Policy Patterns for Usage Control in Data Spaces

Tobias Dam[1], Andreas Krimbacher[2] and Sebastian Neumaier[1]

[1]St. Pölten University of Applied Sciences, Austria
[2]nexyo, Vienna, Austria

Abstract
Data-driven technologies have the potential to initiate a transportation related revolution in the way we travel, commute and navigate within cities. As a major effort of this transformation relies on Mobility Data Spaces for the exchange of mobility data, the necessity to protect valuable data and formulate conditions for data exchange arises. This paper presents key contributions to the development of automated contract negotiation and data usage policies in the Mobility Data Space. A comprehensive listing of policy patterns for usage control is provided, addressing common requirements and scenarios in data sharing and governance. The use of the Open Digital Rights Language (ODRL) is proposed to formalize the collected policies, along with an extension of the ODRL vocabulary for data space-specific properties.

Keywords
usage control policies, ODRL, data spaces, policy enforcement

1. Introduction

The vision of a seamlessly connected mobility ecosystem, where data-driven technologies optimize various aspects of transportation, has the potential to revolutionize the way we travel, commute, and navigate within cities. However, the realization of such a transformative paradigm relies heavily on the establishment of a Data Space for the mobility and transportation domain, wherein data flows effortlessly through fully automated integration processes.

One such effort is the Mobility Data Space (MDS),[1] founded in 2021 as a non-profit organization by "DRM Datenraum Mobilität GmbH." The MDS serves as a virtual marketplace, facilitating the exchange of mobility data among businesses and government entities. It operates as a decentralized infrastructure, allowing providers to list their data offerings in a catalogue while enabling purchasers to search for specific datasets. Importantly, members of the MDS retain ownership of their data, and the data exchange occurs directly between peers, reducing the involvement of the MDS itself. To facilitate this process, the MDS utilizes the Eclipse Data Space Connector (EDC),[2] which provides a framework for sovereign, inter-organizational data exchange.

SEMANTICS 2023 EU: 19th International Conference on Semantic Systems, September 20-22, 2023, Leipzig, Germany
tobias.dam@fhstp.ac.at (T. Dam); andreas.krimbacher@nexyo.io (A. Krimbacher);
sebastian.neumaier@fhstp.ac.at (S. Neumaier)
https://nexyo.io/ (A. Krimbacher)
0000-0002-2463-5831 (T. Dam); 0000-0002-9804-4882 (S. Neumaier)

[1]https://mobility-dataspace.eu/, last accessed 07-07-2023
[2]https://github.com/eclipse-edc/Connector, last accessed 07-07-2023

The EDC is a Free and Open Source Software solution developed and released under the Apache 2 License within the Eclipse Foundation. It serves as a concrete implementation of the protocols defined by the International Data Spaces (IDS) standard and aims to ensure compatibility with the requirements of the GAIA-X project. The EDC facilitates the exchange of data through defined data contracts, which are automatically negotiated to regulate access to data assets. While the Mobility Data Space sees the EDC being responsible for automated contract negotiation and the definition and enforcement of usage policies, it is important to note that the EDC is still in an early phase of development at the time of writing. Currently, there is no agreed-upon set of policy patterns implemented by the EDC that supports enforceability by the connector. To contribute to the further development of automated contract negotiation through data usage policies in the Mobility Data Space, we make the following key contributions:

- *Collection of Policy Patterns*: Based on a literature review, we identify a number of policy patterns for usage control that are relevant in the context of data spaces, with a specific focus on Mobility Data Spaces. These policy patterns capture common and recurring policy requirements and scenarios encountered in data sharing and governance within the mobility and transportation domain. We discuss concrete examples to illustrate how these identified policy patterns can address real-world requirements. The full list of policies is available online.[3]
- *Representation using ODRL*: To support implementation and interoperability, we propose the use of the Open Digital Rights Language (ODRL) for representing the collected policies. Since ODRL is a widely adopted language for expressing permissions, obligations, and conditions related to digital rights, we facilitate standardized and machine-readable policy definitions within the context of data spaces. In cases where the ODRL vocabulary cannot be used to express the policy, we propose an extension of the ODRL vocabulary with Data Space-specific properties, available as ODRL Profile.[4]

In summary, this paper contributes to the development of data spaces, particularly the Mobility Data Space, by providing a comprehensive list of policy patterns for usage control, exemplifying their practical application, and proposing the use of ODRL for representing these policies.

