DOR
312	2022-07-15T11:06	PDL
711	2022-07-15T11:30	PDL
312	2022-07-15T13:06	SD
711	2022-07-15T13:30	SD

Specialized clinic		
Case	Timestamp	Activity
312	2022-07-16T00:06	PAFH
312	2022-07-16T01:06	PIA
312	2022-07-16T03:06	PT
312	2022-07-16T04:06	VRT
312	2022-07-16T05:06	TPB

operations in inter-company contexts without revealing their input data to other parties included in the context. Like our work Shareprom aims to protect the data of the companies involved in the mining operation. Shareprom is only capable of performing operations with directed acyclic graphs that are exchanged in a protected manner between parties. Unlike our work in which logs are exchanged in a protected manner. Using this type of graph restricts the possible use of Shareprom in many contexts, although they are widely used as process representations in process mining, other types of data or representations may be needed in many process mining contexts. In addition, the technology used by Shareprom is secure multiparty computation which does not guarantee high scalability. Our work solves this problem by using trusted applications that execute inside trusted execution environments owned by all parties involved in the inter-organizational context.

3 Motivating Scenario

For our motivating scenario, we focus on a simplified hospitalization process for the treatment of rare diseases that involves the cooperation of three parties: the Hospital, the Pharmaceutical organization, and the Specialized clinic. The process scheme is depicted in the BPMN diagram shown in Fig. 1. For the sake of simplicity, we describe the process through two cases. Alice's journey (case 312) begins when she enters the hospital for the preliminary examinations (the *patient hospitalized* event, PH). The Hospital then places to the Pharmaceutical company an order for the drugs (OD) needed to treat Alice's specific condition. Afterwards, the Pharmaceutical company acknowledges that the drugs order is received (DOR), proceeds to produce the drugs in the laboratory (PDL), and ships the drugs (SD) back to the Hospital. Upon receiving the medications, the Hospital administer the drug (AD), and conducts an assessment to determine if Alice can be treated internally. If specialized care is required, Alice is moved from the Hospital to the Specialized clinic (PAFH). When the patient arrives from the Hospital (PAFH), the Specialized clinic performs in-depth analyses (PIA) and proceeds with the treatment (PT). Once the Specialized clinic had completed the evaluations and verified the response to the alternative treatment (VRT), it transfers the patient back TPB. The Hospital receive the Alice patient back (RPB) and prepares the necessary clinic documentation (PCD). If Alice has successfully recovered, declares her as healed (DPH). When Alice's treatment is complete, the Hospital discharges the patient (DP). Bob enters the Hospital a few hours later than Alice. His hospitalization process is similar to Alice's. However, he does not need specialized care, and his case (711) is only treated by the Hospital. Therefore, the Hospital perform the response to treatment analyses (PRTA) instead of transferring him to the Specialized clinic. Both the National Institute of Statistics of the country in which the three organizations reside, together with the University that hosts/manages the hospital, wish to uncover information on this inter-organizational process for reporting and auditing purposes [**?**] via process analytics. The involved organizations share the urge for such an analysis, and wish to be able to repeat the mining task also in-house. The Hospital, the Specialized clinic, and the Pharmaceutical company have a partial view of the overall unfolding of the inter-organizational process as they record the events stemming from the parts of their pertinence. In Table 1, e.g., we show the traces 312 and 711 recorded by the Hospital (i.e., T^H_{312} and T^H_{711}), the Specialized clinic (i.e., T^S_{312} and T^S_{711}), and the Pharmaceutical company (i.e., T^C_{312} and T^C_{711}). Those traces are projections of the two combined ones for the whole inter-organizational process: T_{312}=⟨PH, COPA, OD, DOR, PDL, SD, RD, AD, TP, PAFH, PIA, PT, VRT, TPB, RPB, DPH, PCD, DP⟩ and T_{711}=⟨PH, COPA, OD, DOR, PDL, SD, RD, AD, TP, PAFH, DPH, PCD, DP⟩. Results stemming from the analysis of the local traces would not provide a full picture. Data should be merged. However, to preserve the privacy of the people involved and safeguard the confidentiality of the information, the involved parties cannot give open access to their traces to other organizations. The diverging interests (being able to conduct process mining on data from multiple sources without giving away the local event logs in-clear) motivate our research. In the following, we describe the design of our solution.

