Leveraging a Microservice Architecture, Access Control and Interoperability Patterns to Manage Privacy-related User Consents

Selena Lamari[1,2], Nadjia Benblidia[2], Chouki Tibermacine[3], Christelle Urtado[4], and Sylvain Vauttier[4]

[1] LIRMM, Univ. Montpellier, CNRS, Montpellier, France
selena.lamari@lirmm.fr
[2] LRDSI Laboratory, Faculty of Sciences, Blida 1 University, Algeria
nadjia.benblidia@univ-blida.dz
[3] Université Bretagne Sud, UMR CNRS 6074, IRISA, F-56000 Vannes, France
chouki.tibermacine@univ-ubs.fr
[4] EuroMov Digital Health in Motion, Univ. Montpellier & IMT Mines Ales, Ales, France – {Christelle.Urtado,Sylvain.Vauttier}@mines-ales.fr

Abstract. Consent management is of paramount importance when processing personal data. It is now prescribed as mandatory by regulations such as the GDPR. This article presents a micro-service architecture providing a protocol for access control including authentication, authorization management, and externalized consent management based on the ABAC (Attribute-Based Access Control) and the side-car architecture patterns. The experimentation on a case example validates that its integration is light and non intrusive for existing applications.

***Keywords*—Software engineering, Privacy, GDPR regulation, Personal data, Consent maangement, Microservice architecture, Interoperability**

1 Introduction

The massive collection of data endangers individuals' privacy, raising crucial questions about how this potentially sensitive information is managed and used. To regulate illegal use of personal data, standards and laws have been put in place, such as the GDPR (General Regulation on the Protection of Personal Data) [2]. The GDPR defines consent and its specific characteristics: it must be clear, explicit, free and unambiguous (Art. 4). Furthermore, the withdrawal of consent must be as easy as its collection. All these rules and directives provide regulatory frameworks that are designed to protect user privacy. However, technical solutions are missing. One of the solutions widely adopted in practice is the introduction of General Conditions of Use (GCU) and privacy policies. These are generally perceived negatively by users, due to their length and the complexity of the legal jargon. Moreover, the process of withdrawing consent often proves to be difficult. Users may have to send an e-mail or follow complex procedures. Managing consent in software applications therefore is a challenge. It often requires extensive refactoring of applications, which can potentially compromise their proper functioning.

To tackle this issue, this paper proposes a solution consisting of an access manager based on data subjects' consents [1], which instantiates the ABAC (Attribute-Based Access Control) architecture pattern [5]. Consents are handled by a microservice architecture (PRIAM-MSA) which covers all the functional requirements induced by the GDPR. This architecture is also independent from the application, and requires a secure connection. To this end, it relies on the OAuth2 protocol to enforce secure, standardized management of authorizations and access.
The remainder of the paper is structured as follows. Section 2 presents the proposed consent management solution. It shows in particular (a) the proposed microservices architecture, (b) an OAuth2 protocol for access control including authentication and authorization management, (c) the consents-based access controller. Section 3 details the case study that illustrates the ability to effectively integrate our solution into an existing application. Section 4 discusses the related work. Section 5 summarizes the contributions of this paper and draws some perspectives.

2 Use of the ABAC pattern for consent management

This section presents a GDPR-compliant architecture for data subject consent management in RESTful applications. The proposal is based on three key elements: a microservices architecture aligned with GDPR requirements, a secure access protocol based on OAuth2 for data subject authentication and authorization, and an access controller based on the ABAC architecture pattern to manage data access according to data subject's consent. This combination creates a consent management solution that adapts to any non-privacy-aware RESTful application, while not being intrusive.

The following sub-sections explain each element of the proposed architecture.