2. Data Space Concepts and Policy Framework

The initial and most significant initiative on data spaces is the International Data Spaces Association (IDSA), initiated in 2015 by the Fraunhofer Society. The non-profit association published a reference architecture [1] and an information model [2].

The Eclipse Dataspace Connector (EDC) project is the most mature instantiation of the IDSA architecture. Figure 1 displays the high-level architecture of the EDC components, as described in the connector's documentation.[5] In alignment with the IDSA architecture, the connectors require a protocol implementation for usage control policy exchange and enforcement among participants.

[3] https://github.com/fhstp/dataspaces-policies
[4] https://w3id.org/dataspaces-policies/
[5] https://eclipse-edc.github.io/docs/#/README, last accessed 07-07-2023

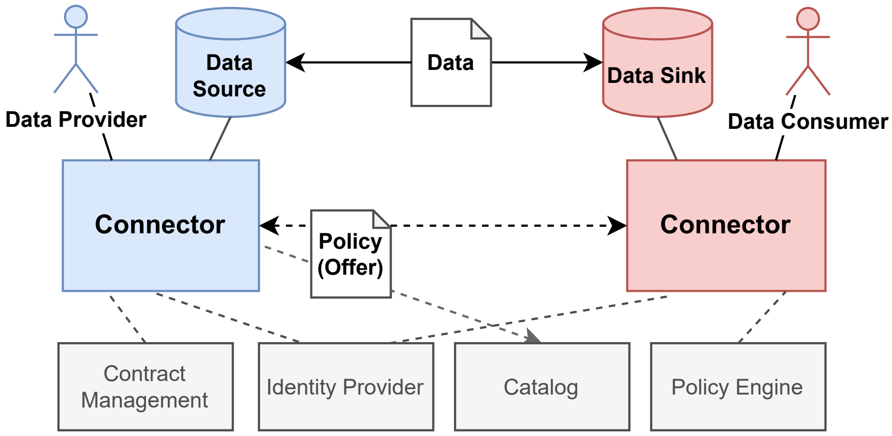

Figure 1: Interactions between the data provider and consumer via connectors and respective components, according to the architecture of the EDC.

2.1. Stakeholder

In a usage control scenario, there are two key entities: the *data provider* and the *data consumer*. In data spaces, these two roles can be interchangeable, as a data consumer may also function as a data provider and vice versa.

An example in the context of a Mobility Data Space is the scenario where a transportation company "TransConnect" collects real-time data from its fleet of vehicles (e.g., GPS coordinates, speed, passenger counts) and shares this data with the analysis firm "TrafficInsights", who processes and analyses it (e.g., identifying traffic patterns, congestion hotspots, travel times). In this scenario, TransConnect acts as the data provider by sharing its real-time vehicle data, while TrafficInsights acts as the data consumer by utilizing the provided data.

However, the roles can be reversed in certain situations: TrafficInsights can act as a data provider by sharing its processed insights and analysis with TransConnect. For instance, TrafficInsights may provide TransConnect with real-time traffic congestion information or recommended alternative routes based on their analysis; TransConnect, in this case, acts as the consumer by utilizing the insights.

In the context of data sharing between a data provider and a data consumer, a trusted *third party* can play a crucial role by providing certificates/guarantees. For instance, consider a data storage provider that acts as a trusted certification authority in our example scenario. Its certificates could verify that TrafficInsights has appropriate data protection protocols in place, such as encryption during data transfer and storage, data anonymization techniques, and adherence to relevant privacy regulations.