4 Design

In this section, we present the high-level architecture underlying our solution. We consider the main functionalities of each component, avoiding details on the

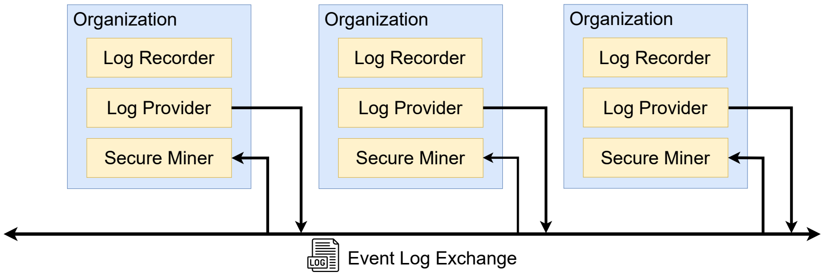


Fig. 2: High-level architectural overview.

employed technologies discussed in the next sections. Once we introduced the architecture, we focus on the `Secure Miner` component that represents the core of our contribution.

4.1 Architecture at large

Our architecture involves different organizational ecosystems characterized by one or more machines. An `Organization` may assume one of the following two different roles or both: *provisioner* if it delivers local event logs to be collaboratively mined; a *miner* whenever it applies process mining algorithms using local event logs retrieved from provisioners. Provisioner `Organizations` collaborate to achieve common objectives and compose inter-organizational business processes whose event logs are scattered across multiple places. Each provisioner produces event logs, recording the operations executed to complete its part in the inter-organizational business process. In Fig. 2, we propose the high-level schematization of our solution. `Organizations` embed three main components, which we describe next: the `Log Recorder`, the `Log Provider`, and the `Secure Miner`. The maintenance of event logs is the core task performed by the `Log Recorder`. This component registers the events taking place in provisioner `Organizations`. The Hospital and the other parties in our running example record Alice and Bob's traces using their `Log Recorders`. The `Log Recorder` is queried by local `Log Providers` of the same `Organization` for event logs to be fed into remote `Secure Miners`. The `Log Provider` component delivers on-demand data to `Secure Miners`. It controls access to local event logs by authenticating data requests generated by miners. `Log Providers` reject demands from unauthorized parties and only permit `Secure Miners` to use the data. In our motivating scenario, the Specialized clinic, Pharmaceutical company, and the Hospital leverage `Log Providers` to authenticate the miner party before sending their logs. The `Secure Miner` shelters external event logs inside a miner ecosystem by preserving data confidentiality and integrity. We provide an in-depth focus on the `Secure Miner` as follows.

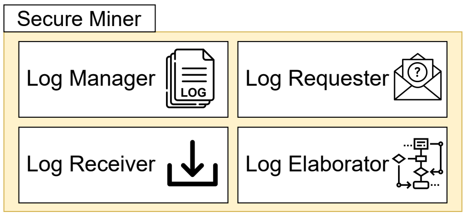


Fig. 3: Subcomponents of the Secure Miner.

4.2 Secure Miner

The primary objective of the `Secure Miner` is to allow miners to securely execute process mining algorithms using event logs retrieved from provisioners such as the Specialized clinic, Pharmaceutical company, and the Hospital of our running example. `Secure Miner`s are isolated components that guarantee tamper-proofing and data confidentiality. In Fig. 3, we show a schematization of a `Secure Miner` in which we distinguish four different subcomponents: the `Log Manager`, the `Log Requester`, the `Log Receiver`, and the `Log Elaborator`. Event logs belonging to provisioners are locked in the `Secure Miner`. We handle these data via the `Log Manager` which prevents malicious parties from having direct access to event logs. These unauthorized entities include any component of the miner `Organization` outside the `Secure Miner`. Referring to our motivating scenario, the `Log Manager` of the miner isolates the traces of Alice and Bob from secrecy-attempting actions generated outside the `Secure Miner`. The `Log Requester` and the `Log Receiver` are the subcomponents that we employ during the event log exchange. `Log Requester`s send authenticable data requests to the `Log Provider` component of provisioners. The `Log Receiver` collects event logs sent by `Log Providers` and entrusts them to the `Log Manager`. The miner of our motivating scenario employs these two components to retrieve the traces of Alice and Bob from the provisioners and to collect this information in the `Secure Miner`. The `Log Elaborator` provides the functionality to securely execute process mining algorithms inside the `Secure Miner`. When activated, the `Log Elaborator` merge the traces locked in the `Secure Miner` in order to have a global view on the inter-organizational process comprensive of activities executed by each the party involved. Aggregated data is employeed by the `Log Elaborator` as input of process mining procedures. Mentioning our motivating scenario, the `Log Elaborator` combine the traces referring to the cases of Alice (i.e., T^{H}_{312}, T^{S}_{312}, and T^{C}_{312}) and Bob (i.e, T^{H}_{711}, T^{S}_{711}, and T^{C}_{711}) generating the chronologically sorted traces T_{312} and T_{711} to be fed into mining algorithms.

