A Blockchain-driven Architecture for Usage Control in Solid

Davide Basile, Claudio Di Ciccio, and Valerio Goretti
Department of Computer Science
Sapienza University of Rome, Italy

Sabrina Kirrane
Department of Information Systems and Operations
Vienna University of Economics and Business, Austria

***Abstract*—Decentralization initiatives like Solid and Digi.me enable data owners to control who has access to their data and to stimulate innovation by creating both application and data markets. Once data owners share their data with others, though, it is no longer possible for them to control how their data are used. To address this issue, we propose a usage control architecture to monitor compliance with usage control policies. To this end, our solution relies on blockchain and trusted execution environments. We demonstrate the potential of the architecture by describing the various workflows needed to realize a motivating use case scenario for data markets. Additionally, we discuss the merits of the approach from privacy, security, integrateability, and affordability perspectives.**

***Index Terms*—Decentralized applications; Blockchain; Smart contracts; Trusted execution environment; Distributed architectures.**

I. Introduction

Decentralized projects like *Solid*[1] and *Digi.me*[2] seek to increase data owners' control over their data while also giving people and small organizations access to information that is typically managed by centralized platforms. The Solid community aims to achieve this objective by building web standards and best practices that make data integration simple and encourage the creation of decentralized social apps based on Linked Data concepts. To provide individuals with more control over their data, Digi.me develops technologies thanks to which users can encrypt and collect their information from centralized platforms in personal datastores.

In both cases, there is potential for new data and application markets. Protocols that design interactions with distributed data stores are essential to work with various data resources that may come with distinct terms and conditions specified by data owners for data sharing. Those terms and conditions typically come in two forms. *Access control* takes place *before* granting information access [1]. *Usage control* extends the former as its enforcement requires *runtime* monitoring of data consumption at a remote location.

A large body of research work improves control and transparency in personal data processing by utilizing blockchain-based distributed application platforms [2]. Ayoade et al. [3] propose a framework wherein blockchain applications are used to manage access to data that are stored off-chain in a trusted execution environment. Zhaofeng et al. [4] introduce a secure usage control scheme for Internet of Things (IoT) data that are built upon a blockchain-based trust management approach. Khan et al. [5] present the *DistU* distributed usage control framework, which applies the UCON$_{ABC}$ [6] model to the Hyperledger Fabric[3] permissioned blockchain. Xiao et al. [7] propose a system called *PrivacyGuard*, which leverages blockchain technologies to share usage policies, records resource usage, and monitors policy compliance in a data market scenario. Furthermore, several research studies propose integrations of the Solid protocol with blockchain technologies. Ramachandran et al. [8] show three possible configurations which combine blockchain with Solid to verify resource integrity, represent resources as smart contracts, and manage crypto wallets through off-chain personal online datastores (*pods*). Cai et al. [9] present a blockchain-assisted system implementing access control policies as a secure authentication mechanism for Solid. Becker et al. [10] propose a blockchain-based payment protocol to build a monetization framework for data stored in Solid personal online datastores. Havur et al. [11] show a decentralized layered architecture supporting the intersection of the SPECIAL[4] policy language with Solid standards integrated into personal online datastores.

Despite these efforts, Solid currently only supports basic access control, and thus it is not possible to ensure that data consumers adhere to usage restrictions specified by data owners.

To overcome this limitation, we propose a decentralized usage control architecture that resorts to a blend of blockchain applications and trusted execution environments. We extend the state of the art by demonstrating (i) how *blockchain oracles* [12] allow for seamless communication between these entities, and (ii) how Solid applications [13] can be enhanced with usage control mechanisms. In the proposed architecture, users' data are kept in Solid personal online datastores. Access is administered through a component named *pod manager*.

The work of S. Kirrane is funded by the FWF Austrian Science Fund and the Internet Foundation Austria under the FWF Elise Richter and netidee SCIENCE programmes as project number V 759-N. The work of D. Basile, C. Di Ciccio and V. Goretti was partly supported by projects SERICS (PE00000014) under the NRRP MUR program funded by the EU-NGEU, PINPOINT (B87G22000450001) under the MUR PRIN programme, and by the Sapienza project "Drones as a Service for First Emergency Response".

