Formal Policy-based Provenance Audit

Denis Butin, Denise Demirel, and Johannes Buchmann

TU Darmstadt, Germany
{dbutin,ddemirel,buchmann}@cdc.informatik.tu-darmstadt.de

Abstract. Data processing within large organisations is often complex, impeding both the traceability of data and the compliance of processing with usage policies. The chronology of the ownership, custody, or location of data — its provenance — provides the necessary information to restore traceability. However, to be of practical use, provenance records should include sufficient expressiveness by design with a posteriori analysis in mind, e.g. the verification of their compliance with usage policies. Additionally, they ought to be combined with systematic reasoning about their correctness. In this paper, we introduce a formal framework for policy-based provenance audit. We show how it can be used to demonstrate correctness, consistency, and compliance of provenance records with machine-readable usage policies. We also analyse the suitability of our framework for the special case of privacy protection. A formalised perspective on provenance is also useful in this area, but it must be integrated into a larger accountability process involving data protection authorities to be effective. The practical applicability of our approach is demonstrated using a provenance record involving medical data and corresponding privacy policies with personal data protection as a goal.

1 Introduction

Data *provenance* [10,29,30] has been defined disparately, for instance as the origin and history of data or as process-related metadata. It is often associated with the semantic web [19], used as a tool to facilitate content curation by restoring trustworthiness in the quality of digital data, or providing replication recipes [16]. In this work, we adopt the W3C perspective on provenance as *(...) a record that describes the people, institutions, entities and activities, involved in producing, influencing, or delivering a piece of data or a thing* [24]. Hence the key idea of provenance — as taken up here — is to paint a comprehensive picture clarifying the life cycle of data, from its creation to its destruction, including the way it is used, and how other data is derived or linked from it.

This makes provenance valuable to verify compliance with usage policies when sensitive data is collected, stored, and processed by several entities within enterprises or governmental institutions. In such large networks, data flow easily becomes opaque, making simple audit methods — such as log analysis limited to a single component — insufficient and leading to failure to comply with usage policies. Here provenance has the potential to underpin accountability, i.e. each data processing entity must not only comply with data protection rules, but

actively demonstrate compliance[1]. This allows an auditor to detect failures within the system. Furthermore, these usage policy rules can be part of the provenance data, making it possible to globally declare them and to prevent inconsistencies. Thus, provenance allows to build powerful audit tools for large networks, such as hospitals and research institutions. In addition, systems using cloud computing typically involve multiple entities, and can therefore benefit from this approach.

An important class of usage policies addresses the privacy of sensitive data [20]. In the context of accountability, such *privacy policies* are often modelled as machine-readable sets of rules in a formally specified privacy policy language such as PPL [32]. While the idea of using provenance as a tool to audit the compliance with privacy regulations is not new, so far formal approaches have been lacking. Both the consistency of provenance records and their compliance with associated privacy policies ought to be stated precisely to pave the way for automated analyses. Some aspects such as the enforcement of processing purposes are not fully amenable to automation, but provenance records allow to collect enough information for complementary manual verification.

Contribution We present a formal framework modelling provenance events and their compliance with respect to usage policies. We first introduce usage policies and model them as tuples including forwarding policies, authorised purposes for different types of operations, deletion delays, and linking and derivation policies (Sec. 2.1). A data category is not necessarily always associated with the same usage policy. The relevant policy depends on the component and on the log events under consideration. Afterwards, we model provenance records as sequences of discrete events. Each record refers to a single data subject, e.g. a patient in the eHealth scenario, but can accommodate an arbitrary number of entities and processed data categories. The granularity of the events is chosen in accordance with the policies (Sec. 2.2). Next, we establish additional notation (Sec. 3.1) and formalise the correctness (Sec. 3.2) and compliance of logs of provenance events when dealing with privacy policies (Sec. 3.3). Correctness pertains to the internal consistency of the provenance record, independently of the usage policies under consideration. Conversely, compliance relates to the relation between the provenance record and the associated usage policies. This formalisation can serve as a basis for a posteriori provenance analysis by an auditor. Next, we discuss the limitations of the approach when dealing with privacy policies and show how it can be included in a global accountability process (Sec. 4). We consider the case of privacy separately because the guarantees at stake are especially critical when personal data is involved. Furthermore, the application of provenance records to this use case immediately raises the question of whether logging creates additional privacy risks. We show under which conditions our framework for provenance-based auditing is applicable to privacy policies. The framework is then evaluated

[1] In the scope of accountability as a data protection principle [3], this proof-of-compliance requirement is not limited to high-level statements of intent, but is often seen as incorporating a practical level as well, i.e. concrete data handling actions [9].

through a scenario involving personal medical data processed among different entities (Sec. 5). Finally, we provide a review of related work (Sec. 6) and offer our conclusions (Sec. 7).

2 Usage Policies and Provenance Events

In this section we introduce the concept of usage policies and provenance events. We refer to an entity collecting or processing data as *component*. In the scope of this paper, a *usage policy* is assumed to be a machine-readable set of rules regarding the use of data. Depending on the scenario under consideration, the values of usage policies are assigned either by a component, or by a data subject[2] if the processed data is of personal nature. With respect to the data processed, a *data category* is a designation of the type of data in natural language, i.e. `postal address`. By *purpose*, we mean the finality or goal of an operation. Each event is assigned one or two timestamps, corresponding to the start and (if applicable) end of the event.

2.1 Usage Policies for Provenance

To evaluate provenance records systematically, we introduce a formalisation of usage policies. Alternatively, policies can be described involving text in natural language. However, this approach presents usability challenges, especially in the subcase of privacy policies [28].

Furthermore, most existing machine-readable policy languages [20] have limited expressiveness, often impeding automated compliance checking [8]. Thus, to the best of our knowledge, our method is the first to provide sufficient expressiveness to be used in conjunction with provenance records, by covering the broad range of provenance activities.

A number of data handling operations used here are absent from usual policy languages (such as PPL [32]), but essential to provenance modelling. Provenance events will be formalised later (Sec. 2.2), but it is already helpful to introduce these operations intuitively in this section to put the usage policies elements in context.

In the W3C PROV ontology [21], *derivation*, is defined as "a transformation of one entity into another". We use *linking* to refer to the combination of different categories of data from a single source. The *forwarding* of data refers to the sending of data from one component to another, with the assumption that the original component also retains control of the data.

We now define provenance usage policies formally in terms of constraints for deletion, forwarding, linking, derivation, and use for specific purposes.

Definition 1 (Usage Policy). *Provenance usage policies are defined as tuples:*

[2] In practice, data subjects may delegate the power of negotiating policies to a third party they trust.

$$\mathcal{P} = \mathsf{D_g} \times \mathsf{D_{rf}} \times \mathsf{Fw} \times \mathsf{Li} \times \mathsf{De} \times \mathsf{P_{use}} \times \mathsf{P_{der}}$$

and address rules for deletion, forwarding, linking, derivation, and authorised purposes. These usage policy components are defined as follows:

$\mathsf{D_g}$ *The global deletion delay* $\mathsf{D_g}$ *is the maximal delay after which data of any category must have been deleted. Time is measured from the initial collection of each piece of data. This deletion does not need to be requested explicitly — the policy specifies that it should happen unconditionally. However, earlier deletion may occur following an explicit request.*

$\mathsf{D_{rf}}$ *The request fulfilment delay* $\mathsf{D_{rf}}$ *is the maximal delay for components to fulfil requests, e.g. deletion requests. The delay is measured from the timestamp corresponding to the start of the requested event. We do not account for delays between the moment the request is sent and the moment it is received by the component(s) holding the data.*

Fw *The forwarding policy* Fw *is a pair* (rest, List)*. Possible values for* rest *are* $\bot$ *(data may be forward to any component),* $\top$ *(data may be forwarded to no other component),* bl *(a list of forbidden components is declared) and* wl *(a list of authorised components is declared). If rest* $=\bot$ *or rest* $=\top$*, the set* List *must be empty. If* rest $=$ bl*, the set* List *is a blacklist of components, i.e. to no component on the list may be forwarded any data. If* rest $=$ wl*,* List *is a whitelist of components, i.e. any data may be forwarded to any components on the list.* List *then contains identifiers of all components for which data forwarding is forbidden or permitted, respectively.*

Li *The linking policy* Li *is a set of pairs limiting the linking of different data categories by components. If two data categories are in the same pair of this set, pieces of data of these categories may never be linked.*[3] *This policy component can be used to restrict profiling [4] based on correlation.*

De *The derivation policy* De *is a set of data categories. Data categories in* De *may never be used to derive new data. This policy component can be used, for instance, to prevent the derivation of personal data from perturbed data [25].*

$\mathsf{P_{use}}$ *The authorised use purposes* $\mathsf{P_{use}}$ *are a set of pairs (data category, acceptable purpose for use).*

$\mathsf{P_{der}}$ *The authorised derivation purposes* $\mathsf{P_{der}}$ *are a set of pairs (data category, acceptable purpose for derivation).*

Purposes are assumed to be taken from a fixed ontology, i.e. a centralised taxonomy, as in PPL [32]. The adequacy of using a purpose ontology is discussed in Sec. 4.

