















ContinuumConductor: Decentralized Process Mining on the Edge-Cloud Continuum

Hendrik Reiter¹ , Janick Edinger² , Martin Kabierski³ , Agnes Koschmider⁴ ,
, Olaf Landsiedel^{1,5} , Arvid Lepsien¹ , Xixi Lu⁶ , Andrea Marrella⁷ ,
Estefania Serral⁸ , Stefan Schulte⁵ , Florian Tschorsch⁹ , Matthias
Weidlich¹⁰ , and Wilhelm Hasselbring¹ 

¹ Kiel University, Kiel, Germany

{hendrik.reiter,hasselbring}@email.uni-kiel.de

² University of Hamburg, Hamburg, Germany

³ University of Vienna, Vienna, Austria

⁴ University of Bayreuth, Bayreuth, Germany

⁵ Hamburg University of Technology, Hamburg, Germany

⁶ Utrecht University, Utrecht, Netherlands

⁷ Sapienza University of Rome, Rome, Italy

⁸ KU Leuven, Leuven, Belgium

⁹ Dresden University of Technology, Dresden, Germany

¹⁰ Humboldt University of Berlin, Berlin, Germany

Abstract. Process mining traditionally assumes centralized event data collection and analysis. However, modern Industrial Internet of Things (IIoT) systems increasingly operate over distributed, resource-constrained edge-cloud infrastructures. This paper proposes a structured approach for decentralizing process mining by enabling event data to be mined directly within the IoT system’s edge-cloud continuum. We introduce *ContinuumConductor* a layered decision framework that guides when to perform process mining tasks such as preprocessing, correlation, and discovery centrally or decentrally. Thus, enabling privacy-preserving, responsive and resource-efficient process mining. For each step in the process mining pipeline, we analyze the trade-offs of decentralization versus centralization across these layers and propose decision criteria. We demonstrate *ContinuumConductor* at a real-world use-case of process optimization in inland ports. Our contributions lay the foundation for computing-aware process mining in cyber-physical and IIoT systems.

Keywords: Process Mining · Distributed Computing · IoT · Edge-Cloud Continuum.

1 Introduction

The proliferation of sensors and actuators forms the backbone of modern Industrial Internet of Things (IIoT) environments. These systems leverage vast amounts of sensor data to monitor processes, while actuators perform actions often in direct response to the insights derived from this data. This complex interaction

requires the design of robust and efficient computing architectures that can meet three critical objectives: firstly, responsive analysis ensuring that data analysis is performed in real-time to allow timely actions; secondly, privacy-preservation, particularly important in scenarios involving human interaction where sensitive behavioral data, such as that captured by cameras, must be handled with care and third resource-efficiency, since transferring and processing large data volume may exceed the devices capacities.

The edge-cloud continuum [28] offers a promising solution to achieve real-time responsiveness, enhanced privacy protection and resource-efficiency. By distributing computational tasks closer to the data source (edge) while leveraging the scalability of centralized cloud resources, these architectures can mitigate latency, reduce the exposure of sensitive information and minimize transferred data. In this setting, process mining stands out as a powerful technique. Traditionally applied to centralized event logs for retrospective business insights, process mining in the dynamic, data-rich IIoT context requires a full pipeline, from preprocessing raw sensor data to visualizing processes and extracting actionable insights.

This paper delves into the benefits and challenges of transforming the conventional process mining pipeline into a distributed paradigm across the edge-cloud continuum. Specifically, this paper contributes by:

1. Presenting a real-world use case that highlights the practical relevance and requirements of decentralized process mining in an IIoT setting.
2. Discussing the key challenges inherent in implementing decentralized process mining on unstructured IoT sensor data streams.
3. Introducing *ContinuumConductor* a decision framework for the placement of computational steps within the process mining pipeline, determining where computations should be executed within the edge-cloud-continuum.