[1] https://solidproject.org/. Accessed: August 4, 2023.

[2] https://digi.me/. Accessed: August 4, 2023.

[3] https://www.hyperledger.org/use/fabric. Accessed: August 4, 2023.

[4] https://ai.wu.ac.at/policies/policylanguage/ Accessed: August 4, 2023.

The usage control is handled by blockchain executable applications that are capable of (i) recording where data resides, (ii) declaring what the usage restrictions are, and (iii) monitoring compliance with these policies. Applications that leverage data stored in Solid pods run in a *trusted execution environment* [14], which enables users to revoke access if data consumers do not adhere to the usage policies. Finally, blockchain oracles enable pod managers and trusted execution environments to communicate with the blockchain and vice versa. We illustrate the application of our architecture and highlight its effectiveness in the in the context of data markets.

Next, Section II describes a motivating scenario we employ as a running example throughout this paper. Section III presents the software architecture at the core of our solution. Section IV illustrates the application of our approach to the motivating scenario. Section V evaluates our approach through the lens of four key properties. Finally, Section VI concludes the paper and outlines possible endeavors for future work.

II. Motivating Use Case Scenario

Alice and Bob sign up for a new decentralized data market service for data trading across datastores. Their accounts include contact details, subscription details, and a username and password for the service. They set up a personal datastore on a server of their choosing, wherein they add the data that they would like to trade. Alice and Bob employ usage policies to set usage restrictions on their data. Bob's dataset contains medical data to be used only for medical purposes. Alice's dataset contains internet-browsing datasets, which must be deleted one month after their storage. Alice and Bob send metadata associated with the data that they would like to trade alongside the usage policies to the decentralized data market.

Alice is a researcher in the healthcare domain. She is interested in Bob's medical dataset. She asks the service for a data reference and a certificate proving she has paid the market fee. Alice uses the reference to contact Bob's personal datastore and check the certificate's validity. Thereafter, it returns Bob's medical dataset and the associated usage policy. Similarly, Bob, a web data analyst, wishes to retrieve Alice's internet-browsing dataset from her personal online datastore. Alice and Bob only use the data obtained from the market on their trusted devices (which is part of the terms and conditions stipulated by the market), in which a trusted software component enforces policies. This ensures that Alice's dataset is deleted after one week of usage and Bob's healthcare data are only used for medical purposes. Alice asks the market service to check that the usage policy associated with her datasets is being adhered to. In this case, trusted devices storing her resources provide her with evidence of the policies' compliance. At any point, Alice and Bob can change the rules associated with their datasets. In particular, after two days, Alice changes the maximum storage time of her internet-browsing data to one week. In the meantime, Bob modifies the allowed purpose of use of his medical resources to academic pursuits. Trusted devices guarantee ongoing policies update after the information retrieval, thus catching Alice and Bob's policy updates from the market service. As a result, Alice's data are erased from Bob's device after the new expiry time lapses. As Alice is using an application in the medical research domain for a university hospital, changes do not affect her access grants.

III. Decentralized Usage Control Architecture

To cater for our motivating use case scenario, we propose an architecture that extends Solid with usage control capabilities. More specifically, we build upon the existing Solid infrastructure, which we enhance to (i) continuously monitor compliance with usage policies and (ii) enforce the fulfillment of usage policy obligations after access to data has been granted. Figure 1 depicts the proposed architecture. We describe it in detail below.

A. Pods, Pod Managers, and the Solid Protocol

Our architecture extends the Solid protocol, whose main goal is to support decentralized data storage and application development [13]. Solid applications communicate with personal data stores called `Pods`, according to the Solid communication rules, via `Pod Managers`. The `Pod Manager` is a web application that allow users to retrieve, modify and control data that are stored in a Solid `Pod`. Thus, the `Pod Manager` determines whether access can be granted by checking the access control policies that are stored locally.

However, once data are retrieved from the Solid `Pod`, it is not possible for Solid to control how data are subsequently used. Thus, we combine the Solid infrastructure with a distributed blockchain application that facilitates usage control after data have been retrieved.