As an illustration, we describe in turn the components of a concrete usage policy π:

[3] For instance, a linking policy can prevent de-anonymisation.

- $\pi.\mathsf{D}_\mathsf{g}$ = 3 months — data of any category must be deleted within 3 months after its collection;
- $\pi.\mathsf{D}_\mathsf{rf}$ = 1 day — requests received by components must be fulfilled within one day;
- $\pi.\mathsf{Fw} = (\mathsf{wl}, \mathsf{List})$ with $\mathsf{List} = \{\texttt{Hospital}, \texttt{ResearchInstitute}\}$ — a (white)list of authorised components is declared: data of any category may be forwarded to the components `ResearchInstitute` and `Hospital`. No data exports to other components are allowed;
- $\pi.\mathsf{Li} = \{(\texttt{Treatment}, \texttt{Status}), (\texttt{ID}, \texttt{Drug})\}$ — pieces of data with respective categories `Treatment` and `Status` may never be linked, directly or indirectly. Otherwise, information about the stage of a disease may be extrapolated. In addition, pieces of data with respective categories `ID` and `Drug` may never be linked, directly or indirectly, so as not to reveal which patient is taking which drug;
- $\pi.\mathsf{De} = \{\texttt{Frequency}, \texttt{Risk}, \texttt{Drug}\}$ — new data may never be derived from pieces of data with category `Frequency`, `Risk`, or `Drug`, directly or indirectly. This prevents that sensitive information about the frequency, e.g. of treatments or hospital visits, the risk of getting a certain disease, or drugs taken by a patient is extracted;
- $\pi.\mathsf{P}_\mathsf{use} = \{(\texttt{Treatment}, \texttt{Logistic}), (\texttt{ID}, \texttt{Marketing})\}$ — data with category `Treatment` may be used (directly or indirectly) only for the purpose `Logistic`. This allows, for instance, hospitals to reserve operating theaters for patients. Data with category `ID`, e.g. the name, address, and family status of patients, may be used (directly or indirectly) only for the purpose `Marketing`. Data of other categories may not appear in `Use` events;
- $\pi.\mathsf{P}_\mathsf{der} = \{(\texttt{History}, \texttt{Statistic})\}$ — Data with category `History` may be derived from (directly or indirectly) only for the purpose `Statistic`, to allow statistical evaluation of medical data. Data of other categories may not appear as the first argument of a `Derive` event.

2.2 Provenance Events

The usage policies defined above allow to define requirements that must be fulfilled by components when processing data. Their compliance with these policies can be evaluated using the available provenance data. However, to this end, provenance information ought to be represented in a unified way.

Provenance is often visualised as a graph. Instead, we model it as a sequence of discrete events, i.e. $\lambda = \{\lambda_1, \ldots, \lambda_n\}$, allowing us to reason over sets of such events. Since all events include timestamps, they can be represented a posteriori in a chronological fashion. However, log indexes are not assumed to be ordered chronologically, i.e. $i < j$ does not imply that λ_i occurred before λ_j.

Definition 2 (Provenance Events). *Let $\boldsymbol{Pu}$ be a purpose ontology, seen as a set of purposes in natural language. Let $\mathbb{P}(\boldsymbol{Pu})$ be the power set of $\boldsymbol{Pu}$. Similarly,*

$\boldsymbol{Ca}$ is an ontology of data categories and $\mathbb{P}(\boldsymbol{Ca})$ is its power set. Furthermore, let $\boldsymbol{C}$ be a set of components. For all events and independently, $\Theta \in \mathbb{P}(\boldsymbol{Ca})$, $\rho \in \boldsymbol{Pu}$, $P \in \mathbb{P}(\boldsymbol{Pu})$, $C, C' \in \boldsymbol{C}$, and the parameters t, s, and e are timestamps. While t marks the start of discrete events, s and e is used to specify the start and end time for the `Use` *event, which occurs over a time period. Data circulates together with its associated usage policy $\pi \in \mathcal{P}$; we follow the* sticky policy *approach advocated by Pearson [27]. In a sequence of discrete events $\lambda = \{\lambda_1, \ldots, \lambda_n\}$ a single event λ_i for $i \in [1, n]$ can be one of the following provenance events:*

- **(`Acquire`,** Θ, C, π, P, t**)** *A set of data with a corresponding set of data categories Θ was collected by a component C for a set of specific purposes P. The values of the data do not appear in the event, only its categories. In addition, this event contains the usage policy π set for the collected data and a timestamp t.*
- **(`Use`,** Θ, C, ρ, R, s, e**)** *This event marks the use of a set of data categories Θ, e.g.* `{Postal address, Age group}` *by a component C. Reason R is a justification of the single chosen purpose designation ρ in natural language. The timestamps s and e mark the start and the end of this event. Note that for this to be meaningful Θ must be a subset of previously collected data categories.*
- **(`Export`,** Θ, C, C', π, P, t**)** *A set of data, with a corresponding set of data categories Θ, was sent by component C to component C' at time t. The associated usage policy is π and the data export was performed for a specific set of purposes P.*
- **(`Link`,** $\theta, \theta', \theta'', C, \pi, \rho, R, t$**)** *At time t component C linked two data elements with categories θ and θ'. The result is a single piece of data with category θ'', now available to component C. θ'' does not have to be new, but it may be (i.e. has not appeared before). This event also contains a reason R justifying the single chosen purpose designation ρ. The usage policy π is associated with the new data.*
- **(`Derive`,** $\theta, \theta', C, \pi, \rho, R, t$**)** *At time t, component C derived data with category θ' from data with category θ. In addition, the event contains reason R, the corresponding single purpose ρ, and a usage policy π.*
- **(`ReqRemove`,** Θ, t**)** *At time t, total deletion of the set of data categories Θ was requested. Removal requests are assumed to be sent simultaneously to all components relevant to the set of data categories under consideration.*
- **(`Remove`,** Θ, C, t**)** *The set of data categories Θ were deleted by component C at time t.*

As an example, consider the event λ_1 = (`Export`, {`Treatment`, `ID`, `Status`}, `Hospital`, `ResearchInstitute`, π_1, {`Logistic`, `Statistic`}, 2016-05-12T12:17) and the event λ_2 = (`Link`, `Frequency`, `Treatment`, `Risk`, `ResearchInstitute`, π_2, `Statistic`, `Correlation study`, 2016-05-20T12:14). In λ_1, a data set with categories `Treatment`, `ID`, and `Status` is forwarded from the component `Hospital` to the component `ResearchInstitute`. The associated usage policy π_1 restricts how the component `ResearchInstitute` can use the forwarded data set, but is not relevant for this event. The purposes `Logistic` and `Statistic` are provided for this data export, and the timestamp of the event is 2016-05-12T12:17. In

λ_2, component `ResearchInstitute` links two pieces of data with respective categories `Frequency` and `Treatment`. The result of this linking is a piece of data with category `Risk`. The usage policy now associated with `Risk` for the component `ResearchInstitute` is π_2, which, again, restricts how the research institute can use the forwarded data set, but is not relevant for this event. The provided reason for this linking operation is `Correlation study`, meant to justify the included purpose designation `Statistic`. and the last parameter 2016-05-20T12:14 is, again, the timestamp.

3 Formalising Correctness and Compliance

Having defined both usage policies and provenance events enables two types of checks:

- Provenance correctness: a number of conditions must be fulfilled for provenance information to be coherent, independently of any usage policy. Sanity checks are possible and can be seen as a category of minimal guarantees.
- Compliance of provenance with usage policies: the compliance of recorded provenance can be analysed with regard to the predefined policies.