The remainder of this paper is structured as follows: Section 2 describes the problem by introducing the use-case of the automation in inland ports, describing its additional goals for process mining at the edge-cloud-continuum as well as related work to achieve them. Section 3 demonstrates the process mining pipeline. Moreover, it proposes techniques how the steps of the process mining pipeline can be executed within the edge-cloud-continuum. Section 4 presents *ContinuumConductor* the decision framework for the placement of those steps and applies it on the use case. Section 5 concludes the paper.

2 Problem Description

We illustrate the need for decentralized process mining by first outlining a specific use-case (Section 2.1), from which then derive general requirements (Section 2.2). Then, we review related work in the light of these requirements (Section 2.3).

2.1 Use-case: Decentralized Process Mining in Inland Ports

To illustrate the benefits of decentralized process mining, we consider the *IntegratedDrones* project [29], which aims to modernize data collection and operational

transparency in inland port terminals. Inland multi-purpose terminals often operate in highly dynamic environments, characterized by frequent changes in cargo types, varying throughput, and a limited degree of process standardization. As a result, systematic monitoring and process optimization are challenging.

In this use case, a sensor ecosystem is deployed across the terminal to capture fine-grained operational data. The system combines heterogeneous data sources: *1. Fixed Cameras:* Permanently installed cameras cover predefined areas of the terminal, capturing video streams that document incoming and outgoing goods as well as vehicle movements. Due to their stable network connections, these cameras can stream high-resolution data directly to central edge servers. *2. Vehicle-Mounted Cameras:* Mobile terminal vehicles, such as reach stackers, straddle carriers, and trucks, are equipped with cameras that record their activities and immediate surroundings. These data streams provide context on the movement of cargo and the utilization of terminal equipment but are only intermittently connected to the terminal's IT infrastructure via wireless links. *3. Autonomous Drone Cameras:* A fleet of drones autonomously patrols the terminal area, generating aerial video data to monitor operational zones that are otherwise difficult to cover. Drones can dynamically focus on areas of interest, for example, to track specific handling operations or perform targeted inspections. *4. Sensor Boxes on Vehicles:* Selected vehicles are equipped with sensor boxes that record GPS position, acceleration, vibration, and the current height above ground of the spreader beam. These readings enable a precise reconstruction of vehicle behavior and cargo handling sequences. Data processing in this use case has to cope with *heterogeneous connectivity* and *massive data volumes*. While some sensors (e.g., fixed cameras) provide continuous data streams over wired connections, others (e.g., drones) rely on variable wireless connectivity. To mitigate bandwidth constraints and latency, sensor nodes perform *local preprocessing* on edge computing resources, which includes data filtering, aggregation, anonymization, and transformation into structured intermediate representations. For example, drone video streams are locally analyzed to extract object trajectories and anonymize sensitive information before transmitting results to the central infrastructure.

The sensor data is consolidated in a middleware platform deployed across the terminal's edge-cloud continuum. From this platform, *event logs* are generated that describe the lifecycle of each cargo unit, including timestamps for arrival, intermediate handling steps, storage movements, and final departure. Moreover, the activities and states of vehicles, such as loading, unloading, idle time, and maintenance-related events, are captured. Based on these logs, process mining can improve situational awareness, process compliance, and operational efficiency.

2.2 Goals for Process Mining on the Edge-Cloud-Continuum

From the above application scenario, we derive three additional goals for the process mining analysis:

G1) Privacy preservation: Sensitive data must be anonymized close to the source to comply with privacy regulations and to maintain stakeholder trust.

G2) Real-time responsiveness: Immediate detection of process deviations (e.g., unauthorized access) requires near-sensor computation.

G3) Resource efficiency: Raw sensor data from high-resolution video and telemetry streams exceed available network bandwidth if transmitted unprocessed.

These requirements can be fulfilled by employing the edge-cloud continuum and decentralized process mining techniques. By performing process mining tasks closer to the data sources with cloud-based aggregation and analytics, the approach balances latency, data protection, and process insight in a dynamic, resource-constrained environment. Nevertheless, for each step within the pipeline it has to be discussed where the task are placed within the edge-cloud continuum.