2.1 PRIAM-MSA: Privacy Driven Microservices

Based on the principles and requirements of the GDPR, we propose PRIAM-MSA which stands for PRIAM (PRIvacy Assessment Model) MicroService Architecture. The architecture comprises six microservices. Microservices are customizable to fit any RESTful application that processes personal data in order to make the application GDPR-compliant. Each microservice has its own database and distinct set of features:

- **Actor Management (AM)** manages all actors involved in the application whether they have direct or indirect access to data subjects' personal data.
- **Data & Processing management (DM)** manages the list of personal data used in the application. It also manages the list of all the personal data processing. This information is needed, for example, to maintain the records of processing (Art. 30), implement the right to know and manage the consents required to carry out new processing.
- **Right Management (RM)** records and monitors all requests made by data subjects for the exercise of their rights as mentioned in the GDPR (Art. 15–19). It also manages the corresponding answers given by the application provider. The RM microservice needs to access the list of processings and personal data according to the type of requests. It has a dependency with the DM microservice.

[1] Data subjects are users whose personal data is collected and processed by an application

- **Consent Management (CM)** manages consents, including their collection and withdrawal by data subjects. The CM microservice calls the DM microservice to list personal data processings and display their descriptions.
- **Notification (N)** ensures communication between the different involved actors (application provider, data subjects, etc.). Notifications are requested by RM in order to inform the actors concerned with the requests issued. They are also requested by CM when a processing is modified in order to inform the data subjects and allow them to revise their consents.
- **Breach Management (BM)** manages all personal data breaches. Depending on the risk level, it notifies users about the data that has been breached. BM microservice hence depends on DM, AM and N microservices. Similar to the records of processings, records of breaches are automatically generated by BM.

All these microservices are provided through an API Gateway which centralizes access to the individual API of each microservice.

The **PRIAM front-end** implements a basic front-end for PRIAM-MSA so that the application holds UIs to access the PRIAM-MSA privacy services. This front-end can be easily integrated into any application. The application developer can simply add a button, a link or a menu item that gives access to the PRIAM front-end main page.

The solution meets the characteristics of consents cited in Art. 4(11) of the GDPR, which defines consent as a free, specific, informed and unambiguous indication of the will of the data subject. In addition, it complies with Art. 7, which mentions the conditions applicable to consents, including the ability to demonstrate that the data subject has given consent and the right to withdraw consent at any time.
PRIAM meets these requirements by enabling data subjects to give and withdraw their consent simply and transparently with toggle buttons via an easily accessible UI. Data subjects are informed as required by Art. 13 (purposes, categories of data, etc.) to make informed decisions. Each consent metadata is recorded to demonstrate that it has been given in compliance with legal requirements.
To secure data subjects' access to PRIAM and enable them to exercise their rights, authentication and authorizations are needed.

2.2 Authentication / Authorization service

Since PRIAM is application independent, adding an authentication layer is key to secure data subjects' access. The proposed solution enables data subjects to connect to PRIAM without additional credentials, simplifying their user experience. It promotes interoperability with the application and guarantees a high level of security, at least equivalent to that of the application.
PRIAM uses OAuth2 [12], a standard authorization protocol that secures access to online resources without sharing login credentials. OAuth2 is used to secure access to both the PRIAM front-end and microservices. Each access to PRIAM-MSA requires verification at the API Gateway level. Identity and authorization clients are added as sidecars to the API Gateway so that it communicates with the identity and authorization server which may be present on the application side. Figure 1 details the authentication and authorization process based on OAuth2:
{1,2}: The data subject accesses the PRIAM front-end via her / his browser.
{3}: The PRIAM front-end checks the data subject's authorization to access resources.