2.2. Enforcement

Enforcement involves employing mechanisms to ensure compliance with and control the management of usage policies throughout the different phases of usage: before, during, and after. Akaichi and Kirrane [3] highlight several important aspects of enforcement:

- *Preventive Mechanisms* are dynamic and proactive enforcement mechanisms that allow or prohibit requests for data usage, revoke access in case of policy violations, delay usage requests until obligations are fulfilled, update user or object attributes based on usage decisions, and execute actions like sending notifications to data owners.
- *Detective Mechanisms* can be applied in situations where the usage control framework cannot enforce policy restrictions or prevent policy violations. Various detective mechanisms, such as auditing, logging, or user notifications, can be employed to provide evidence or indications of executed commands.
- *Continuous Mechanisms* involve managing attributes, conditions, and obligation actions that ensure the validity of ongoing data usage.

2.3. Components of a Policy Enforcement Framework

To distribute and modularize the tasks involved in enforcing usage control policies, the framework can be divided into distinct components, where each component focuses on a specific aspect of the process. The eXtensible Access Control Markup Language (XACML) is an OASIS Open standard designed specifically for Attribute-Based Access Control (ABAC). It provides a declarative fine-grained access control policy language, an architecture, and a processing model for evaluating access requests based on defined policies [4]. In particular, the core components of the XACML reference architectures are widely used and can also be applied in the context of data spaces to identify the main responsibilities of the stakeholders:

- The *Policy Enforcement Point (PEP)* serves as the entry point for enforcement within the XACML framework. It forms a request based on attributes of the requester, the resource being accessed, the requested action, and other relevant information, which is sent to the Policy Decision Point for evaluation. The PEP then enforces the decisions made by the Policy Decision Point [4]. *In the data space scenario, this point is typically located at the data provider's connector since it administers access to the data source.*
- The *Policy Decision Point (PDP)* is responsible for making access control decisions by evaluating the request sent by the PEP and considering the applicable policies. The PDP examines the attributes of the requester, the target resource, and contextual information to determine whether access should be granted or denied. The result of the evaluation is then sent back to the PEP for enforcement [4]. *Depending on the concrete policy pattern, the data space stakeholder responsible for determining the fulfilment can be either the data consumer or the provider.*
- The *Policy Information Point (PIP)* is a component within the XACML framework that provides contextual information during policy evaluation. It supplies the PDP with the necessary context information required to make access control decisions, such as geographical location. The availability of contextual information enhances the granularity of access control decisions [4]. *In a data space scenario, such contextual information can be either provided by the connector of the provider or consumer, or by a trusted third party.*
- The *Policy Administration Point (PAP)* is not directly involved in enforcement but plays a crucial role in the specification and management of usage policies within the XACML framework. The PAP handles the management of policies, including creation, activation,

deactivation, and deletion and may provide a user-friendly graphical interface for policy specification and creation [4]. *In a data spaces scenario, the PAP is located at the stakeholder responsible for deploying the policy.*

3. Collection of Policy Patterns

In the following we describe the steps applied to collect applicable policy patterns:

1. *Survey of Usage Control Approaches*: We base our work on the comprehensive survey of different usage control approaches by Akaichi and Kirrane [3]. The survey includes a list of existing usage policy frameworks which we considered for the collection.
2. Collection of Policy Examples: From these frameworks, we collected any kind of policy examples, patterns and concrete instantiations included.
3. *Policy Analysis and Aggregation*: The collected policies were subjected to analysis and aggregation. Similar policies were identified and grouped together, while policies specific to very particular use cases were filtered out. General-purpose policies with wide-ranging applicability were simplified and included in our final collection.
4. *Classification of Policies*: We classified the policy patterns based on their enforcement type, as described in Section 2.2.
5. *Stakeholder Description*: We identify the stakeholders involved in the implementation of the policy pattern; stakeholders are providers, consumers, and third parties, as described in Section 2.1.
6. *Specification of Policy Information Point and Policy Administration Point/Policy Decision Point*: For each policy pattern in our collection, we specified the corresponding PIP and PAP/PDP, as described in Section 2.3. This specification identifies the stakeholder responsible for deploying (PAP) and evaluating (PDP) the policy, and the stakeholder responsible for providing information about its fulfilment (PIP).
7. *Identification of Additional Patterns*: During the compilation of our policy collection, we observed that certain use cases, presumably relevant for policies used in data spaces, were not adequately covered. Consequently, we appended some rules of our own to address these gaps.