5 Realization

In this section, we outline the technical aspects concerning the realization of our approach. Therefore we first present the enabler technologies through which we

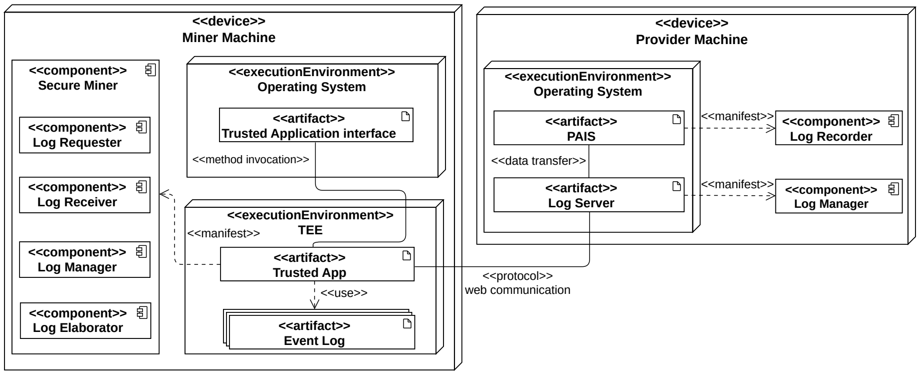


Fig. 4: UML deployment diagram.

instantiate the design principles presented in Section 4. After that, we discuss the interaction workflow between the instantiated technologies. Finally, we show the implementation details.

5.1 Deployment

As follows, we bridge the gap between high-level system architecture and its practical realization. Fig. 4 depicts a *UML deployment diagram* [5] that aims to help with understanding the instantiated infrastructure.

The `Organization Machine` represents the physical computation *device* embracing the software and hardware entities of the company. The `Log Recorder`, the `Log Provider`, and `Secure Miner` are included in the `Organization Machine` as abstract *components*. These logical elements incorporate the core functionalities already discussed in Section 4. The `Organization Machine` is characterized by two *execution environment*s, namely the `Operative System` and the `TEE`.

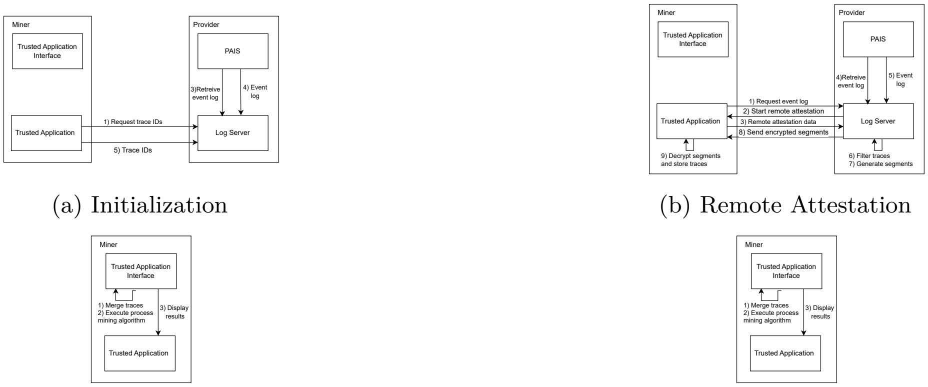
Software entities we expose to the users of the `Organization Machine` run inside the `Operative System`. We manifest the functionalities offered by the `Log Recorder` in the `PAIS` [1]. These systems help users to handle business processes, including accounting and resource management. In our solution, the `PAIS` provides the `Log Server` access to event logs. `Log Servers` are web services that process remote data request and provides event log to miners. We build these entities upon existing web standards such as HTTP[1], FTP[2], and Goopher[3].

`TEE`s create a separated context from the normal `Operating System` to protect code and data through hardware-based security features in a reserved zone of the `Organization Machine`'s CPU. We leverage the security guarantees offered by

[1] https://www.w3.org/Protocols/rfc2616/rfc2616.html. Accessed: September 29, 2023.

[2] https://www.w3.org/Protocols/rfc959/. Accessed: September 29, 2023.