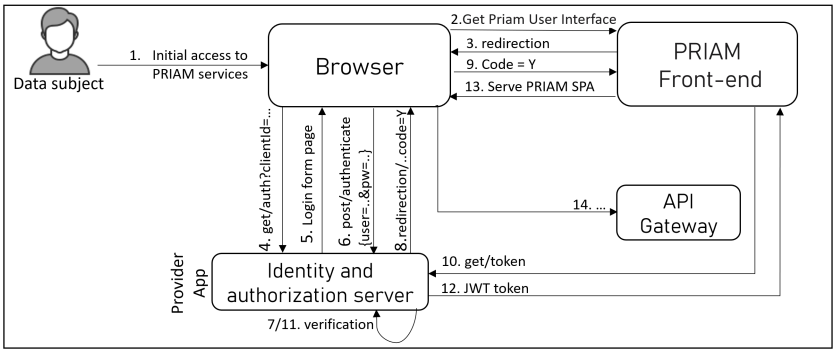

Fig. 1. Illustrating authorization to PRIAM flow

{4}: Since the data subject is not yet logged into PRIAM, he / she is redirected to the login page of the application's identity and authorization server.
{5}: The identity and authorization server returns a login form to the data subject.
{6-9}: After successful authentication, the identity and authorization server provides an authentication code to the PRIAM front-end.
{10,11}: The PRIAM front-end then requests an OAuth2 access token from the authorization server with the previously retrieved authentication code.
{12-13}: The access token is returned to the PRIAM front-end. The data subject is now connected and authorized to access resources.
{14...}: Once the data subject is authorized to access the PRIAM front-end, he / she can exercise his / her rights. Access to microservices is carried out via the API Gateway which is secured by the same protocol.
The Single Sign-On (SSO) capability provided by our solution allows to access all secured resources as a single active "session".

2.3 Consent-based access controller

Consent management is a crucial issue both for application provider, who must comply with regulations, and data subjects, who want their personal data to be protected. Authorization to access and use data depends on the fine-grained preferences of each data subject.
Once consents have been collected from data subjects in a secure manner, our solution manages the permissions to run processings and access personal data based on these consents. Access control models have been proposed, such as RBAC [14] and ABAC [5]. RBAC defines roles that hold access rights to functionalities or services. Users can then be given one or more roles that authorize access to services accordingly. ABAC defines authorizations based on attributes and their values. These authorizations are finer grained than those defined in RBAC. They best suit our needs as access to personal data depends not only on the data subjects, but also on the fine-grained consents they give for specific data and processings. We thus propose an access controller based on ABAC. The architecture of ABAC is composed of four services: the Policy Administration Point (PAP), the Policy Information Point (PIP), the Policy Decision Point (PDP) and the Policy Enforcement Point (PEP) [5].

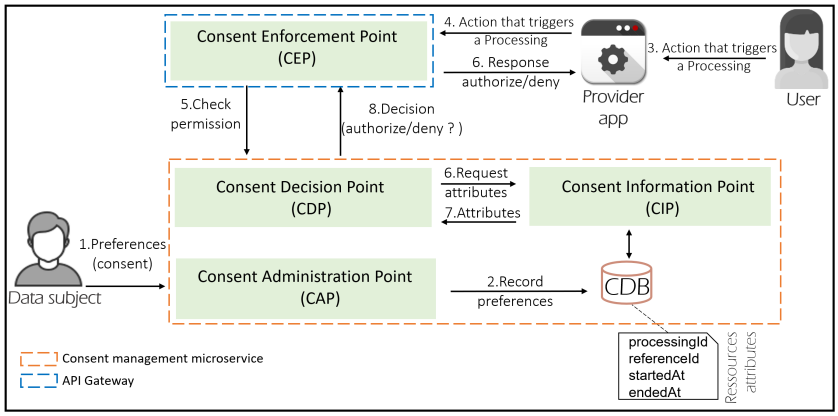

Fig. 2. Instance of the ABAC architecture pattern for consent management

In the following, we explain how this architecture is integrated to our microservice architecture to propose a design centered on data subjects' preferences (*see* Fig.2). As in the ABAC architecture pattern, we define four distinct services:

1. **Consent Administration Point (CAP)**: the CAP serves as a centralized command center for managing access consents. Its main function is to empower data subjects with a unified interface for giving, updating and withdrawing their consents. This ensures that data subjects' preferences and authorizations are correctly taken into account, in line with legal and regulatory requirements.
   Once data subjects have chosen their preferences, CAP records the consents in the consent database (CDB), the database of the consent management microservice.
   The CAP plays a crucial role in consent management, as the recorded attributes (referenceId, processingId, start and end date of consent) serve as the basis for the Consent Decision Point (CDP) to manage access control to personal data.
   referenceId refers to the identifier of the data subject in the application database. This attribute is used to link data subject's data in the application and the corresponding privacy metadata in the the PRIAM database.
2. **Consent Information Point (CIP)**: in an ABAC architecture pattern, access conditions are defined by attributes. In our architecture, they include the identifiers of the data subject, the identifier of the consented processing, as well as the dates of validity of consent: start date (date on which consent is given) and end date (case of withdrawal of consent or end of processing).
   The CIP is responsible for providing the CDP with these attributes in real time, ensuring that it has up-to-date, accurate information to make informed and contextual decisions.
3. **Consent Decision Point (CDP)**: the CDP is the heart of the assessment of access rights. It is responsible for making decisions about access to the personal data of data subjects, based on the resources returned by the CIP.
   Consent is considered valid if it exists in the CDB database, with a valid start date and a non expired end-date. A full consent history is kept. When a data subject gives consent for processing, the consent is created with a start date. In the event of withdrawal, an end date is assigned. If the data subject decides to modify a consent, a new consent is created to trace modifications over time.

The consent history is essential for transparent auditing and compliance with data protection regulations. By maintaining a detailed record of consents, the application can demonstrate that data has been processed legitimately.

4. **Consent Enforcement Point (CEP)**: the CEP receives requests from the application (processing that a user wishes to carry out) and transmits them to the CDP, which evaluates the requests in the light of the data subject's consents, and decides whether or not to grant access to the personal data in order to carry out their processing. Depending on the response from the CDP, immediate action is taken to either execute or block the request.

The CAP, CIP and CDP points all are integrated into the consent management microservice. The CEP is integrated into the API Gateway, since its role is to intercept requests and return results to the application.
Our consent management solution facilitates the exercise of data subjects' rights, in particular with regard to the collection and withdrawal of consents. It also enables application providers to comply with current regulations by managing data access authorizations and providing a tangible proof of given consents. In the following, we will illustrate how it can be integrated into an existing application.

3 Case study: Adding consent management in the TeaStore application

This section depicts and discusses an experimentation about the integration of our solution into a reference RESTful web application, the "TeaStore" [2]. This application, developed by the University of Wurzburg [19], is used in various benchmarks and tests. TeaStore does not provide any personal data protection mechanisms.
To integrate the microservices of our solution, the application has to provide a standardized set of REST API endpoints to PRIAM-MSA in order to read, update or delete personal data, when data subjects exercise their rights.[3]. This kind of API exists in most RESTful applications. Otherwise, it needs to be developed but the effort necessary to develop it is marginal as compared to the features brought out by the integration of PRIAM-MSA. Interacting through a REST API makes PRIAM-MSA independent from the application's technologies. This solution does not require direct access to the application's databases which makes it compatible with any database server where personal data is stored. In the case study, 4 API endpoints are exported from the TeaStore application to PRIAM-MSA in order to manage RUD operations and respond to access, rectification and to be forgotten rights [3].
We also integrated the basic PRIAM front-end supplied with PRIAM-MSA to provide the application with a Web UI for privacy management. Granting and withdrawal of consent is done via the same UI, where a list of processing operations is displayed with toggles. Each processing operation is described in detail, including the data required to run it and its purpose.
In the `TeaStore` application (*see* Fig.3), the recommender processing is optional and can be activated or deactivated by the data subject. "Place an order" processing, on the other hand, is essential to the functioning of the application; we cannot refuse

[2] https://github.com/DescartesResearch/TeaStore, accessed on July 19, 2024
[3] The full Open-API specification and the explanation of the configuration are both available at: `https://github.com/PRIAM-solution/PRIAM-Teastore.git`