B. Blockchain, DistExchange Application, and Usage Policies

Modern blockchain technologies offer trusted and secure environments not only for classical data storage but also for the execution of applications that run on distributed virtual machines [15]. The correctness of the executed code is validated by the consensus mechanism of the blockchain.

In Fig. 1, we enclose the multiple software elements we deploy on the blockchain infrastructure in a dedicated macro-component labeled as `Blockchain`, which we leverage for multiple aims. First of all, we resort to its ledger to store references to the physical location of Solid `Pods`, as well as specific `Resource Location` and applicable `Usage Policies`. Additionally, we resort to the distributed virtual machine running *smart contracts* to develop a `DistExchange Application` (`DE App`) that is capable of monitoring compliance with usage control policies. For instance, a `Usage Policy` may specify temporal obligations that state the duration of usage for a particular resource (e.g., the one-week expiry of Alice's web data) and purpose obligations that constrain resource usage to a given purpose (e.g., medical research as the sole access aim for Bob's data).

The `DE App` is responsible for monitoring compliance with every `Usage Policy` and detecting policy violations. It relies on the `Trusted Execution Environment` hosted by data consumer devices to enforce `Usage Policy`.

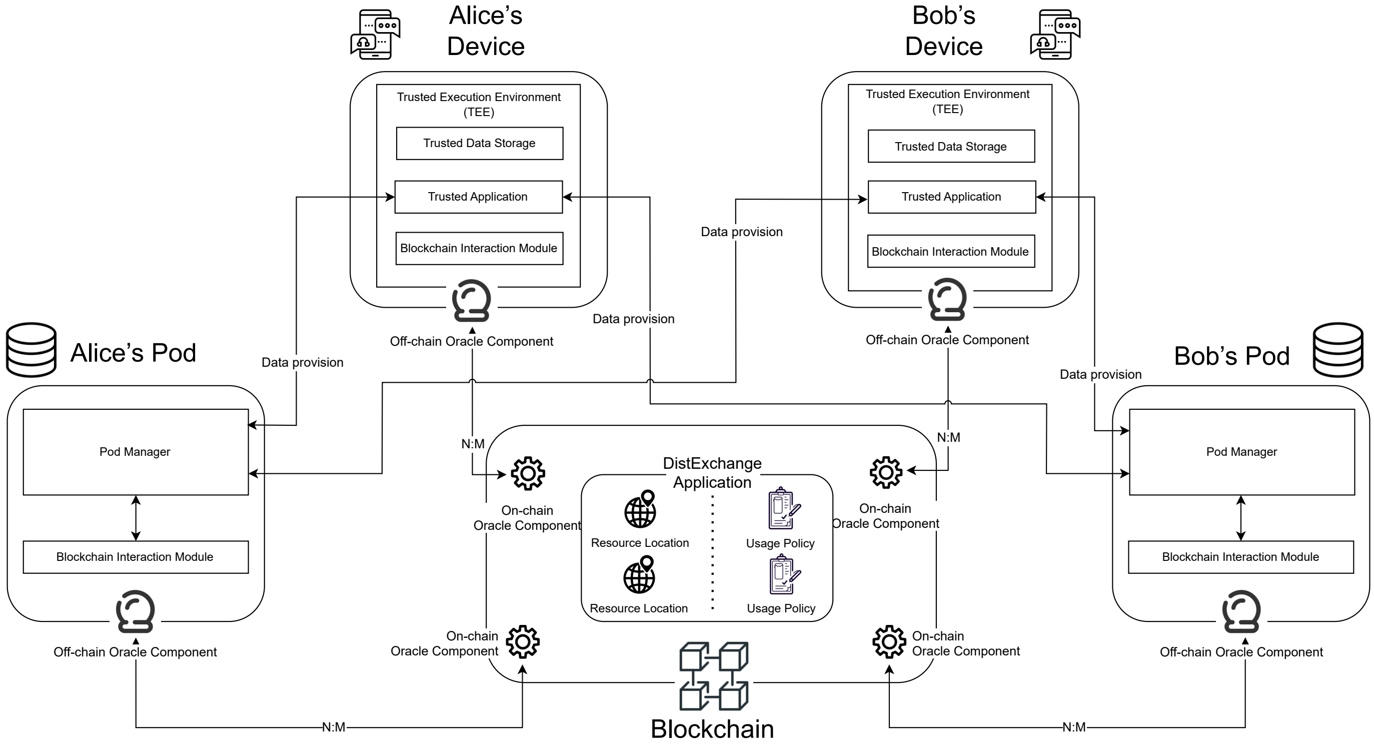

Fig. 1. A Decentralised Usage Control Architecture

C. Trusted Execution Environment

A Trusted Execution Environment is composed of hardware and software that ensures the protection of sensitive data by providing isolated execution, application integrity, and data confidentiality [14]. A Trusted Application is a software object running in a Trusted Execution Environment. Our infrastructure imposes that Solid client requests are generated by Trusted Applications. A copy of the requested data is stored locally and managed by the Trusted Execution Environment through the Trusted Data Storage. Local access to the Trusted Data Storage is controlled by the Trusted Execution Environment according to the Usage Policy. For instance, consider the temporal obligation on Alice's data. In this case, the Trusted Execution Environment