Complexities of Identity Provenance Metadata

Radovan Semančík[1]

Abstract: Data provenance information is an important part of personal data protection mechanisms. However, capabilities of existing identity management systems are severely limited when it comes to maintaining and processing data provenance information. This paper describes an effort to design and implement capability to process provenance information in midPoint, an open source identity management and governance system. The solution used value metadata for the purposes of storage and processing of provenance information. Resulting prototype was fully integrated into midPoint code base. The solution dealt with all layers of provenance information processing, from data acquisition to user interface. The prototype uncovered a relation between provenance information and other metadata types, as well as potential use of provenance-enriched metadata in conjunction with data protection mechanisms.

Keywords: Identity management, Data provenance, Data modeling, Metadata, Personal data protection

1 Introduction

We live in a connected world, where information flows from system to system. This is certainly true for identity information, that moves around the world even more than we would like to admit. Processing of the data does no longer depend only on the content of the data. *Provenance* (origin) [Gy10] of the data has to be considered as well. Provenance of the data may be needed to evaluate trustworthiness of the data, considering aspects such as trust in the source system, declared level of assurance and similar metadata. However, tracking data provenance is crucial for data protection mechanisms as well. Provenance metadata are needed to prove that the data came from a legitimate source and that the data are up-to-date. Data provenance is also closely related to concept of *basis for data processing* [GD16], making provenance a crucial element in evaluating whether we can process particular data at all.

However, there are hidden complexities in processing provenance metadata, complexities that seem to extend to processing of all metadata. The complexities were discovered during work on *Data Provenance Prototype* [MP20a] for midPoint, under the umbrella of midPrivacy initiative [MP21a]. MidPoint [MP21b] is a comprehensive open source identity management and governance platform. The *Data Provenance Prototype* project focused on maintenance of provenance metadata for identity information for data

[1] Evolveum s.r.o., Vendelínska 109, Lozorno, 90055, Slovakia, radovan.semancik@evolveum.com,
https://orcid.org/0000-0002-4903-1436

protection purposes, and its prototype implementation in midPoint. Goal of the project was to design a data provenance schema (as metadata schema), and to apply the concept to identity data managed by midPoint.

Due to the limited capabilities of existing identity and access management (IAM) protocols, the goal of the project was focused on "local" provenance information. Existing IAM protocols are currently not capable to convey full, end-to-end provenance information, tracking information lineage from its ultimate source. Therefore, the project was limited to considered provenance from the point of view of a single organization. The goal was to track information provenance from the "previous hop" only, the last external system that provided the information to our organization. Intended result of the project was a practical prototype, maintaining limited provenance information as metadata. Even such a limited provenance information, is still useful for maintaining appropriate data protection mechanisms, which is an ultimate goal of midPrivacy initiative. However, maintaining even such a relatively simple "local" provenance information has proven to be a challenge.

2 Metadata Multiplicity Problem

Simply speaking, MidPoint's responsibility is to move identity data between systems. MidPoint can feed identity data from source systems, correlate them, transform them to a common data model, apply policies (such as role-based access control, RBAC), transform data to foreign data models and provision the data to external systems by using a generic connector framework. This is an essential identity management functionality, needed by almost every organization. As midPoint is already in the middle of identity data interchange in the organization, it is a natural point to control policies, including data protection policy. Implementing automated data protection mechanism has been an ambition of midPoint development team for many years. However, there is necessary work on technological foundations, before full data protection mechanisms can be implemented.

MidPoint is a schema-based system, internal data model is based on formal data model definition. The data model can be expressed in XML, JSON and YAML. Data examples in this paper are based on JSON notation based on simplified midPoint schema. For clarity, some examples are shortened to JSON snippets.