Table 1 contains our compiled set of policies, our classifications as well as where they originated from. Our self-defined policies can be identified via the “-” in the reference column.

Table 1: Derived policy patterns classified by enforcement mechanism as well as the respective PIP, PAP/PDP.

Policy Pattern	**Description / Example**	**PIP**	**PAP/PDP**	**Enforcement**	
Allow access	Provider allows access to the data for a specific consumer.	Provider	Provider	Preventive	[5, 6]
Location / Regional access restriction	Consumer can access data only if located in allowed region.	Provider	Provider	Detective	[5, 6, 7]
Location / Regional storage restriction	Consumer can store data only if the storage is located in allowed region.	Consumer, Third Party	Provider	Detective	[5, 6, 7]
Time restriction	Consumer can access data only in pre-defined time period.	Provider	Provider	Preventive	[5]
Access count	Consumer can access the data a fixed number of times.	Provider	Provider	Preventive	[5, 6]
Rate limit	Consumer can access the data only a limited number of times within a period.	Provider	Provider	Preventive	[8]
Concurrent active connections	The number of concurrent connections to retrieve the data is limited.	Provider	Provider	Preventive	[8]
Amount of data	The amount of data that can be transferred/ streamed is limited.	Provider	Provider	Preventive	[6]
Processing power	The processing power to prepare/ provide the data is limited.	Provider	Provider	Preventive	[6, 9]
Bandwidth	The bandwidth to transfer/ stream the data is limited.	Provider, Third Party	Provider	Preventive	[6, 9]
Billing / Credit points	The consumer will be charged for the data accesses.	Provider	Provider	Preventive	[5, 6]

Data quality	The consumer demands certain data quality standards, e.g., schema conformance, data consistency, etc.	Consumer	Consumer	Detective	-
Deletion	Consumer is required to delete data after specific period.	Consumer, Third Party	Provider	Detective	[5, 10]
Purpose / Application	Consumer is restricted to use data for specific purpose only.	Consumer, Third Party	Provider	Detective	[5]
Provable attribute	Provider demands a certificate/guarantee, e.g. of membership.	Provider, Third Party	Provider	Preventive	[5, 11]
Encryption by consumer	Provider demands consumer to store data encrypted.	Consumer	Provider	Detective	[5, 7]
Encryption by provider	Consumer demands provider to transfer data encrypted.	Consumer	Consumer	Preventive	-
Aggregation	Provider demands consumer to aggregate data before processing.	Consumer	Provider	Detective	[12]
Anonymization	Provider demands consumer to anonymize data before processing.	Consumer	Provider	Detective	[12, 13]
Activity logging	Provider demands shared activity logging of the data processing.	Consumer	Provider	Detective	[5, 10]
Delegation of permission	Provider demands to attach a certain policy when distributing the data.	Consumer	Provider	Detective	[5, 6]
Up-to-dateness	Consumer demands that the data is updated with a specified frequency.	Consumer	Provider	Detective	-

4. Formalization of Policy Patterns

This section showcases the applicability of the policy patterns in table 1 and provides a formalization using the Open Digital Rights Language (ODRL) [14]:

- The policies follows the ODRL Information Model [14];
- we make use of the ODRL Core Vocabulary and ODRL Common Vocabulary [15] to describe the actions and operands of the respective policies, where possible;
- our representations follow the example use cases from the specification [14];
- the corresponding Github repository[6] lists a separate ODRL instantiation for each policy pattern;
- in case we could not model the policy using the ODRL vocabulary terms, we propose extensions in an ODRL Profile. The profile is available in a dedicated ontology.[7]

The following examples present selected, combined instantiations of the policy patterns:

Provider-administered policy patterns. Listing 1 combines the "Amount of data", "Deletion" and "Anonymization" patterns of Table 1. In this example, the data provider grants the consumer the permission to *read* an asset.[8] In our approach, to be able to limit the amount of data that can be acquired, first the unit of the action *read* is defined to be measured in "MiB" (cf. lines 32 to 38). Then the actual limitation is specified via a constraint that requires the count to be less than 1024 (lines 40 to 43). Each execution of the action increases the count, effectively permitting a read of 1 MiB per execution and a total of 1024 MiB.