automatically deletes the resource from the Trusted Data Storage after one week has passed, as per the policy. The Trusted Execution Environment logs resource usage, too. This feature facilitates policy monitoring whereby the Blockchain regularly interacts with the Trusted Execution Environment in order to ensure that usage policies are being adhered to. For instance, Bob can routinely check who the granted users to his data are and what use they are making of his shared information. Pod Managers and Trusted Execution Environments communicate with the Blockchain and vice versa via blockchain oracles.

D. Communication via Blockchain Oracles

Given that blockchains are closed environments, applications running in the blockchain ecosystem cannot natively communicate with entities located outside the network. For this reason, communication mechanisms called oracles are needed in order to connect the on-chain to the off-chain world [16]. Oracles are trusted entities used to facilitate data flow from the on-chain apps to real-world software and vice versa. We classify oracles according to two criteria: flow direction (in-bound/out-bound) and data operation (pull-based/push-based). Considering these criteria, it is possible to distinguish four types of oracles: *push-in*, *push-out*, *pull-in*, and *pull-out*. To be realized, oracles are split into two core parts. One lies off-chain and the other lies on-chain [12].

In the proposed architecture, the off-chain entities that communicate with the Blockchain are Pod Managers and the Trusted Execution Environments hosted on data consumer devices. These applications interact with the Blockchain via Blockchain Interaction Modules and the respective Off-chain Oracle Components. We assume that each off-chain entity has the credentials necessary to sign transactions and send data to the Blockchain.

IV. An Instantiation of the Architecture

We demonstrate the effectiveness of the proposed architecture by revisiting our motivating use case scenario. To this end, we analyze the data flows and interactions among components, which we separate into subsequent processes. We specifically focus on the interaction between Pod Managers, Trusted Execution Environments and the DE App. Thus, we do not go into specifics in terms of setting up the market and

