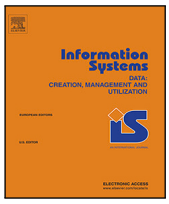
Information Systems 131 (2025) 102525

Contents lists available at ScienceDirect

Information Systems

journal homepage: www.elsevier.com/locate/is



Federated conformance checking

Majid Rafiei, Mahsa Pourbafrani *, Wil M.P. van der Aalst

Chair of Process and Data Science, RWTH Aachen University, Aachen, Germany

ARTICLE INFO

Keywords:
Event data
Federated process mining
Conformance checking
Inter-organizational process mining

ABSTRACT

Conformance checking is a crucial aspect of process mining, where the main objective is to compare the actual execution of a process, as recorded in an event log, with a reference process model, e.g., in the form of a Petri net or a BPMN. Conformance checking enables identifying deviations, anomalies, or non-compliance instances. It offers different perspectives on problems in processes, bottlenecks, or process instances that are not compliant with the model. Performing conformance checking in federated (inter-organizational) settings allows organizations to gain insights into the overall process execution and to identify compliance issues across organizational boundaries, which facilitates process improvement efforts among collaborating entities. In this paper, we propose *a privacy-aware federated conformance-checking approach* that allows for evaluating the correctness of overall cross-organizational process models, identifying miscommunications, and quantifying their costs. For evaluation, we design and simulate a supply chain process with three organizations engaged in purchase-to-pay, order-to-cash, and shipment processes. We generate synthetic event logs for each organization as well as the complete process, and we apply our approach to identify and evaluate the cost of pre-injected miscommunications.

1. Introduction

Process mining encompasses a range of techniques aimed at discovering, analyzing, and improving business processes [1]. By leveraging event logs, process mining provides evidence-based and actionable insights into the actual execution of processes. The field of process mining comprises three fundamental types of analysis: (1) *process discovery*, where the goal is to learn real process models from event logs, (2) *conformance checking*, where the aim is to find commonalities and disconformities between a process model and the corresponding event log, and (3) *process enhancement*, where the aim is to extend or improve process models using different aspects of the available data. In this paper, our main focus is on conformance checking, which plays a pivotal role in process mining. It enables the detection of deviations, anomalies, and instances of non-compliance.

The sub-discipline of process mining focusing on the collaborative discovery, monitoring, analysis, and improvement of cross-organizational processes is referred to by various terms such as *inter-organizational process mining*, *cross-organizational process mining*, and *federated process mining* [2–5]. In inter-organizational collaborations, organizations work together to achieve shared goals, often requiring the exchange of information, goods, or services across organizational boundaries. Conformance checking, in this context, offers organizations insights into the overall execution of processes and facilitates the identification of compliance issues that span organizational boundaries. This, in turn, aids in driving process improvement efforts among collaborating entities, leading to enhanced operational efficiency and alignment. However, due to its distributed nature and the need to share and analyze sensitive data, performing conformance checking in federated or inter-organizational settings presents several challenges. It raises privacy concerns given that conformance checking should be done without revealing the entire internal process of each involved organization in the overall inter-organizational process, while also taking into account the entire process and communication points.

In this paper, we propose a privacy-aware federated conformance-checking approach. Our approach addresses privacy concerns and enables organizations to perform conformance checking while safeguarding sensitive information. Our approach provides a comprehensive analysis of process conformance in federated environments by evaluating the correctness of cross-organizational process models, identifying and quantifying miscommunication costs, and computing the overall fitness of cross-organizational process instances.

To validate the effectiveness of our proposed approach, we simulate a supply chain process involving three organizations. Specifically, we focus on purchase-to-pay, order-to-cash, and shipment processes, which are critical components of the supply chain domain. By designing this synthetic supply chain process, we have full control over its

* Corresponding author.
E-mail addresses: majid.rafiei@pads.rwth-aachen.de (M. Rafiei), mahsa.bafrani@pads.rwth-aachen.de (M. Pourbafrani).

https://doi.org/10.1016/j.is.2025.102525
Received 24 January 2024; Received in revised form 1 September 2024; Accepted 17 January 2025
Available online 25 January 2025

characteristics, allowing us to evaluate our approach in a controlled environment. We generate synthetic event logs, capturing the activities and interactions within the supply chain process. These event logs serve as the basis for our conformance-checking analysis. By applying our proposed approach to these synthetic event logs, we can assess the accuracy of our method for detecting and quantifying pre-injected miscommunications.

The remainder of this paper is structured as follows. In Section 2, we discuss related work. In Section 3, the preliminaries are explained. Section 4 provides details of the proposed approach. In Section 5, we evaluate our approach, and Section 6 concludes the paper.

2. Related work

In this section, we present an overview of the research conducted in the field of federated process mining, which is also known as inter-organizational process mining or cross-organizational process mining. We reference previous studies that have explored the privacy and confidentiality aspects related to this discipline.

2.1. Federated process mining

In [2], inter-organizational process mining is explained, and various categories of inter-organizational data flows are characterized. In [3], the authors utilize EDI messages to illustrate an effective case study of inter-organizational process mining in the automobile industry. In [4], the authors focus on enhancing process performance by leveraging insights gained from cross-organizational process mining. In [6], an approach is proposed to compare collections of process models and event logs recorded in different Dutch municipalities. In [7], an artifact-driven approach for process monitoring is introduced. This approach exploits the Internet of Things (IoT) paradigm to monitor business process by capturing and analyzing the status of physical artifacts. Although such an approach can go beyond the scope of a single organization to monitor business processes, it highly depends on sensor equipped physical artifacts and ignores the message passing phase of collaboration, which is essential to capture inter-organizational miscommunications.

The potential of cloud computing [8] and blockchains [9] has also been acknowledged within the context of inter-organizational process mining. In [10], the authors present an approach for discovering distributed processes in supply chains. In [11], the authors describe fundamental patterns to capture modeling concepts commonly encountered in supply chains. In [12], a process mining approach is introduced to uncover coordination patterns and workflow models from resource allocation logs of different organizations. Then, a process integration approach combines these models and coordination patterns to create a cross-organizational workflow model. In [13], the authors propose an algorithm to discover Intra-department Healthcare Process (IHP) models based on different patterns among medical departments. In [14], a formal method is proposed to systematically model and verify cross-department processes, taking into account various coordination patterns among different departments. Finally, in [5], a framework is introduced that recommends event log abstractions to enable cross-organizational process mining.

2.2. Privacy-aware federated process mining

Recent research in this field has increasingly focused on addressing privacy and confidentiality issues, which are widely recognized as the primary barriers to federated process mining. In [15], a technique based on secure multi-party computation is proposed for discovering directly follows graphs, which focuses on two parties. In [16], the authors propose a framework for sharing public process models and discovering organization-specific process models from multiple parties, relying on a trusted third party. In [17], the authors suggest an approach for discovering process models in inter-organizational settings, utilizing differential privacy and secure multi-party computation. In [18], the authors propose an abstraction-based approach for privacy-aware federated process mining that focuses on process discovery. In [19], the authors present an approach that enables process discovery on multi-actor event data while preserving the confidentiality and integrity of the original records in an inter-organizational business context. This approach prioritizes data confidentiality at the organizational level and imposes computational overhead on mining algorithms.

While previous work primarily focused on process discovery in process mining, our work, to the best of our knowledge, is the first to propose an approach for federated conformance checking, taking into account privacy and confidentiality concerns.

3. Preliminaries

In this section, we provide definitions of event logs, Petri nets, and alignments that are used in the rest of the paper. We start with introducing basic notations and mathematical concepts. Let A be a set. $B(A)$ is the set of all multisets over A. Given A and B as two multisets, $A \uplus B$ is the sum over multisets, e.g., $[a^2, b^3] \uplus [b^2, c^2] = [a^2, b^5, c^2]$, and $A \setminus B$ is the multiset (set) difference, e.g., $[a^2, b^3] \setminus [b^2, c^2] = [a^2, b]$. We define a finite sequence over A of length n as $\sigma = \langle a_1, a_2, \dots, a_n \rangle$ where $\sigma(i) = a_i \in A$ for all $i \in \{1, 2, \dots, n\}$. The set of all finite sequences over A is denoted with A^*.

3.1. Event log

An event log is a collection of events. An event corresponds to an activity execution and can be characterized by various attributes, e.g., an event often has a timestamp attribute referring to the time at which the event happened.

Definition 1 (*Event*). Let $\mathcal{E}$ be the universe of event identifiers and $\mathcal{N}$ be the universe of attribute names. For any $e \in \mathcal{E}$ and attribute name $n \in \mathcal{N}$, $\pi_n(e) \in \mathcal{U}_n$ is the value of attribute n for event e, where $\mathcal{U}_n$ is the universe of attribute values for attribute n. If e does not have an attribute named n, then $\pi_n(e) = \perp$ (*null*). We assume the following attributes to always have a value for any event e:

- $\pi_{cid}(e) \in \mathcal{U}_{cid}$; capturing the *case identifier* of the process instance,
- $\pi_{act}(e) \in \mathcal{U}_{act}$; capturing the *activity* executed, and
- $\pi_{time}(e) \in \mathcal{U}_{time}$; capturing the *timestamp* at which the event happened.

Definition 2 (*Event Log*). Let $\mathcal{E}$ be the universe of events. An event log L is a collection of events, i.e., $L \subseteq \mathcal{E}$.

Table 1 shows a fragment of an event log recorded by a manufacturer's information system by executing a purchase-to-pay process, where case identifiers refer to order numbers.

Definition 3 (*Trace, Simple Event Log*). Let $L \subseteq \mathcal{E}$ be an event log. $cases(L) = \{\pi_{cid}(e) | e \in L\}$ is the set of case identifiers, and $events(L, c) = \{e \in L | \pi_{cid}(e) = c\}$ is the set of events in case c. $trace(L, c) = \langle e_1, \dots, e_n \rangle \in L^*$ is a trace such that $\{e_1, \dots, e_n\} = events(L, c)$, and $\forall_{1 \le i < n}\ e_i \prec e_{i+1}$, where $\prec$ is a total order on events such that $\forall_{e_1, e_2 \in L}\ e_1 \prec e_2 \implies \pi_{time}(e_1) \le \pi_{time}(e_2)$. $variant(L, c) = \langle \pi_{act}(e_1), \pi_{act}(e_2), \dots, \pi_{act}(e_n) \rangle \in \mathcal{U}_{act}^*$ is called the corresponding trace variant of $trace(L, c)$. $\bar{L} = \{trace(L, c) | c \in cases(L)\}$ is the set of traces in L and $\tilde{L} = [variant(L, c) | c \in cases(L)]$ is the multiset of trace variants in L, which is called a *simple event log*.

