Towards an Architecture for Policy-Aware Decentral Dataset Exchange

Sebastian Neumaier
St. Pölten University of Applied Sciences
St. Pölten, Austria
orcid.org/0000-0002-9804-4882
email: sebastian.neumaier@fhstp.ac.at

Giray Havur
Siemens AG Österreich
Vienna, Austria
orcid.org/0000-0002-6898-6166
email: giray.havur@siemens.com

Tassilo Pellegrini
St. Pölten University of Applied Sciences
St. Pölten, Austria
orcid.org/0000-0002-0795-0661
email: tassilo.pellegrini@fhstp.ac.at

***Abstract*—In the production of digital artefacts, components, such as software libraries, datasets, data streams, and content items are typically provided and used under various policies, such as licenses, terms of trade, or disclaimers. Ensuring policy compliance is a mandatory requirement for legally secure commercialization. However, manual clearance of rights is time-consuming, costly, and error-prone, especially when multiple stakeholders and contractual dependencies are involved. In this *position paper* we present an architecture for a trusted exchange in a shared data ecosystem. This includes the modelling of transparent, interoperable, and customizable data sharing policies; methods for collection and monitoring of metadata against the respective policies; and the automated validation and compliance checking of the modelled policies in a secure and trusted environment.**

***Keywords*—multi-lateral data sharing, policy-aware systems, policy languages**

I. Introduction

New data-sharing practices stimulated by phenomena like open data, open innovation, and crowdsourcing initiatives as well as the increasing interconnectivity of services, sensors, and (cyber physical) systems have nurtured an environment, in which the effective handling of policies has become key to legally secure innovation, productivity and value creation. Herein, policies shall be understood as a documented set of guidelines for ensuring the accountable management and intended usage of information. Policy-compliant data sharing becomes especially challenging when multiple stakeholders are involved. From the *user's perspective*, general problems associated with policy compliance are: (1) a massive information overload and high efforts/costs in acquiring and understanding the service provider's policy; (2) a lack of interoperability between policies due to device, application and service dependent frameworks; (3) a loss of transparency and control over data; and (4) a loss of trust into the data provider. From the *data provider's perspective*, problems associated with policy management are: (5) high efforts in ensuring legal compliance and accountability as conforming with regulations; (6) missed opportunities to use data usage preferences for service and business model innovation; and (7) missed opportunities to use the user's data sensitivity for service improvements and customer relationship management.

To tackle the problems (1-7), we aim to develop a decentralized, trustable policy negotiation framework which enables transparent, flexible and legally compliant creation and processing of data usage policies in a service ecosystem.

In Section II, we argue for the necessity of various policy types to facilitate data exchange. In Section III, we identify key challenges of policy-aware data exchange. In Section IV, we introduce three policy types (cf. Section IV-A) processed by our envisioned architecture model (cf. Section IV-B). In Section V, we provide the related work. In Section VI, we conclude with an outlook on the next research steps.

II. Policy Representation and Policy-Types

Rights Expression Languages (RELs) are a subset of Digital Rights Management technologies that are used to explicate machine-readable policies for the purpose of automated Digital Asset Management. Recent research conducted on the genealogy of RELs indicates that since 1989 more than 60 RELs have been developed from which just a small fraction is constantly maintained [1]. Among these, the most prominent RELs used to represent policies are the MPEG-21 Rights Expression Language [2], the W3C Open Digital Rights Language (ODRL) [3] and the Creative Commons Rights Expression Language (ccREL) [4]. Chong et al. [5] distinguish six policy types that appear in the context of asset management: 1) revenue policies, 2) provision policies, 3) operational policies, 4) contract policies, 5) copyright policies, and 6) security policies. While general-purpose RELs, such as MPEG-21 or ODRL support all of these policies but come with limitations concerning semantic expressivity, complementary special-purpose RELs allow to express more complex policies [6].

Enabling automated policy-based data exchange requires at least three preconditions: (i) policies, such as dataset usage licenses should be available *trust-based*; (ii) policy validation should be achieved through *proactive monitoring, control and access mechanisms* [7][8]; and (iii) reactive checks should be applied to prevent policy violations [7][8] i.e., by applying dataset watermarking techniques [9]. We can conclude that automated policy clearance requires various policies types and compliance mechanisms to specify the conditions under which digital assets are being utilized and exploited, especially when multiple stakeholders are involved in the commercialization strategy [10][11].

