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

´CDC: Status corrente secondo me. Abbiamo. . . Un'intro affabile ma che dobbiamo ricalibrare per toglierla dal pantano della TEE e puntare dritto al nostro obiettivo dichiarato e mantenuto, con vista sul potenziale inespresso.
Un motivating scenario che però resta staccato dal resto.
Un related work che però dovrebbe spostarsi sul far capire cosa facciamo di più e di diverso, o di simile ma rivisto.
Una overview eccellente dell'architettura che però non è connessa con l'esempio.
Una overview accurata delle interazioni dei moduli (e quando lo spazio è poco è il caso di fare il merge tra le due cose per risparmiare spazio, come ci dicemmo) – anche qui senza esempio, dunque capire cosa si fa dove è arduo. Non per me o per te, ma perché conosciamo il lavoro. Chi non lo conosce, non ha assolutamente idea di cosa voglia dire fare il merge degli eventi da tracce provienienti da actor diversi rimettendoli in ordine così da preservare integrità temporale, per esempio.
Una discussion/evaluation ben congegnata in cui però manca la parte quantitativa e la qualitativa è ancora da dettagliare.
Una conclusione.

In today's business landscape, organizations are constantly seeking ways to enhance their operational efficiency, increase their 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 are required to execute process mining methodologies. 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 foreign devices. *Trusted Applications* are tamper-proof software objects running in these contexts. In this paper, we employ trusted applications running in TEEs to enable companies to execute process mining techniques and exchange sensitive information by preserving privacy and integrity of the shared data. In terms of contribution, we extend the state of the art by: (i) proposing a TEE-based infrastructure that enables process mining in inter-organizational settings; (ii) designing the core components of trusted applications providing organizations confidential data exchange and utilization.

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

It can be reduced. EDIT: Already reduced. MISSING: what do we do similarly to and what do we do differently from / improve on the cited papers? A comparison is offered only with the work of Müller et al.

Our work revolves around the following areas: 1, 2 and 3. Next, we position our contribution against the existing body of literature.

The work of Müller et al. [11] is the first contribution that considers TEEs in combination with process mining techniques. This research proposes a con-

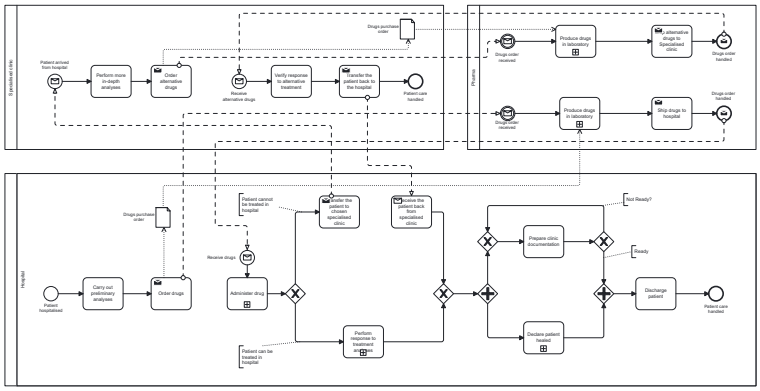

Fig. 1: BPMN Healthcare Scenario

ceptual architecture in which process mining algorithms are executed inside centralized third-party services. Inspired by this preliminary contribution, we design a decentralized approach context where each organization can run process mining algorithms without involving external stakeholders. The theme of inter-organizational process mining is discussed in the literature from different perspectives. Van der Aalst [1] highlights the challenges of inter-organizational process extraction through a categorization. Elkoumy et al. [5] introduce Shareprom, a secure tool for inter-organizational process mining based on secure multi-party computation (MPC). Engel et al. [7] present the EDImine Framework, which applies process mining to inter-organizational processes using the EDI standard. In inter-organizational contexts, merging event logs from different organizations is essential. Hernandez-Resendiz et al. [8] propose a methodology for log merging, while Claes et al. [3] provide techniques for merging data to support process mining algorithms. Data exchange in business collaboration environments has already been explored in various works. Engel et al. [6] extend process mining by analyzing interaction sequences based on EDI documents. Lo et al. [10] present a blockchain-based framework for secure data exchange, even within inter-organizational scenarios. Lastly, there are solutions for secure data sharing with third parties. Xie et al. [12] propose an IoT architecture using TEE and blockchain. Basile et al. [2] introduce ReGov for controlled data utilization in decentralized web contexts.