Table 1
A fragment of an event log recorded by a manufacturer's information system by executing a purchase-to-pay process.

cid	act	Time
c1	create purchase order (*po*)	01.01.2023-08:00:00
c1	send order request (*so*)	01.01.2023-08:30:00
c1	goods receipt (*gr*)	02.01.2023-09:00:00
c1	invoice receipt (*ir*)	02.01.2023-09:30:00
c1	payment (*pa*)	02.01.2023-11:00:00
c2	create purchase order (*po*)	02.01.2023-11:30:00
c2	send order request (*so*)	02.01.2023-11:40:00
c2	invoice receipt (*ir*)	02.01.2023-16:30:00
c2	goods receipt (*gr*)	02.01.2023-17:30:00
c2	payment (*pa*)	03.01.2023-10:30:00
c3	create purchase order (*po*)	03.01.2023-10:40:00
c3	send order request (*so*)	03.01.2023-10:50:00
c3	invoice receipt (*ir*)	03.01.2023-16:30:00
c3	payment (*pa*)	03.01.2023-17:00:00
c3	goods receipt (*gr*)	03.01.2023-17:30:00

Definition 4 (*Applying Functions to Sequences*)**.** Let $f : X \nrightarrow Y$ be a partial function. f can be applied to sequences of X using the following recursive definition: (1) $f(\langle\,\rangle) = \langle\,\rangle$ and (2) for $\sigma \in X^*$ and $x \in X$:

$$f(\langle x \rangle \cdot \sigma) = \begin{cases} f(\sigma) & \text{if } x \notin dom(f) \\ \langle f(x) \rangle \cdot f(\sigma) & \text{if } x \in dom(f) \end{cases}$$

3.2. Petri nets

A Petri net is represented by a directed graph, with nodes representing *places* and *transitions*, and edges representing the flow of *tokens*[1] between places and transitions. Each transition is represented by a square, and each place is represented by a circle. Transitions are used to model activities, and they are connected via places that model possible states of the process. The state of a Petri net is determined by the distribution of tokens over places and is referred to as its *marking*.

Definition 5 (*Petri Net*)**.** A Petri net is a triplet $N = (P, T, F)$, P is a finite set of places, T is a finite set of transitions which it allows $P \cap T = \emptyset$, and $F \subseteq (P \times T) \cup (T \times P)$ is a set of directed arcs. A marked Petri net is a pair (N, M), where $N = (P, T, F)$ is a Petri net and $M \in \mathbb{B}(P)$ is a multiset over places denoting the markings of the net.

Given a Petri net $N = (P, T, F)$, for any $x \in P \cup T$, $\bullet x = \{y \mid (y, x) \in F\}$ denotes the set of input nodes and $x\bullet = \{y \mid (x, y) \in F\}$ denotes the set of output nodes. A transition $t \in T$ is enabled in marking M of net N, denoted as $(N, M)[t\rangle$, if each of its input places $\bullet t$ contains at least one token. It is possible for an enabled transition t to occur, in which case one token will be consumed from each input location in $\bullet t$ and one token will be produced for each output place in $t\bullet$. To represent with formal notations, the marking resulting from firing enabled transitions t in marking M of Petri net N is $M' = (M \setminus \bullet t) \uplus t\bullet$. $(N, M)[t\rangle(N, M')$ denotes that t is enabled in M and firing t results in marking M'. Given $\sigma = \langle t_1, t_2, \ldots, t_n \rangle \in T^*$ as a sequence of transitions, $(N, M)[\sigma\rangle(N, M')$ denotes that there is a set of markings $M_0, M_1, \ldots, M_n$ such that $M_0 = M$, $M_n = M'$, and $(N, M_i)[t_{i+1}\rangle(N, M_{i+1})$ for $0 \leq i < n$. M' is called a reachable marking from M if there exists a $\sigma \in T^*$ such that $(N, M)[\sigma\rangle(N, M')$.

Definition 6 (*Labeled Petri Net*)**.** A labeled Petri net is a tuple $N = (P, T, F, l)$ where (P, T, F) is a Petri net as defined in Definition 5, with labeling function $l : T \nrightarrow \mathcal{U}_{act}$. Given $\sigma_v = \langle a_1, a_2, \ldots, a_n \rangle \in \mathcal{U}_{act}^*$, $(N, M)[\sigma_v\rangle(N, M')$ if and only if there is a sequence of transitions $\sigma = \langle t_1, t_2, \ldots, t_n \rangle \in T^*$ such that $(N, M)[\sigma\rangle(N, M')$ and $\sigma_v = l(\sigma)$.

[1] Tokens are represented as black dots.

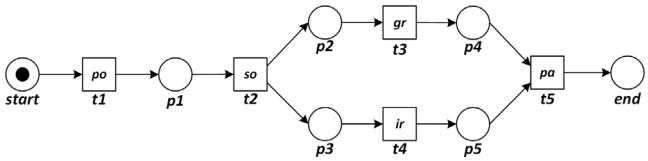


Fig. 1. A system net that models the behavior seen in Table 1.
P={$start, p1, p2, p3, p4, p5, end$}, T={$t1, t2, t3, t4, t5$}, F={$(start, t1), (t1, p1), (p1, t2), (t2, p2), (t2, p3), (p2, t3), (t3, p4), (p3, t4), (t4, p5), (p4, t5), (p5, t5), (t5, end)$}, M_{init}=[$start$], M_{final} = [end].

If $t \notin dom(l)$, it is called an invisible (silent) transition. We write $l(t) = \tau$ if $t \notin dom(l)$. A visible transition $t \in dom(l)$ corresponds to an observable activity $l(t) \in \mathcal{U}_{act}$. In the domain of process mining, we always study processes that begin in one state and conclude in another. As a result, we develop the concept of a system net.

Definition 7 (*System Net*)**.** A system net is a triplet $SN = (N, M_{init}, M_{final})$ where $N = (P, T, F, l)$ is a labeled Petri net, $M_{init} \in \mathbb{B}(P)$ is the initial marking, and $M_{final} \in \mathbb{B}(P)$ is the final marking. $\mathcal{U}_{SN}$ denotes the universe of system nets.

Definition 8 (*System Net Traces*)**.** Let $SN = (N, M_{init}, M_{final})$ be a system net. $\phi_v(SN) = \{\sigma_v \mid (N, M_{init})[\sigma_v\rangle(N, M_{final})\}$ is the set of visible traces starting in M_{init} and ending in M_{final}. $\phi_f(SN) = \{\sigma \mid (N, M_{init})[\sigma\rangle(N, M_{final})\}$ is the corresponding set of complete firing sequences.

Fig. 1 shows a system net that models the behavior seen in Table 1, where $\phi_v(SN) = \{\langle po, so, gr, ir, pa\rangle, \langle po, so, ir, gr, pa\rangle\}$ and $\phi_f(SN) = \{\langle t1, t2, t3, t4, t5\rangle, \langle t1, t2, t4, t3, t5\rangle\}$.

3.3. Conformance checking

Conformance checking techniques investigate how well an event log L and a system net $SN = (N, M_{init}, M_{final})$ fit together. Note that SN may have been discovered through process mining or may have been made by hand. In any case, it is interesting to compare the observed example behavior in L with the potential behavior of SN. L is perfectly fitting SN if and only if $\bar{L} \subseteq \phi_v(SN)$. There are different techniques to quantify fitness such as token-based replay [20] and alignments [21]. In this paper, we focus on alignments as the most common and well-investigated technique. An alignment between a log and a model is defined based on *moves*.

Given a log L and a model $SN = (N, M_{init}, M_{final})$, where $N = (P, T, F, l)$. A *move* refers to a pair $(x, (y, t))$, where the first element $x \in \mathcal{U}_{act}$ refers to an activity in the log and the second element (y, t) refers to a transition $t \in T$ and its corresponding label $y = l(t)$ in the model. For example, $(a, (a, t1))$ means that both log and model make an "a move", a so-called *synchronous move*, where the move in the model is caused by the occurrence of transition $t1$, $(\gg, (a, t1))$ means that an "a move" in the model is not mimicked by a corresponding move in the log, a so-called *model move*, and $(a, \gg)$ means that an "a move" in the log is not followed by the model, a so-called *log move*.

Definition 9 (*Legal Moves*)**.** Let $L \subseteq \mathcal{E}$ be an event log, $A_L = \{\pi_{act}(e) \mid e \in L\}$ be the set of activities in L, and $SN = (N, M_{init}, M_{final})$ be a system net, where $N = (P, T, F, l)$. $LM_{SN,L} = \{(x, (x, t)) \mid x \in A_L \wedge t \in T \wedge l(t) = x\} \cup \{(\gg, (x, t)) \mid t \in T \wedge l(t) = x\} \cup \{(x, \gg) \mid x \in A_L\}$ is the set of legal moves.

Definition 10 (*Alignment*). Let $\sigma_L \in \tilde{L}$ be a log trace variant and $\sigma_M \in \phi_f(SN)$ be a complete firing sequence of system net SN. An alignment of σ_L and σ_M is a sequence of legal moves $\gamma \in LM^*_{SN,L}$, s.t., the projection on the first elements of legal moves (ignoring $\gg$) results in σ_L and the projection on the last element (ignoring $\gg$ and transition labels) results in σ_M.

For instance, given $\sigma_L = \langle po, gr, ir, so, pa \rangle$ as a log trace and the model shown in Fig. 1, a corresponding alignment is as follows:

$\gamma =$

po	$\gg$	gr	ir	so	pa
po	so	gr	ir	$\gg$	pa
$t1$	$t2$	$t3$	$t4$		$t5$

Given a trace variant in a log and a process model, there may be many (if not infinitely many) alignments. To select the most appropriate one(s), so-called *optimal alignments*, one needs to associate costs to undesirable moves, i.e., non-synchronous moves (misalignments), and select an alignment with the lowest total costs.