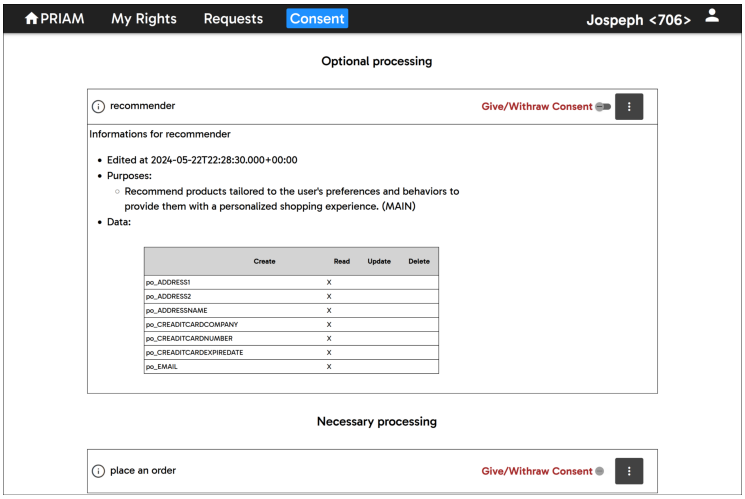

Fig. 3. Example of the UI for giving and withdrawing consent

consent for this kind of processing (it is displayed with a grayed toggle blocked on "On" position). To integrate the PRIAM front-end, the application developers can simply add a button, a link or a menu item, redirecting to the PRIAM UI main page. In our case example, a button is added to the `TeaStore` main page.

Access the PRIAM front-end must be authorized. Our consent manager thus integrates identity and authorization clients as a sidecar to PRIAM front-end and the API Gateway. These clients interact with the application's identity and authorization server. If it does not have this server, the developer will have to integrate it. The TeaStore app does not provide this kind of authorization server. We therefore use Keycloak, an open-source Identity and Access Management server [1].

Last, to integrate access control, developers must add a filter or a middleware mechanism that intercepts processing execution requests and sends them to the CEP, to check access rights according to consents. In this case study, we added at the "service" layer of the application HTTP requests sent to PRIAM API for checking access rights when functions are called. The request includes the user ID (embedded in the ID token) and the processing ID, which is by default the name of the function. If consent is granted the normal workflow of the function is executed. If not, an INFO-level log is produced and a notification message is returned. This message is displayed to the user by the `TeaStore` front-end. As `TeaStore` is a Java Servlet application, a more elegant solution for intercepting processing requests would be to use security filters, if they were part of the original implementation of the application. Nevertheless, as we may notice, the integration of our solution is very simple and has a minimal impact on the application.

Discussion. In a recently article, Smirnova *et al.* [16] highlight the challenges of GDRP compliance. Our solution addresses these challenges by being:

- **Independent and interoperable:** One of the strengths of our solution is to keep the business logic linked to GDPR requirements separated from the rest of the application. This means that compliance with the regulation does not involve any major modifications or refactoring to the application code.

- **Dynamic and automatic**: Updates to the purpose, data or other information related to any processing triggers a notification to the data subject, thus ensuring transparency and offering her / him the possibility of changing his / her choices if necessary. Moreover, it adjusts access rights automatically as consents are collected and withdrawn.
- **Proactive and reactive:** The recording of consent, with the date of collection and withdrawal, is particularly useful for audit purposes. This history is used, for example, to verify compliance after a complaint has been reported. In addition, it consists in anticipating and implementing measures to ensure that user consents are obtained, maintained and respected, thus avoiding any breach of regulations and sanctions.
- **Enhanced user experience**: The PRIAM UI does not display a GCU (general conditions of use). It guarantees transparency by highlighting all processings and collects consents explicitly. Withdrawing consent is as simple as obtaining it, as required by Art. 7, and is done via the same UI, thus meeting requirements for consent management from Art. 4.
- **Minimum cost and effort**: GDPR compliance is costly in terms of time, human and financial resources [16]. Our solution provides an off-the-shelf implementation of the software requirements that can be derived from the GDPR regulation documents. Beyond development effort, it also saves the need for domain experts to collaborate with development teams to assert their right understanding of the legal concepts of the GDPR.
- **Actors notification**: The notification microservice informs any processor or third parties (Art. 4). who have access to and process data subjects' personal data of any changes concerning their decisions, for example in the event of consent withdrawal. The contact details of all the processors are managed by the actor microservice.