3 Motivating Scenario

It can be reduced

In the medical field, cooperation between different structures is crucial and many processes are frequently outsourced. In our running example, we consider an healthcare scenario in which a patient care process involve the cooperation of three organizations namely, the `Hospital`, the `Specialized Clinic` and a

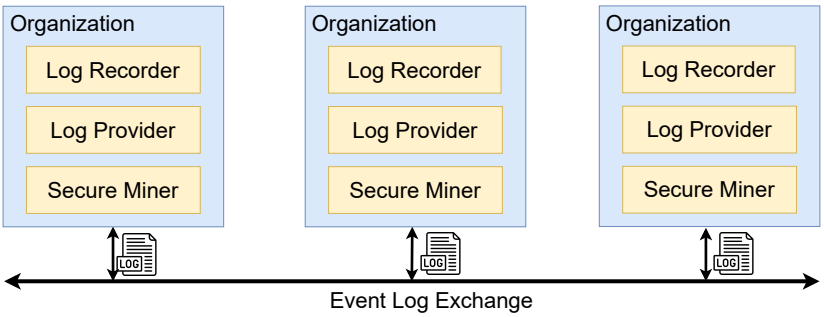

Fig. 2: High-level architectural overview.

`Pharmaceutical Organization`. When a patient enters the hospital, preliminary examinations are carried out and then the hospital company orders the drugs needed to treat the patient from the pharmaceutical company. The pharmaceutical company receives the order and prepares it in the laboratory if the drugs are unavailable, otherwise sends it to the requesting hospital. The hospital will manage the drugs received and check whether the patient can be treated in the hospital or not. If patients require special care, they are transferred to a specialized clinic where more in-depth checks will be carried out. If necessary, the specialized clinic will order drugs from the pharmaceutical company, which will supply them, depending on availability. Once the response to the alternative treatment has been verified, the patient is transferred to the hospital to prepare the dehospitalization. Before discharging the patient, the hospital prepares the necessary clinical documentation. In addition, the hospital also carries out analysis checks and then declares the patient cured to be discharged.

Any organization that is part of the collaboration environment can make the request to perform process mining operations. For instance, the hospital cooperates with the specialized clinic and the pharmaceutical company, it can decide at any time to analyze the entire process by considering data from all three partners to provide an overview. The hospital requests the necessary data from all companies participating in the cooperation. All companies will send their process data in event log form. In order to mine all data together, the hospital must merge the event logs received. Once the event log has been merged, the hospital can proceed with the execution of the mining algorithm to analyze the entire process.

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

CDC: Do not use underscores in any circumstance, also NOT in labels or filenames. Look instead at Fig. 2, for example. Underscores create troubles in most of the cases as LaTeX reserves that character for subscripts in math environments. We have gone through this too, so we should act as we know.

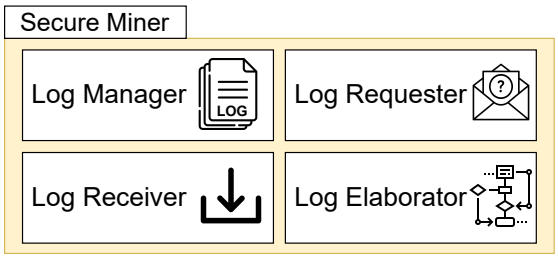

Fig. 3: Subcomponents of the Secure Miner.

4.1 Architecture at large

Our architecture involves networks of nodes controlled by different `Organizations` exchanging their event logs. `Organizations` in the same network collaborate to achieve a common objective and compose business processes whose event logs are scattered across multiple places. Therefore, each `Organization` produces event logs recording the operations executed to complete a business process. The hospital, the specialized clinic, and the pharmaceutical company mentioned in the running example provide an example of partner `Organizations`.

CDC: This is the last time we write something about the motivating scenario to exemplify our architecture until Section 5.3. Clearly, this is not OK. I have rejected papers for less.

An `Organization` may assume one of the following two different roles or both: *provider*, if it delivers local event logs to be collaboratively mined; a *miner* whenever it applies process mining algorithms using local event logs in combination with ones generated by providers. In Fig. 2, we propose a high-level schematization of our solution. Each `Organization` embeds four 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 `Log Recorder`. This component registers the events taking place in the `Organization`. The `Log Recorder` is queried by the local `Log Provider` for event logs to be fed into `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 unauthorizated parties and only permit `Secure Miners` of partner `Organizations` to use the data. The `Secure Miner` shelters external event logs inside an `Organization`'s system by preserving data confidentiality and integrity. We provide an in-depth focus on this component as follows.