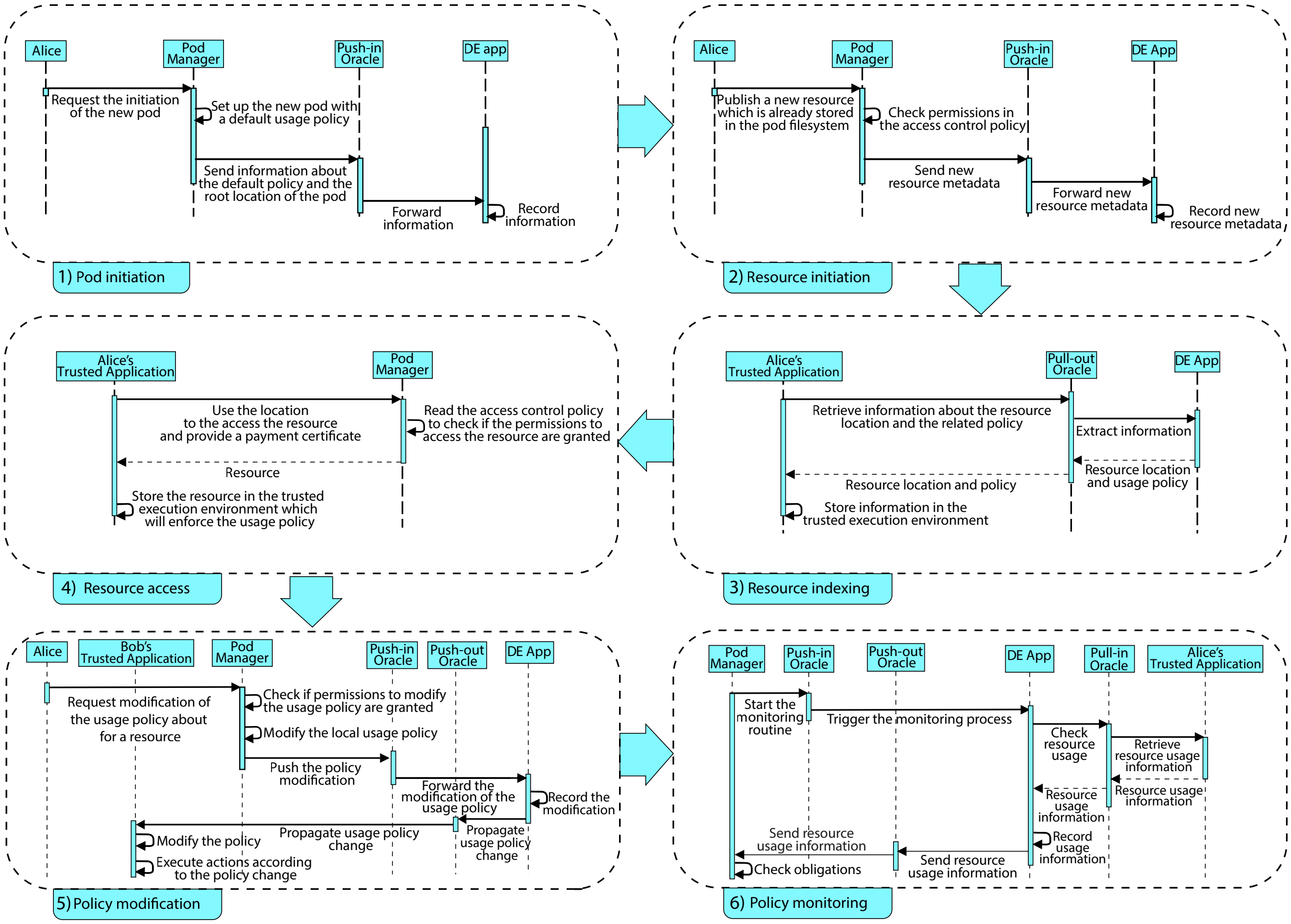
Alice
Pod Manager
Push-in Oracle
DE app
Request the initiation of the new pod
Set up the new pod with a default usage policy
Send information about the default policy and the root location of the pod
Forward information
Record information
1) Pod initiation
Alice
Pod Manager
Push-in Oracle
DE App
Publish a new resource which is already stored in the pod filesystem
Check permissions in the access control policy
Send new resource metadata
Forward new resource metadata
Record new resource metadata
2) Resource initiation
Alice's Trusted Application
Pull-out Oracle
DE App
Retrieve information about the resource location and the related policy
Extract information
Resource location and usage policy
Resource location and policy
Store information in the trusted execution environment
3) Resource indexing
Alice's Trusted Application
Pod Manager
Use the location to the access the resource and provide a payment certificate
Read the access control policy to check if the permissions to access the resource are granted
Resource
Store the resource in the trusted execution environment which will enforce the usage policy
4) Resource access
Alice
Bob's Trusted Application
Pod Manager
Push-in Oracle
Push-out Oracle
DE App
Request modification of the usage policy about for a resource
Check if permissions to modify the usage policy are granted
Modify the local usage policy
Push the policy modification
Forward the modification of the usage policy
Record the modification
Propagate usage policy change
Propagate usage policy change
Modify the policy
Execute actions according to the policy change
5) Policy modification
Pod Manager
Push-in Oracle
Push-out Oracle
DE App
Pull-in Oracle
Alice's Trusted Application
Start the monitoring routine
Trigger the monitoring process
Check resource usage
Retrieve resource usage information
Resource usage information
Resource usage information
Record usage information
Send resource usage information
Send resource usage information
Check obligations
6) Policy monitoring