III. Challenges

Challenge 1 – Policies for external data exchange in scalable, multilateral settings: The first challenge we identified

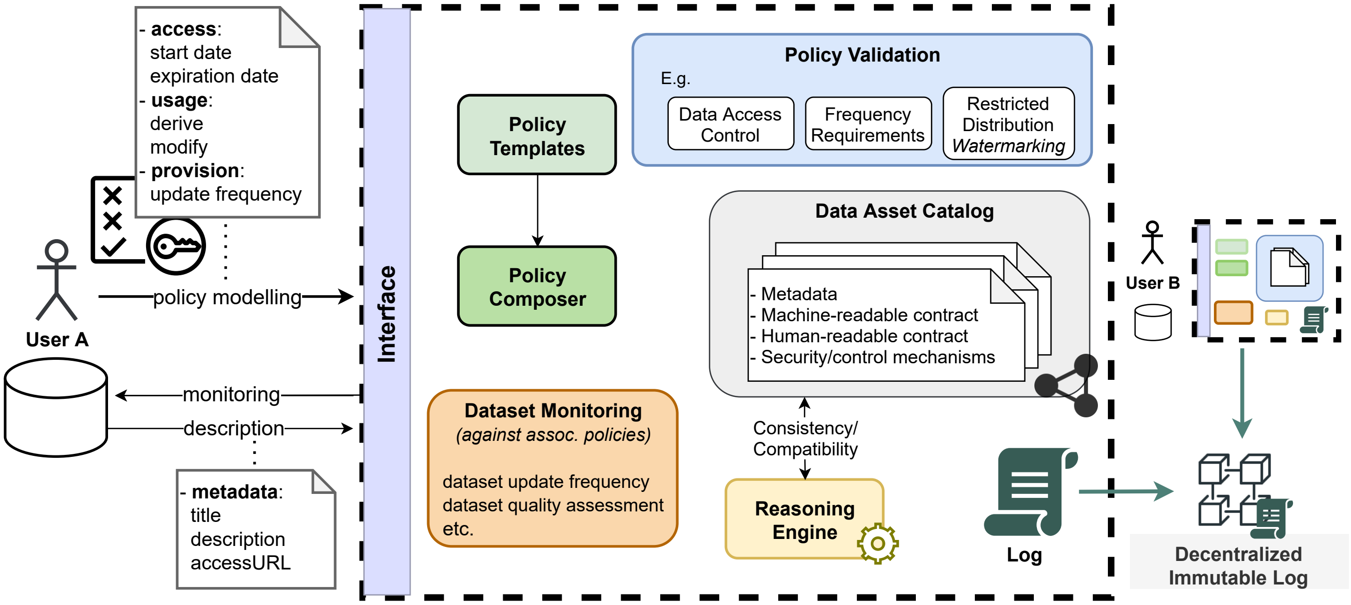

Figure 1. Architecture model of the components and interactions.

is the need for extensible machine-readable but also verbalisable/understandable policies that allow both automated contracting and compliance checking approved by legal experts. This requires auditable processes for policy modelling, adaption and modification. In particular, the process of policy modelling gets increasingly complex when more than two parties are involved: many data contracting and policy reasoning frameworks so far have focused on bilateral contracts only.

Challenge 2 – Develop and extend reasoning routines to support policy creation and ensure policy conformance: A set of formalised and modelled policies can be translated into rules derived from their machine-readable representations (e.g., RDF). These rules (often conditionally) permit or prohibit the execution of an action on certain subjects and may affect other rules, e.g., that govern the execution of the same action on the other subject(s). Accordingly, a declarative (logic-programming-style) reasoning mechanism is required to infer conformance of a created policy and test the compliance with defined terms and conditions.

Challenge 3 – Metadata catalogues for data exchange under specified policies: Current data catalogues so far only organise basic descriptive metadata, i.e., they allow a listing of datasets, provide metadata (in standard vocabularies) and offer search functionalities over the metadata; however, they do not integrate any policy management. The challenge is to incorporate machine-readable policies and contracts in current data catalogues.

Challenge 4 – Automated policy checking and service-level validation: An essential requirement for data users is a guaranteed high quality and reliability of data sources. Quality control and policy management within a data catalogue governed by well defined and modelled machine-readable policies would allow to automate the control and checking of these agreements and policies. The challenge that we identify is the use of monitoring information, such as quality measurements and collected metadata in policies.