In the following, we first establish some useful definitions in Sec. 3.1, followed by rules for internal correctness of logs in Sec. 3.2 and rules for the compliance of logs with usage policies in Sec. 3.3.

3.1 Definitions

Some formalism is needed to model both aspects: the correctness of logs, and their compliance with usage policies. Let $\lambda = \{\lambda_1, \ldots, \lambda_n\}$ be a log of n provenance events and $i \in [1, n]$.

Definition 3 (Event type). *Let* EvType *be the function mapping an event to its type. That is, if* $X \in \{$`Acquire`, `Use`, `Export`, `Link`, `Derive`, `ReqRemove`, `Remove`$\}$ *and* $\lambda_i = (X, \ldots)$, *then* $\text{EvType}(\lambda_i) = X$.

Definition 4 (Event time). EvTime *is defined as the function returning the starting time of an event.* $\text{EvTime}(\lambda_i) = s$, *if* $\text{EvType}(\lambda_i) = \texttt{Use} \wedge \lambda_i = (\texttt{Use}, \ldots, s, e)$. $\text{EvTime}(\lambda_i) = t$, *if* $\text{EvType}(\lambda_i) \neq \texttt{Use} \wedge \lambda_i = (\text{EvType}(\lambda_i), \ldots, t)$. *We assume that different events always feature different starting times, i.e.* $\lambda_i \neq \lambda_j \implies \text{EvTime}(\lambda_i) \neq \text{EvTime}(\lambda_j)$.

Definition 5 (Active component). $\text{Active}(\lambda_i)$ *is the acting component for a given event, i.e.*

$$\begin{cases} \lambda_i = (\texttt{Acquire}, \Theta, C, \pi, P, t) \implies \text{Active}(\lambda_i) = C \\ \lambda_i = (\texttt{Use}, \Theta, C, \rho, R, s, e) \implies \text{Active}(\lambda_i) = C \\ \lambda_i = (\texttt{Export}, \Theta, C, C', \pi, P, t) \implies \text{Active}(\lambda_i) = C \\ \lambda_i = (\texttt{Link}, \theta, \theta', \theta'', C, \pi, \rho, R, t) \implies \text{Active}(\lambda_i) = C \\ \lambda_i = (\texttt{Derive}, \theta, \theta', C, \pi, \rho, R, t) \implies \text{Active}(\lambda_i) = C \\ \lambda_i = (\texttt{Remove}, \Theta, C, t) \implies \text{Active}(\lambda_i) = C \end{cases}$$

The active component is undefined for `ReqRemove`, since this event is not initiated by any component but triggered externally.

Definition 6 (Set of controllers). $\text{Control}(\lambda, \theta)$ *is the set of components that have gained control over data with category* θ *in log* λ*, i.e.* $\text{Control}(\lambda, \theta) = \big\{C \mid \exists\, \lambda_i \in \lambda, \Theta, \theta', \theta'', \pi, \rho, P, R, t, C' \mid \big(\, \theta \in \Theta \wedge \big(\lambda_i = (\texttt{Acquire}, \Theta, C, \pi, P, t) \vee \lambda_i = (\texttt{Export}, \Theta, C', C, \pi, P, t)\big)\big) \;\vee\; \lambda_i = (\texttt{Derive}, \theta', \theta, C, \pi, \rho, R, t) \;\vee\; \lambda_i = (\texttt{Link}, \theta', \theta'', \theta, C, \pi, \rho, R, t)\big\}$.

Definition 7 (Associated data categories). *The function* DataCat *takes as input an event different from* `ReqRemove` *or* `Remove` *and returns the set of data categories appearing in the event, in any form:*

- $\lambda_i = (\texttt{Acquire}, \Theta, \ldots) \;\vee\; \lambda_i = (\texttt{Use}, \Theta, \ldots) \;\vee\; \lambda_i = (\texttt{Export}, \Theta, \ldots) \implies \text{DataCat}(\lambda_i) = \Theta$.
- $\lambda_i = (\texttt{Link}, \theta, \theta', \theta'', \ldots) \implies \text{DataCat}(\lambda_i) = \{\theta, \theta', \theta''\}$.
- $\lambda_i = (\texttt{Derive}, \theta, \theta', \ldots) \implies \text{DataCat}(\lambda_i) = \{\theta, \theta'\}$.

Definition 8 (Descended data categories). *The function* $\text{Dsc}(\lambda, \theta)$ *returns the set of data categories generated from the data category* θ*, directly or indirectly, through linking or derivation. It is defined recursively as follows:*

$$\begin{cases} \theta \in \mathit{Dsc}(\lambda, \theta) \\ \lambda_i = (\texttt{Derive}, X, \theta', C, \pi, \rho, R, t) \;\wedge X \in \mathit{Dsc}(\lambda, \theta) \implies \theta' \in \text{Dsc}(\lambda, \theta) \\ \lambda_i = (\texttt{Link}, X, X', \theta', C, \pi, \rho, R, t) \;\wedge X \in \mathit{Dsc}(\lambda, \theta) \implies \theta' \in \text{Dsc}(\lambda, \theta) \\ \lambda_i = (\texttt{Link}, X, X', \theta', C, \pi, \rho, R, t) \;\wedge X' \in \mathit{Dsc}(\lambda, \theta) \implies \theta' \in \text{Dsc}(\lambda, \theta) \end{cases}$$

Definition 9 (Relative strength of usage policies). *Let* π *and* π' *be two usage policies as defined in Def. 1.* π' *is said to be stronger or equal than* π*, denoted* $\pi' \geq \pi$*, if all of the following conditions hold: (1)* $\pi'.\mathsf{D_g} \leq \pi.\mathsf{D_g}$*; (2)* $\pi'.\mathsf{D_{rf}} \leq \pi.\mathsf{D_{rf}}$*; (3)* $\pi'.\mathsf{Fw} = (\top, \varnothing) \;\vee\; \pi'.\mathsf{Fw} = \pi.\mathsf{Fw} = (\bot, \varnothing) \;\vee\; \big(\pi'.\mathsf{Fw} = (\mathsf{bl}, \mathsf{List}') \;\wedge\; \pi.\mathsf{Fw} = (\mathsf{bl}, \mathsf{List}) \wedge \mathsf{List} \subseteq \mathsf{List}'\big) \vee \big(\pi'.\mathsf{Fw} = (\mathsf{wl}, \mathsf{List}') \wedge \pi.\mathsf{Fw} = (\mathsf{wl}, \mathsf{List}) \wedge \mathsf{List}' \subseteq \mathsf{List}\big)$*; (4)* $\pi.\mathsf{Li} \subseteq \pi'.\mathsf{Li}$*; (5)* $\pi.\mathsf{De} \subseteq \pi'.\mathsf{De}$*; (6)* $\pi'.\mathsf{P_{use}} \subseteq \pi.\mathsf{P_{use}}$*; (7)* $\pi'.\mathsf{P_{der}} \subseteq \pi.\mathsf{P_{der}}$*. In particular,* $\pi = \pi' \iff \pi \geq \pi' \;\wedge\; \pi' \geq \pi$.

Definition 10 (Extracting the usage policy associated with a data category). *The usage policy relevant for a data category* θ *depends both on the component* C *under consideration and on the latest relevant event of the log* $\lambda = \{\lambda_1, \ldots, \lambda_n\}$*. We define* $\lambda_*(\lambda, \theta, C)$ *to be the latest event defining a usage policy for* θ*. It is the event such that* $\text{EvTime}(\lambda_*(\lambda, \theta, C)) = \max\big\{t \mid \exists\; \lambda_i \in \lambda, \Theta, \theta_1, \theta_2, C, C', \pi, P, \rho, R \mid \big(\lambda_i = (\texttt{Acquire}, \Theta, C, \pi,\; P, t) \;\wedge\; \theta \in \Theta\big) \;\vee\; \big(\lambda_i = (\texttt{Export}, \Theta, C', C, \pi, \rho, R, t) \;\wedge\; \theta \in \Theta\big) \;\vee\; \lambda_i = (\texttt{Link}, \theta_1, \theta_2, \theta, C, \pi, \rho, R, t) \vee \lambda_i = (\texttt{Derive}, \theta_1, \theta, C, \pi, \rho,\; R, t)\big\}$*. Based on the value of this event* λ_**, we now define the associated usage policy* $\pi_*(\lambda, \theta, C)$ *as follows.*

$$\begin{cases} \lambda_* = (\texttt{Acquire}, \Theta, C, \pi, P, t) \implies \pi_* = \pi \\ \lambda_* = (\texttt{Export}, \Theta, C', C, \pi, \rho, R, t) \implies \pi_* = \pi \\ \lambda_* = (\texttt{Link}, \theta_1, \theta_2, \theta, C, \pi, \rho, R, t) \implies \pi_* = \pi \\ \lambda_* = (\texttt{Derive}, \theta_1, \theta, C, \pi, \rho, R, t) \implies \pi_* = \pi \end{cases}$$

Since different events feature different timestamps (Def. 4), λ_* and consequently π_* are uniquely defined for a given triple (λ, θ, C).