4.2 Secure Miner

The primary objective of the `Secure Miner` is to allow `Organizations` to execute process mining algorithms using event logs retrieved from partner `Organizations`, ensuring fair data utilization to log providers. `Secure Miners` leverage isolated execution contexts that guarantee tamper-proofing and data confidentiality. In Fig. 3,

CDC: Figure widths should NEVER be absolute in terms of centimeters or so. They should always be a fraction of `textwidth` or `pageheight`. Otherwise, if we change a template (which is very likely, either because of extended versions of the paper or because sooner or later you might end up writing your PhD thesis), dimensions could all be screwed up. Look at Figs. 2 and 3 instead. We have gone through this too, so we should act as we know.

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 partner `Organizaition`s are stored in the isolated execution context of the `Secure Miner`. We handle these data via the `Log Manager` that makes event log access not practicable from outside the `Secure Miner`'s execution context. Thus, the `Log Manager` prevents external parties from having direct access to event logs. These unauthorized entities include the owner of the miner `Organization` system. The `Log Requester` and the `Log Receiver` are the subcomponents that we employ during the event log exchange. `Log Requester`s initializes the exchange procedure and sends authenticable data requests to the `Data Provision` module of log providers. The `Log Receiver` collects event logs sent by `Log Providers` and entrusts them to the `Log Manager`. When collecting data, `Log Receiver`s prove their trustworthiness to `Log Provider`s by delivering evidence that certifies the `Secure Miner`'s execution context. The `Log Elaborator` is the core module of the `Secure Miner`. It collects the logic to securely execute process mining algorithms. When activated, the `Log Elaborator` accesses external event logs inside the `Secure Miner` and integrates them with the local event log of the `Organization`. We refer to this procedure as *merging*. During the merging, the `Log Elaborator` enriches local traces with events belonging to logs from partner `Organization`s.

5 Realization

CDC: If the section is Realization, why is the label "sec:implementation"? This has the potential to drive the authors nuts when trying to rewrite things. Please fix the issue.

In this section, we outline the technical aspects concerning the realization of our approach. Therefore we first present the enabler technologies through which we 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 follow, we bridge the gap between high-level system architecture and its practical realization. Fig. 4 depicts a *UML deployment diagram* [9] 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` [4]. 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

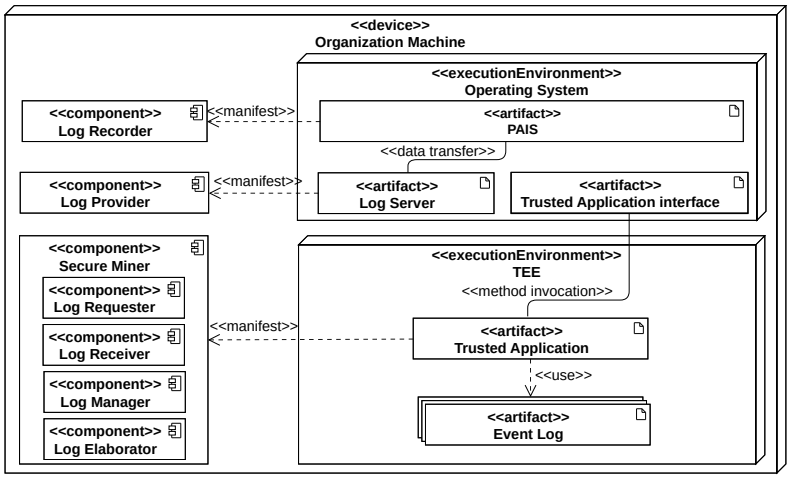

Fig. 4: UML deployment diagram.

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 these technologies to instantiate a `Trusted Application` to fulfill the functionalities of the `Secure Miner` and its subcomponents. The `Trusted Application` collect the logic to generate verifiable data request, receive event external 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 Log`s 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 Log`s 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`.

[1] https://www.w3.org/Protocols/rfc2616/rfc2616.html. Accessed: August 31, 2023.
[2] https://www.w3.org/Protocols/rfc959/. Accessed: August 31, 2023.
[3] https://datatracker.ietf.org/doc/html/rfc1436. Accessed: August 31, 2023.