Challenge 5 – Towards a framework for decentral data exchange: Current data sharing platforms have mainly centralised and monolithic architectures and potentially build complex environments to serve datasets. These platforms need efficient and scalable management of policies and data access to manage data exchange between multiple partners under several policies and agreements. However, to ensure the synchronisation of the relevant information between the stakeholders (e.g., policies and monitoring results), the architecture model needs to consider a decentral "logging" component.

IV. Solution Approach

Herein, we present our envisioned policy-aware dataset exchange platform (depicted in Figure 1). It processes three policy types, which we derived from the above-stated challenges.

A. Policy Types

In the following, we identify and discuss three different policy types: *(i) usage policies* that regulate distribution and modification of the resource; *(ii) provision policies*, such as a service-level agreement where the provider supplies data, compliant with a specific schema and defined quality metrics (e.g., availability and up-to-dateness); *(iii) access policies* applied to the data by the dataset provider, such as restricted access based on time constraints, version, anonymisation, or subsetting of data.

(i) Usage policies – agreements wrt. permissions, prohibitions and obligations:

Usage policies typically state *trust-based aspects*, as the transmission of data always implies some loss of control over the resource. Any further modification and distribution are possible without the knowledge of the publisher, and it is open for research what is actually (technically/contractually) enforceable in this respect. The example given below depicts a usage policy – using the ODRL vocabulary and RDF Turtle syntax – which prohibits re-distribution of a dataset:

```
<http://example.com/usagePolicy> a odrl:Agreement ;
  odrl:prohibition [
    odrl:action odrl:distribute ;
    odrl:assigner <http://ex.com/OrgaA> ;
    odrl:assignee <http://ex.com/OrgaB> ;
    odrl:target <http://ex.com/doc1> ] .
```

There is recent research on watermarking [9] and fingerprinting [12] of digital resources, which allows a reactive checking of the stated usage policies.

(ii) Provision policies – guaranteed Quality-of-Service / Quality-of-Data: High quality of data – and equally important, metadata – is a crucial requirement for successful data publishing and data sharing via platforms. Provision policies, such as data quality agreements, can be modelled by using (and potentially extending) standard vocabularies. To support an automated validation of provision policies the data-sharing platform needs quality control based on monitoring and quality assessments of the data sources. The following example of a provision policy contains an obligation clause which requires daily updates to the dataset (expressed by using the “odrl:modify” property):

```
<http://example.com/provisionPolicy> a odrl:Agreement ;
  odrl:obligation [
    odrl:action [
      rdf:value odrl:modify ;
      odrl:refinement [
        odrl:leftOperand odrl:elapsedTime ;
        odrl:operator odrl:lt ;
        odrl:rightOperand "P1D" ;
        odrl:unit xsd:duration
                      ]
                ] ;
    odrl:assigner <http://ex.com/OrgaC> ;
    odrl:assignee <http://ex.com/OrgaA> ;
    odrl:target <http://ex.com/doc1> ] .
```

In a real-world setting, such provision policies need additional provenance information, such as a validity period and applicable region.

(iii) Access policies – restricted and monitored access control: In a conditional data sharing scenario, the data provider needs to explicate the access and authorisation conditions. Defining a set of access policies allow the automation of such authorisation and access requirements. Example access policies are time-restricted data access, subsetting or aggregation of data, anonymisation of attributes, etc. Here we give an example of an access policy which permits read-access for a restricted time period:

```
<http://example.com/accessPolicy> a odrl:Agreement ;
  odrl:permission [
    odrl:assigner <http://ex.com/OrgaA> ;
    odrl:assignee <http://ex.com/OrgaD> ;
    odrl:action odrl:read ;
    odrl:constraint [
      odrl:leftOperand odrl:dateTime ;
      odrl:operator odrl:lt ;
      odrl:rightOperand "2022-01-01"^^xsd:date
                    ] ;
    odrl:target <http://example.com/document1> ] .
```

B. Platform Architecture

Figure 1 displays *Data Owner* (User A, at the left of the figure) – potentially also a data user – who interacts with the system in three ways: first, the owner brings in metadata descriptions of the datasets, second, allows monitoring of the datasets, and third, describes the policies under which the dataset is entered into the framework, e.g., restricted access by a start and expiration date, modification policies, and guaranteed update frequency of the resource. The *Policy Composer* and *Policy Templates* components support modelling and ingestion of new policies.