Below is an example of pure JSON data, describing a person:

```
{
  "givenName" : "Lawrence",
  "fullName" : "Lawrence Long"
}
```

This data structure can be enriched with metadata using a special notation with @

character:

```
{
  "givenName" : {
    "@value" : "Lawrence",
    "@metadata" : {
      ...
    }
  },
  "fullName" : {
    "@value" : "Lawrence Long",
    "@metadata" : {
      ...
    }
  }
}
```

A naïve design of a data provenance solution would use a simple metadata structure specifying where particular value came from:

```
"fullName" : {
  "@value" : "Lawrence Long",
  "@metadata" : {
    "provenance" : "user entry"
  }
}
```

Obviously, any single value of a data item may come from several sources. In the above example of user's full name, it may be entered manually by the user during registration process. Then the same value may come from the human resource (HR) system, or it may be retrieved from an identity provider in an identity federation. The provenance value must not be simply overwritten when the new information comes, e.g. the `user entry` value must not be overwritten with `HR` value when the data are correlated to the HR system. In case of the overwrite we would lose information that the user entered manually, which may be a problem in case that user's employment contract ends and the HR information is removed. In such case we may still want to retain user's name, as it was entered manually outside of the employment context. Therefore we have come to a conclusion that the provenance information has to be multi-valued:

```
"fullName" : {
  "@value" : "Lawrence Long",
  "@metadata" : {
    "provenance" : [ "user entry", "HR" ]
  }
}
```

However, even this solution has proven to be unsatisfactory. The value may be a result combining inputs from several information sources. For example, this particular person may want to be called "Larry" instead of "Lawrence", therefore he changes his given name in his user profile. The system then automatically computes full name by using the user-entered given name and surname that originated in the HR system. Considering this possibility, the provenance metadata become a complex data structure. For simplicity, we will denote this combination using a simple plus sign (+), that is a placeholder for complex data structure used in a real-world system. Naturally, the provenance metadata still need to be multi-valued. The following example illustrates metadata for a value which is a combination of two sources, and at the same time the complete value was obtained from the identity provider acting on behalf of "The Institute":

```
"fullName" : {
  "@value" : "Larry Long",
  "@metadata" : {
    "provenance" : [ "user entry + HR", "The Institute" ]
  }
}
```

The example above provides simplified data structure, the real metadata schemas used in the data provenance prototype project are considerably more complex. Modeling of metadata structures has proven to be a challenge of its own, motivating the design of Axiom [Ax20], a new data modeling language.

However, even this solution can be further improved. Aside from provenance metadata, there are other metadata types. E.g. there are *storage metadata* that describe when was the value stored, when it was updated and so on. There are *assurance metadata* that describe level of assurance or trust in the particular value. There are *transformation metadata*, *process metadata*, *policy-related metadata* and other metadata types. We have found that all the metadata which we need to process in midPoint are tightly bound to a particular provenance. For example, any particular value may have high assurance when it originates in HR system, as such information comes from an employment contract that is checked against identity documents. However, if the same value is entered by the user in a registration form, the level of assurance is likely to be low. It would be incorrect to store only the information about high assurance level, as that may be incorrectly applied to the low-assurance data entered by the user, e.g. in case that the HR record is removed. Therefore both assurance values have to be stored, each of them tightly bound to the provenance metadata. This thinking has been applied in the design of *yield* concept (see below).

Detailed description of this "metadata multiplicity problem" is beyond the scope of this paper. The explanation with detailed examples can be found on the project website [MP20b].

Although we have conducted research of available literature at the beginning of the project, we have not found any mention of this problem in a literature related to identity management or management of metadata. The fact that *provenance* information (and hence the metadata) have a multi-valued character comes as a natural consequence of provenance character [SPG05]. However, the relation of provenance metadata to all the other metadata types was not obvious (e.g. [MD20]). Therefore, unaware of the problem, we have started the project with a simplistic provenance metadata model, which caused significant re-design and re-engineering later in the project. We find this lack of problem coverage surprising, and it was a partial motivation to author this paper.

3 Yield

The design decision in the project was to make the entire metadata structure multi-valued. There is one complete value for all the metadata structures for each individual data source. In midPoint parlance, we use term *yield* to refer to each such source. *Yield* may correspond to external data source, such as HR database, remote identity provider or manual user entry. Yet, *yield* may also represent an internal source, such as mapping that combines data from several sources or a value generator.