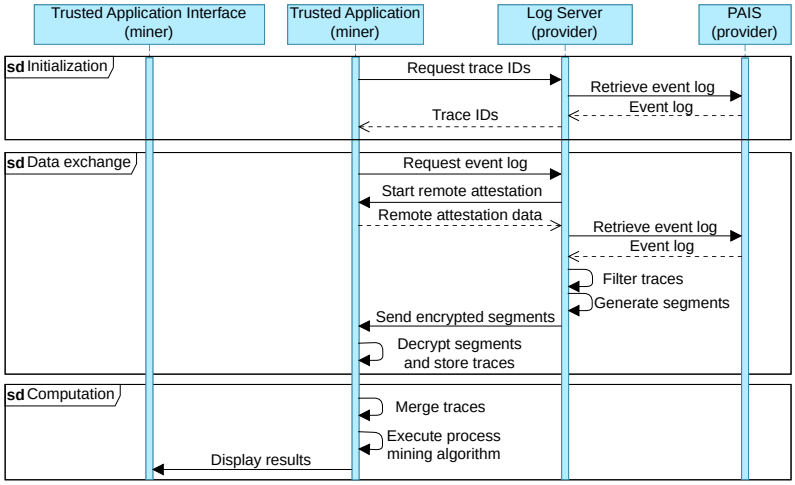

Fig. 5: UML sequence diagram.

5.2 Workflow

As follow, 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).

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.

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 Server`s 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, the `Trusted Application` collects the responses and stores them in the `TEE`.

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 Server`s 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 Log`s 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

In order to generate the logs for the execution of the trusted application, a process model was created based on BPMN notation[4]. Subsequently, the model was imported into the BIMP[5] software, which made it possible to generate the synthetic event logs. 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 [6]. We were able to filter the logs based 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 specialised 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 organisations.

Referring to the trusted execution environment, we used a framework called EGo[7], which makes it possible to develop trusted applications programmed in GO[8]. We developed the Trusted Application (TA) within the TEE with the same language. 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[9](Petri Net Markup

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

[4]https://www.bpmn.org
[5]https://bimp.cs.ut.ee
[6]https://promtools.org
[7]https://www.edgeless.systems/products/ego/
[8]https://go.dev
[9]https://www.pnml.org

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[10] software was used, which takes as input a PNML file and provides the graphic representation of the Petri net.

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

CDC: MISSING: Nella evaluation, dobbiamo riportare 1) la memoria usata e 2) il tempo richiesto dall'algoritmo per fare (a) merge e (b) elaborazione. Se potessimo mostrare un grafico che illustra come la memoria richiesta aumenta pian pianino mentre si raccolgono i pezzi di log e infine quanta ne serva per l'elaborazione (parlo solo della memoria riservata in TEE) sarebbe ancora meglio. L'uso di memoria qui conta perché è un parametro fondamentale.
Dobbiamo informare il lettore sulle caratteristiche del log (numero eventi, numero tracce, dimensione totale in KB una volta salvato in formato XES), come lo abbiamo creato, perché il modello non è uguale a quello di Fig. 1.

In the preceding section, we expounded upon the functioning of the secure event log sharing mechanism through the use of a Trusted Execution Environment (TEE). In order to determine the robustness and effectiveness of the framework, it is important to thoroughly evaluate diverse requirements of its operation. Of particular significance is the assessment of the accuracy and reliability of the event log generated through the amalgamation of logs received from disparate providers. This evaluation is pivotal in establishing the efficacy of the collaborative data exchange process. The primary objective the Section 6 is to delve into a convergence analysis by evaluating the precision and structure of the Petri Net derived from the merged event log with that of the Petri Net originating from the individual event logs. This comparison serves as a critical means to gauge the reliability of the generated process model and the extent to which it accurately represents the collaborative workflow. We then took into analysis other requirements such as integrity and confidentiality in Section 6 and validation on real data in Section 6.

CDC: This is not a Petri net but a Workflow net. Not the same thing.

CDC: We should give the numbers of the subsections, not the section here. Otherwise, it is always Section 6 mentioned all over the place.

6.1 Validation

We conducted a practical validation of our framework using the BPIC 2013 dataset, specifically the Volvo IT incident management system event log. As this log pertains to an intra-company setting, we strategically filtered via ProM the data to isolate department-specific segments. This allowed us to rigorously test the functionality of our framework by securely exchanging partial logs, effectively simulating an inter-organizational context