2.3 Related Work

Privacy preservation. Privacy considerations in process mining received considerable attention in recent years, especially for IIoT applications [24]. To address these privacy risks, encryption techniques can facilitate confidentiality [27] and a multitude of data sanitization techniques have been proposed, adopting group-based privacy notions [13] or differential privacy [10]. These techniques are not limited to the control-flow of a process, but may be lifted to contextual information contained in an event log [14]. Most of the existing techniques for protecting privacy in process mining have not been designed for distributed environments that continuously produce event data. However, some notable proposals include the use of multiparty computation [10] for simple process mining tasks and control-flow abstractions of distributed event logs [26].

Real-time responsiveness. Real-time considerations in process mining are addressed in the field of streaming process mining [7]. Streaming process mining algorithms perform on continuously generated, potentially infinite event streams instead of complete and static event logs. These algorithms require bound runtime and memory usages. Utilizing techniques such as filtering, sampling or windowing process mining techniques process discovery, conformance checking and process enhancement have been transferred to a streaming domain. Although initial approaches [12,5] exist to perform streaming process mining tasks in a distributed manner, the approach to perform them within the computing infrastructure close to the data source still lacks further research.

introduces a technique for computing partial and overlapping process models as Petri nets from partial event logs and defines a way to merge them to a Petri net of the complete event log.

Resource Efficiency and Distributed Mining. In process discovery, van der Aalst [1] introduced a method for computing and merging partial Petri nets from partial event logs. Techniques like Map-Reduce [12] parallelize discovery algorithms across multiple compute nodes to manage large event logs. Similarly, for conformance checking, the Single-Entry Single-Exit approach breaks down large models and logs into smaller sub-processes for independent analysis [25], distributing the workload. While these methods efficiently handle large event logs, they remain centralized, lacking both scalability and real-time responsiveness.

Recently, EdgeMiner [5] introduced a distributed, resource-efficient algorithm that operates directly on resource-constrained sensor nodes with real-time event data. This demonstrated the feasibility of near real-time process mining, though it's currently limited to classic footprint-matrix-based algorithms like the alpha miner.

3 IIoT Process Mining on the Edge-Cloud Continuum

The use of process mining allows to get insights for efficiently coordinating objects (drones, vehicles) in the port terminal use-case, in order to, for example, shorten the loading and unloading process or to predict the readiness or deviation of objects before breakdown. However, due to the nature of the distributed IoT, e.g., video data stream used in this use case, several challenges must be addressed to apply process mining efficiently. In this section, we use the use-case presented earlier to discuss these challenges as well as the opportunity the edge cloud continuum brings for enabling goals such as privacy-preservation and real-time analysis.

3.1 IIoT Process Mining Pipeline

To transform raw unstructured data streams into valuable process insights, the steps of *preprocessing*, *aggregation*, *correlation*, *discovery* and *insights* have to be performed [20]. Figure 1 shows these steps which we refer to as the process mining pipeline in the following.

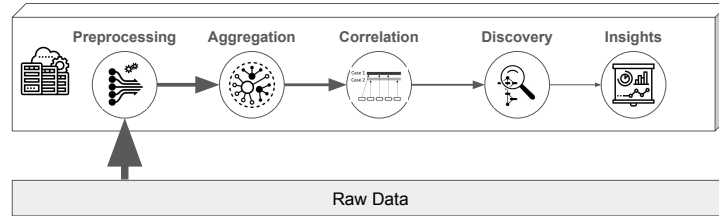


Fig. 1. Process Mining Pipeline. Raw sensor data is preprocessed to low-level events. These are aggregated to high-level events and correlated with case/object ids. Further, process models are discovered and insights are extracted from them.

1. *Preprocessing*: Usually, IoT data is unstructured and heterogeneous (C1), and exists at a lower level of abstraction compared to the event data traditionally used as input for process mining. IoT devices (i.e., cameras, drones and sensor boxes) generate various data formats such as JSON, video, or time-series sensor readings, which cannot directly be applied by process mining techniques. Although numerous methods exist to abstract unstructured data into a structured form such as an event log, these methods are not generalizable and must be adapted to the specific application purpose [31]. Transforming video data

into recognized (low-level) activities computationally complex machine learning, which may exceed compute capabilities (C4), need to be employed [21]. Due to the probabilistic output of such models an uncertainty (C2) is introduced [22]. Moreover, data quality issues [16] such as noise increase this uncertainty further. Distribution may be instrumental in reducing uncertainty. For instance, the use of multiple complementary or redundant sensors may be used to resolve ambiguity and mitigate concerns about data quality by covering previously unrecorded parts of the process and ensuring data availability in the presence of sensor failures [20].