Following example shows `fullName` user property with two yields in a very simplified form:

```
"fullName" : {
  "@value" : "Larry Long",
  "@metadata" : [
    {
      "provenance" : {
        ... data are combination of HR and user entry ...
      }
    },
    {
      "provenance": {
        ... data came from The Institute ...
      }
    }
  ]
},
```

Each *yield* may contain a complex data structure that describes fine details about the value. For example, following data structure may provide details how and when particular data were transformed:

```
"fullName" : {
  "@value" : "Larry Long",
  "@metadata" : [
    {
      "provenance" : {
        ... data are combination of HR and user entry ...
      },
      "transformation" : {
        ... data produced by a dynamic expression in
midPoint ...
      }
    },
    {
      "provenance": {
        ... data came from The Institute ...
      },
      "transformation" : {
        ... data copied directly from the source ...
      }
    }
  ]
},
```

While *yield* may contain a lot of metadata structures that describe exhaustive details about data, *provenance metadata* still have a prime position among all other details. Provenance metadata work as an identifier, allowing the processor to identify the correct *yield* to work with. E.g., user interface can use provenance metadata to identify the *yield* that describes metadata related to manual user entry, making sure the correct metadata are used or updated. Of course, provenance metadata still describe the origin of data, yet this purpose is almost secondary in this case. For this approach to work, the provenance metadata have to describe data origin on a conceptual level, without excessive details, to allow algorithmic comparison of data provenance information. For example, midPoint is using a simple data structure called *acquisition* to represent provenance. The *acquisition* data structure contains a reference to *origin*, which is one of objects in internal midPoint database. This *origin* works as a conceptual representation of data source, such as external system, organization or even purely abstract concept as "user entry". The *origin* has two purposes. Firstly, it identifies the logical origin of the data, for the purposes of machine-processing of the metadata. The origin identifies a particular *yield*. Secondly, the *origin* can be used to display the source of data in user-friendly way. This design is illustrated in the following example.

```
"fullName" : {
  "@value" : "Larry Long",
  "@metadata" : [
    {
      "provenance" : {
        "acquisition" : [
          {
            "timestamp" : "2020-06-22T10:52:03Z",
            "originRef" : {
              ... reference to "User entry" origin ...
              .... indicates data entered by user ...
            }
          },
          {
            "timestamp" : "2020-03-06T23:05:42Z",
            "originRef" : {
              ... reference to "HR" origin ...
              ... indicates data taken from HR Database
...
            }
          }
        ]
      }
    },
    {
      "provenance": {
        "acquisition" : [
          {
            "timestamp" : "2020-08-17T14:45:12Z",
            "originRef" : {
              ... reference to "The Institute" origin ...
            }
          }
        ]
      }
    }
  ]
},
```

The example describes situation illustrated in the previous section. The value "Larry Long" comes from two sources. First source is a combination of data entered by the user with HR data. Second source is "The Institute" organization.

Conceptualization of the `provenance` structure does not leave place for excessive details. Additional details can be stored in other metadata structures, such as `transformation` or `storage` that can be placed at the same level as `provenance` structure. Such separation has additional benefit of separating metadata schemas into encapsulated data structures.

```
"fullName" : {
  "@value" : "Larry Long",
  "@metadata" : [
    {
```

```
  "provenance" : {
    "acquisition" : [
      {
        "timestamp" : "2020-06-22T10:52:03Z",
        "originRef" : {
          ... reference to "User entry" origin ...
          .... indicates data entered by user ...
        }
      },
      {
        "timestamp" : "2020-03-06T23:05:42Z",
        "originRef" : {
          ... reference to "HR" origin ...
          ... indicates data taken from
              HR Database ...
        }
      }
    ]
  },
  "storage" : {
    "createTimestamp" : "2020-06-22T10:52:03Z",
    "modifyTimestamp" : "2020-03-06T23:05:42Z",
    ...
  },
  "transformation" : {
    ... detailed description of data mappings ...
  }
},
{
  "provenance": {
    "acquisition" : [
      {
        "timestamp" : "2020-08-17T14:45:12Z",
        "originRef" : {
          ... reference to "The Institute" origin ...
        }
      }
    ]
  },
  "storage" : {
    ...
  },
  "transformation" : {
    ...
  }
```

```
      }
    ]
  },
```

This design was prototyped in midPoint, implemented and integrated in midPoint codebase and released in midPoint 4.2. The prototype reached all layers of the system, from storage to presentation. The prototype was a success, although some challenges and open questions remain.