The following correctness and compliance rules are stated $\forall\ \lambda_i \in \lambda$.

3.2 Rules for Internal Correctness of Logs

Correctness rules ensure the internal consistency of event logs and are independent of the associated usage policy.

(Cor1) For every `Use` or `Export` event and for every data category appearing in the event the data was acquired, derived, or linked somewhere. Note that we only consider "complete" provenance histories. $\lambda_i = (\texttt{Use}, \Theta, C, \rho, R, t, e) \ \vee\ \lambda_i = (\texttt{Export}, \Theta, C, C', \pi, P, t) \Longrightarrow \forall\ \theta \in \Theta, \exists\ \lambda_j \in \lambda, C'', \Theta', \theta_1, \theta_2, \theta_3, \rho', P', R', t, \pi' \mid (\lambda_j = (\texttt{Acquire}, \Theta', C'', \pi',\ P', t') \ \wedge\ \theta \in \Theta') \ \vee\ \lambda_j = (\texttt{Derive}, \theta_1, \theta, C'', \pi', \rho', R', t') \vee \lambda_j = (\texttt{Link},\ \theta_2, \theta_3, \theta, C'', \pi', \rho', R', t') \ \wedge\ (\ t' < t)$.

(Cor2) A similar rule holds for `Derive` events: $\lambda_i = (\texttt{Derive}, \theta, \theta', C, \pi, \rho, R, t) \Longrightarrow \exists\ \lambda_j \in \lambda, \Theta, C', \pi', \rho', P', t', \theta'', R', \theta_1, \theta_2 \mid \big(\lambda_j = (\texttt{Acquire}, \Theta, C', \pi', P', t') \wedge \theta \in \Theta\big) \vee \lambda_j = (\texttt{Derive}, \theta'', \theta, C', \rho', R', t') \vee \lambda_j = (\texttt{Link}, \theta_1, \theta_2, \theta, C', \pi', \rho', R', t') \ \wedge\ (t' < t)$.

(Cor3) Similar rules hold for both source arguments of `Link` events:

- $\lambda_i = (\texttt{Link}, \theta, \theta', \theta'', C, \pi, \rho, R, t) \Longrightarrow \exists\ j, \Theta, C', \pi', \rho', P', t', R', \theta_1, \theta_2, \theta_3 \mid \big(\lambda_j = (\texttt{Acquire}, \Theta, C', \pi', P', t') \wedge \theta \in \Theta\big) \vee \lambda_j = (\texttt{Derive}, \theta_1, \theta, C', \pi', \rho', R', t') \ \vee\ \lambda_j = (\texttt{Link}, \theta_2, \theta_3, \theta, C', \pi', \rho', R, t') \ \wedge\ (t' < t)$.
- $\lambda_i = (\texttt{Link}, \theta, \theta', \theta'', C, \pi, \rho, R, t) \Longrightarrow \exists\ j, \Theta, C', \pi', \rho', P', t', R', \theta_1, \theta_2, \theta_3 \mid \big(\lambda_j = (\texttt{Acquire}, \Theta, C', \pi', P', t') \wedge \theta' \in \Theta\big) \vee \lambda_j = (\texttt{Derive}, \theta_1, \theta', C', \pi', \rho', R', t') \ \vee\ \lambda_j = (\texttt{Link}, \theta_2, \theta_3, \theta', C', \pi', \rho, R', t') \ \wedge\ (t' < t)$.

(Cor4) For non-instantaneous events (i.e. data use), starting and ending timestamps are well-formed: $\lambda_i = (\texttt{Use}, \Theta, C, \rho, R, s, e) \Longrightarrow s < e$.

(Cor5) Successive data derivations exhibit monotonous timestamps: $\lambda_i = (\texttt{Derive}, \theta, \theta', C, \pi, \rho, R, t) \ \wedge\ \lambda_j = (\texttt{Derive}, \theta', \theta'', C', \pi', \rho', R', t') \ \wedge\ i \neq j \ \wedge\ \lambda_j \in \lambda \Longrightarrow t' > t$.

(Cor6) For a given component, the usage policies for a partial log λ of log λ' associated with a given data category are consistent, i.e. the policy may not become weaker. $\lambda \subseteq \lambda' \Longrightarrow \pi_*(\theta, C, \lambda') \geq \pi_*(\theta, C, \lambda)$.

(Cor7) Data of a given category is not processed in any form after the data with this category has been removed: $\lambda_i = (\texttt{Remove}, \Theta, C, t) \ \wedge\ \lambda_j \in \lambda \ \wedge\ \theta \in \Theta \ \wedge\ \theta \in \mathrm{DataCat}(\lambda_j) \wedge\ \mathrm{EvType}(\lambda_j) \in \{\texttt{Use}, \texttt{Export}, \texttt{Link}, \texttt{Derive}\} \Longrightarrow \mathrm{EvTime}(\lambda_j) < t$.

(Cor8) No data forwarding is permitted once a removal request has been received, even before the request fulfilment delay is reached: $\lambda_i = (\texttt{ReqRemove}, \Theta, t) \ \wedge\ \theta \in \Theta \ \wedge\ \lambda_j \in \lambda \ \wedge\ \lambda_j = (\texttt{Export}, \Theta', C, C', \pi,\ P, t') \Longrightarrow \theta \notin \Theta' \ \vee\ t' < t$.

(Cor9) Similarly, no data use may start after a removal request has been sent: $\lambda_i = (\texttt{ReqRemove}, \Theta, t) \ \wedge\ \theta \in \Theta \ \wedge\ \lambda_j \in \lambda \ \wedge\ \lambda_j = (\texttt{Use}, \Theta', C, \rho, R, s, e) \Longrightarrow \theta \notin \Theta' \ \vee\ s < t$.

(Cor10) Likewise, new data may not be derived from data for which deletion has already been requested: $\lambda_i = (\texttt{ReqRemove}, \Theta, t) \ \wedge \ \theta \in \Theta \ \wedge \ \lambda_j \in \lambda \wedge \lambda_j = (\texttt{Derive}, \theta, \theta', C, \pi, \rho, R, t') \Longrightarrow t' < t$.

(Cor11) The new usage policy affecting data generated by linking must be stronger or equal than the policies associated with each of the source data elements: $\lambda_i = (\texttt{Link}, \theta_1, \theta_2, \theta, C, \pi, \rho, R, t) \wedge \lambda_j = (\texttt{Acquire}, \Theta_1, C, \pi_1, P, R', t') \wedge \theta_1 \in \Theta_1 \ \wedge \ \lambda_k = (\texttt{Acquire}, \Theta_2, C, \pi_2, P', R'', t'') \ \wedge \ \theta_2 \in \Theta_2 \Longrightarrow \pi \geq \pi_1 \ \wedge \ \pi \geq \pi_2 \ \wedge \ t' < t \ \wedge \ t'' < t$.

(Cor12) A similar property holds for derived data: $\lambda_i = (\texttt{Derive}, \theta, \theta', C, \pi, \rho, R, t) \ \wedge \lambda_j = (\texttt{Acquire}, \Theta, C, \pi', P, R', t') \ \wedge \ \theta \in \Theta \Longrightarrow \pi \geq \pi' \ \wedge \ t' < t$.