2. *Aggregation*: It is crucial that the aggregation process takes into account the semantics of events in the form of temporal patterns as well as the context (e.g., operating mode of the cameras, shift time of the sensor boxes). Without considering context, the same event can represent different activities (C3). Processing distributed IoT data at the edge in terms of storage, pre-processing, and real-time is also very computationally heavy (C4). Additionally, sensor readings can be inaccurate, incomplete, delayed, or lost, which can lead to incorrect process models or misinterpretations (C5) [6].

3. *Correlation*: The correlation of recognized activities to specific cases or objects, such as within object-centric process mining [2] frameworks, requires a global shared notation (C6), especially in distributed environments. At this stage, the temporal dimension and sequential order of events become paramount (C5). Distinguishing the precedence of events is fundamental for accurate activity correlation. This temporal ordering enables the construction of direct-follow relations within individual cases or objects, a prerequisite for numerous process discovery algorithms. To address the completeness of available data [19], dynamic windowing techniques can be used to balance the trade-off between accuracy and responsiveness [18].

4. *Discovery*: The task of transforming an event log towards a process model is called process discovery. Process discovery algorithms transform underlying process behavior into abstract formal representations such as Petri nets or process trees. The models capture control-flow relations and provide a formal basis for further analysis. In a decentralized setting [3] multiple actors/organizations can discover process fragments. If the algorithms require a complete event log central processing is beneficial while some algorithms such as the Inductive Miner may work better on smaller, localized event logs (C5). In terms of privacy, the challenge is to merge local fragments and clarify interfaces (C6) to other actors and organizations to create a whole view of the complete process.

5. *Insights*: The final step of the process mining pipeline is extracting insights from the process model. This includes techniques such as visualization, conformance checking, performance diagnostics (service times, waiting times), root-cause analysis, simulation (e.g., digital twin), or operations research (e.g., resource planning) [4]. Some techniques require computationally complex machine learning algorithms to be executed on specialized hardware (C4). In a distributed setting each actor may have a partial view of the data due to e.g. access rights. Here, insight extraction must be robust to incomplete data (C5).

Table 1. Challenges and Goals of IIoT process mining

Challenges	Goals
C1) Large volume of unstructured data	G1) Privacy preservation
C2) Uncertainty	G2) Real-time responsiveness
C3) Sensitivity to ambiguous context	G3) Resource efficiency
C4) Network and computing limitations	
C5) Erroneous and incomplete data	
C6) Necessity of shared case/object notion	

3.2 Edge-Cloud Continuum

Edge Computing is a computing paradigm that brings computing and storage closer to data sources of sensors or mobile devices [28]. For this, the computing capabilities of the end devices, IoT gateways, the network infrastructure, or local micro data centers are utilized. Edge computing promises responsive and efficient services due to lower network latencies and network utilization. Additionally, privacy can be improved by not sharing all data immediately at a central place. In contrast, the cloud provides centralized, scalable, and high-performance computing infrastructure capable of handling large-scale data storage. The edge-cloud continuum spans a hierarchical tree of computing resources out of edge, cloud, and fog (in between-edge and cloud) nodes. Each node can either store or compute locally on-device, on a higher computing tier, or on the same computing tier in a peer-to-peer manner. The continuum between edge and cloud enables a more efficient, responsive, and privacy-aware process mining architecture. It supports real-time analytics at the edge while leveraging the cloud for deeper, long-term insights, thereby achieving a balance between latency, resource usage, bandwidth consumption, and data privacy [23].