3.1 Variations, Challenges and Further Development

The multi-valued metadata and the concept of *yield* worked very well in midPoint environment. Yet, it is still an open question whether these concepts can work for other systems. E.g. some systems may need single-valued ("global") metadata structures in addition to *yields*, even though we have not identified such need in midPoint. We have also considered an alternative approach that does not require multi-valued metadata. However, in this case the system must allow to repeat the same value of a data item several times. This means that all data items in the system must be (technically) multi-valued, it complicates operations with data, presentation and it is likely to confuse the developers. Even though we have rejected this approach in midPoint, it may still be a feasible solution for other systems.

Comprehensive implementation of data provenance requires non-trivial data structures in the metadata. Modeling of metadata schema was one of the early challenges in the project. However, this challenge was expected. It was addressed by designing Axiom [Ax20], a new data modeling language capable of metadata schema definition. We have decided to design a new data modeling language as our research of data modeling languages revealed no existing solution. The midPoint project had been using W3C XML Schema Definition (XSD) [XSD04] as a data modeling language for almost a decade, gaining considerable experience, especially regarding the limitation of traditional data modeling languages. Although we believe that traditional data modeling languages could be used to model metadata, design and maintenance of such solution is likely to be very difficult. This lead to the design and implementation of Axiom, which was an enabler for rich metadata schemas. In hindsight, Axiom was a major factor contributing to the success of data provenance prototype.

Moreover, the project focused on local provenance metadata. Global, end-to-end provenance metadata are likely to be significantly more complex. This was one of the reasons of investing into a modeling language that can support efficient evolution of metadata schemas.

The complexity of data provenance makes it a major challenge to present the provenance information to users. Our project was focused on presenting the information to system administrators, who are expected to handle some levels of complexity. However, even that has proven to be a challenge. The metadata presentation components went through several design iterations during the project. Even though the final result is acceptable, it is still not ideal. Presenting provenance information to common users is likely to a major user experience challenge.

Data provenance information has a value on its own. However, the full value of the provenance is realized when it is combined with data protection concepts. Data provenance is related to the concept of *basis for data processing* (see Art. 6 of GDPR [GD16]), which is central to the data protection mechanisms. Personal data can be processed only if there is a valid *basis* for their processing. Data for which there is no valid *basis* must be erased. Processing of *bases* such as employment or study may seem straightforward. However, in reality various *bases* for data processing overlap. For example, a person may be both employee and student of the same university, or a contractor and customer of the same company. This complexity is often addressed by the use of *personas*, e.g. segregating student and employee data into separate user accounts. While this approach can be very efficient in some cases, it may also be confusing and uncomfortable for users in other cases. Obviously, there is a value in a system that can properly track many overlapping *bases for data processing* on a single persona.

The concepts of data *origin* (provenance) and *basis for processing* are related, however, they are not equivalent. for example, employee data may originate from the HR system, but they may also be entered by an administrator in emergency situations (e.g. outages). HR data may be manually corrected by the user. Those are three different origins of the data, but we are processing the data on the same basis (employment).

The exact role of the *origin* in the data protection mechanisms is not clear yet. It is clear the concept of *origin* is helpful in demonstrating the data were acquired by lawful means (accountability). However, the *origin* itself is not an entitlement for data processing. Therefore, it is questionable whether *origin* plays an essential role in data erasure, or whether the erasure can be determined using only the *basis*. Such questions are an inspiration for further research.

3.2 Conclusion

Data provenance prototype provided valuable insights into practical aspects of identity provenance. The prototype was implemented in midPoint - an established open source identity management system. Therefore the prototype has to fit into an existing ecosystem, adapt to practical limitations of real-world systems. Many of the limitations are arguably given by the design choices made when existing systems were created. However, as many existing systems are built in a similar fashion as midPoint, any identity provenance solution is likely to encounter similar challenges.

Several important lessons were learned during the development and testing of the prototype. Perhaps the most unexpected lesson was the multi-valued nature of metadata. The multi-valued nature of provenance information was no big surprise. However, the fact that this multi-valued character seems to apply to all the metadata was unexpected, and it caused significant rework during the prototype. It is still an open question whether this is an inherent characteristic of the metadata or whether it is merely a design choice. Either way, there seems to be a major engineering advantage to model metadata in multi-valued fashion, identified by provenance information.