Definition 11 (*Alignment Cost*). Let $LM_{SN,L}$ be a set of legal moves. A cost function $\delta : LM_{SN,L} \to \mathbb{N}$ assigns costs to legal moves. The cost of an alignment $\gamma \in LM^*_{SN,L}$ is the sum of all costs: $\delta(\gamma) = \sum_{(x,(y,t)) \in \gamma} \delta((x,(y,t)))$.

An alignment $\gamma \in LM^*_{SN,L}$ for a trace variant $\sigma_L \in \tilde{L}$ is optimal, if there is no other alignment for the trace whose cost is lower than $\delta(\gamma)$. Synchronous moves have no costs, i.e., $\delta(x,(y,t)) = 0$. Model moves only have no costs if the transition is invisible, i.e., $\delta(\gg,(\tau,t)) = 0$ if $l(t) = \tau$. $\delta(\gg,(y,t)) > 0$ is the cost of a model move with a visible transition. $\delta(x,\gg) > 0$ is the cost for a log move. These costs may depend on the nature of the activity, e.g., skipping a payment may be more severe than sending too many letters. It is worth noting that misaligned costs may be converted into a fitness value ranging from 0 (bad fitness, i.e., maximal costs) to 1 (perfect fitness, zero costs). We refer to [21] for details.

4. Approach

Fig. 2 depicts the general conceptual framework of federated conformance checking. A collection of organizations collaboratively execute a process model. Each organization executes a part of the collaborative process model. Such an execution emits a data footprint, i.e., a *private event log*. Moreover, each organization has a private reference model, specifying the intended process behavior. A private reference model is either created manually or discovered based on the private event log. Both the private reference process model and the private event log are company-internal and cannot be publicly shared.

Definition 12 (*Private Event Log*). Let $\mathcal{E}$ be the universe of events and $\mathcal{O}$ be a collection of organizations. A private event log $L^{prv}_o \subseteq \mathcal{E}$ belonging to an organization $o \in \mathcal{O}$ is a collection of events where each event $e \in L^{prv}_o$ has *in*, *out*, and *oid* as message passing and organization identifier attributes in addition to the default attributes *cid*, *act*, and *time*.

- $\pi_{in}(e) \in (\mathcal{O} \setminus \{o\}) \cup \{\bot\}$ indicates that the event required input from an organization in order to be executed.
- $\pi_{out}(e) \in (\mathcal{O} \setminus \{o\}) \cup \{\bot\}$ indicates that the execution of the event was required as input from an organization.
- $\pi_{oid}(e) \in \mathcal{O}$ indicates the identifier of the organization owning the log.
- $\{\pi_{in}(e), \pi_{out}(e)\} \cap \{\bot\} = \{\bot\}$.

Note the following constraints in Definition 12 without loss of generality: (1) an organization has no message passing with itself and (2) an event e cannot be executed as a sender, i.e., $\pi_{out}(e) \neq \bot$, and receiver, i.e., $\pi_{in}(e) \neq \bot$, at the same time. Table 2 shows a private version of the event log shown in Table 1, where the manufacturer with identifier $m1$ exchanges messages with a supplier organization with identifier $s1$.

Table 2
A private version of the sample event log shown in Table 1.

cid	act	time	in	out	oid
c1	create purchase order (*po*)	01.01.2023-08:00:00	$\bot$	$\bot$	m1
c1	send order request (*so*)	01.01.2023-08:30:00	$\bot$	s1	m1
c1	goods receipt (*gr*)	02.01.2023-09:00:00	$\bot$	$\bot$	m1
c1	invoice receipt (*ir*)	02.01.2023-09:30:00	s1	$\bot$	m1
c1	payment (*pa*)	02.01.2023-11:00:00	$\bot$	s1	m1
c2	create purchase order (*po*)	02.01.2023-11:30:00	$\bot$	$\bot$	m1
c2	send order request (*so*)	02.01.2023-11:40:00	$\bot$	s1	m1
c2	invoice receipt (*ir*)	02.01.2023-16:30:00	s1	$\bot$	m1
c2	goods receipt (*gr*)	02.01.2023-17:30:00	$\bot$	$\bot$	m1
c2	payment (*pa*)	03.01.2023-10:30:00	$\bot$	s1	m1
c3	create purchase order (*po*)	03.01.2023-10:40:00	$\bot$	$\bot$	m1
c3	send order request (*so*)	03.01.2023-10:50:00	$\bot$	s1	m1
c3	invoice receipt (*ir*)	03.01.2023-16:30:00	s1	$\bot$	m1
c3	payment (*pa*)	03.01.2023-17:00:00	$\bot$	s1	m1
c3	goods receipt (*gr*)	03.01.2023-17:30:00	$\bot$	$\bot$	m1

For a given collection of organizations, $\mathcal{O}=\{o_1, o_2, \ldots o_k\}$, we assume that the corresponding private event logs $L^{prv}_1, \ldots, L^{prv}_k$ do not share any events, i.e., $\forall_{1 \leq i < j \leq k} L^{prv}_i \cap L^{prv}_j = \emptyset$. Organizations share a public version of their event log, only consisting of *interaction events*.

Definition 13 (*Public Event Log*). Given a private event log of an organization o, L^{prv}_o, its corresponding public version is defined as follows: $L^{pub}_o = \{e \in L^{prv}_o \mid \pi_{in}(e) \neq \bot \vee \pi_{out}(e) \neq \bot\}$. Clearly, $L^{pub}_o \subseteq L^{prv}_o$.

Note that private and public event logs can easily be converted to their corresponding simple versions (see Definition 3). A *private process model* is a reference model, e.g., a Petri net, that specifies the intended behavior of an organizational process including its interaction with other organizations, i.e., message-passing information modeling an interface. We use *open nets*, as a specific type of Petri nets, to define a private process model [22,23].

Definition 14 (*Private Process Model - Open Net*). Let I be a set of input places and O be a set of output places. $ON^{prv} = (P, T, F, l, M_{init}, M_{final}, I, O)$ is an open net, modeling a private process model, where:

- $(P \cup I \cup O, T, F, l, M_{init}, M_{final})$ is a system net such that P, I, O are pairwise disjoint,
- for all $p \in I \cup O$, $M_{init}(p) = 0$ and $M_{final}(p) = 0$,
- for all $p \in I$, $\bullet p = \emptyset$, for all $p \in O$, $p\bullet = \emptyset$,
- $T_{int} = \{t \in T \mid (\bullet t \cup t\bullet) \cap (I \cup O) = \emptyset\}$, $T_{com} = T \setminus T_{int}$, and
- there exists $t \in T_{int}$, s.t., $l(t) \neq \tau$.

Note that in Definition 14, T_{int} refers to *internal transitions* that are not involved in any message-passing activities. A set of transitions in a private process model is composed of internal and *communicational transitions* T_{com}, i.e., $T = T_{int} \cup T_{com}$. An open net turns to a *closed net* if $I = O = \emptyset$. Given an open net ON, $inner(ON)$ is its corresponding system net resulting from removing the interface places and their adjacent arcs from ON. Fig. 3 shows a private process model corresponding to the private event log shown in Table 2.

Organizations are by default not willing to share their private process models containing their sensitive internal activities. However, we assume that sharing a public version of their private internal process models is of less sensitivity, and organizations are willing to do this to collaboratively analyze the overall collaborative process model. A public process model only models message-passing activities.

Definition 15 (*Public Process Model*). The public version of a private process model $ON^{prv} = (P, T, F, l, M_{init}, M_{final}, I, O)$ is obtained by making all the internal transitions invisible, i.e., $ON^{pub} = (P, T, F, l, M_{init}, M_{final}, I, O)$ such that for all $t \in T_{int}$, $l(t) = \tau$.

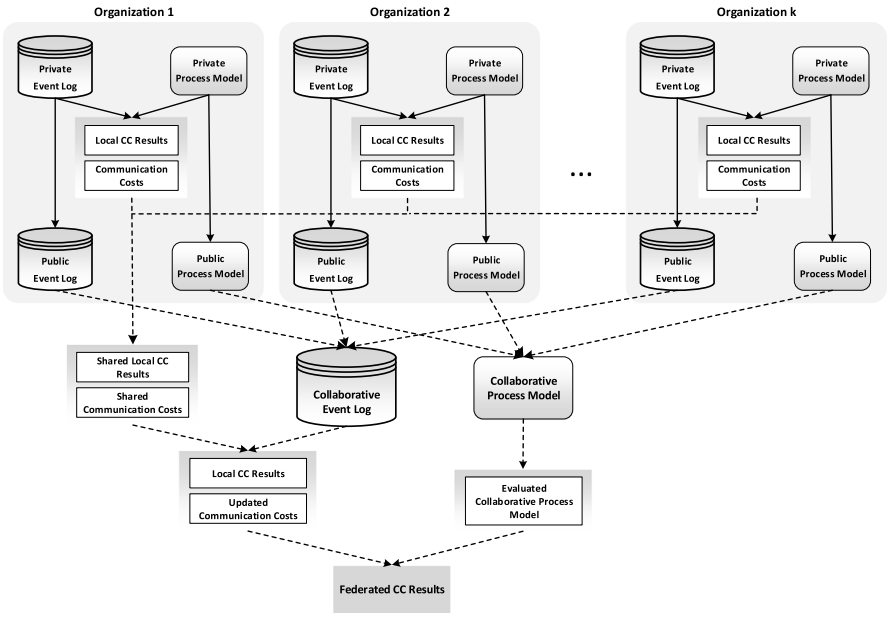


Fig. 2. General conceptual framework of federated conformance checking.

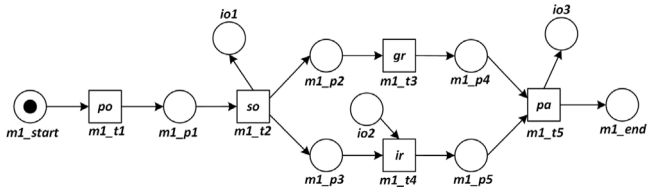


Fig. 3. A private process model corresponding to the private event log shown in Table 2. $I = \{io2\}$, $O = \{io1, io3\}$, $T_{int} = \{m1\_t1, m1\_t3\}$, and $T_{com} = \{m1\_t2, m1\_t4, m1\_t5\}$.

