Trusted Applications for Inter-organizational Process Mining

Davide Basile[1] [0000−1111−2222−3333], Luca Barbaro[1] [0000−0002−2975−5330], Valerio Goretti[1] [0000−0001−9714−4278], and Claudio Di Ciccio[1] [2222−−3333−4444−5555]

Sapienza University of Rome

Abstract. Through process mining tecniques, organizations enhances their operational efficiency, improve performances, and deepen the understanding of their business processes. While most process mining research focuses on intra-organizational settings, the emerging importance of inter-organizational collaborations for operational excellence cannot be ignored. Inter-organizational business processes involve multiple independent organizations collaborating to achieve mutual interests. However, inter-organizational process mining faces substantial challenges, primarily centered on confidentiality concerns. In this paper, we introduce a novel approach based on the adoption of trusted applications running in Trusted Execution Environments (TEEs). Our research work aims at ensuring privacy preservation and safeguarding the integrity of sensitive information during process mining procedures in inter-organizational contexts. Therefore, we introduce a TEE-based infrastructure supporting the execution of trusted applications through which partner organizations securely share operational information and apply process mining techniques. We show the feasibility of our solution by exposing an healthcare scenario that serve as running example. Our contribution includes a discussion of the proposed research work that addresses strengths and areas for improvement.

1 Introduction

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 tecniques to discover, monitor and improve business processes through the extraction of knowledge from chronological record known as *event logs*. Organizations record in this ledgers events referring to activities and interactions occurring within a business process. The vast majoriy of process mining contributions considers *intra-organizational* settings, in which, business processes are executed inside individual organizations. However, organizations are increasingly recognizing the value of collaboration and synergy in achieving operational excellence. *Inter-organizational* business processes involve several indipendent organizations that actively cooperate to achieve a shared objective. Several ways of collaborative setting are possible: *subcontracting*, *chained execution*, *capacity sharing*, *case transfer*, *loosely coupled* [13].

Despite the advantages in terms of transparency, performance optimization and benchmarking that companies can gain from such practices, inter-organizational process mining raises challanges that make it still hardly applicable. The major issue concerns confidentiality. Companies are reluctant to outsource with 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 enabler to balance the need for insights with the imperative to protect sensitive information in inter-organizational settings. TEEs offer a secure contexts which guarantee code integrity and data confidentiality in foreign devices. *Trusted Applications* are tamper-proof software objects running in these contexts. In this paper, we employee trusted applications running in TEEs to enable companies to execute process mining techniques and exchange sensitive information, by preserving privacy and integrity on the shared data. In terms of countribution, we extend the state of the art by: (i) proposing a TEE-based infrastructure that enable 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 an healthcare scenario. The high level architecture of our solution is presented in Section 4. Following on from this, we instatantiate the addressed design principles in Section 5 focusing on the employeed 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 literature proposes several studies that consider process mining techniques in inter-organizational environments. Van Der Aalst [1] shows that inter-organizational processes can be divided according to different dimensions making identifiable challenges of inter-organizational process extractions. Elkoumy et al. [6] propose a tool that allows independent parts of an organization to perform process mining operations by revealing only the result. This tool is called Shareprom and exploits the features of secure multi-party computation (MPC). Engel et al. [8] present EDImine Framework, which allows to apply process mining operations for inter-organizational processes supported by the EDI standard[1] and evaluate their performance using business information. Elkoumy et al. [5] propose an MPC-based architecture that aims to perform process mining operations without sharing their data or trusting third parties.

[1] https://edicomgroup.com/learning-center/edi/standards

Applying process mining techniques in intra-organizational contexts requires merging the event logs of the organizations participating in the process. The literature offers several study in this area. For instance, Hernandez-Resendiz et al. [9] present a methodology for merging logs at the trace and activity level using rules and methods to discover the process. Claes et al. [3] provide techniques for performing merge operations in inter-organizational environments. This paper indicates rules for merging data in order to perform process mining algorithms.

The state of the art provides some studies that investigate issues and possible solutions regarding data exchange, more specifically in an business collaboration context. EDI standards enable the communication of business documents. Among these standards, the notion of process is not explicitly specified. This inhibits organizations from applying Business Process Management (BPM) methods in business collaboration environments. Engel et al.[7] extended process mining techniques by discovering interaction sequences between business partners based on EDI exchanged documents. Lo et al.[12] have provided and developed a framework for data exchange designed even in intra-organizational situations. This framework is based on blockchain and decentralized public key infrastructure technologies which ensure scalability, reliability data security, and data privacy.