[3] https://datatracker.ietf.org/doc/html/rfc1436. Accessed: September 29, 2023.



(a) Initialization

(b) Remote Attestation

(c) Data Exchange

(d) Computation

these technologies to instantiate a `Trusted Application` to fulfill the functionalities of the `Secure Miner` and its subcomponents. The `Trusted Application` collects the logic to generate verifiable data requests, receive external event logs, store them in the `TEE`, and apply process mining algorithms. Procedures executed by the `Trusted Application` are tamperproof. The `TEE` ensures that the code of the `Trusted Application` executed within it is protected from unauthorized access and malicious manipulations. We employ the isolated context of `TEE` to store `Event Logs` of partner organizations inside the miner machine. The `TEE` provides a mechanism to protect this sensitive information without exposing it to the `Operative System`. The `Trusted Application` is the only entity that can access the `Event Logs` and feed them to process mining algorithms. Users can communicate with the `Trusted Application` via the `Trusted Application Interface`. The `Trusted Application` offers secure methods to safely receive information from the `Operative System` and present the outputs of the computation. These methods are invoked by the `Trusted Application Interface` and instantiate the only communication channel to the `Trusted Application`.

5.2 Interaction flow

As follows, we analyze the data flows and interactions among the introduced technologies. We separate the workflow into subsequent processes, namely *initialization*, *data exchange*, and *computation*. The parties involved in the workflow are a miner (i.e., an organization that executes process mining algorithms) and one or more providers (i.e., partner organizations that serve their event logs).

Initialization. In the initialization, the miner's `Trusted Application` requests preliminary information from the providers' `Log Server` concerning the event logs of an inter-organizational business process. After authenticating the sender, the involved `Log Servers` retrieve the local event log from the `PAIS` and respond to the miner by providing the list of trace IDs in the event log. Hence,

CDC: We need an example. We must provide an example of (simplified) event log excerpt, with a few traces taken from the scenario, show which bits reside in the different machines, and show the final merged event log. Then, discuss it during the description of the workflow, showing how we get from the original pieces to the final one. This description takes time.

the Trusted Application collects the responses and stores them in the TEE.

Remote Attestation.

Talk specifically of remote attestation

Data exchange. Once recorded the preliminary information, the miner starts the data exchange. Therefore, its Trusted Application sends data requests to the Log Servers. The requests include as parameters the list of trace ids and the segment size. Subsequently, the Log Servers starts the *remote attestation* procedure, thanks to which they can verify that the sender of the log request: is a Trusted Application running inside a TEE; comes from a partner organization. This operation involves the exchange of additional messages between the Log Server and the Trusted Application. If the procedure is successful, the miner's identity is verified. Subsequently, the Log Servers retrieve the local event log and filter its traces according to the trace IDs sent by the Trusted Application. Filtered event logs are split into several segments containing traces whose dimension does not exceed the segment size parameter. Log Servers encrypts the segments and send each of them to the Trusted Application. The Trusted Application decrypts the received segments, extracts the traces, and stores them in Event Logs inside the TEE.

Computation. To start a computation routine, the Trusted Application needs all partner organizations to have delivered traces having the same ID. When this occurs, the Trusted Application merges external traces with the owned one. Assembled traces are used as parameters of process mining algorithms executed by the Trusted Application that presents the computation results to the users via the Trusted Application Interface.

5.3 Implementation

It can be reduced

In this section, we describe the implementation of our paper. The implementation proposed integrates a trusted application running in a trusted execution environment and some event logs generated to address the solution proposed in the motivating scenario. The code is available at the following address: https://github.com/dave0909/TEExProcessMining/

We have encoded a well-known process discovery algorithm within the Secure Miner component to demonstrate the capability of conducting process analytics tasks with our approach.

Modify and reposition as needed

This is part of the evaluation. Move it wherever appropriate in the next section.

In order to generate the logs for the execution of the trusted application, we produced a simulation model based on our running example (see Section 3).fed as input into the BIMP tool.[4] The number of log traces generated through BIMP aligns with other works in the state of the art; the generation software was set to 1000 traces. Following the generation, the synthetic event log relating to the process model was filtered via ProM.[5] We were able to filter the logs based

Mention Sepsis here

[4] https://bimp.cs.ut.ee

[5] https://promtools.org

on attribute values, which allowed us to filter the synthetic log according to the resource involved in the activities. Referring to the motivating scenario, the resources involved are the hospital, the specialized clinic, and the pharmaceutical company. In this way, we created three separate event logs from the initial event log, which were used to exchange data between the organizations.