Definition 11 (Correctness). *A log λ is said to be correct if all correctness properties* Cor1 *...* Cor12 *hold for λ.*

To illustrate this notion, consider the following example log $\lambda = \lambda_1 \ldots \lambda_7$:

λ_1: (Acquire, {Treatment, ID, Frequency}, Hospital, π_2, {Logistic, Statistic}, 2016-05-01T08:07)
λ_2: (Use, {Treatment, ID}, Hospital, Logistic, Patient registration mandatory, 2016-05-01T10:25, 2016-05-09T17:54)
λ_3: (Export, {Treatment, ID, Frequency}, Hospital, ResearchInstitute, π_1, {Logistic, Statistic}, 2016-05-12T12:17)
λ_4: (Link, ID, Treatment, History, ResearchInstitute, π_1, Statistic, Insurance billing requested, 2016-05-14T22:33)
λ_5: (Derive, History, Frequency, ResearchInstitute, π_1, Statistic, Quantitative research, 2016-05-18T09:41)
λ_6: (Use, {Treatment, Frequency}, ResearchInstitute, Logistic, Workflow optimisation, 2016-05-19T14:41, 2016-05-19T15:03)
λ_7: (Link, Frequency, Age, Risk, ResearchInstitute, π_1, Statistic, Correlation study, 2016-05-20T12:14)

λ is not correct, since Cor3 is violated. No acquisition, derivation or linking event yielding data category Age, appearing in λ_7, is part of λ. Thus, Age cannot be linked with Frequency in λ_7.

Note that provenance for a category of data may not necessarily extend over the entire data life cycle. In particular, the fact that the data may not have been deleted at the end of a log does not falsify the correctness of the log. From this perspective, the existence of deletion events become policy-dependent and is therefore covered by compliance rules, not by correctness rules.

3.3 Rules for Compliance of Logs with Usage Policies

Compliance rules depend on the values of associated usage policies.

Global and requested data deletion

(Com1) The global deletion delay $\mathsf{D_g}$ of usage policy $\pi_*(\lambda, \theta, C)$ extracted as defined in Def. 10 holds for all categories of data. No category of data can therefore appear in a log λ after the expiration of this global delay:
- $\lambda_i = (\texttt{Acquire}, \Theta, C, \pi, P, t) \ \wedge \ \theta \in \Theta \ \wedge \ \theta \in \text{DataCat}(\lambda_j) \ \wedge \lambda_j \in \lambda \wedge \text{Active}(\lambda_j) = C \implies \text{EvTime}(\lambda_j) - t < \pi_*(\lambda, \theta, C).\mathsf{D_g}$.
- $\lambda_i = (\texttt{Export}, \Theta, C, C', \pi, P, t) \ \wedge \ \theta \in \Theta \ \wedge \ \theta \in \text{DataCat}(\lambda_j) \ \wedge \lambda_j \in \lambda \wedge \text{Active}(\lambda_j) = C' \implies \text{EvTime}(\lambda_j) - t < \pi_*(\lambda, \theta, C).\mathsf{D_g}$.
- $\lambda_i = (\texttt{Link}, \theta', \theta'', \theta, C, \pi, \rho, R, t) \ \wedge \ \theta \in \text{DataCat}(\lambda_j) \ \wedge \lambda_j \in \lambda \wedge \text{Active}(\lambda_j) = C' \implies \text{EvTime}(\lambda_j) - t < \pi_*(\lambda, \theta, C).\mathsf{D_g}$.

(Com2) Deletion requests are fulfilled in a delay compatible with the associated policy's request fulfilment delay $\mathsf{D_{rf}}$: $\lambda_i = (\texttt{ReqRemove}, \Theta, t) \ \wedge \ \theta \in \Theta \implies \forall \ C \in \text{Control}(\lambda, \theta), \ \exists \ \lambda_j \in \lambda, \Theta', t' \mid \lambda_j = (\texttt{Remove}, \Theta', C, t') \ \wedge \ \theta \in \Theta' \wedge t' - t < \pi_*(\lambda, \theta, C).\mathsf{D_{rf}}$.

Data forwarding

(Com3) If all forwarding is forbidden, no data exports are allowed to take place: $\pi_*(\lambda, \theta, C).\mathsf{Fw} = (\top, \varnothing) \wedge \lambda_j \in \lambda \wedge \text{EvTime}(\lambda_j) \geq \text{EvTime}(\lambda_*(\lambda, \theta, C)) \wedge \theta \in \text{DataCat}(\lambda_j) \implies \text{EvType}(\lambda_j) \neq \texttt{Export}$, where $\lambda_*(\lambda, \theta, C)$ is the latest event defining a usage policy for θ (see Def. 10).

(Com4) If the associated forwarding policy defines a whitelist, all data exports are destined to components on the list: $\lambda_i = (\texttt{Export}, \Theta, C, C', \pi, P, t) \ \wedge \ \theta \in \Theta \ \wedge \ \pi_*(\lambda, \theta, C).\mathsf{Fw} = (\mathsf{wl}, \mathsf{List}) \wedge \text{EvTime}(\lambda_i) \geq \text{EvTime}(\lambda_*(\lambda, \theta, C)) \implies C' \in \mathsf{List}$.

(Com5) In case a blacklist is defined by the associated forwarding policy, no data exports towards components in the list take place: $\lambda_i = (\texttt{Export}, \Theta, C, C', \pi, P, t) \ \wedge \ \theta \in \Theta \ \wedge \ \pi_*(\lambda, \theta, C).\mathsf{Fw} = (\mathsf{bl}, \mathsf{List}) \wedge \text{EvTime}(\lambda_i) \geq \text{EvTime}(\lambda_*(\lambda, \theta, C)) \implies C' \notin \mathsf{List}$.

Data linking

(Com6) If the first data category in the argument list of a `Link` event also appears directly or indirectly in the data category set of the associated linking policy, then the second data category in the argument list of the event may not appear directly or indirectly in the same set of the linking policy: $\lambda_i = (\texttt{Link}, \theta', \theta'', \theta, C, \pi, \rho, R, t) \ \wedge \ \exists \ A \in \pi_*(\lambda, \theta, C).\mathsf{Li} \mid \theta_A \in A \ \wedge \ \theta' \in \text{Dsc}(\lambda, \theta_A) \wedge \text{EvTime}(\lambda_i) \geq \text{EvTime}(\lambda_*(\lambda, \theta, C)) \implies \forall \ \theta'_A \in A \mid \theta'_A \neq \theta_A, \theta'' \notin \text{Dsc}(\lambda, \theta'_A)$.

Data derivation

(Com7) Data categories in the associated derivation policy may neither be used to directly derive new data, nor indirectly: $\lambda_i = (\texttt{Derive}, \theta', \theta'', C, \pi, \rho, R, t) \ \wedge \ \theta' \in \text{Dsc}(\lambda, \theta) \wedge \text{EvTime}(\lambda_i) \geq \text{EvTime}(\lambda_*(\lambda, \theta, C)) \implies \theta \notin \pi_*(\lambda, \theta, C).\mathsf{De}$.

Purposes for data use and derivation

(Com8) Only data categories for which use is authorised for a specific purpose appear in `Use` events: $\lambda_i = (\texttt{Use}, \Theta, C, \rho, R, s, e) \ \wedge \ \theta' \in \Theta \ \wedge \ \theta' \in \mathrm{Dsc}(\lambda, \theta) \wedge \mathrm{EvTime}(\lambda_i) \geq \mathrm{EvTime}(\lambda_*(\lambda, \theta, C)) \Longrightarrow (\theta, \rho) \in \pi_*(\lambda, \theta, C).\mathsf{P}_{\mathsf{use}}$.

(Com9) Similarly, data is only derived, directly or indirectly, for purposes authorised by the associated derivation policy: $\lambda_i = (\texttt{Derive}, \theta', \theta'', C, \pi, \rho, R, t) \ \wedge \ \theta' \in \mathrm{Dsc}(\lambda, \theta) \wedge \mathrm{EvTime}(\lambda_i) \geq \mathrm{EvTime}(\lambda_*(\lambda, \theta, C)) \Longrightarrow (\theta, \rho) \in \pi_*(\lambda, \theta, C).\mathsf{P}_{\mathsf{der}}$.

We can now define compliance for an entire log of provenance events:

Definition 12 (Compliance). *A log λ is said to be compliant if all compliance properties* Com1 *...* Com9 *hold for λ.*

To illustrate this notion, we first show in Sec. 4 the applicability of our framework to privacy accountability, and then provide an example in Sec. 5.