4.1. Local conformance checking

Local conformance checking refers to the alignment of a private event long (i.e., observed behavior) and a private process model (i.e., modeled behavior) in each organization. In order to align observed behavior and modeled behavior, we need as input a simple event log $L \in \mathbb{B}(\mathcal{U}^*_{act})$ and a system net $SN = (N, M_{init}, M_{final})$. A simple event log can easily be obtained from a private event log (see Definition 3). A system net of a private process model ON^{pru} is obtained by $inner(ON^{pru})$. Note that private process models represented by open nets cannot simply be taken as input for conformance checking. That is because in an open net with input places I and output places O, transitions consuming from I are dead, and tokens produced on places in O cannot be removed by the net.

4.2. Local communication costs

To compute federated conformance checking, each organization needs to calculate the local communication cost associated with a communication point of its private process model (i.e., an input or an output place). The local communication cost of a communication point refers to the *potential* cost of misalignment between a trace and a private process model (i.e., an open net) due to a miscommunication between two organizations. We introduce three different types of miscommunications: *sender move*, *receiver move*, and *asynchronous communication*. In the following, we define different types of (mis)communications.

Definition 16 (*Communication Types*)**.** Let o_s and o_r be sender and receiver organizations communicating through a place p_1 used as output place in the process model of o_s and as input place in the process model of o_r. A **sender move** refers to a situation where a message (i.e., a token in a Petri net) produced by o_s in p_1 is never consumed by o_r. A **receiver move** refers to a situation where o_r consumes a message from p_1 that is never produced by o_s. An **asynchronous communication** refers to a situation where o_r consumes a message from p_1 before getting produced by o_s.

Note that miscommunications are not observable in an organization without having access to the communication-related data of organizations involved in communications. Thus, local communication costs are considered potential costs and calculated by assuming that traces involved in communications misalign with the organization's private process model due to miscommunications. To calculate local communication costs (i.e., potential miscommunications), we replace the label of communicational transitions on the private model with a label that cannot get synched with the corresponding activity in the trace, e.g., the silent label τ.

We consider two specific types of labels that can be replaced with communicational transitions: input labels and output labels.

- **Input labels**: These are associated with transitions that connect to an input place in the process model. We replace an input label with the label of a communicational transition that is linked to this input place.
- **Output labels**: These are tied to transitions that connect to an output place. Similarly, an output label is replaced with the label of a communicational transition associated with this output place.

In practice, different misalignment costs can be assigned to input labels and output labels, depending on the severity or significance of the misalignment in the context of the process being analyzed. To represent these labels, we use the labels τ_{in} for input labels and τ_{in} for output labels. We treat these labels as silent transitions. This means that when these labels are used, they do not incur any cost in terms of a corresponding model move. However, it is important to note that when the label of a communicational transition is replaced with specific silent labels (τ_{in} or τ_{out}), it leads to a log move in the event logs. This log move reflects the occurrence of an event in the trace without a corresponding action in the model, effectively indicating a communication-related discrepancy. The minimal cost that can be assigned to this type of miscommunication is based on this log move, representing the least

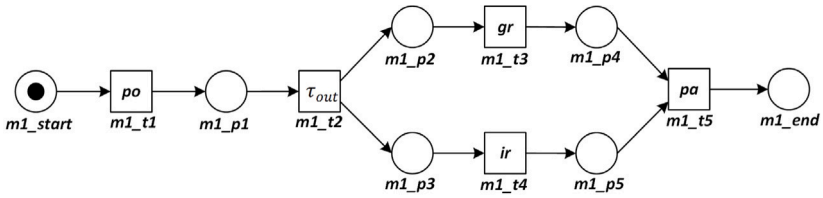


Fig. 4. The system net obtained from Fig. 3 by replacing the label of communication transition $m1\_t2$ with τ_{out} as an output label and applying the *inner*(.) function.

penalty for such a misalignment.

Moreover, as discussed in Section 4.1, alignments cannot be computed over open nets. Thus, after replacing the label of the communicational transition associated with a communication point, for which we want to calculate the local communication cost, we obtain the corresponding system net by applying the inner function *inner*(.) to the open net. Note that the local communication cost of a trace includes the local alignment cost of the trace. Thus, we need to subtract the local conformance cost from the local communication cost to obtain the cost solely associated with potential miscommunications.

As an example, consider a trace $\sigma = \langle po, so, ir, pa, gr \rangle$ and the private process model in Fig. 3. To calculate the local communication cost associated with communication point $io1$, we replace the label of communication transition $m1\_t2$ with τ_{out} as an output label and apply function *inner*(.) to the open net. Fig. 4 shows the corresponding system net. The alignment of trace σ with the resulting system net is as follows:

$$\gamma = \begin{array}{|c|c|c|c|c|c|c|} \hline po & so & \gg & ir & pa & gr & \gg \\ \hline po & \gg & \tau_{out} & ir & \gg & gr & pa \\ m1\_t1 & & m1\_t2 & m1\_t4 & & m1\_t3 & m1\_t5 \\ \hline \end{array}$$

As can be seen in the alignment of the above-mentioned example, the local communication cost associated with communication point $io1$ appears as a log move associated with activity *so* that cannot be mimicked by the model because of the replacement of the corresponding activity label with a silent transition. Considering no cost for the output label τ_{out} the local communication cost is 1. As mentioned earlier, one can consider different costs for output (input) labels. However, the overall cost of this alignment, which is 3 assuming no cost for the output label, includes the local alignment cost as well, which is due to the order misalignment of activities *pa* and *gr*. To obtain the local communication cost associated with communication point $io1$ excluding the local alignment cost, we need to subtract the local alignment cost from the overall alignment cost. Thus, the local communication cost associated with $io1$ is $3 - 2 = 1$.

In the remainder, the local communication cost of a trace for a specific communication point refers to the cost that does not contain the local conformance cost of the trace.

4.3. Communication between public process models

Communication between two public process models is modeled by composing the respective open nets [23]. To this end, shared input and output places are merged and turned into internal places. A channel is represented by a merged internal place, and a token on such a place corresponds to a waiting message in the channel. For the composition of two open nets, we assume that the sets of transitions and internal places of two nets are pairwise disjoint. However, the interfaces, i.e., the sets of input and output places, overlap. Furthermore, each shared place p has just one open net that sends into p and one open net that receives from p. i.e., communication is *bilateral* and *directed* [23]. Open nets that meet these requirements are called *composable*.

Table 3
A fragment of a private event log recorded by a supplier's information system by executing an order-to-cash process.

cid	act	time	in	out	oid
c1	receive order (*ro*)	01.01.2023-08:40:00	m1	⊥	s1
c1	create invoice (*ci*)	01.01.2023-08:50:00	⊥	⊥	s1
c1	order processing (*op*)	01.01.2023-09:00:00	⊥	⊥	s1
c1	order shipment (*os*)	01.01.2023-15:30:00	⊥	⊥	s1
c1	send invoice (*si*)	02.01.2023-09:00:00	⊥	m1	s1
c1	payment collection (*pc*)	02.01.2023-11:30:00	m1	⊥	s1
c2	receive order (*ro*)	02.01.2023-11:50:00	m1	⊥	s1
c2	order processing (*op*)	02.01.2023-12:30:00	⊥	⊥	s1
c2	order shipment (*os*)	02.01.2023-16:30:00	⊥	⊥	s1
c2	create invoice (*ci*)	02.01.2023-17:30:00	⊥	⊥	s1
c2	send invoice (*si*)	02.01.2023-17:40:00	⊥	m1	s1
c2	payment collection (*pc*)	03.01.2023-10:40:00	m1	⊥	s1
c3	receive order (*ro*)	03.01.2023-10:55:00	m1	⊥	s1
c3	create invoice (*ci*)	03.01.2023-11:00:00	⊥	⊥	s1
c3	order processing (*op*)	03.01.2023-11:30:00	⊥	⊥	s1
c3	send invoice (*si*)	03.01.2023-16:20:00	⊥	m1	s1
c3	order shipment (*os*)	03.01.2023-17:00:00	⊥	⊥	s1

4.4. Running example

As a running example, we consider a simplified supply chain process, where a collaborative process is executed by two parties: a *manufacturer*, executing a purchase-to-pay process, and a *supplier* executing an order-to-cash process. We consider the process model shown in Fig. 3 as the private process model of the manufacturer and the process model shown in Fig. 5 as the private process model of the supplier, with a corresponding private event log shown in Table 3.

Definition 17 (*Collaborative Process Model - Open Nets Composition*)**.** Let $ON_1^{pub} = (P_1, T_1, F_1, l_1, M_{init_1}, M_{final_1}, I_1, O_1)$ and $ON_2^{pub} = (P_2, T_2, F_2, l_2, M_{init_2}, M_{final_2}, I_2, O_2)$ be two open nets modeling two public process models. ON_1^{pub} and ON_2^{pub} are composable if $(P_1 \cup T_1 \cup I_1 \cup O_1) \cap (P_2 \cup T_2 \cup I_2 \cup O_2) = (I_1 \cap O_2) \cup (I_2 \cap O_1)$. The composition of ON_1^{pub} and ON_2^{pub} is an open net $ON_1^{pub} \bigoplus ON_2^{pub} = (P, T, F, l, M_{init}, M_{final}, I, O)$ modeling a collaborative process model where:

- $P = P_1 \cup P_2 \cup (I_1 \cap O_2) \cup (I_2 \cap O_1)$,
- $T = T_1 \cup T_2$, $F = F_1 \cup F_2$, $M_{init} = M_{init_1} \uplus M_{init_2}$, $M_{final} = M_{final_1} \uplus M_{final_2}$,
- $l : T \nrightarrow \mathcal{U}_{act}$ with $dom(l) = dom(l_1) \cup dom(l_2)$, $l(t) = l_1(t)$ if $t \in dom(l_1)$ and $l(t) = l_2(t)$ if $t \in dom(l_2)$,
- $I = (I_1 \cup I_2) \setminus (O_1 \cup O_2)$, and $O = (O_1 \cup O_2) \setminus (I_1 \cup I_2)$.

Note that open net composition models *asynchronous* message-passing. Asynchronous message-passing implies non-blocking communication, where a process can proceed with its execution without waiting for a sent message to be received. Additionally, messages can be overtaken by one another, meaning that the order in which messages are sent may not align with the order in which they are received. Fig. 6 shows the collaborative model obtained by composing the public versions of the private process models shown in Fig. 3 and Fig. 5.