To process the policies (i.e., to check the consistency and compatibility of new entries), there is a *Reasoning Engine* component required, supporting logical reasoning operations. The *Dataset Monitoring* component collects information, such as quality assessments and monitoring results. The central component of the architecture depicted in Figure 1 is the catalogue: it holds the descriptions of the resources, the machine-readable policies and agreements, and the associated control and validation mechanisms that are applied.

Eventually, if Data Consumer (User B, at the right of Figure 1) wants to access a dataset, there is a *Policy Validation* layer which tests and validates the defined policies. For instance, the layer consists of a control mechanism that restricts access based on the defined constraints. To ensure the synchronisation of the relevant information in the *Data Asset Catalog* between the stakeholders (e.g., policies and monitoring results), the architecture includes a shared log component, which synchronises with a *decentralised immutable ledger*.

V. Related Work

There have been several initiatives and approaches to enable efficient and new use of data for small and medium sized companies, to generate new products and services in recent years. Data Markets try to solve these needs: the goal is to enable the distribution and transfer of data – raw, processed, anonymised, etc. – and therefore support a business model based on the exchange of data. A prominent example is the Data Market Austria (DMA) [13] that devised a national-level Data-Services Ecosystem supported by algorithms, tools, and methods for data analytics along the data value chain, and providing data curation, discovery and preservation services through the use of cloud-based approaches. However, in DMA, standard – *non-machine-processable* – licenses for data use and re-use can be defined when datasets are added to the system; and if data providers provide data that is licensed by third parties, they are responsible for disclosing and specifying the licensing terms. Our architecture aims at vastly reducing the tedious contracting efforts.

A survey by Kirrane et al. on existing access control models and policy languages can be found in [10]; a very recent overview of existing policy languages and vocabularies in the context of data protection and GDPR in [14] (under review).

Regarding license management, proof of concepts combining software and data licenses were provided by the Ontology Engineering Group [15] of the University of Madrid and the

IPTC working group on RightsML [16]. Both approaches are still in an experimental phase and lack a sufficient level of usability and legal validation to be suitable for commercial purposes. Villata and Gandon [17] and Governatori et al. [18] describe the formalization of a license composition tool for derivative works. They also provide a demo called Licentia [19] that exemplifies the practical value of such a service. The pitfall of their approach is that license compatibility can just be checked against a bundle of selected permissions, obligations and prohibitions and not against a selection of two or more other licenses containing these or other conditions. Additionally, their compatibility check assumes a reciprocal relationship between licenses instead of a directed relationship as given under real-world circumstances.

In prior work, we developed a framework for automated compatibility checks of these licenses: the DALICC software framework [20] supports the automated license clearance of rights issues in the creation of derivative digital assets (e.g., datasets, software, images, videos, etc.). However, extending these to customized usage policies, such as the examples given above, and provide an automated clearance of these, is still an open research question. The proposed architectures extends DALICC in three main points: (i) it provides a domain-specific licence contract management environment specialized for data sharing among multiple parties, (ii) it focuses on permanence and enforceability of contracts via a distributed trusted environment and an immutable log and (iii) aims at the validation of service-level policies, such as the checking of data quality agreements.

VI. Conclusion and Future Work

In this position paper, we have proposed an architecture that allows stakeholders (users, service providers and third parties) to define customised, machine-processable policies for data exchange that supports automated clearance of usage restrictions, automated validation of data provision and quality agreements, and enforcement and control of data restriction requirements.

Future work will be dedicated to developing methods to validate provision policies to *enforce access restrictions*, and to validate usage policies (e.g., based on digital fingerprinting [12]). Eventually, the results will lead to a platform that allows defining usage, access and provision policies for their resources, to make the resources available to others in decentral organised instances, and to check for potentially conflicting policies and validate the compliance if available ones.

References

[1] T. Pellegrini *et al.*, “A genealogy and classification of rights expression languages–preliminary results,” in *Data Protection/LegalTech-Proceedings of the 21st International Legal Informatics Symposium IRIS*, 2018, pp. 243–250.