Complexity of data provenance information, and the information associated with it was another major challenge. Even though this challenge was expected, it took significant effort to address it. Existing technology does not provide adequate tools to deal with complex metadata structures. Therefore the solution included design and prototype implementation of a completely new data modeling language: Axiom [Ax20]. Axiom has a native support for influencing the underlying concepts of the data model, dubbed *inframodel* by Axiom authors. New features of Axiom were used to model the provenance metadata, together with other metadata types. The prototype demonstrated that this is a feasible and efficient method to deal with the complexity of metadata structures.

Complexity of the provenance information poses a major challenge to presentation of data provenance to the users. The prototype included a user interface for presentation of provenance information to system administrators. Even though system administrators can deal with complex information, the design of user interface required several iterations. The result is acceptable for the purposes of the prototype. Yet, there is still a significant room for improvement even in the administration user interface. Designing a user-friendly and intuitive user interface for common users is likely to be much more challenging.

The prototype suggests that there may be a close relationship between data, metadata and data protection mechanisms. Most notably, it looks like data protection mechanisms need to use metadata to properly manage data values, especially with regard to *basis for data processing*. It is not clear what is the exact role of data provenance information (beyond its use for accountability). However, it is clear that similar mechanisms that are used to manage provenance information in the metadata can be used to store data protection information.

Overall, the project of data provenance prototype was an engineering project, focused on solving practical problems rather than theoretical ones. Despite that, there were interesting challenges that are worth sharing with the broader community.

Even though the resulting code is just a prototype, it has interesting potential for the future, especially in the area of data protection. The implemented mechanism allows maintenance of metadata for every value of every data item in the system, maintaining separate metadata for every *origin* of that value. E.g. the system can track separate data protection information for a full name value that was entered by the user, and the same value that originated in the human resource system. Similarly, the system can track metadata for each value of multi-valued property, such as data about user's affiliations. This ability is likely to be an essential enabler in fine-grained tracking and management of *basis for data processing*, as given by European data protection regulations. This seems to be a very promising avenue for future research and development.

3.3 Acknowledgements

Author would like to thank all the participants of the project, Katarína Bolemant, Pavol Mederly, Anton Tkáčik, Slávek Licehammer and Igor Farinič. The work of project participants provided an essential material for this paper. We would also like to thank NGI_TRUST for proving funding for this project, and we would especially like to thank project mentors, Maite Alvarez and Collin Wallis, for their help and encouragement. We would like to express our thanks to open source community and especially all the contributors to midPoint project.

This project has received funding from the European Union's Horizon 2020 research and innovation programme under the NGI_TRUST grant agreement no 825618.

Bibliography

[Ax20] Axiom, https://docs.evolveum.com/midpoint/axiom/, Accessed 27/01/2021.

[GD16] 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). 2016.

[Gy10] Gil, Y. et.al.: Provenance XG Final Report, W3C Incubator Group Report. https://www.w3.org/2005/Incubator/prov/XGR-prov-20101214/, 2010.

[MD20] Metadata 2020, http://www.metadata2020.org/, Accessed 27/01/2021.

[MP20a] MidPrivacy: Data Provenance Prototype, https://docs.evolveum.com/midpoint/midprivacy/phases/01-data-provenance-prototype/, Accessed 27/01/2021

[MP20b] Metadata Multiplicity Problem, https://docs.evolveum.com/midpoint/midprivacy/phases/01-data-provenance-prototype/metadata-multiplicity-problem/, Accessed 27/01/2021.

[MP21a] MidPrivacy, https://docs.evolveum.com/midpoint/midprivacy/, Accessed 27/01/2021.

[MP21b] MidPoint, https://midpoint.evolveum.com/, Accessed 27/01/2021.

[SPG05] Simmhan Y.L., Plale B., Gannon D.: A Survey of Data Provenance Techniques. Technical Report TR618, Department of Computer Science, Indiana University, 2005.

[XSD04] XML Schema Part 1: Structures Second Edition, https://www.w3.org/TR/xmlschema-1/, Accessed 27/01/2021.