Additionally, there are several papers that propose solutions for the correct sharing and use of data by third parties. Xie et al.[14] propose an architecture for the internet of things based on trusted execution environment and blockchain. The proposed architecture aims to solve data and identity security problems in the process of data sharing. Basile et al. [2] in their study created a framework called ReGov that allows the exchange of sensitive information in a decentralized web context, ensuring usage control-based data access and usage. In order to control the consumer's device ReGov uses trusted execution environment that allows storage and utilization management of retrieved resources. Hussain et al.[10] present a tool for privacy protection and data management among multiple collaborating companies. This tool allows data encryption to be configured according to the privacy obligations dictated by the context of a system's use.

3 Motivating Scenario

In the medical field, cooperation between the various structures is crucial, and many processes are outsourced. When a patient enters the hospital, preliminary examinations are carried out and then the hospital company orders from the pharmaceutical company the drugs needed to treat the patient. The pharmaceutical company receives the order and if the drugs is not available, prepares it in the laboratory, otherwise sends it to the requesting hospital. The hospital will manage the drugs received and check whether the patient can be treated in hospital or not. If the patient requires special care, the patients will be transferred to a specialised clinic where more in-depth checks will be carried out. If necessary, the specialised 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 back to the hospital to

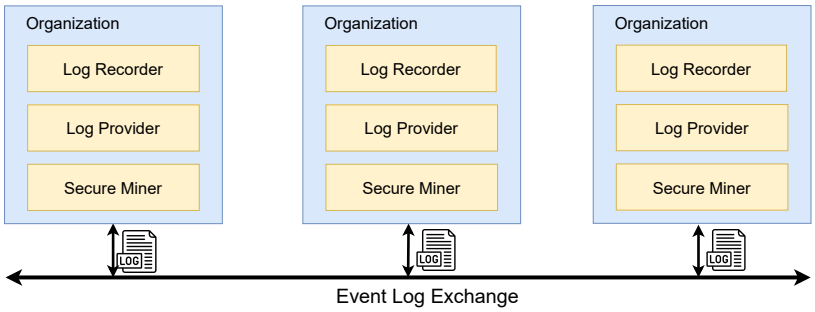

Fig. 1: High-level architectural overview.

prepare for dehospitalisation. Before discharging the patient, the hospital prepares the necessary clinical documentation in order to discharge the patient. it also carries out analysis checks and then declares the patient cured to be discharged.

As the hospital is in cooperation with the specialised clinic and the pharmaceutical company, it is decided one day to analyse the entire process by considering data from all three partners to provide an overview of the process. The hospital makes a request for the necessary data from all companies participating in the co-operation. All companies will proceed with sending their process data in the form of event logs and in order to mine all data together, the hospital must merge the event logs. Once the event log has been merged, the hospital can proceed with the execution of the mining algorithm to analyse the entire process.

4 Design

In this section, we present the high-level architecture underlying our solution. We take into account 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 represent the core of our contribution.

4.1 Architecture at large

Our architecture involves networks of nodes controlled by different `Organization`s exchanging their event logs. `Organization`s 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 `Organization`s. 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

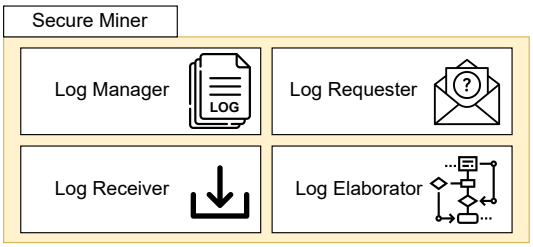

Fig. 2: Modules of the Secure Miner component.

mining algorithms using local event logs in combination with ones generated by providers.

In Fig. 1, 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 Miner`s.

The `Log Provider` component delivers on-demand data to `Secure Miner`s. It controls access to owned event logs by authenticating data requests generated by miners. `Log Provider`s 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 deepth focus on this component as follow.