In terms of process mining, edge nodes can perform initial processing tasks like data filtering, aggregation, or even lightweight mining. The cloud can perform more comprehensive analyses, integrate data from multiple edge sources, and apply advanced algorithms for process discovery and conformance checking when compared to edge devices.

3.3 Goal-Supporting Techniques

Privacy-Preservation. In order to determine whether centralized processing is appropriate or if decentralization is required, privacy threat modeling plays a vital role. Frameworks such as LINDDUN [8] support this process by incorporating attacker models and introducing the notion of trust zones, i.e., areas where data processing is considered safe based on trust assumptions. Beyond these zones, data sharing may require additional safeguards or be avoided altogether.

In addition to threat modeling, privacy design strategies offer guidance for implementing privacy-preserving processing [15]. While applicable to both centralized and decentralized settings, several technical principles, such as separate,

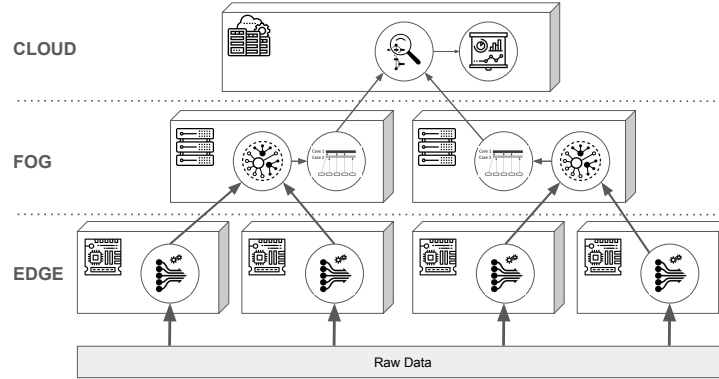


Fig. 2. Process Mining Pipeline placed on the edge-cloud continuum. The preprocessing is performed distributedly on the edge nodes. On fog nodes, the data gets abstracted and correlated to cases. This results are then centralized to perform discovery and insight extraction.

minimize, and aggregate, naturally align with decentralized architectures. Processing data closer to its source, as enabled by edge-cloud infrastructures, facilitates early minimization and aggregation, reducing data exposure and granularity before transmission. Separation can be achieved by limiting data merging across trust zones or device boundaries, helping to mitigate linkage risks. The hide principle further supports broader privacy goals such as confidentiality and anonymity. Privacy-enhancing technologies (PETs) like secure multi-party computation [30] and local differential privacy [11], which often rely on distributed architectures, achieve these goals by processing sensitive data locally and sharing only obfuscated or aggregated outputs. This further underscores the privacy benefits of decentralized processing.

Real-time and Resource Efficient Analysis. Edge-side real-time data preprocessing and complex event processing (CEP) [17] are critical to reduce data volume and enable instant insights. By performing filtering, aggregation, and transformation directly at the edge, the amount of data sent to the cloud is drastically reduced. Next, robust, low-latency data ingestion and stream processing frameworks [12] ensure a swift and reliable event transfer, maintaining real-time fidelity. Incremental process discovery and online conformance checking algorithms are also essential, allowing process models to continuously update and enabling immediate deviation detection. Furthermore, accurate time synchronization across the continuum ensures reliable event ordering, which is fundamental for data integrity. Finally, activity recognition and event abstraction using on-edge AI/ML [9] improves insight quality. This method generates richer event logs from sensor data, providing precise and actionable real-time insights with the resource efficiency of TinyML.

4 Continuum Conductor

In previous sections, we gathered challenges, goals and techniques associated with distributing process mining tasks across an Edge-Cloud Continuum. To facilitate decision-making regarding the optimal placement of pipeline steps for individual use cases, we introduce the *ContinuumConductor*.