We want a collaborative process model to be *valid*. A valid collaborative process model is a closed net, where $I = O = \emptyset$. That is, after

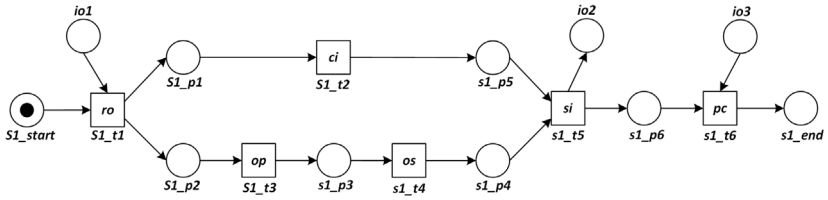


Fig. 5. A private process model corresponding to the private event log shown in Table 3. $I = \{io1, io3\}$, $O = \{io2\}$, $T_{int} = \{s1\_t2, s1\_t3, s1\_t4\}$, and $T_{com} = \{s1\_t1, s1\_t5, s1\_t6\}$.

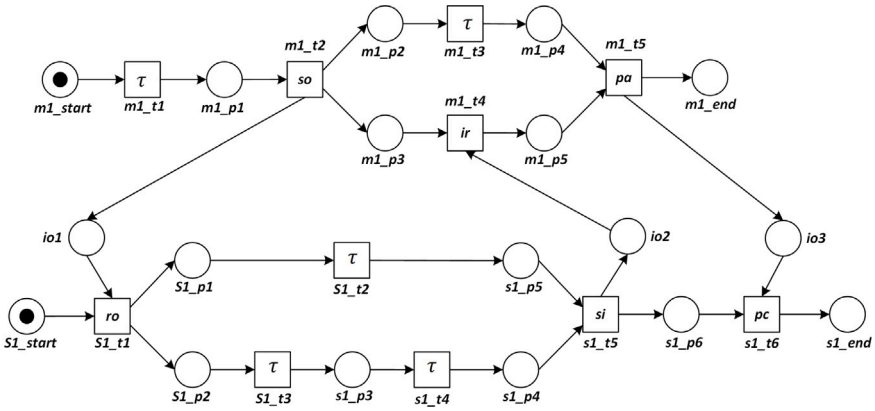


Fig. 6. The collaborative model obtained by composing the public versions of the private process models shown in Figs. 3 and 5.

composing public process models there must not be input or output places remaining unmerged, i.e., not turned into internal places of the collaborative process model. The validity checking of a collaborative process model is the first step of federated conformance checking that can disclose communicational problems. Assuming that the collaborative process model is a valid closed net, we can compute federated conformance checking based on shared local conformance checking results, shared local communication costs, and the collaborative event log which is a collection of public event logs.

Definition 18 (*Collaborative Event Log*). Given $L_1^{pub}, L_2^{pub}, \ldots, L_k^{pub}$ as public logs of k organizations, $L^{col} = L_1^{pub} \cup L_2^{pub} \cup \cdots \cup L_k^{pub}$ is their corresponding collaborative event log. Given L^{col}, the public event log of an organization o is $L_o^{pub} = \{e \in L^{col} \mid \pi_{oid}(e) = o\}$.

Table 4 shows the collaborative log obtained by combining the public versions of the private event logs shown in Tables 2 and 3.

The first step of calculating federated alignment costs is to use the collaborative event log to realize (real) communication costs of local communication costs calculated by each organization. As mentioned in Section 4.2, a local communication cost of a trace refers to the cost associated with a potential miscommunication related to a communication point.

The local communication cost of a trace σ_c belonging to a case c and related to a communication point io becomes a real communication cost if the type of communication associated with io *is not synchronous*. We consider **no cost** for a local communication cost if it is not realized after sharing public event logs. The federated alignment cost of a case involved in communication between two organizations is calculated as Definition 19.

Definition 19 (*Federated Alignment Cost*). Let org_1 and org_2 be two organizations involved in communication via a joint case c. Given $LaC^{org_1}(c)$ as the local alignment cost of c in org_1, $LaC^{org_2}(c)$ as the local alignment cost of c in org_2, $LcC^{org_1}(c)$ as the real communication cost of c in org_1 for all the communication points, and $LcC^{org_2}(c)$ as the real communication cost of c in org_2 for all the communication points, the federated alignment cost of case c is $FaC(c) = LaC^{org_1}(c) + LaC^{org_2}(c) + LcC^{org_1}(c) + LcC^{org_2}(c)$.

Table 4
The collaborative log obtained by combining the public versions of the private event logs, shown in Tables 2 and 3.

cid	act	time	in	out	oid
c1	send order request (*so*)	01.01.2023-08:30:00	⊥	s1	m1
c1	receive order (*ro*)	01.01.2023-08:40:00	m1	⊥	s1
c1	send invoice (*si*)	02.01.2023-09:00:00	⊥	m1	s1
c1	invoice receipt (*ir*)	02.01.2023-09:30:00	s1	⊥	m1
c1	payment (*pa*)	02.01.2023-11:00:00	⊥	s1	m1
c1	payment collection (*pc*)	02.01.2023-11:30:00	m1	⊥	s1
c2	send order request (*so*)	02.01.2023-11:40:00	⊥	s1	m1
c2	receive order (*ro*)	02.01.2023-11:50:00	m1	⊥	s1
c2	send invoice (*si*)	02.01.2023-17:40:00	⊥	m1	s1
c2	invoice receipt (*ir*)	02.01.2023-18:30:00	s1	⊥	m1
c2	payment (*pa*)	03.01.2023-10:30:00	⊥	s1	m1
c2	payment collection (*pc*)	03.01.2023-10:40:00	m1	⊥	s1
c3	send order request (*so*)	03.01.2023-10:50:00	⊥	s1	m1
c3	receive order (*ro*)	03.01.2023-10:55:00	m1	⊥	s1
c3	send invoice (*si*)	03.01.2023-16:20:00	⊥	m1	s1
c3	invoice receipt (*ir*)	03.01.2023-16:30:00	s1	⊥	m1
c3	payment (*pa*)	03.01.2023-17:00:00	⊥	s1	m1

As an example, consider $\sigma_{c3}^{m1} = \langle po, so, ir, pa, gr \rangle$ and $\sigma_{c3}^{s1} = \langle ro, ci, op, si, os \rangle$ as the trace of case $c3$ in organizations $m1$ and $s1$, respectively. Tables 5 and 6 show the local alignment and communication costs with respect to the system nets of the private models of two organizations shown in Figs. 3 and 5, considering no cost for input and output labels.

The alignment of the collaborative trace $\sigma_{c3}^{col} = \langle so, ro, si, ir, pa \rangle$, which can be obtained from the collaborative event log with the collaborative process model is as follows:

$$\gamma = \begin{array}{|c|c|c|c|c|c|c|c|c|c|c|} \hline \gg & so & ro & \gg & \gg & \gg & si & ir & \gg & pa & \gg \\ \hline \tau & so & ro & \tau & \tau & \tau & si & ir & \tau & pa & pc \\ \hline m1\_t1 & m1\_t2 & s1\_t1 & s1\_t2 & s1\_t3 & s1\_t4 & s1\_t5 & m1\_t4 & m1\_t3 & m1\_t5 & s1\_t6 \\ \hline \end{array}$$

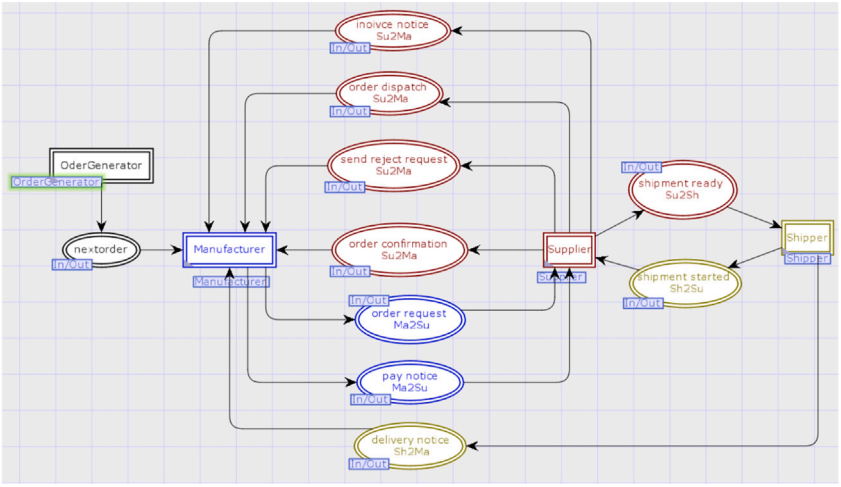


Fig. 7. The high-level designed process and communications among three organizations in the CPN Tools.

Table 5
The local alignment and communication costs for trace $\langle po, so, ir, pa, gr\rangle$ with respect to the system net of the private model of organization m1 shown in Fig. 3, considering no cost for input and output labels.

case id	local align. cost $LaC^{m_1}(c3)$	local com. cost io1 $LcC^{m_1}(io1)$	local com. cost io2 $LcC^{m_1}(io2)$	local com. cost io3 $LcC^{m_1}(io3)$
c3	2	1	1	1

Table 6
The local alignment and communication costs for trace $\langle ro, ci, op, si, os\rangle$ with respect to the system net of the private model of organization s1 shown in Fig. 5, considering no cost for input and output labels.

case id	local align. cost $LaC^{s_1}(c3)$	local com. cost io1 $LcC^{s_1}(io1)$	local com. cost io2 $LcC^{s_1}(io2)$	local com. cost io3 $LcC^{s_1}(io3)$
c3	3	1	1	0

In the alignment of a collaborative trace with the collaborative process model, each visible non-synchronous move (see Definition 10 and different types of moves) is associated with a type of miscommunication. In the above alignment, there exists only one visible non-synchronous move, which is associated with a sender move type of miscommunication related to the communication point $io3$. Thus, the only local miscommunication that is realized is the one associated with the communication point $io3$. The federated alignment cost of case $c3$ is calculated as follows: $LaC^{m1}(c3) = 2$, $LaC^{s1}(c3) = 3$, $LcC^{m1}(c3) = 0+0+1$, $LcC^{s1}(c3) = 0+0+0$, and $FaC(c3) = 3+2+1+0$.

Note that assuming that miscommunication costs are not customized by individual organizations, the federated alignment cost for a case is equal to the local alignment costs of each organization plus the alignment costs of the corresponding collaborative trace with respect to the collaborative process model.