To implement the trusted applications, we used the EGo,[6] a framework to encode programs for TEEs in GO.[7] Within the TA there is the "Secure Miner" module, which allows logs from other organisations to be requested, managed, and processed. Log processing is made possible by the implementation of the "Heuristc Miner" process mining algorithm**??**, which takes the log traces as input and performs a discovery operation. The output of the algorithm is a PNML[8](Petri Net Markup Language) which allows the representation of Petri nets that graphically illustrate the model calculated by the algorithm. In order to generate the graphic image of the Petri net, the WoPed[9] software was used, which takes as input a PNML file and provides the graphic representation of the Petri net.

CDC: Double quotes MUST be written as “ and ”. Besides, why are we using quotes to indicate components here?

Another fundamental module within the TA is that of the Log Provider. This part of the TA is also written in Go and is listening on one of the ports set up by the organisation owning the application. It accepts requests made by other organisations and forwards its log.

6 Discussion

In this section, we evaluate the proposed approach. The primary objective of the Section 6.2 is to delve into a convergence analysis by evaluating the efficacy of the collaborative data exchange process. We then took into assessment the memory usage by measuring the RAM usage in diverse parameter configurations in Section 6.3. In **??**, we validate the addressed approach using real-world data to conclude the discussion.

6.1 Datasets

6.2 Convergence

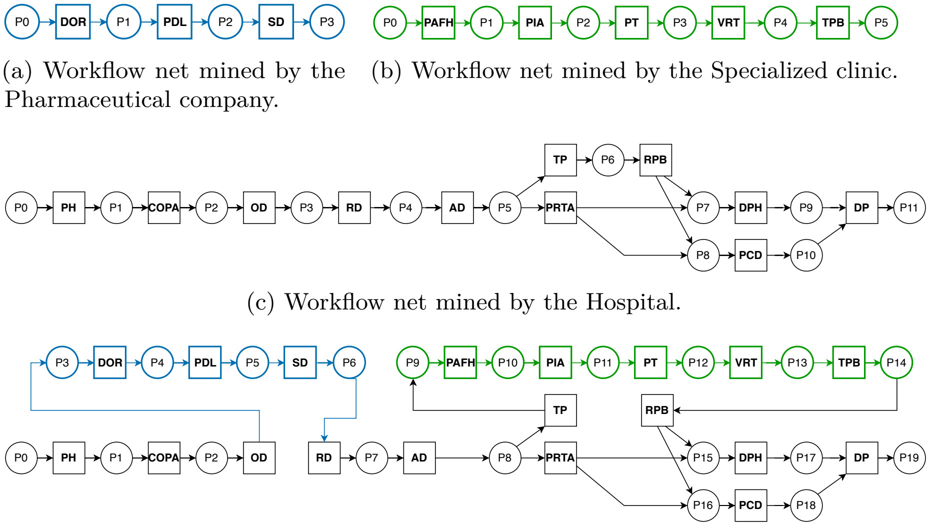
We take into analysis the convergence of the process discovery outputs, as a way of validating the correct operation of the event log exchange mechanism, . Specifically, we generated the workflow nets computed in an intra-organizational setting, in which each organization directly mines its own event log. Subsequently, we employed our approach with a miner actor that computes the same discovery algorithm using an inter-organizational event log obtained as a result of the log exchange and the merging mechanisms.

[6] https://www.edgeless.systems/products/ego/

[7] https://go.dev

[8] https://www.pnml.org

[9] https://woped.dhbw-karlsruhe.de



(a) Workflow net mined by the Pharmaceutical company.

(b) Workflow net mined by the Specialized clinic.

(c) Workflow net mined by the Hospital.

(d) Workflow net mined by the Hospital in inter-organizational setting.

Fig. 6: Outputs used for the convergence test.

To run the test, we used the synthetic event logs devised from our motivating scenario whose BPMN is depicted in Fig. 1. The size of the `Hospital`, `Specialized clinic`, and `Pharmaceutical company` event logs are 4.8 MB, 1.1 MB, and 1.6 MB respectively. Each log contains the standard value of 1000 traces, in accordance with the Sepsis Cases [6] event log.