4.2 Secure Miners

The primary objective of the `Secure Miner` is to allow `Organization`s to execute process mining algorithms using event logs retrieved from partner `Organization`s, ensuring fair data utilization to log providers. `Secure Miner`s leverage isolated execution contexts that guarantee tamperproofing and data confidentiality. In Fig. 2, we show an high level schematization of `Secure Miner`s in which we distinguish four different modules: the `Log Manager`, the `Log Requester`, the `Log Receiver`, and the `Log Elaborator`

Event logs belonging to partner `Organizaitions` 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 fundamental modules that we employee during the event log exchange. `Log Requester`s initalize the

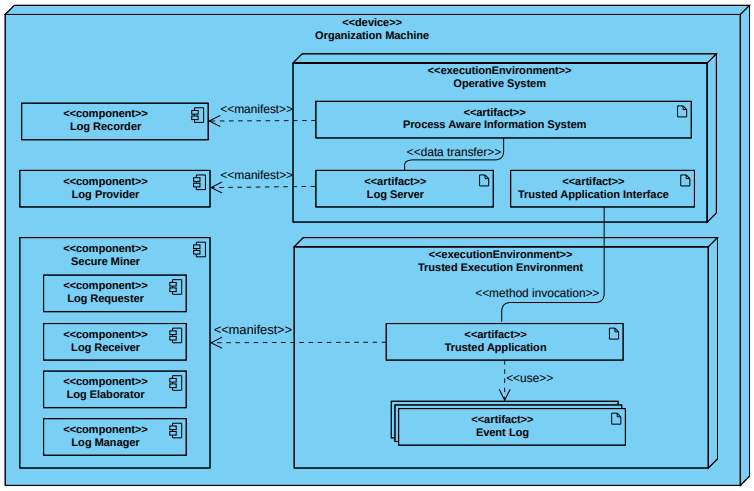

Fig. 3: UML deployment diagram.

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 entrust them to the `Log Manager`. When collecting data, `Log Receivers` prove their trustworthiness to `Log Providers` delivering evidences that certifies the `Secure Miner`'s execution context.

The `Log Elaborator` is the core module of the `Secure Miner`. It collects the logic to safely execute process mining algorithms. It support the integration of *process discovery* [**?**], *conformance checking* [**?**] and *performance analysis* [**?**] tecniques. 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 `Organizations`.

5 Realization

In this section we outline the technical aspects concerning the realization of our framework. 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. 3 depicts a *UML deployment diagram* [11] that aims to help with understanding the instantiated infrastrucuture.

The `Organization Machine` represent 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 `Trusted Execution Environment`.

Software entities that we expose to the users of the `Organization Machine` run inside the `Operative System`. We mainfest the functionalities offered by the `Log Recorder` in the `Process Aware Information System` [4]. These systems help users to handle business processes including accounting and resource management. In our solution, the `Process-Aware Information System` provides the `Log Server` access to event logs. `Log Servers` are web services which processes remote data request and provides event log to miners. We build this entities upon existing web standards such as HTTP, FTP and Goopher.

`Trusted Execution Environment`s are the core technologies of our solution. It creates 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 this techonologies to instantiate a `Trusted Application` to fulfill the functionalities of the `Secure Miner` and its subcomponents. The `Trusted Application` collect the logic generate verifiable data request, receive event external logs, store them in the `Trusted Execution Environment`, and apply process mining algorithms. Procedures executed by the `Trusted Application` are tamperproof. The `Trusted Execution Environment` ensures that the code of the `Trusted Application` executed within it is protected from unauthorized accesses and malicious manipulations. We employ the isolated context of `Trusted Execution Environment` to store `Event Log`s of partner orgnizations inside the miner machine. The `Trusted Execution environment` provide mechanism to protect this sensitive information withoutg 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` offer 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 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 execute process mining algorithms) and one or more providers (i.e., partner organizations that serve their event logs).