4.5. Privacy considerations

In general, two main levels for privacy concerns can be considered in federated process mining: *individual-level* and *organizational-level* [18]. The former aims to protect private data belonging to individuals in organizations. The latter considers the sensitive internal activities of an organization as private information that should not be revealed [16,18]. As described below, our approach considers both levels of privacy concerns.

Focusing on the control-flow aspect of event logs, the sensitive individual-related data that need to be protected are complete sequences of activities performed for individuals (cases) [24–28]. In our approach, we never share a complete sequence of activities performed for a case. We only share the local alignment results per case and public models that cannot be exploited to infer complete sequences of activities.

The sensitive organizational-level data that need to be protected, focusing on the control-flow aspect of event logs, are internal organization-specific activities that are never shared in our approach. We only share communicational activities that are involved in sending (receiving) messages to (from) other organizations.

5. Experimental results

In order to validate the effectiveness of our proposed approach, we devised a simulated supply chain process involving three organizations: a manufacturer, a supplier, and a shipper. Specifically, we focused on purchase-to-pay (executed by the manufacturer), order-to-cash (executed by the supplier), and shipment (executed by the shipper) processes, which are critical components of the supply chain domain.

We used the CPN Tools [29] and SML functions to simulate this scenario. The source code is available in our GitHub repository[2]. The abstract high-level design and communication among three organizations in the explained supply chain process is shown in Fig. 7. Fig. 8 represents the detailed process inside the manufacturer, where it starts with the creation of a purchase order and is followed by the order confirmation or rejection. This process ends with payment. The designed processes of the supplier and shipper are shown in Fig. 10. By designing this synthetic supply chain process, we have full control over its characteristics, allowing us to evaluate our approach in a controlled environment (see Fig. 9).

Using the designed process models in the CPN Tools, we generate synthetic event logs capturing the activities and interactions within

[2] https://github.com/m4jidRafiei/Federated_Conformance_Checking/tree/main/CPN

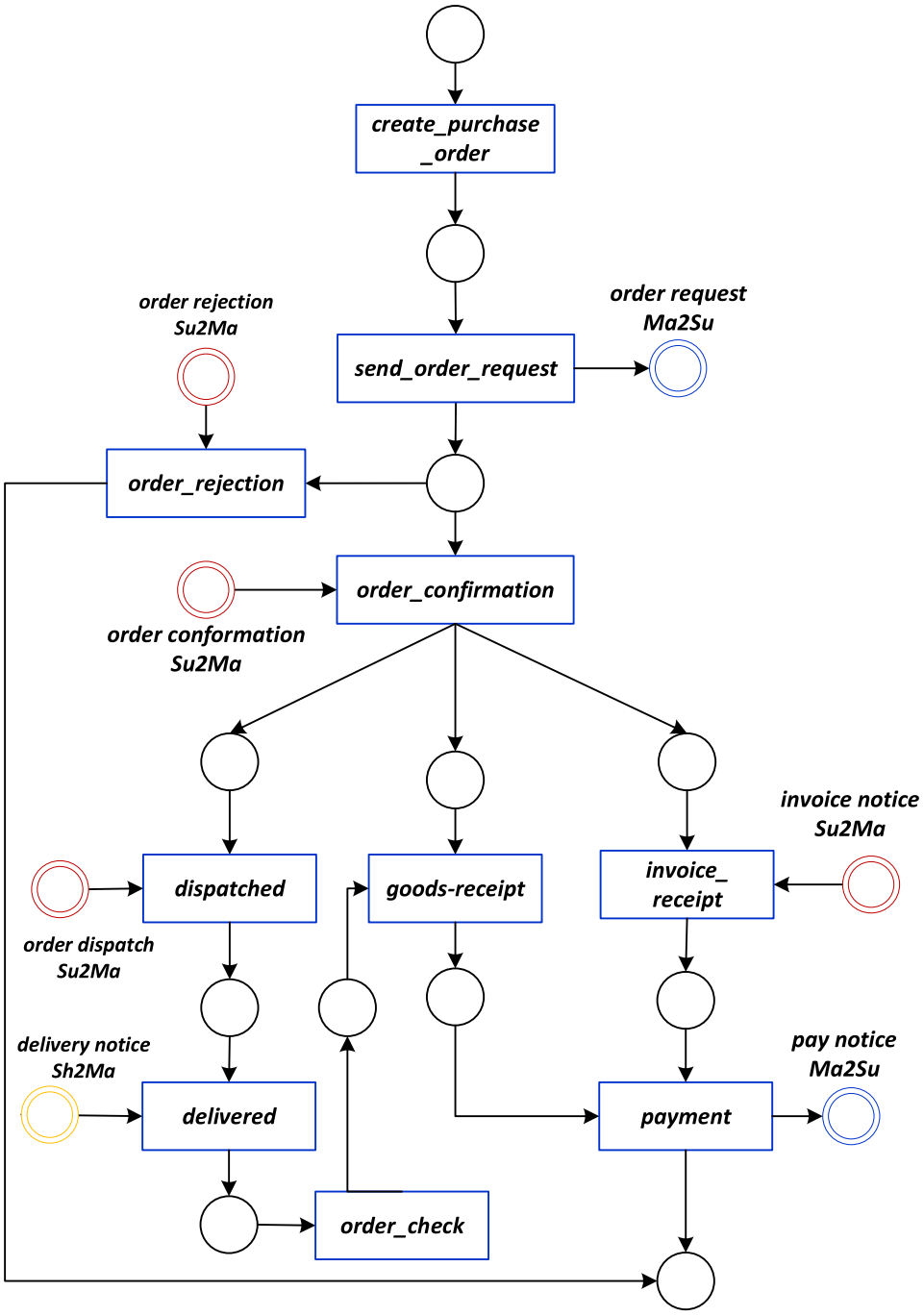


Fig. 8. The designed and implemented process of the manufacturer.

the supply chain process. These event logs served as the basis for our conformance checking analysis. The generated event logs are also available in the GitHub repository.[3] Table 7 shows the general statistics of the event logs.

By deliberately injecting miscommunications into the synthetic event logs, we introduced various scenarios representing communication breakdowns or deviations from the expected process flow. Our approach then enabled us to identify the costs incurred due to these miscommunications. This analysis provides valuable insights into the impact of miscommunications on the overall process performance and allows us to evaluate the effectiveness of our approach in capturing such costs.

Table 7
The general statistics of the generated event logs.

	#cases	#events	#activities	#communication points
Manufacture	297	2517	10	7
Supplier	297	2788	11	8
Shipper	271	1355	5	3
Overall	297	6660	26	9

[3] https://github.com/m4jidRafiei/Federated_Conformance_Checking/tree/main/Logs

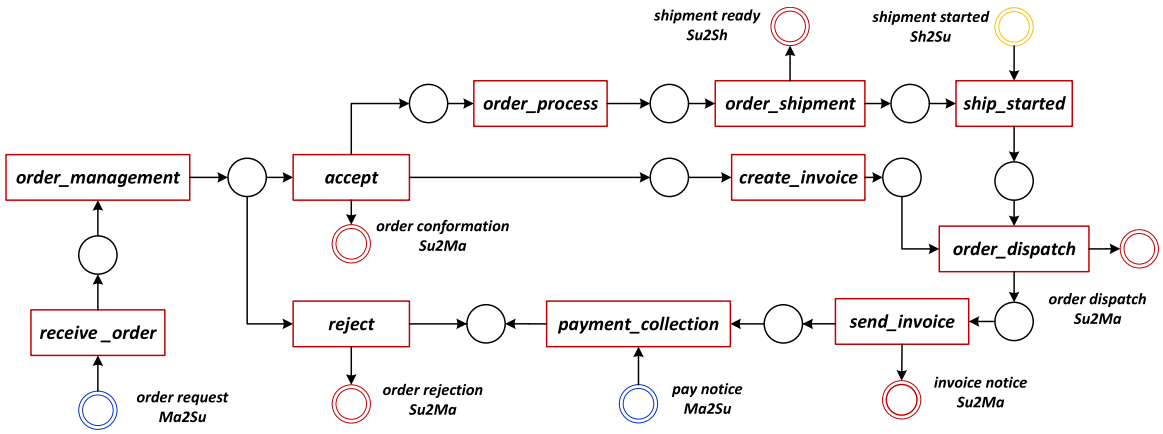


Fig. 9. The designed and implemented processes of the supplier.

Fig. 10. The designed and implemented processes of the shipper.

5.1. Miscommunication scenarios

We inject three miscommunication scenarios corresponding to the three miscommunication types including *sender move*, *receiver move*, and *asynchronous communication*.

- **Sender move**: To simulate this type of miscommunication, we focus on the communication between the manufacturer and the supplier through the communication point *order dispatch*. A sender move happens when the supplier as the sender executes the order dispatch activity but the manufacturer does not execute the buddy activity that is dispatched. Thus, we deliberately removed the dispatched activity from 10 cases in the event log of the manufacturer.
- **Receiver move**: To simulate this type of miscommunication, we focus on the communication between the supplier and the shipper via the communication point *shipment started*. A receiver move happens when the supplier executes the activity ship_started while the shipper does not execute the buddy activity which is preparation. We intentionally removed the preparation activity from the shipper's event log for 19 cases.
- **Asynchronous communication**: This type of miscommunication refers to the situation where the communication happens in the wrong order. We focus on the communication between the shipper and the manufacturer to simulate this type of communication. Particularly, we swap the order of the *delivery* and *delivered* activities that are connected via the communication point delivery notice such that the delivered activity is executed by the manufacturer earlier before the delivery activity is executed by the shipper. To this end, we shifted the execution time of the delivery activity by the shipper for 5 cases.

The altered event logs with the inserted miscommunications for three organizations are available in the GitHub repository.[4]

5.2. Local and federated conformance checking

As shown in Fig. 2, the first step is to compute the local alignments and communication costs by each organization. We use no-cost by default for input and output labels associated with the communicational transitions. Thus, potential communication costs associated with the communication points are equal to 1, which refers to a log move, for all the traces where the communication points get involved.

Fig. 11 shows the local alignment results produced by ProM 6.2 for the manufacturer, the supplier, and the shipper. The manufacturer can see a model move on the dispatched activity in its local view due to the cases for which the dispatched activity was not executed, which refers to the sender move scenario. The supplier sees no misalignment in its local view, even though it involves in two miscommunication scenarios, i.e., the sender and receiver moves. The shipper can see a model move on the preparation activity in its local view due to the cases for which the preparation activity was not executed, which refers to the receiver move scenario.