Upon visual examination of Fig. 6, we observe that the workflow net computed through our approach, displayed in Fig. 6(d), encapsulates the structure and behavior of the workflow nets derived from the intra-organizational discovery procedures depicted in Fig. 6(a),Fig. 6(b), and Fig. 6(c). In detail, Fig. 6(a), colored in blue, depicts the process of the `pharmaceutical company` from the moment the drug order is received to its fulfillment. Figure 6(b), colored in green, depicts the process of the `Specialized clinic` from the moment the patient arrives from the `Hospital` to his transfer.

6.3 Memory Usage

grafici memory usage: segment size,

7 Conclusion and Future Work

It can be reduced

In our implementation, we have focused on process discovery tasks. However, our approach has the potential to seamlessly cover a wider array of process mining functionalities such as *conformance checking*, and *performance analysis* techniques. Implementing them and show their integrability with our approach paves the path for future work.

To be rephrased and repositioned wherever it fits best.

Confidentiality is paramount in inter-organizational process mining, as sensitive data traverses organizational boundaries. Preserving the privacy and confidentiality of operational data becomes a critical concern in this regard. Our research explores a secrecy-preserving approach in which trusted applications allow organizations to apply process mining techniques using event logs from various organizations while ensuring the preservation of partners' privacy. Our solution still has room for improvement. Currently, we assume that providers act fairly, and we do not expect to have injected or maliciously manipulated event logs. In addition, we do not handle TEE crashes and suppose that miners and providers exchange messages in perfect communication channels where no loss, no snap, and no bit corruption take place. We also make assumptions on event log data. We assume the existence of a universal clock for event timestamps across various systems, eliminating the need for synchronization procedures. Additionally, we presume that traces from different organizations relating to the same process instance share a common case identifier. However, this assumption is unrealistic in real-world scenarios, where organizations might employ different case notations. To address this challenge, we should explore alternative event log representations. Future work includes the elaboration of an interaction protocol that formalizes the communication workflow between data providers and miners. Additionally, we plan to integrate usage control policies containing terms and conditions on event log utilization. To achieve this goal, we will design dedicated mechanisms inside trusted applications for monitoring usage rules and enforcing their fulfillment. The presented solution embraces model process mining techniques in a general way. However, we believe that the presented approach is particularly compatible with declarative model representations. Therefore, trusted applications could compute and store the entire set of rules representing a business process, and users may interact with them via trusted queries. We plan to extend the discussion in Section 6 by integrating threat modeling analysis and quantitative assessments concerning scalability, throughput, and performances on real-world event logs.

CDC: The bibliography entries are too rich. Look at [4]. Do we really care that the conference was in Toulouse? And look at [5]: the acronym is enough for the conference name. Also, the volume number is useless if we do not have the series (anyway, we could not care less about either of the two). We have already gone through this, so we should shorten the entries as we know.

References

1. Dumas, M., La Rosa, M., Mendling, J., Reijers, H.A.: Fundamentals of Business Process Management, Second Edition. Springer (2018)
2. Elkoumy, G., Fahrenkrog-Petersen, S.A., Dumas, M., Laud, P., Pankova, A., Weidlich, M.: Secure multi-party computation for inter-organizational process mining. In: Enterprise, Business-Process and Information Systems Modeling: 21st International Conference, BPMDS 2020, 25th International Conference, EMMSAD 2020, Held at CAiSE 2020, Grenoble, France, June 8–9, 2020, Proceedings 21. pp. 166–181. Springer (2020)
3. Elkoumy, G., Fahrenkrog-Petersen, S.A., Dumas, M., Laud, P., Pankova, A., Weidlich, M.: Shareprom: A tool for privacy-preserving inter-organizational process mining. BPM (PhD/Demos) **2673**, 72–76 (2020)

4. Engel, R., Krathu, W., Zapletal, M., Pichler, C., van der Aalst, W.M.P., Werthner, H.: Process mining for electronic data interchange. In: EC-Web. pp. 77–88 (2011)
5. Koch, N., Kraus, A.: The expressive power of uml-based web engineering. In: Second International Workshop on Web-oriented Software Technology (IWWOST02). vol. 16, pp. 40–41. Citeseer (2002)
6. Mannhardt, F.: Sepsis cases - event log (2016). https://doi.org/10.4121/UUID:915D2BFB-7E84-49AD-A286-DC35F063A460, https://data.4tu.nl/articles/_/12707639/1
7. Müller, M., Simonet-Boulogne, A., Sengupta, S., Beige, O.: Process mining in trusted execution environments: Towards hardware guarantees for trust-aware inter-organizational process analysis. In: International Conference on Process Mining. pp. 369–381. Springer International Publishing Cham (2021)