[2] *MPEG-21*, https://mpeg.chiariglione.org/standards/mpeg-21, [Online; accessed 19-August-2021].

[3] *ODRL Information Model 2.2*, https://www.w3.org/TR/odrl-model/, [Online; accessed 19-August-2021].

[4] *ccREL: The Creative Commons Rights Expression Language*, https://www.w3.org/Submission/ccREL/, [Online; accessed 19-August-2021].

[5] C. Chong *et al.*, “LicenseScript: A logical language for digital rights management,” *Annales des Télécommunications*, vol. 61, pp. 284–331, Apr. 2006. DOI: 10.1007/BF03219910.

[6] J. Prados, E. Rodriguez, and J. Delgado, “Interoperability between different rights expression languages and protection mechanisms,” in *International Conference on Automated Production of Cross Media Content for Multi-Channel Distribution*, IEEE Computer Society, 2005, pp. 145–152. DOI: 10.1109/AXMEDIS.2005.28.

[7] B. Agreiter *et al.*, “A technical architecture for enforcing usage control requirements in service-oriented architectures,” in *Proceedings of the 4th ACM Workshop On Secure Web Services*, ACM, 2007, pp. 18–25. DOI: 10.1145/1314418.1314422.

[8] S. Pearson and M. C. Mont, “Sticky policies: An approach for managing privacy across multiple parties,” *Computer*, vol. 44, no. 9, pp. 60–68, 2011. DOI: 10.1109/MC.2011.225.

[9] A. S. Panah, R. G. van Schyndel, T. K. Sellis, and E. Bertino, “On the properties of non-media digital watermarking: A review of state of the art techniques,” *IEEE Access*, vol. 4, pp. 2670–2704, 2016. DOI: 10.1109/ACCESS.2016.2570812.

[10] S. Kirrane, A. Mileo, and S. Decker, “Access control and the resource description framework: A survey,” *Semantic Web*, vol. 8, no. 2, pp. 311–352, 2017. DOI: 10.3233/SW-160236.

[11] M. Hilty, A. Pretschner, D. A. Basin, C. Schaefer, and T. Walter, “A policy language for distributed usage control,” in *12th European Symposium On Research In Computer Security*, ser. Lecture Notes in Computer Science, vol. 4734, Springer, 2007, pp. 531–546. DOI: 10.1007/978-3-540-74835-9_35.

[12] P. Kieseberg, S. Schrittwieser, M. Mulazzani, I. Echizen, and E. R. Weippl, “An algorithm for collusion-resistant anonymization and fingerprinting of sensitive microdata,” *Electron. Mark.*, vol. 24, no. 2, pp. 113–124, 2014. DOI: 10.1007/s12525-014-0154-x.

[13] B.-P. Ivanschitz, T. J. Lampoltshammer, V. Mireles, A. Revenko, S. Schlarb, and L. Thurnay, “A semantic catalogue for the Data Market Austria,” in *Proceedings of the Posters and Demos Track of the 14th International Conference on Semantic Systems - SEMANTiCS2018*, 2018.

[14] B. Esteves and V. Rodrıguez-Doncel, “Analysis of ontologies and policy languages to represent information flows in GDPR,” 2021, *Under review*. [Online]. Available: http://www.semantic-web-journal.net/system/files/swj1280.pdf.

[15] *ODRLAPI: A Java API to manipulate ODRL2.0 RDF expressions*, http://oeg-upm.github.io/odrlapi/, [Online; accessed 19-August-2021].

[16] *RightsML - Implementation Examples*, http://dev.iptc.org/RightsML-Implementation-Examples, [Online; accessed 19-August-2021].

[17] S. Villata and F. Gandon, “Licenses compatibility and composition in the web of data,” in *Third International Workshop on Consuming Linked Data (COLD2012)*, 2012.

[18] G. Guido, L. Ho-Pun, R. Antonino, V. Serena, and G. Fabien, “Heuristics for licenses composition,” *Frontiers in Artificial Intelligence and Applications*, vol. 259, pp. 77–86, 2013. DOI: 10.3233/978-1-61499-359-9-77.

[19] *Licentia*, http://licentia.inria.fr/, [Online; accessed 19-August-2021].

[20] T. Pellegrini *et al.*, “DALICC: A license management framework for digital assets,” *Internationales Rechtsinformatik Symposion (IRIS)*, vol. 10, 2019.