As can be seen in the local alignment results, a sender (receiver) move can be reflected in the local alignments of the receiver (sender) organization. However, such local costs cannot be realized as miscommunication without the collaborative log and model. Moreover, the asynchronous type of miscommunication cannot be reflected in local alignments. This type of miscommunication can only be realized by aligning the collaborative log with the collaborative model.

To perform alignments on the collaborative model and log, we first need to obtain them as described in Definitions 17 and 18. The collaborative event log and model are available in the GitHub repository.[5] Fig. 12 shows the alignment result produced by ProM 6.2 for the collaborative log and model. One can see the sender and receiver moves on the dispatched and preparation activities, respectively. Moreover, the asynchronous communication scenario is reflected by 5 model moves and 5 log moves on the delivery and delivered activities, respectively.

Note that in this section, we focused on the deliverable insights provided by the proposed framework that can be obtained by the existing tools rather than calculating the federated alignment costs for each case that has already been shown via the running example.

[4] https://github.com/m4jidRafiei/Federated_Conformance_Checking/tree/main/Modified_Logs

[5] https://github.com/m4jidRafiei/Federated_Conformance_Checking/tree/main/Collaborative

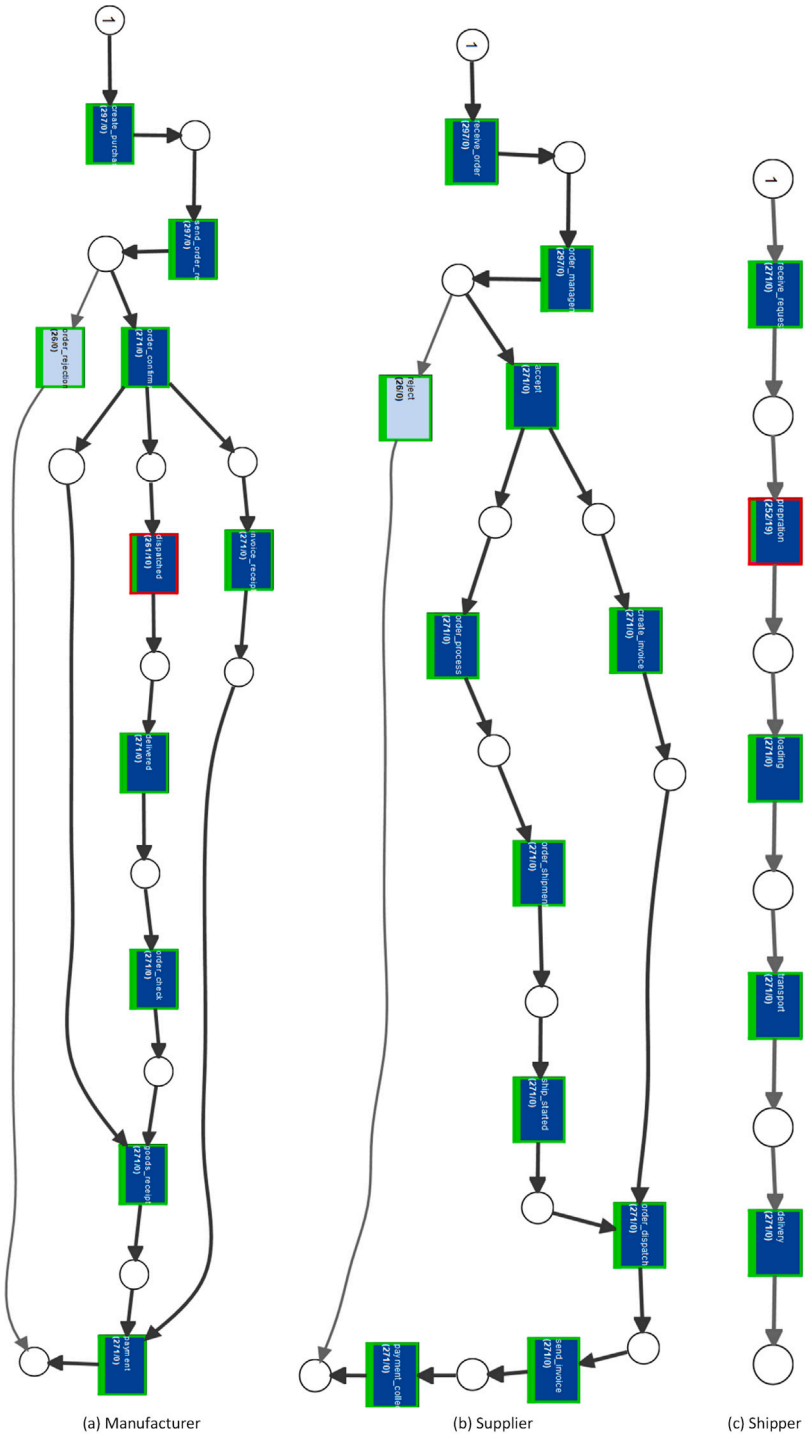


Fig. 11. The local alignment results produced by ProM 6.2 for the manufacturer, the supplier, and the shipper. A red border for a transition indicates a model move.

6. Conclusion

In conclusion, this paper presented a novel contribution to the field of process mining by addressing the critical aspect of conformance checking in the context of federated (inter-organizational) settings. Our proposed privacy-aware federated conformance checking approach offered a comprehensive framework for evaluating correctness and uncovering non-compliance instances across organizational boundaries.

The value of our approach lies in its ability to provide insights into process compliance issues while respecting the privacy concerns of individual organizations. We introduced public and private models and logs and showed how to compute federated alignment costs by only sharing public models and logs to protect sensitive information, allowing collaborating entities to share and analyze cross-organizational process data without compromising confidentiality.

The empirical evaluation conducted in this paper underscored the efficacy of our approach. Through the design and simulation of a supply chain process involving three distinct organizations, we generated synthetic event logs and showed how our proposed approach could be employed to assess and highlight miscommunications. These insights not only shed light on potential inefficiencies and deviations but also guided process improvement initiatives among collaborating

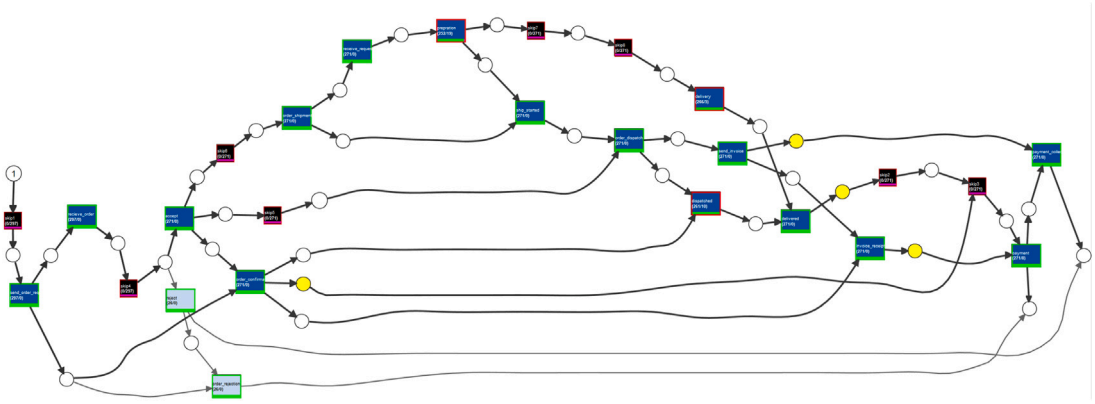


Fig. 12. The alignment result produced by ProM 6.2 for the collaborative log and model. A red border for a transition indicates a model move. A yellow place indicates a log move.

organizations, fostering enhanced coordination and performance in the inter-organizational context.

CRediT authorship contribution statement

Majid Rafiei: Writing – review & editing, Writing – original draft, Visualization, Validation, Resources, Methodology, Investigation, Formal analysis, Conceptualization. **Mahsa Pourbafrani:** Writing – review & editing, Writing – original draft, Visualization, Validation, Data curation, Conceptualization. **Wil M.P. van der Aalst:** Writing – review & editing, Supervision, Resources, Funding acquisition, Formal analysis.

Declaration of competing interest

The authors declare that they have no known competing financial interests or personal relationships that could have appeared to influence the work reported in this paper.

Acknowledgments

Funded under the Excellence Strategy of the Federal Government and the Länder, Germany. We also thank the Alexander von Humboldt (AvH) Stiftung for supporting our research.

Data availability

The link to the shared data is included in the paper and is publicly available.

References

[1] W.M.P. van der Aalst, Process Mining - Data Science in Action, Second Edition, Springer, 2016, http://dx.doi.org/10.1007/978-3-662-49851-4.

[2] W.M.P. van der Aalst, Intra- and inter-organizational process mining: Discovering processes within and between organizations, in: P. Johannesson, J. Krogstie, A.L. Opdahl (Eds.), The Practice of Enterprise Modeling - 4th IFIP WG 8.1 Working Conference, PoEM 2011 Oslo, Norway, November 2–3, 2011 Proceedings, in: Lecture Notes in Business Information Processing, vol. 92, Springer, 2011, pp. 1–11, http://dx.doi.org/10.1007/978-3-642-24849-8_1.

[3] R. Engel, W.M.P. van der Aalst, M. Zapletal, C. Pichler, H. Werthner, Mining inter-organizational business process models from EDI messages: A case study from the automotive sector, in: J. Ralyté, X. Franch, S. Brinkkemper, S. Wrycza (Eds.), Advanced Information Systems Engineering - 24th International Conference, CAiSE 2012, Gdansk, Poland, June 25–29, 2012. Proceedings, in: Lecture Notes in Computer Science, vol. 7328, Springer, 2012, pp. 222–237, http://dx.doi.org/10.1007/978-3-642-31095-9_15.

[4] O. Yilmaz, P. Karagoz, Generating performance improvement suggestions by using cross-organizational process mining, in: P. Ceravolo, S. Rinderle-Ma (Eds.), Proceedings of the 5th International Symposium on Data-Driven Process Discovery and Analysis, SIMPDA 2015, Vienna, Austria, December 9–11, 2015, in: CEUR Workshop Proceedings, vol. 1527, CEUR-WS.org, 2015, pp. 3–17, URL http://ceur-ws.org/Vol-1527/paper1.pdf.