Additionally, this policy has two obligations that the consumer has to fulfil. The first obligation (lines 16 to 26) requires the consumer to delete the file before a specific date; the second obligation (26 to 30) requires the consumer has to anonymise the acquired file.

Consumer-administered policy patterns. Listing 2 combines the "Up-to-dateness" and "Data quality" patterns of Table 1 and gives an example for use cases where the consumer requires the provider to perform specific tasks. I.e., in such a scenario the enforcement of usage control policies is not only relevant for data leaving the provider's domain but also for regulating usage within the consumer's domain.

Our example (Listing 2) requires the provider to continuously update the provided data as well as to conform their data to a specified format. Such a conformance specification could be provided using an explicit schema description, e.g., our example uses a Shape Constraint Language (SHACL) schema [16]. The policy is modelled using the terms *update*, *qualityControl*, and *conformsTo* (line 9, 19, and 21) which are self-defined extensions, described in our ODRL profile.[9]

[6] https://github.com/fhstp/dataspaces-policies/tree/main/example-policies
[7] https://w3id.org/dataspaces-policies/
[8] The W3C's ODRL Vocabulary specification defines the action *read* as "obtain data from the Asset" [15]. While this action is a general approach covering any kind of data access, more specific types of access could be defined in a dedicated profile.
[9] https://fhstp.github.io/dataspaces-policies/index.html#update

```
@prefix odrl: <http://www.w3.org/ns/odrl/2/> .
@prefix dc11: <http://purl.org/dc/elements/1.1/> .
@prefix xsd: <http://www.w3.org/2001/XMLSchema#> .
@prefix rdf: <http://www.w3.org/1999/02/22-rdf-syntax-ns#> .
@prefix dsp: <http://www.w3id.org/dataspaces-policies/> .

<http://example.com/policies#consumer-administered>
  a odrl:Policy ; odrl:profile odrl:core ;
  odrl:permission _:perm .

_:perm a odrl:Permission ;
  odrl:target <http://example.com/files/file1> ;
  odrl:assigner <https://www.example.com/provider> ;
  odrl:assignee <https://www.example.com/consumer> ;
  odrl:action _:act1 ;
  odrl:constraint _:constr1 ;
  odrl:obligation [
    a odrl:Obligation ;
    odrl:target <http://example.com/files/file1> ;
    odrl:action odrl:delete ;
    odrl:constraint [
      a odrl:Constraint ;
      odrl:leftOperand odrl:dateTime ;
      odrl:operator odrl:lt ;
      odrl:rightOperand "2023-07-10T00:00:00Z"^^xsd:dateTime
    ]
  ], [
    a odrl:Obligation ;
    odrl:target <http://example.com/files/file1> ;
    odrl:action odrl:anonymize
  ] .

_:act1 a odrl:Action ;
  rdf:value odrl:read ;
  odrl:refinement [
    odrl:leftOperand odrl:unitOfCount ;
    odrl:operator odrl:eq ;
    odrl:rightOperand "MiB"
  ] .

_:constr1 a odrl:Constraint ;
  odrl:leftOperand odrl:count ;
  odrl:operator odrl:lteq ;
  odrl:rightOperand "1024" .

<http://example.com/files/file1> a odrl:Asset ;
  dc11:title "File 1" .
```

Listing 1: The provider limits the amount of data the consumer can acquire, obliges the consumer to delete the data after a specific date as well as requires the consumer to anonymise the data.

The constraint to the *update* obligation (line 4 to 15) specifies that the action has to be executed every 30 seconds.[10] The *qualityControl* obligation (lines 16 to 25) consists of a refinement to determine what quality control is actually to be applied, i.e., it states that the asset needs to conform to a specific SHACL shape. Additionally, the obligation contains a constraint (line 26