Framework: The *ContinuumConductor* is a decision framework designed to determine whether each step in a process mining pipeline should be executed centrally or in a distributed manner. Comprising 16 questions that address the previously identified challenges, the current version of the ContinuumConductor aligns with each of the five steps of the process mining pipeline. This tool does not aim to offer a comprehensive list of all potential challenges. Instead, it is designed to initiate a discourse and provide a foundational framework for exploring further dimensions. The model operates by posing a series of questions, each with four possibilities to perform the compute: centralized (critical), centralized (favorable), decentralized (favorable), decentralized (critical). Table 2 presents the *ContinuumConductor* questions. A unique scenario arises when both a ‘critical centralized’ and a ‘critical decentralized’ evaluation are present within the assessment of a single pipeline step. This represents a conflict, necessitating the application of specialized algorithms or hardware adaptations to resolve the architectural dilemma. To address these conflicts, we present two examples. First, when raw data is privacy-sensitive but on-device processing is too slow, a better solution is to deploy more powerful hardware closer to the device. Second, if a process mining algorithm requires a complete event log, but parts of the model are privacy-critical, it is necessary to design new algorithms. New algorithms require the quantification and optimization of the privacy-utility trade-off based on the specific application’s requirements.

Application to use-case: We demonstrate the *ContinuumConductor* by applying it to the InteGreatDrones project. Thereby, we discuss the strategic placement of computational tasks across the edge-cloud continuum. In the project context, data is processed across four computing layers: 1) directly at or within sensors (e.g., smart cameras), 2) at edge devices such as mini-computers or gateways, 3) within edge clusters formed by GPU-equipped servers, and 4) in a cloud instance primarily responsible for application and visualization services. Data sources include inertial, distance, and position (e.g., GPS) sensors, as well as cameras, the latter generating the largest data volumes and requiring the most intensive preprocessing. Specialized tasks such as license plate detection can be executed directly on smart cameras, whereas video streams from cameras mounted on terminal vehicles must be transferred to edge servers for *preprocessing*, as local processing is not feasible (Pre1, Pre4). Since these video streams may contain personal information, anonymization is performed at the edge (Pre2). Given the large volume of video data and the risk of intermittent connectivity in moving vehicles, intelligent filtering of relevant images is also necessary (Pre3). Hence, the decentralized preprocessing is mandatory. During the *aggregation* step, data such as the position of the container handler and the camera-based identification of containers must be combined. Certain events can be extracted

locally, such as a trailer entering the terminal. However, to detect more complex events fusion of proximate sensors is required (Agg3). Single sensor outages may be tolerated (Agg4), hence a distributed protocol for sensor fusion is possible. The low level-events are not necessarily privacy-critical (Agg1), but still occur in high-volume for high frame-rate video streams (Agg2). In the *correlation* phase, a global case/object ID is ensured, as both trailers and containers possess unique identifiers, which must be recorded at every relevant process step (Cor1). Although timestamps are generally synchronized across devices, the abundance of heterogeneous equipment and potential delays in data synchronization can lead to temporal inconsistencies, highlighting the importance of robust time synchronization mechanisms (Cor2). Handling out-of-order events is critical: Each new entry of a cargo unit into the terminal should be recognized as a distinct case, making the detection and prevention of such events essential (Cor3). In the *discovery* step, no privacy-critical data remains, as all sensitive information is removed beforehand (Dis1). Since process discovery benefits from consistent and complete event logs (Dis3), and there is no clear advantage to local processing in this case (Dis2), central execution is appropriate. *Insight extraction* requires a set of complex reasoning algorithms capable of handling conflicting data and determining the most likely outcomes. To achieve this, a comprehensive view of all data sources is necessary (Ins1, Ins2), making centralized processing preferable.

In conclusion *ContinuumConductor* suggests distributed preprocessing near the sensors. For abstraction and correlation distributed processing remains possi-

Table 2. *ContinuumConductor* questions to decide on the placement of every step in the process mining pipeline within the cloud-edge continuum.