[5] W.M.P. van der Aalst, Federated process mining: Exploiting event data across organizational boundaries, in: N.L. Atukorala, C. Chang, E. Damiani, M. Fu, G. Spanoudakis, M. Srivatsa, Z. Wang, J. Zhang (Eds.), IEEE International Conference on Smart Data Services, SMDS 2021, Chicago, IL, USA, September 5–10, 2021, IEEE, 2021, pp. 1–7, http://dx.doi.org/10.1109/SMDS53860.2021.00011.

[6] J.C.A.M. Buijs, B.F. van Dongen, W.M.P. van der Aalst, Towards cross-organizational process mining in collections of process models and their executions, in: F. Daniel, K. Barkaoui, S. Dustdar (Eds.), Business Process Management Workshops - BPM 2011 International Workshops, Clermont-Ferrand, France, August 29, 2011, Revised Selected Papers, Part II, in: Lecture Notes in Business Information Processing, vol. 100, Springer, 2011, pp. 2–13, http://dx.doi.org/10.1007/978-3-642-28115-0_2.

[7] G. Meroni, Artifact-driven process monitoring: A viable solution to continuously and autonomously monitor business processes, in: A. Polyvyanyy, M.T. Wynn, A.V. Looy, M. Reichert (Eds.), Business Process Management - 19th International Conference, BPM 2021, Rome, Italy, September 06–10, 2021, Proceedings, in: Lecture Notes in Computer Science, vol. 12875, Springer, 2021, pp. 37–43, http://dx.doi.org/10.1007/978-3-030-85469-0_5.

[8] W.M.P. van der Aalst, Configurable services in the cloud: Supporting variability while enabling cross-organizational process mining, in: R. Meersman, T.S. Dillon, P. Herrero (Eds.), On the Move To Meaningful Internet Systems: OTM 2010 - Confederated International Conferences: CoopIS, IS, DOA and ODBASE, Hersonissos, Crete, Greece, October 25–29, 2010, Proceedings, Part I, in: Lecture Notes in Computer Science, vol. 6426, Springer, 2010, pp. 8–25, http://dx.doi.org/10.1007/978-3-642-16934-2_5.

[9] S. Tönnissen, F. Teuteberg, Using blockchain technology for cross-organizational process mining - concept and case study, in: W. Abramowicz, R. Corchuelo (Eds.), Business Information Systems - 22nd International Conference, BIS 2019, Seville, Spain, June 26–28, 2019, Proceedings, Part II, in: Lecture Notes in Business Information Processing, vol. 354, Springer, 2019, pp. 121–131, http://dx.doi.org/10.1007/978-3-030-20482-2_11.

[10] L. Maruster, J.C. Wortmann, A.J.M.M. Weijters, W.M.P. van der Aalst, Discovering distributed processes in supply chains, in: H. Jagdev, J.C. Wortmann, H.J. Pels (Eds.), Collaborative Systems for Production Management, IFIP TC5/WG5.7 Eighth International Conference on Advances in Production Management Systems, September 8–13, 2002, Eindhoven, The Netherlands, in: IFIP Conference Proceedings, vol. 257, Kluwer, 2002, pp. 219–230.

[11] R. Liu, A. Kumar, W.M.P. van der Aalst, A formal modeling approach for supply chain event management, Decis. Support Syst. 43 (3) (2007) 761–778, http://dx.doi.org/10.1016/j.dss.2006.12.009.

[12] Q. Zeng, S.X. Sun, H. Duan, C. Liu, H. Wang, Cross-organizational collaborative workflow mining from a multi-source log, Decis. Support Syst. 54 (3) (2013) 1280–1301, http://dx.doi.org/10.1016/J.DSS.2012.12.001.

[13] C. Liu, H. Li, S. Zhang, L. Cheng, Q. Zeng, Cross-department collaborative healthcare process model discovery from event logs, IEEE Trans. Autom. Sci. Eng. 20 (3) (2023) 2115–2125, http://dx.doi.org/10.1109/TASE.2022.3194312.

[14] Q. Zeng, F. Lu, C. Liu, H. Duan, C. Zhou, Modeling and verification for cross-department collaborative business processes using extended Petri nets, IEEE Trans. Syst. Man Cybern. Syst. 45 (2) (2015) 349–362, http://dx.doi.org/10.1109/TSMC.2014.2334276.

[15] G. Elkoumy, S.A. Fahrenkrog-Petersen, M. Dumas, P. Laud, A. Pankova, M. Weidlich, Secure multi-party computation for inter-organizational process mining, in: S. Nurcan, I. Reinhartz-Berger, P. Soffer, J. Zdravkovic (Eds.), 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, in: Lecture

Notes in Business Information Processing, 387, Springer, 2020, pp. 166–181, http://dx.doi.org/10.1007/978-3-030-49418-6_11.

[16] C. Liu, H. Duan, Q. Zeng, M. Zhou, F. Lu, J. Cheng, Towards comprehensive support for privacy preservation cross-organization business process mining, IEEE Trans. Serv. Comput. 12 (4) (2019) 639–653, http://dx.doi.org/10.1109/TSC.2016.2617331.

[17] A. Khan, A. Ghose, H.K. Dam, Cross-silo process mining with federated learning, in: Service-Oriented Computing - 19th International Conference, ICSOC 2021, Virtual Event, November 22–25, 2021, Proceedings, in: Lecture Notes in Computer Science, vol. 13121, Springer, 2021, pp. 612–626, http://dx.doi.org/10.1007/978-3-030-91431-8_38.

[18] M. Rafiei, W.M.P. van der Aalst, An abstraction-based approach for privacy-aware federated process mining, IEEE Access 11 (2023) 33697–33714, http://dx.doi.org/10.1109/ACCESS.2023.3263673.

[19] V. Goretti, D. Basile, L. Barbaro, C.D. Ciccio, Trusted execution environment for decentralized process mining, in: G. Guizzardi, F.M. Santoro, H. Mouratidis, P. Soffer (Eds.), Advanced Information Systems Engineering - 36th International Conference, CAiSE 2024, Limassol, Cyprus, June 3–7, 2024, Proceedings, in: Lecture Notes in Computer Science, vol. 14663, Springer, 2024, pp. 509–527, http://dx.doi.org/10.1007/978-3-031-61057-8_30.

[20] A. Rozinat, W.M.P. van der Aalst, Conformance checking of processes based on monitoring real behavior, Inf. Syst. 33 (1) (2008) 64–95, http://dx.doi.org/10.1016/j.is.2007.07.001.

[21] A. Adriansyah, B.F. van Dongen, W.M.P. van der Aalst, Conformance checking using cost-based fitness analysis, in: Proceedings of the 15th IEEE International Enterprise Distributed Object Computing Conference, EDOC 2011, Helsinki, Finland, August 29 – September 2, 2011, IEEE Computer Society, 2011, pp. 55–64, http://dx.doi.org/10.1109/EDOC.2011.12.

[22] N. Lohmann, P. Massuthe, K. Wolf, Operating guidelines for finite-state services, in: J. Kleijn, A. Yakovlev (Eds.), Petri Nets and Other Models of Concurrency - ICATPN 2007, 28th International Conference on Applications and Theory of Petri Nets and Other Models of Concurrency, ICATPN 2007, Siedlce, Poland, June 25–29, 2007, Proceedings, in: Lecture Notes in Computer Science, vol. 4546, Springer, 2007, pp. 321–341, http://dx.doi.org/10.1007/978-3-540-73094-1_20.

[23] C. Stahl, W.M.P. van der Aalst, Behavioral service substitution, in: A. Bouguettaya, Q.Z. Sheng, F. Daniel (Eds.), Web Services Foundations, Springer, 2014, pp. 215–244, http://dx.doi.org/10.1007/978-1-4614-7518-7_9.

[24] M. Rafiei, G. Elkoumy, W.M.P. van der Aalst, Quantifying temporal privacy leakage in continuous event data publishing, in: Cooperative Information Systems - 28th International Conference, CoopIS 2022, Bozen-Bolzano, Italy, October 4–7, 2022, Proceedings, in: Lecture Notes in Computer Science, vol. 13591, Springer, 2022, pp. 75–94, http://dx.doi.org/10.1007/978-3-031-17834-4_5.

[25] G. Elkoumy, M. Dumas, Libra: High-utility anonymization of event logs for process mining via subsampling, in: A. Burattin, A. Polyvyanyy, B. Weber (Eds.), 4th International Conference on Process Mining, ICPM 2022, Bolzano, Italy, October 23–28, 2022, IEEE, 2022, pp. 144–151, http://dx.doi.org/10.1109/ICPM57379.2022.9980619.

[26] S.A. Fahrenkrog-Petersen, M. Kabierski, F. Rösel, H. van der Aa, M. Weidlich, SaCoFa: Semantics-aware control-flow anonymization for process mining, in: C.D. Ciccio, C.D. Francescomarino, P. Soffer (Eds.), 3rd International Conference on Process Mining, ICPM 2021, Eindhoven, The Netherlands, October 31 – Nov. 4, 2021, IEEE, 2021, pp. 72–79, http://dx.doi.org/10.1109/ICPM53251.2021.9576857.

[27] M. Rafiei, F. Wangelik, W.M.P. van der Aalst, TraVaS: Differentially private trace variant selection for process mining, in: Process Mining Workshops, Springer, 2023, pp. 114–126, http://dx.doi.org/10.1007/978-3-031-27815-0_9.

[28] M. Rafiei, F. Wangelik, M. Pourbafrani, W.M.P. van der Aalst, TraVaG: Differentially private trace variant generation using GANs, in: S. Nurcan, A.L. Opdahl, H. Mouratidis, A. Tsohou (Eds.), Research Challenges in Information Science: Information Science and the Connected World - 17th International Conference, RCIS 2023, Corfu, Greece, May 23–26, 2023, Proceedings, in: Lecture Notes in Business Information Processing, vol. 476, Springer, 2023, pp. 415–431, http://dx.doi.org/10.1007/978-3-031-33080-3_25.

[29] A.V. Ratzer, L. Wells, H.M. Lassen, M. Laursen, J.F. Qvortrup, M.S. Stissing, M. Westergaard, S. Christensen, K. Jensen, CPN tools for editing, simulating, and analysing coloured Petri nets, in: Applications and Theory of Petri Nets 2003, Springer Berlin Heidelberg, Berlin, Heidelberg, 2003, pp. 450–462.