4 Related Work

Privacy has been intensively studied after the adoption of regulations such as the GDPR. Among the key issues, managing user consent is crucial and requires effective solutions to be integrated to applications. Existing work has for instance proposed ontologies and conceptual models [9] [10] [11] [17]. Despite the importance of domain knowledge formalization and standardization, these work mainly propose conceptual solutions that do not tackle application development with concrete software artifacts. Other works have proposed solutions to check the compliance of applications to the GDPR, based on privacy policies descriptions [3] [7]. Both works propose a corrective approach that does prevent privacy breaches, as a right management architecture. Blockchains have also been studied as consent management solutions [4], [6], [13], [15], [18]. However, blockchain does not address directly access control requirements. It is nonetheless a technical solution that is relevant for the unforgeable, auditable recording of important legal information such as consents.

A microservice model including a data access control layer is proposed in [8]. It ensures GDPR compliance by managing status, consent validity, and roles. This solution thus requires personal data to be stored within the microservices, implying deep re-engineering of existing applications.

5 Conclusion

This paper proposes the architectural design of a solution for managing privacy by leveraging access control. Regulations like GDPR provide clear and precise recommendations for guaranteeing privacy through personal data protection. One of the pillars of privacy is consent management. Consents should be obtained from individuals in a transparent and comprehensible manner. Their withdrawal should be performed in a simple and accessible way.
To address these issues, this work stresses the importance of providing "an administration console" for managing the consents of data subjects. This console enables:

1. applications providers to list the processing operations and the personal data used to run applications;
2. data subjects to give their consents in a fine-grained way, i.e. for each processing operation, thanks to clear and detailed user interfaces that list the purpose of the processing, the involved personal data, etc. Conversely, consents can be withdrawn in as simple and accessible way using toggles;
3. developers to be assisted in integrating consent management in a seamless and non-intrusive way, by leveraging interoperability patterns, like OAuth2 and OIDC, and the ABAC access control patterns.

The proposed solution has been furthermore specifically designed to minimize integration effort in applications. Feasibility of its integration has been concretely experimented on a use case based on a reference third-party RESTful application.
As the reader may notice, the solution we propose is intended to be as least intrusive as possible.
As a future work, we plan to complete the architecture, building on the consent management prototype developed for this paper. This includes finalizing the remaining microservices and conducting large-scale experiments to evaluate the solution's effectiveness in managing real-world business processes. We will test across various application architectures, such as distributed systems and monolithic applications, with a focus on scalability, performance, and complexity. In the long term, we aim to automate the identification of personal data processing for consent collection and adapt our approach to support other legal regulations.

References

1. Keycloak. `https://www.keycloak.org/`. (Accessed on 07/18/2024).
2. What is GDPR, the EU's new data protection law? `https://gdpr.eu/what-is-gdpr/`. (Accessed on 12/06/2024).
3. P. A. Bonatti, L. Sauro, and J. Langens. Representing consent and policies for compliance. In *IEEE Europ. Symp. on Security and Privacy Wkshps*, pages 283–291, virtual event, Sept. 2021. IEEE.
4. O. Can, T. Dag, and M. Kantarcioglu. A blockchain based hybrid architecture for auditable consent management. *IEEE Access*, 2024.
5. V. C. Hu, D. Ferraiolo, R. Kuhn, and A. Schnitzer, *et al*. Guide to attribute based access control (ABAC) definition and considerations. *NIST special publication*, 800(162):1–37, Jan. 2014.