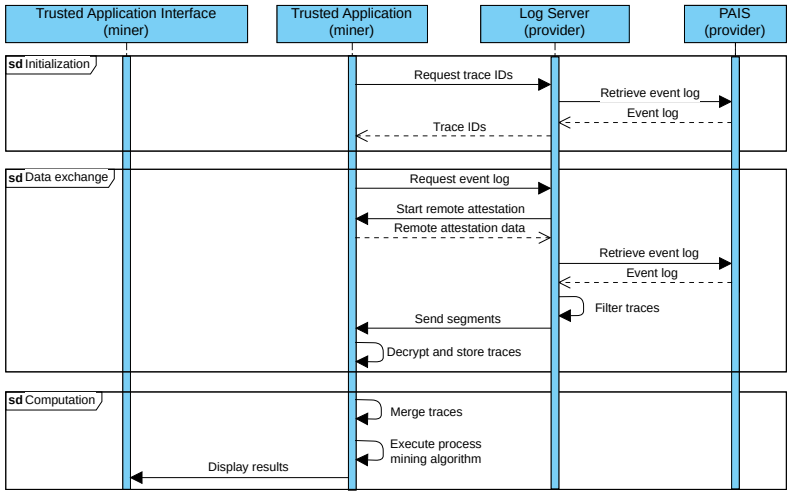

Fig. 4: UML sequence diagram.

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 `Process-Aware Information System` and respond the miner by providing the list of trace IDs in the event log. Hence, the `Trusted Application` collect the responses and store them in the `Trusted Execution Environment`.

Data exchange. Once recorded the preliminary information, the miner starts the data exchange. Therfore, 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 `Trusted Execution Environment`; comes from a partner organization. This operation involve the exchange of additional messages between the `Log Server` and the `Trusted Application`. If the procedure is successful, the identity of the miner 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 splitted in several segments containg 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` decrypt the received segments, extract the traces and store them in `Event Logs` inside the `Trusted Execution Environment`.

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. Assebled traces are used as parameter of process mining algorithms executed by the `Trusted Application` that presents the result of the computation to the users via the `Trusted Application Interface`.

5.3 Implementation

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/

In order to generate the logs for the execution of the trusted application, a process model was created based on BPMN notation[2]. Subsequently, the model was imported into the BIMP[3] 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 [4]. 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[5], which makes it possible to develop trusted applications programmed in GO[6]. 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[7](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[8] 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.

[2]https://www.bpmn.org
[3]https://bimp.cs.ut.ee
[4]https://promtools.org
[5]https://www.edgeless.systems/products/ego/
[6]https://go.dev
[7]https://www.pnml.org
[8]https://woped.dhbw-karlsruhe.de

6 Evaluation

7 Conclusion and Future Work

Our solution has still room for improvement in several areas. We assume that providers act fairly and we do not expect to have injected or maliciously manipulated. In addition, we do not handle TEE crashes and assume that miners and providers exchange messages in perfect communication channels where no loss, no snapp and no bit corruption take place. We make assumption on event log data too. Our solution suppose an universal clock for event timestamp in different systems and no synchronization procedure is actually considered. Moreover traces of different organizations, referring to the same process instance, are characterized by the same case identifier. This scenario is implausible in real world data and organizations may adopt different case notations. To overcome this issue, alternative event log representation should be considered.

Future work include 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 enforce their fulfillment. The presented solution embraces model process mining techniques in a general way. However, we believe that the technologies employed are particularly suitable for declarative model representations where trusted applications compute and store the entire set of rules representing a business process and users interact with them via trusted queries. We plan to extend the discussion in Section 6 by integrating threat modeling analysis and quantitative assesments concerning scalability, throughput and performances on real-world event logs.

References

1. van der Aalst, W.M.: 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., Ciccio, C.D., Goretti, V., Kirrane, S.: Blockchain based resource governance for decentralized web environments. Frontiers in Blockchain **6** (may 2023). https://doi.org/10.3389/fbloc.2023.1141909, https://doi.org/10.3389%2Ffbloc.2023.1141909
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). https://doi.org/10.1007/978-3-662-56509-4
5. 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)
6. 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)
7. Engel, R., Krathu, W., Zapletal, M., Pichler, C., van der Aalst, W.M., 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)
8. Engel, R., Krathu, W., Zapletal, M., Pichler, C., Bose, R.J.C., van der Aalst, W., 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)
9. 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)
10. Hussain, A., Lasrado, L.A., Mukkamala, R.R., Tanveer, U.: Sharing is caring–design and demonstration of a data privacy tool for interorganizational transfer of data. Procedia Computer Science **181**, 394–402 (2021)
11. 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)
12. 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)
13. Van Der Aalst, W.M.: Process-oriented architectures for electronic commerce and interorganizational workflow. Information systems **24**(8), 639–671 (1999)
14. 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). https://doi.org/https://doi.org/10.1016/j.future.2022.10.016, https://www.sciencedirect.com/science/article/pii/S0167739X22003326