4 Applicability to Privacy Accountability

We now discuss under which conditions our framework can be applied to the particular case of personal data protection. We argue that, while provenance can be of great benefit here, special care must be taken due to the sensitive nature of the involved data. Furthermore, it is necessary to combine the framework with a global accountability process, since not all verification aspects can be automated. In the following we use the wording introduced in Sec. 2, making three exceptions. We refer to an entity collecting or processing data as *data controller*, following the usual European terminology. Since personal data is usually assigned to an individual, we call this data owner *data subject*. Furthermore, to reflect the specificities of dealing with personal data, the *usage policies* under consideration are named *privacy policies*.

Motivation The large-scale dissemination of personal data rightfully causes grave concerns. As data subjects are not informed clearly about the processing and distribution of their data, loss of control prevails [22]. The case of outsourced medical data is particularly problematic. It combines highly sensitive categories of personal data with strong data sharing incentives for data controllers such as hospitals, pharmacies, research institutes, and private firms [6]. With an audit-centric approach, provenance can be not only a tool for (non-personal) data processing fault detection, but can also help restoring some clarity to data subjects regarding the whereabouts of their personal data. In this specific context, provenance can support privacy, and more precisely accountability [7].

Policies One specificity of privacy-oriented scenarios is the choice of the applicable policies. Even for a given data processing use case, a one-size-fits-all approach is not possible here. Privacy preferences are a matter of personal choice, since data subjects exhibit varying levels of sensitivity with respect to data protection, leading some privacy frameworks to incorporate different trust models corresponding to typical user profiles [14].

Time In addition, temporal aspects become indispensable in the privacy case. While provenance models do not always include time [23], our framework incorporates this aspect. Once a data subject requests deletion of their data, the applicability of this decision depends on a clear distinction between prior and ulterior events. Note that also other established models, such as PROV-O [21], include temporal aspects.

Logs A common concern about privacy protection through log auditing is that the log itself may become a threat to privacy. One mitigating feature of our framework is the fact that only data categories, not actual values, are logged. This reduces the threat of leaks to metadata. However, even metadata is known to be potentially sensitive [18]. This raises the question of secure log storage for which solutions do exist [5].

Process integration The provenance record constitutes the evidence at the centre of the accountability process, but the panoramic view provided by provenance records goes hand in hand with limited guarantees about their trustworthiness. Since numerous entities are involved, there can be no mapping, realised in a controlled environment, between system events and log items. Adopting an external view as we do, provenance records cannot be guaranteed to match actual system processing. For such a decentralised perspective, inclusion in a wider process is critical and pressure on entities to declare data handling truthfully must come from a different direction. The legal implementation of the principle of accountability, a core principle in the recently adopted European General Data Protection Regulation [15], would help to address this issue. Contributing to correct provenance records is a way for data controllers to be accountable to data subjects.

Processing purposes in the privacy case Another specificity of privacy-oriented use cases is the central importance of stated data processing purposes. Since personal data is involved, processing purposes are actually data handling *finalities*. Declaring such finalities is legally required in the European Union. It is demanded that the purpose designations be in line with actual processing purposes, but the verification of this coupling is not amenable to automation. We assumed the existence of a purpose ontology, but in the case of personal data protection, no convincing real-world and broadly accepted equivalent of such an ontology exists. As a result, there is a realistic risk of data controllers employing purpose designations abusively, stretching commonly accepted meanings to give the appearance of compliance. Data Protection Authorities such as the UK ICO, the Swedish Data Inspection Board, or the French CNIL can play an active role in enforcing a reasonable mapping between processing purposes and involved categories of personal data. Principles such as the legitimacy of processing finalities have already been put forward in this context [2]. More stringent and systematic checks must be enforced for data subjects to fully trust declared processing purposes.

5 Evaluation: A Medical Scenario

To evaluate our formal model, we describe a scenario involving medical data about an individual. Real provenance logs involving multiple components are expected to be much more complex than this simple example.

Personal data from the patient is processed by health professionals from three different organisations on different occasions. The data controllers involved are a hospital `Hospital`, a research institute `ResearchInstitute`, and a pharmacy `Pharmacy`. The following categories of personal data are involved: a patient's full name `ID`, treatment `Treatment`, treatment frequency `Frequency`, status of the treated pathology `Status`, medical history `History`, drug group `Drug`, and risk categorisation `Risk`. Used purpose designations include logistics `Logistic`, statistics `Statistic`, business operations `Business` and marketing `Marketing`. Let π_1 and π_2 be two privacy policies, with values as in Fig. 1 (the process leading to the definition of these values is beyond the scope of this scenario).

$$\begin{cases} \pi_1.\mathsf{D_g} = 3 \text{ months} \\ \pi_1.\mathsf{D_{rf}} = 1 \text{ day} \\ \pi_1.\mathsf{Fw} = (\mathsf{wl}, \mathsf{List}) \\ \mathsf{List} = \{\texttt{Hospital}, \\ \texttt{ResearchInstitute}\} \\ \pi_1.\mathsf{Li} = \{(\texttt{Treatment}, \texttt{Status}), \\ (\texttt{ID}, \texttt{Drug})\} \\ \pi_1.\mathsf{De} = \{\texttt{Frequency}, \texttt{Risk}, \texttt{Drug}\} \\ \pi_1.\mathsf{P_{use}} = \{(\texttt{Treatment}, \texttt{Logistic}), \\ (\texttt{ID}, \texttt{Logistic}), \\ (\texttt{Frequency}, \texttt{Logistic}), \\ (\texttt{History}, \texttt{Logistic}), \\ (\texttt{Status}, \texttt{Logistic})\} \\ \pi_1.\mathsf{P_{der}} = \{(\texttt{History}, \texttt{Statistic})\} \end{cases} \qquad \begin{cases} \pi_2.\mathsf{D_g} = 6 \text{ months} \\ \pi_2.\mathsf{D_{rf}} = 2 \text{ days} \\ \pi_2.\mathsf{Fw} = (\mathsf{wl}, \mathsf{List}) \\ \mathsf{List} = \{\texttt{Hospital}, \\ \texttt{ResearchInstitute}, \\ \texttt{Pharmacy}\} \\ \pi_2.\mathsf{Li} = \{(\texttt{Treatment}, \texttt{Status}), \\ (\texttt{ID}, \texttt{Drug})\} \\ \pi_2.\mathsf{De} = \{\texttt{Frequency}, \texttt{Drug}\} \\ \pi_2.\mathsf{P_{use}} = \{(\texttt{Treatment}, \texttt{Marketing}), \\ (\texttt{Treatment}, \texttt{Logistic}), \\ (\texttt{ID}, \texttt{Logistic}), \\ (\texttt{Frequency}, \texttt{Logistic}), \\ (\texttt{History}, \texttt{Logistic}), \\ (\texttt{Status}, \texttt{Logistic})\} \\ \pi_2.\mathsf{P_{der}} = \{(\texttt{History}, \texttt{Statistic}), \\ (\texttt{ID}, \texttt{Logistic}), \\ (\texttt{Treatment}, \texttt{Business})\} \end{cases}$$

Fig. 1. Example privacy policies π_1 and π_2.

Note that $\pi_1 \geq \pi_2$, but $\neg\pi_2 \geq \pi_1$, i.e. π_1 is strictly stronger than π_2. We now consider the log $\lambda = \lambda_1 \dots \lambda_{15}$ in Fig. 2. Its corresponding provenance graph is depicted in Fig. 3.