[10]The time interval is specified as a XML Schema Definition (XSD) duration [17].

```
1  <http://example.com/policies#consumer-administered>
2    a odrl:Policy ;
3    odrl:profile <http://www.w3id.org/dataspaces-policies/> ;
4    odrl:obligation [
5      a odrl:Obligation ;
6      odrl:target <http://example.com/files/file1> ;
7      odrl:assigner <https://www.example.com/consumer> ;
8      odrl:assignee <https://www.example.com/provider> ;
9      odrl:action dsp:update ;
10     odrl:constraint [
11       a odrl:Constraint ;
12       odrl:leftOperand odrl:timeInterval ;
13       odrl:operator odrl:eq ;
14       odrl:rightOperand "P30S"^^xsd:duration
15     ]
16   ], [
17     odrl:action [
18       a odrl:Action ;
19       rdf:value dsp:qualityControl ;
20       odrl:refinement [
21         odrl:leftOperand dsp:conformsTo ;
22         odrl:operator odrl:eq ;
23         odrl:rightOperand <http://example.com/shacl-shape>
24       ]
25     ] ;
26     odrl:constraint [
27       odrl:leftOperand odrl:event ;
28       odrl:operator odrl:lt ;
29       odrl:rightOperand odrl:policyUsage
30     ]
31   ] .
```

Listing 2: The consumer demands the provider to update their data every 30 seconds and requires the data to conform to a specified schema.

to 30) that requires the asset to conform to the schema before the action is executed.

5. Conclusion

The increasing exchange of data in mobility ecosystems, specifically through the use of data space connectors such as the Eclipse Dataspace Connector (EDC), creates the need of automated negotiation of data exchange contracts. To effectively protect valuable data by declaring who can access data under which conditions, participating members of data spaces require policy patterns to implement robust access and usage controls.

To address this need, we have contributed a collection of usage and access control policies, classified according to their enforcement mechanism, and discussed the respective components of the policy enforcement framework in a data space setup. We have demonstrated the application of our policy patterns via example ODRL scenarios for each of the policy pattern.

In some of the scenarios, we have found that the existing ODRL core and common vocabulary are not sufficient to create ODRL instantiations for the collected policy pattern. To overcome this limitation, we have proposed an initial set of complementing terms as an ODRL Profile;[11] this extension needs further definition, extension and evaluation in future work.

While our collection of policy patterns is based on a comprehensive survey of policy frameworks [3], we acknowledge that it may be incomplete. To provide a more comprehensive set of policy patterns that aligns with the needs of data sharing companies, future research will focus on gathering real-world use cases and obtaining feedback from these companies.

To ensure an efficient adoption of our policy patterns, companies require a complete, well-tested and easy-to-use implementation of a data space connector, e.g. of the EDC. Additionally, a graphical user interface to create and administer patterns is essential; there is already existing work in this respect that can serve as a basis.[12]

In conclusion, our research emphasizes the importance of automated negotiation of data exchange contracts and the need for comprehensive policy patterns to enforce usage control in data spaces. By providing a collection of policy patterns and outlining the requirements for their practical application, we aim to contribute to the development of data sharing ecosystems in the context of the Mobility Dataspace.

Acknowledgments

This research was funded by the Austrian Research Promotion Agency (FFG) through BRIDGE project 891103 “DiDaMe”. The financial support by the Austrian Research Promotion Agency and the Federal Ministry for Digital and Economic Affairs is gratefully acknowledged.

References

[1] B. Otto, S. Steinbuß, A. Teuscher, S. Lohmann, IDS Reference Architecture Model, 2019. URL: https://internationaldataspaces.org/wp-content/uploads/IDS-Reference-Architecture-Model-3.0-2019.pdf.

[2] S. Bader, J. Pullmann, C. Mader, S. Tramp, C. Quix, A. W. Müller, H. Akyürek, M. Böckmann, B. T. Imbusch, J. Lipp, S. Geisler, C. Lange, The International Data Spaces Information Model – An Ontology for Sovereign Exchange of Digital Content, in: 19th International Semantic Web Conference (ISWC 2020), Springer, 2020, pp. 176–192. doi:10.1007/978-3-030-62466-8_12.

[3] I. Akaichi, S. Kirrane, Usage control specification, enforcement, and robustness: A survey, 2022. arXiv:2203.04800.

[4] OASIS Standard, eXtensible Access Control Markup Language (XACML) Version 3.0, 2013. URL: http://docs.oasis-open.org/xacml/3.0/xacml-3.0-core-spec-os-en.html.