CDC: Missing: citation

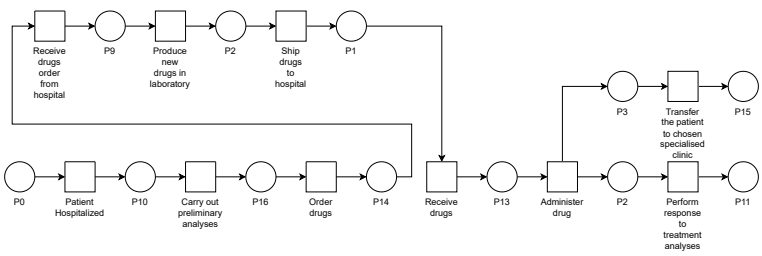

Fig. 6: Workflow Net generated from the merged event log

6.2 Convergence

Matching the two Workflow Nets reveals how their convergence substantiates the efficacy and adequacy of the event log sharing mechanism. The first Workflow Net is crafted from the merged event log, which encapsulates the collective data from the hospital, specialised clinc and the pharmaceutical company. The second Workflow Net originates from the unaltered entire event log. This comparative study aims to elucidate the congruence and comparability of these two Workflow Nets. Upon visual examination, it is possible to notice that the Workflow Net originating from the merged event log closely mirrors the structure and behavior of the Workflow Net derived from the individual event logs. The markings, transitions, and places within both Workflow Nets exhibit a remarkable congruence, indicative of a shared underlying process model. This congruence further extends to the temporal sequencing of events and the causal relationships among process steps. This substantial resemblance between the two Workflow Nets serves as a testament to the seamless convergence of disparate event logs, achieved through the secure exchange mechanism. The fact that the collaborative process model, represented by the Workflow Net generated from the merged event log, aligns closely with the independent models of individual organizations underscores the fidelity and accuracy of the data exchange process.

6.3 Integrity and Confidentiality

An essential aspect of the evaluation pertains to the fundamental principles of integrity and confidentiality upheld by the Trusted Execution Environment (TEE), crucial pillars that underpin the effectiveness and trustworthiness of the event log sharing mechanism. In this framework, the TEE is the cornerstone of data processing, demonstrating praiseworthy abilities to ensure data integrity. Throughout the entire process, from the moment data is ingested from individual organizations to the merging of event logs, the TEE maintains an unyielding grip on data integrity, steadfastly safeguarding against unauthorized modifications. This veracity is established through cryptographic hashing and secure storage

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

mechanisms that ensure that the merged event log remains unaltered and representative of the original data. Concurrent with data integrity, the TEE exercises a robust commitment to confidentiality. The cryptographic measures implemented within the TEE ensure that all data, whether in transit or at rest, is encrypted with a level of security that mitigates the risk of unauthorized access. This cryptographic fortification guarantees that sensitive information encapsulated within event logs remains comprehensively shielded, rendering them inaccessible to any unauthorized entities. As a result, participating organizations can confidently share their event logs, knowing that their proprietary and sensitive information remains impervious to prying eyes.

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 [6]. Do we really care that the conference was in Toulouse? And look at [9]: 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. van der Aalst, W.M.P.: Intra-and inter-organizational process mining: Discovering processes within and between organizations. In: The Practice of Enterprise Modeling: 4th IFIP WG 8.1 Working Conference, PoEM 2011 Oslo, Norway, November 2-3, 2011 Proceedings 4. pp. 1–11. Springer (2011)
2. Basile, D., Di Ciccio, C., Goretti, V., Kirrane, S.: Blockchain based resource governance for decentralized web environments. Frontiers in Blockchain **6** (may 2023)
3. Claes, J., Poels, G.: Merging event logs for process mining: A rule based merging method and rule suggestion algorithm. Expert Systems with Applications **41**(16), 7291–7306 (2014)
4. Dumas, M., La Rosa, M., Mendling, J., Reijers, H.A.: Fundamentals of Business Process Management, Second Edition. Springer (2018)
5. 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)
6. Engel, R., Krathu, W., Zapletal, M., Pichler, C., van der Aalst, W.M.P., Werthner, H.: Process mining for electronic data interchange. In: E-Commerce and Web Technologies: 12th International Conference, EC-Web 2011, Toulouse, France, August 30-September 1, 2011. Proceedings 12. pp. 77–88. Springer (2011)
7. Engel, R., Krathu, W., Zapletal, M., Pichler, C., Bose, R.J.C., van der Aalst, W.M.P., Werthner, H., Huemer, C.: Analyzing inter-organizational business processes: process mining and business performance analysis using electronic data interchange messages. Information Systems and e-Business Management **14**, 577–612 (2016)
8. Hernandez-Resendiz, J.D., Tello-Leal, E., Marin-Castro, H.M., Ramirez-Alcocer, U.M., Mata-Torres, J.A.: Merging event logs for inter-organizational process mining. In: New Perspectives on Enterprise Decision-Making Applying Artificial Intelligence Techniques, pp. 3–26. Springer (2021)
9. 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)
10. Lo, N.W., Chen, S.C., Chang, S.C.: A flexible electronic data exchange framework based on consortium blockchain. Journal of Internet Technology **21**(5), 1313–1324 (2020)
11. 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)
12. Xie, H., Zheng, J., He, T., Wei, S., Hu, C.: TEBDS: A trusted execution environment-and-blockchain-supported IoT data sharing system. Future Generation Computer Systems **140**, 321–330 (2023)