λ_1: (Acquire, {Treatment, ID, Status}, Hospital, π_2, {Logistic, Statistic}, 2016-05-01T08:07)
λ_2: (Use, {Treatment, ID}, Hospital, Logistic, Patient registration mandatory, 2016-05-01T10:25, 2016-05-09T17:54)
λ_3: (Export, {Treatment, ID, Status}, Hospital, ResearchInstitute, π_1, {Logistic, Statistic}, 2016-05-12T12:17)
λ_4: (Link, ID, Status, History, ResearchInstitute, π_1, Statistic, Insurance billing requested, 2016-05-14T22:33)
λ_5: (Derive, History, Frequency, ResearchInstitute, π_1, Statistic, Quantitative research, 2016-05-18T09:41)
λ_6: (Use, {Treatment, Frequency}, ResearchInstitute, Logistic, Workflow optimisation, 2016-05-19T14:41, 2016-05-19T15:03)
λ_7: (Link, Frequency, Treatment, Risk, ResearchInstitute, π_1, Statistic, Correlation study, 2016-05-20T12:14)
λ_8: (Export, {Treatment}, Hospital, Pharmacy, π_2, {Logistic}, 2016-05-22T16:37)
λ_9: (Derive, Treatment, Drug, Pharmacy, π_2, Business, Stock estimation, 2016-05-22T23:12)
λ_{10}: (ReqRemove, {ID, Status}, 2016-06-02T18:23)
λ_{11}: (Remove, {ID, Status}, ResearchInstitute, 2016-06-03T10:46)
λ_{12}: (Remove, {ID, Status}, Hospital, 2016-06-03T11:17)
λ_{13}: (Remove, {Treatment}, Hospital, 2016-06-17T15:40)
λ_{14}: (Remove, {Treatment, History, Frequency, Risk}, ResearchInstitute, 2016-07-02T08:35)
λ_{15}: (Remove, {Treatment}, Pharmacy, 2016-07-25T04:32)

Fig. 2. The provenance log $\lambda = \lambda_1 \dots \lambda_{15}$ for our example.

This log is correct, since it is straightforward to verify that Cor1 ... Cor12 are all respected. However, log λ is not compliant with respect to the involved privacy policies. Indeed, the Link event in λ_7 contradicts Com6. The latest policy-defining event, as in Def. 10, for the data category Risk for the data controller ResearchInstitute is $\lambda_*(\lambda,$ Risk, ResearchInstitute$) = \lambda_7$ = (Link, Frequency, Treatment, Risk, ResearchInstitute, π_1, Statistic, Correlation study, 2016-05-20T12:14). As a consequence, the associated privacy policy is $\pi_*(\lambda,$ Risk, ResearchInstitute$) = \pi_1$. Now recall Def. 8, and notice that Dsc(λ, Treatment) = {Treatment, Risk, Drug} because of the linking in λ_7 and the derivation in λ_9. On the other hand, Dsc(λ, Status) = {Status, History, Frequency, Risk} because of the linking in λ_4, the derivation in λ_5 and the second linking in λ_7. We have Frequency $\in$ Dsc(λ, Status), and Frequency is the first data category being linked by λ_7. Furthermore, Treatment $\in$ Dsc(λ, Treatment) and Treatment is the second data category being linked by this same event.

Since (Treatment, Status) $\in \pi_1$.Li $= \pi_*(\lambda,$ Risk, ResearchInstitute$)$, Com6 is violated and global compliance does not hold as a consequence (Def. 12).

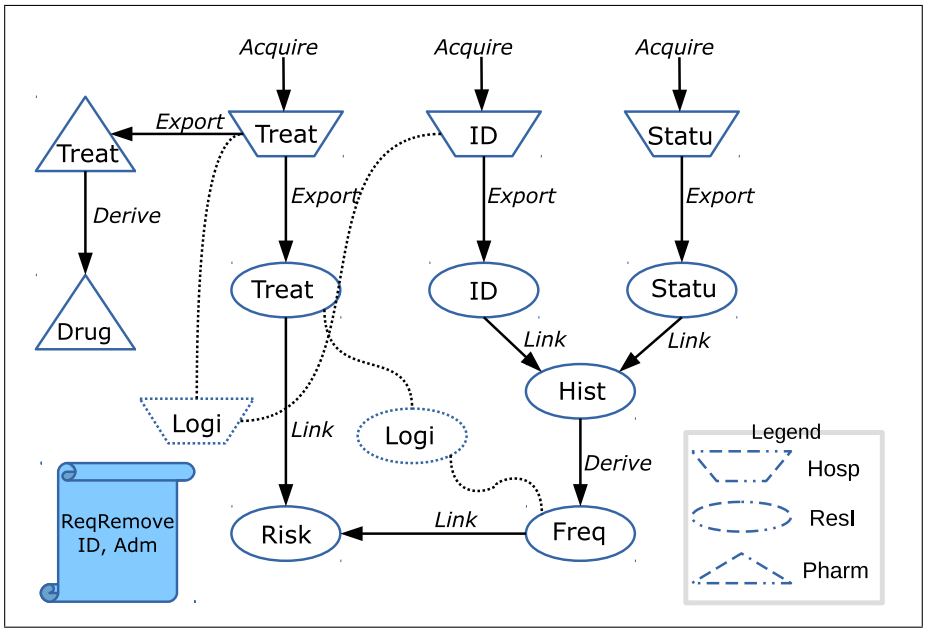

Fig. 3. A provenance record graph depicting the log from Fig. 2. Timestamps and `Remove` events have been omitted for clarity. Different shapes symbolise different data controllers, as shown in the legend. Dotted lines represent `Use` events. Some of the used identifiers are abbreviations.

6 Related Work

The W3C specifies constraints [11] for the PROV data model [24], both for correctness and compliance. Since the PROV data model strives for maximal generality, it is more suited to the semantic web than to the scenario of data processing within an organisation. A high-level discussion (without concrete modelling) of the use of provenance records to support accountability via privacy policies can be found in work by Gil and Fritz [17]. Aldeco-Pérez and Moreau zoom into the specific case of personal data provenance. They introduce a compliance framework for provenance [1], sticking to the usual graph modelling, but only consider simple privacy policies that do not account for derivation, linking, or temporal aspects. Their approach is exemplified by an online shopping scenario and includes a partial implementation in SPARQL. Data handling logs are analysed with respect to privacy policies by Butin and Le Métayer [9], outside of the scope of provenance. Their approach is based on records generated from actual system events, providing stronger assurance with respect to the trustworthiness of logs, but simultaneously limiting the analysis to the evidence made available by a single data controller and losing the panoramic view (encompassing several data controllers) provided by provenance records. Their modelling of purposes is less fine-grained than in our framework, since they do not support different

acceptable purposes for different categories of data. Chong [12] discusses the security of provenance itself, providing semantics to restrict which (potentially sensitive) provenance information is available for consultation. A related issue, the problem of provenance records possibly leaking sensitive or proprietary module information when used in scientific workflows, is tackled by Davidson et al. [13]. Tharaud et al. [31] describe the implementation of a usage control system using provenance in the context of electronic health records, but do not describe policy formats or compliance semantics.

7 Conclusion

We have introduced a formal correctness and compliance framework for provenance, based on a linear view of provenance events and the definition of sticky usage policies. The presented format for policies and rules for correctness and compliance are not meant to be exhaustive or applicable to all scenarios. However, our framework serves as a basis allowing to include other policy components and event types. For instance, authorised purposes can be defined for other event types as well, such as data linking; and more fine-grained policy components can be considered e.g. for data forwarding or derivation. As an example, one could restrict the target data categories of data derivation instead of the source categories. Therefore, the choices presented here in terms of policy components, event types, and rules should only be seen as an instantiation of the framework.

We discussed the possibility of applying our approach to the special case of personal data processing, and pointed out the necessity of combining the formal framework with a global accountability approach involving pressure from data protection authorities and new legal tools. A scenario involving the processing of medical data by different, communicating data controllers was used to exemplify the framework. The use of such a compliance framework is only meaningful within a wider accountability process, since provenance records cannot be intrinsically trustworthy. In combination with such an accountability process it can however contribute to increase transparency about personal data handling, ultimately benefiting data subjects.

Future work In privacy-oriented scenarios, while unconditionally enforcing the global correctness of provenance records seems out of reach, a certification-based approach in coordination with data protection authorities could help improve the accuracy of records provided by data controllers, ultimately leading to differentiated levels of assurance for data handling evidence. Taking into account varying levels of data quality in a provenance-based compliance framework could lead to a more granular analysis of global correctness and compliance. Another open question is how the global provenance record is aggregated securely from the different involved components when they are not all trusted. It would also be interesting to estimate the complexity of correctness and compliance checking. Finally, modelling correctness and compliance rules in a theorem prover like Isabelle/HOL [26] would provide additional guarantees in term of overall consistency.

Acknowledgments

This work has been co-funded by the DFG as part of project "Long-Term Secure Archiving" within the CRC 1119 CROSSING. In addition, it has received funding from the European Union's Horizon 2020 research and innovation program under Grant Agreement No 644962. The authors thank Fanny Coudert for insights about purpose ontologies.