[5] Fraunhofer ISST, IDS Usage Control Policies, 2021. URL: https://international-data-spaces-association.github.io/DataspaceConnector/Documentation/v6/UsageControl.

[11] https://w3id.org/dataspaces-policies

[12] IDS Policy Editor, developed by Fraunhofer, https://odrl-pap.mydata-control.de, last accessed 07-07-2023

[6] X. Zhang, M. Nakae, M. J. Covington, R. Sandhu, Toward a usage-based security framework for collaborative computing systems, ACM Transactions on Information and System Security (TISSEC) 11 (2008) 1–36.

[7] T. Wüchner, S. Müller, R. Fischer, Compliance-preserving cloud storage federation based on data-driven usage control, in: IEEE 5th International Conference on Cloud Computing Technology and Science, CloudCom 2013, Bristol, United Kingdom, December 2-5, 2013, Volume 2, IEEE Computer Society, 2013, pp. 285–288. doi:10.1109/CloudCom.2013.149.

[8] R. Teigao, C. Maziero, A. O. Santin, Applying a usage control model in an operating system kernel, J. Netw. Comput. Appl. 34 (2011) 1342–1352. doi:10.1016/j.jnca.2011.03.019.

[9] G. Costantino, A. L. Marra, F. Martinelli, P. Mori, A. Saracino, Privacy preserving distributed attribute computation for usage control in the internet of things, in: 17th IEEE International Conference On Trust, Security And Privacy In Computing And Communications / 12th IEEE International Conference On Big Data Science And Engineering, TrustCom/BigDataSE 2018, New York, NY, USA, August 1-3, 2018, IEEE, 2018, pp. 1844–1851. doi:10.1109/TrustCom/BigDataSE.2018.00279.

[10] J. Schütte, G. S. Brost, LUCON: data flow control for message-based iot systems, in: 17th IEEE International Conference On Trust, Security And Privacy In Computing And Communications / 12th IEEE International Conference On Big Data Science And Engineering, TrustCom/BigDataSE 2018, New York, NY, USA, August 1-3, 2018, IEEE, 2018, pp. 289–299. doi:10.1109/TrustCom/BigDataSE.2018.00052.

[11] G. Russello, N. Dulay, xducon: Coordinating usage control policies in distributed domains, in: Y. Xiang, J. López, H. Wang, W. Zhou (Eds.), Third International Conference on Network and System Security, NSS 2009, Gold Coast, Queensland, Australia, October 19-21, 2009, IEEE Computer Society, 2009, pp. 246–253. doi:10.1109/NSS.2009.77.

[12] A. Munoz-Arcentales, S. López-Pernas, A. Pozo, Á. Alonso, J. Salvachúa, G. Huecas, Data usage and access control in industrial data spaces: Implementation using fiware, Sustainability 12 (2020) 3885.

[13] Q. H. Cao, G. Madhusudan, R. Farahbakhsh, N. Crespi, Policy-based usage control for a trustworthy data sharing platform in smart cities, Future Gener. Comput. Syst. 107 (2020) 998–1010. doi:10.1016/j.future.2017.05.039.

[14] S. Villata, R. Iannella, ODRL Information Model 2.2, W3C Recommendation, W3C, 2018. URL: https://www.w3.org/TR/2018/REC-odrl-model-20180215/.

[15] R. Iannella, V. Rodríguez-Doncel, M. Steidl, S. Myles, ODRL Vocabulary & Expression 2.2, W3C Recommendation, W3C, 2018. URL: https://www.w3.org/TR/2018/REC-odrl-vocab-20180215/.

[16] H. Knublauch, D. Kontokostas, Shapes Constraint Language (SHACL), W3C Recommendation, W3C, 2017. URL: https://www.w3.org/TR/2017/REC-shacl-20170720/.

[17] H. Thompson, A. Malhotra, S. Gao, P. V. Biron, M. Sperberg-McQueen, D. Peterson, W3C XML Schema Definition Language (XSD) 1.1 Part 2: Datatypes, W3C Recommendation, W3C, 2012. URL: https://www.w3.org/TR/2012/REC-xmlschema11-2-20120405/.