Phase	Question	Challenge
Preprocessing Raw Data → Low-Level Events	Pre1. Are compute resources enough for preprocessing?	C1
	Pre2. Is raw data privacy-critical?	G1
	Pre3. Does raw data transfer need high bandwidth?	C4,G3
	Pre4. Is preprocessing faster on device?	C4,G2
Aggregation Low Level → High Level Events	Agg1. Are low level events still privacy critical?	G1
	Agg2. Are low level events still high-volume?	C1
	Agg3. Can events be build from local context?	C3
	Agg4. Can sensor/network outages be tolerated?	C4,C5
Correlation High Level Events → Event Log	Cor1. Does a global notion of case/object ids exist?	C6
	Cor2. Is the time synchronized between the nodes?	C5
	Cor3. Do out of order events violate real-time objectives?	C5,G2
Discovery Event Log → Process Model	Dis1. Is the process model privacy-critical?	C6,G1
	Dis2. Does the discovery algorithm benefit from locality?	G2,G3
	Dis3. Does the process mining algorithm require consistent and complete event logs?	C5
Insights Process Model → KPIs	Ins1. Does insight extraction need advanced hardware?	C4
	Ins2. Can insight extraction tolerate partial results?	C5,G1

ble for resource efficient computation benefiting from data proximity. For discovery and insight extraction still a central approach is recommended to get an overview on the whole process.

5 Conclusion

This work investigates the challenges and objectives of applying process mining to distributed, high-volume data streams, a field motivated by the InteGreatDrones project. We explore techniques within an edge-cloud continuum and a dedicated process mining pipeline to ensure privacy preservation, real-time insights delivery, and computational efficiency. From this, the *ContinuumConductor* framework has been derived, providing a decision model for centralizing or distributing steps within the process mining pipeline. This framework serves as a foundational discussion point for decentralized process mining within computing infrastructures, necessitating further research into algorithms that balance these identified requirements. Moreover, the question catalog of *ContinuumConductor* needs to be refined by applying it to further scenarios, thereby enhancing its generality and completeness.

Acknowledgments. This work started at Schloss Dagstuhl (Leibniz-Zentrum für Informatik), seminar 25103 “Process Mining on Distributed Event Sources” and received funding from the Deutsche Forschungsgemeinschaft (DFG), grant 496119880 and from the German Federal Ministry for Digital and Transport (BMDV) in the funding program Innovative Hafentechnologien II (IHATEC II).

References

1. van der Aalst, W.M.P.: Decomposing Petri nets for process mining: A generic approach. *Distributed and Parallel Databases* **31**(4) (2013)
2. van der Aalst, W.M.P.: Object-centric process mining: Dealing with divergence and convergence in event data. In: SEFM. LNCS, vol. 11724, pp. 3–25. Springer (2019)
3. van der Aalst, W.M.P.: Federated process mining: Exploiting event data across organizational boundaries. In: SMDS. pp. 1–7. IEEE (2021)
4. van der Aalst, W.M.P.: Process Mining: A 360 Degree Overview (2022)
5. Andersen, J., Rathje, P., Imenkamp, C., Koschmider, A., Landsiedel, O.: Edgeminer: Distributed process mining at the data sources. In: SAC. pp. 705–713. ACM (2025)
6. Bertrand, Y., Schultheis, A., Malburg, L., Grüger, J., Asensio, E.S., Bergmann, R.: Challenges in data quality management for IoT-enhanced event logs. In: RCIS. LNBIP, vol. 547, pp. 20–36. Springer (2025)
7. Burattin, A.: Streaming process mining. In: *Process Mining Handbook*, LNBIP, vol. 448, pp. 349–372. Springer (2022)
8. Deng, M., Wuyts, K., Scandariato, R., Preneel, B., Joosen, W.: A privacy threat analysis framework: supporting the elicitation and fulfillment of privacy requirements. *Requir. Eng.* **16**(1), 3–32 (2011)
9. Ding, A.Y., Peltonen, E., Meuser, T., et al.: Roadmap for edge AI: a dagstuhl perspective. *Comput. Commun. Rev.* **52**(1), 28–33 (2022)