References

1. Aldeco-Pérez, R., Moreau, L.: A Provenance-Based Compliance Framework. In: Berre, A., Gómez-Pérez, A., Tutschku, K., Fensel, D. (eds.) Future Internet — FIS 2010 — Third Future Internet Symposium. Proceedings. Lecture Notes in Computer Science, vol. 6369, pp. 128–137. Springer (2010)
2. Article 29 Data Protection Working Party: Opinion 8/2001 on the processing of personal data in the employment context (2001), http://ec.europa.eu/justice/data-protection/article-29/documentation/opinion-recommendation/files/2001/wp48_en.pdf
3. Article 29 Data Protection Working Party: Opinion 3/2010 on the principle of accountability (2010), http://ec.europa.eu/justice/policies/privacy/docs/wpdocs/2010/wp173_en.pdf
4. Article 29 Data Protection Working Party: Advice paper on essential elements of a definition and a provision on profiling within the EU General Data Protection Regulation (2013), http://ec.europa.eu/justice/data-protection/article-29/documentation/other-document/files/2013/20130513_advice-paper-on-profiling_en.pdf
5. Bellare, M., Yee, B.S.: Forward Integrity for Secure Audit Logs. Tech. rep., University of California at San Diego (1997)
6. Bertino, E., Ooi, B.C., Yang, Y., Deng, R.H.: Privacy and Ownership Preserving of Outsourced Medical Data. In: Aberer, K., Franklin, M.J., Nishio, S. (eds.) Proceedings of the 21st International Conference on Data Engineering, ICDE 2005. pp. 521–532. IEEE Computer Society (2005)
7. Bier, C.: How Usage Control and Provenance Tracking Get Together — A Data Protection Perspective. In: IEEE Symposium on Security and Privacy Workshops. pp. 13–17. IEEE Computer Society (2013)
8. Butin, D., Chicote, M., Le Métayer, D.: Log Design for Accountability. In: 2013 IEEE Security & Privacy Workshop on Data Usage Management. pp. 1–7. IEEE Computer Society (2013)
9. Butin, D., Le Métayer, D.: Log Analysis for Data Protection Accountability. In: Jones, C.B., Pihlajasaari, P., Sun, J. (eds.) FM 2014: Formal Methods — 19th International Symposium. Proceedings. Lecture Notes in Computer Science, vol. 8442, pp. 163–178. Springer (2014)
10. Cheney, J.: A Formal Framework for Provenance Security. In: Proceedings of the 24th IEEE Computer Security Foundations Symposium, CSF 2011. pp. 281–293. IEEE Computer Society (2011)
11. Cheney, J., Missier, P., Moreau, L.: Constraints of the PROV Data Model. Tech. rep., W3C (2013), https://www.w3.org/TR/prov-constraints/

12. Chong, S.: Towards Semantics for Provenance Security. In: Cheney, J. (ed.) First Workshop on the Theory and Practice of Provenance, TaPP'09, Proceedings. USENIX (2009)
13. Davidson, S.B., Khanna, S., Roy, S., Stoyanovich, J., Tannen, V., Chen, Y.: On provenance and privacy. In: Milo, T. (ed.) Database Theory — ICDT 2011, 14th International Conference, Proceedings. pp. 3–10. ACM (2011)
14. Decroix, K.: Model-Based Analysis of Privacy in Electronic Services. Ph.D. thesis, KU Leuven, Faculty of Engineering Science (2015)
15. European Commission: Regulation (EU) 2016/679 of the European Parliament and of the Council of 27 April 2016 on the protection of natural persons with regard to the processing of personal data and on the free movement of such data, and repealing Directive 95/46/EC (General Data Protection Regulation). Official Journal of the European Union 59 (2016), http://eur-lex.europa.eu/eli/reg/2016/679/oj
16. Foster, I.T., Vöckler, J., Wilde, M., Zhao, Y.: The Virtual Data Grid: A New Model and Architecture for Data-Intensive Collaboration. In: First Biennial Conference on Innovative Data Systems Research (CIDR) (2003)
17. Gil, Y., Fritz, C.: Reasoning about the Appropriate Use of Private Data through Computational Workflows. In: Intelligent Information Privacy Management, Papers from the 2010 AAAI Spring Symposium, Technical Report SS-10-05. AAAI (2010)
18. Greschbach, B., Kreitz, G., Buchegger, S.: The devil is in the metadata — New privacy challenges in Decentralised Online Social Networks. In: Tenth Annual IEEE International Conference on Pervasive Computing and Communications, PerCom 2012, Workshop Proceedings. pp. 333–339. IEEE Computer Society (2012)
19. Hartig, O.: Provenance Information in the Web of Data. In: Bizer, C., Heath, T., Berners-Lee, T., Idehen, K. (eds.) Proceedings of the WWW2009 Workshop on Linked Data on the Web, LDOW 2009. CEUR Workshop Proceedings, vol. 538. CEUR-WS.org (2009), http://ceur-ws.org/Vol-538/ldow2009_paper18.pdf
20. Kumaraguru, P., Lobo, J., Cranor, L.F., Calo, S.B.: A Survey of Privacy Policy Languages. In: Workshop on Usable IT Security Management (USM 07): Proceedings of the 3rd Symposium on Usable Privacy and Security. ACM (2007)
21. Lebo, T., Sahoo, S., McGuinness, D.: PROV-O: The PROV Ontology. Tech. rep., W3C (2013), https://www.w3.org/TR/prov-o/
22. Madden, M., Rainie, L., Zickuhr, K., Duggan, M., Smith, A.: Public Perceptions of Privacy and Security in the Post-Snowden Era. Pew Research Center (2014), http://www.pewinternet.org/2014/11/12/public-privacy-perceptions/
23. Moreau, L., Clifford, B., Freire, J., Futrelle, J., Gil, Y., Groth, P., Kwasnikowska, N., Miles, S., Missier, P., Myers, J., Plale, B., Simmhan, Y., Stephan, E., den Bussche, J.V.: The Open Provenance Model Core Specification (V1.1). Future Gener. Comput. Syst. 27(6), 743–756 (2011)
24. Moreau, L., Missier, P.: PROV-DM: The PROV Data Model. Tech. rep., W3C (2013), https://www.w3.org/TR/prov-dm/
25. Okkalioglu, B.D., Okkalioglu, M., Koç, M., Polat, H.: A survey: deriving private information from perturbed data. Artif. Intell. Rev. 44(4), 547–569 (2015)
26. Paulson, L.C.: Isabelle — A Generic Theorem Prover, Lecture Notes in Computer Science, vol. 828. Springer (1994)
27. Pearson, S., Mont, M.C.: Sticky Policies: An Approach for Managing Privacy across Multiple Parties. IEEE Computer 44(9), 60–68 (2011)
28. Proctor, R.W., Ali, M.A., Vu, K.P.L.: Examining Usability of Web Privacy Policies. International Journal of Human-Computer Interaction 24(3), 307–328 (2008)

29. Ram, S., Liu, J.: A New Perspective on Semantics of Data Provenance. In: Freire, J., Missier, P., Sahoo, S.S. (eds.) Proceedings of the First International Workshop on the role of Semantic Web in Provenance Management (SWPM 2009). CEUR Workshop Proceedings, vol. 526. CEUR-WS.org (2009), http://ceur-ws.org/Vol-526/InvitedPaper_1.pdf
30. Sultana, S., Bertino, E.: A Comprehensive Model for Provenance. In: Groth, P.T., Frew, J. (eds.) Provenance and Annotation of Data and Processes — 4th International Provenance and Annotation Workshop, IPAW 2012. Lecture Notes in Computer Science, vol. 7525, pp. 243–245. Springer (2012)
31. Tharaud, J., Wohlgemuth, S., Echizen, I., Sonehara, N., Müller, G., Lafourcade, P.: Privacy by data provenance with digital watermarking — A proof-of-concept implementation for medical services with electronic health records. In: Echizen, I., Pan, J., Fellner, D.W., Nouak, A., Kuijper, A., Jain, L.C. (eds.) Sixth International Conference on Intelligent Information Hiding and Multimedia Signal Processing (IIH-MSP 2010), Proceedings. pp. 510–513. IEEE Computer Society (2010)
32. Trabelsi, S., Njeh, A., Bussard, L., Neven, G.: PPL Engine: A Symmetric Architecture for Privacy Policy Handling. W3C Workshop on Privacy and data usage control (2010)