6. F. Jaafar, D. Amayed, W. Salhab, and H. Bouani, *et al*. Blockchain-based consent management for privacy persevering and transparency in intelligent surveillance systems. In *5th Int. Conf. on Blockchain Computing and Applications*, pages 284–293, Kuwait, Kuwait, Oct. 2023. IEEE.
7. S. Kirrane, J. D. Fernández, P. Bonatti, and U. Milosevic, *et al*. The SPECIAL-K personal data processing transparency and compliance platform. *CoRR*, abs/2001.09461, 2020. https://arxiv.org/abs/2001.09461.
8. B. Mashaly, S. Selim, A. H. Yousef, and K. M. Fouad. Privacy by design: A microservices-based software architecture approach. In *2nd Int. Mobile, Intelligent, and Ubiquitous Computing Conf.*, pages 357–364, Cairo, Egypt, May 2022. IEEE.
9. D. M. Nogueira, C. Maciel, J. Viterbo, and D. Vecchiato. A privacy-driven data management model for smart personal assistants. In T. Tryfonas, editor, *5th Int. Conf. on Human Aspects of Information Security, Privacy and Trust*, volume 10292 of *LNCS*, pages 722–738. Springer, Vancouver, Canada, July 2017.
10. M. Palmirani, M. Martoni, A. Rossi, and C. Bartolini, *et al*. Pronto: Privacy ontology for legal reasoning. In A. Kö and E. Francesconi, editors, *7th Int. Conf. on Electronic Government and the Information Systems Perspective*, volume 11032 of *LNCS*, pages 139–152. Springer, Regensburg, Germany, Sept. 2018.
11. H. J. Pandit, Ch. Debruyne, D. O'Sullivan, and D. Lewis. GConsent - a consent ontology based on the GDPR. In M. Fernández et al. P. Hitzler, editor, *16th Int. Conf. on The Semantic Web*, volume 11503 of *LNCS*, pages 270–282. Springer, Portorož, Slovenia, June 2019.
12. J. Richer and A. Sanso. *OAuth 2 in action*. Manning, March 2017.
13. I. Román-Martínez, J. Calvillo-Arbizu, V. J. Mayor-Gallego, and G. Madinabeitia-Luque, *et al*. Blockchain-based service-oriented architecture for consent management, access control, and auditing. *IEEE Access*, 11:12727–12741, Feb. 2023.
14. R. S. Sandhu. Role-based access control. In M. V. Zelkowitz, editor, *Advances in Computers*, volume 46, pages 237–286. Elsevier, 1998.
15. J. Schramm and T. Eichinger. Towards building GDPR-friendly consent management systems on top of self-sovereign identity ecosystems. In *Open Identity Summit*, pages 93–102, Porto, Portugal, June 2024.
16. Y. Smirnova and V. Travieso-Morales. Understanding challenges of GDPR implementation in business enterprises: a systematic literature review. *Int. Journal of Law and Management*, 66(3):326–344, Apr. 2024.
17. A. Toumia, S. Szoniecky, and I. Saleh. ColPri: Towards a collaborative privacy knowledge management ontology for the Internet of Things. In *5th Int. Conf. on Fog and Mobile Edge Computing*, pages 150–157, Paris, France, Apr. 2020. IEEE.
18. J. C. Vargas. Blockchain-based consent manager for GDPR compliance. In H. Roßnagel *et al.*, editor, *Open Identity Summit*, pages 165–170, Garmisch-Partenkirchen, Germany, March 2019.
19. J. von Kistowski, S. Eismann, N. Schmitt, and A. Bauer, *et al*. TeaStore: A micro-service reference application for benchmarking, modeling and resource management research. In *26th IEEE Int. Symp. on the Modelling, Analysis, and Simulation of Computer and Telecommunication Systems*, pages 223–236. IEEE, Sept. 2018.