10. Elkoumy, G., Pankova, A., Dumas, M.: Mine me but don't single me out: Differentially private event logs for process mining. In: ICPM. pp. 80–87. IEEE (2021)
11. Erlingsson, Ú., Pihur, V., Korolova, A.: RAPPOR: randomized aggregatable privacy-preserving ordinal response. In: CCS. pp. 1054–1067. ACM (2014)
12. Evermann, J.: Scalable Process Discovery Using Map-Reduce. *IEEE Transactions on Services Computing* **9**(3) (2016)
13. Fahrenkrog-Petersen, S.A., van der Aa, H., Weidlich, M.: Optimal event log sanitization for privacy-preserving process mining. *Data Knowl. Eng.* **145** (2023)
14. Hildebrant, R., Fahrenkrog-Petersen, S.A., Weidlich, M., Ren, S.: PMDG: privacy for multi-perspective process mining through data generalization. In: CAiSE. LNCS, vol. 13901, pp. 506–521. Springer (2023)
15. Hoepman, J.: Privacy design strategies - (extended abstract). In: SEC. IFIP Adv. in Inf. and Comm. Tech., vol. 428, pp. 446–459. Springer (2014)
16. ter Hofstede, A.H.M., Koschmider, A., et al.: Process-data quality: The true frontier of process mining. *ACM J. Data Inf. Qual.* **15**(3), 29:1–29:21 (2023)
17. Imenkamp, C., Akili, S., Weidlich, M., Koschmider, A.: Detect and conquer: Template-based analysis of processes using complex event processing. In: ICPM Workshops. LNBIP, vol. 533, pp. 681–692. Springer (2024)
18. Imenkamp, C., Kabierski, M., Reiter, H., Weidlich, M., Hasselbring, W., Koschmider, A.: Determining window sizes using species estimation for accurate process mining over streams. In: CAiSE 2025. p. 109–124. Springer (2025)
19. Kabierski, M., Richter, M., Weidlich, M.: Quantifying and relating the completeness and diversity of process representations using species estimation. *Inf. Syst.* **130** (2025)
20. Koschmider, A., et al.: Process mining for unstructured data: Challenges and research directions. In: Modellierung 2024. LNI, vol. P-348, pp. 119–136. GI (2024)
21. Lepsien, A., Koschmider, A., Kratsch, W.: Analytics pipeline for process mining on video data. In: BPM Forum. LNBIP, vol. 490, p. 196–213. Springer (2023)
22. Lepsien, A., Pegoraro, M., Fonger, F., et al.: Ranking the Top-K Realizations of Stochastically Known Event Logs. In: ICPM Workshops. LNBIP, Springer (2025)
23. Meuser, T., Lovén, L., Bhuyan, M.H., Patil, S.G., Dustdar, S., et al.: Revisiting edge AI: opportunities and challenges. *IEEE Internet Comput.* **28**(4), 49–59 (2024)
24. Michael, J., Koschmider, A., Mannhardt, F., Baracaldo, N., Rumpe, B.: User-Centered and Privacy-Driven Process Mining System Design for IoT. In: CAiSE Forum. LNBIP, vol. 350, pp. 194–206. Springer (2019)
25. Munoz-Gama, J., Carmona, J., van der Aalst, W.M.P.: Single-entry single-exit decomposed conformance checking. *Inf. Syst.* **46**, 102–122 (2014)
26. Rafiei, M., van der Aalst, W.M.P.: An abstraction-based approach for privacy-aware federated process mining. *IEEE Access* **11**, 33697–33714 (2023)
27. Rafiei, M., von Waldthausen, L., van der Aalst, W.M.P.: Supporting confidentiality in process mining using abstraction and encryption. In: SIMPDA (2019)
28. Satyanarayanan, M.: The emergence of edge computing. *Computer* **50**(1), 30–39 (2017)
29. Teegen, J., Kelm, A., Grasse, O., Hillemann, M., Gülsoylu, E., Frintrop, S.: Drone-based identification of containers and semi-trailers in inland ports. *EasyChair Preprint 14025* (EasyChair, 2024)
30. Yao, A.C.: Protocols for secure computations (extended abstract). In: FOCS. pp. 160–164. IEEE Computer Society (1982)
31. van Zelst, S.J., Mannhardt, F., de Leon, M., Koschmider, A.: Event abstraction in process mining : literature review and taxonomy. *Granular Computing* **6**(3), 719–736 (2021)