Fig. 2. Decentralised Usage Control Architecture Processes.

catering for new registrations. The various processes, which are depicted in Fig. 2, are described in detail below.

*1) **Pod initiation**:* Once Alice and Bob have registered with the DE App, they need to link their Solid Pods to their respective data market accounts. The process starts when Alice makes a request to the Pod Manager to initialize a new Pod. The Pod Manager sets up the Pod with its default policy (e.g., only subscribed users have access to the data). Thereupon, it invokes a Push-in Oracle to send information about the Pod's web reference and its default policy to the blockchain smart contract that is part of the DE App. The process for Bob is analogous.

*2) **Resource initiation**:* The resource initiation process is used in order to add a new resource to the DE App. The process begins when Alice asks her Pod Manager to add a resource to the market that has already been uploaded in her Pod's filesystem via the Solid protocol. The Pod Manager first checks that Alice is permitted to perform this action. If so, the Pod Manager uses the Push-in Oracle to forward the necessary metadata to the DE App (i.e., a reference to the resource and possibly a resource-specific usage policy), which adds the resource's metadata to the index and publishes the applicable Usage Policy.

*3) **Resource indexing**:* With this process, users retrieve a link to a resource that is initialized in the DE App. Alice is interested in the medical data that Bob has added to the market. Given that Alice does not know the exact web location of the resource, she asks the DE App for a link to it alongside the corresponding usage policy. The process is initiated when Alice requests information about the resource (i.e., the aforementioned web link and policy). Alice's Trusted Application generates the request, running in the Trusted Execution Environment. It uses the Pull-out Oracle to read this piece of information directly from the DE App running in the Blockchain. The retrieved information is stored in Alice's Trusted Execution Environment and can subsequently be used to retrieve the resource physically.

*4) **Resource access**:* The resource access process allows data consumers to retrieve information stored in a Solid Pod. In order to collect Bob's data, Alice's Trusted Application makes a request from within the Trusted Execution Environment to the Pod Manager. The request includes a certificate that proves she has paid the market fee. The Trusted Application provides the Pod Manager with a reference for the resource that it obtained from the DE App via the above process of resource indexing. The Pod Manager first checks that Alice is permitted to perform the read action. If so, the Pod Manager returns the resource to Alice's Trusted Application. In turn, Alice's Trusted Application stores it within its Trusted Data Storage.

*5) **Policy modification**:* The policy modification process enables users to update usage policies after resources have been deployed to the DE App. For instance, Alice shortens the time lapse for the usage of her internet browsing data to one week, whereas it was initially set to one month. Here we assume that such updates are permitted according to the general rules of the market. Alice makes a request to her Pod Manager to change the Usage Policy for that resource. The Pod Manager checks whether Alice is granted the permission to change the policy. If so, it proceeds with the update locally. The Pod Manager, then, uses the Push-in Oracle to send the updated policy to the DE App, which replaces the policy accordingly. The DE App uses a Push-out Oracle to notify those users that have a copy of the resource (e.g., Bob) that the policy has been updated. The Trusted Applications bearing a copy of the resource (e.g., Bob's Trusted Application) update their local policies, check if the change requires any actions to be executed locally, and if so, execute them. In Bob's case, the consequent action to be taken is the erasure of the collected data if the check happens after one week from the first download. Notice that this mechanism is automatically enforceable since data are entirely and solely stored in the Trusted Data Storage as described above.

*6) **Policy monitoring**:* The policy monitoring process regularly checks usage policy compliance once data are accessed. The Pod Manager uses the Push-in Oracle to start the monitoring (for instance, via a scheduled job). The Push-in Oracle forwards the request to the DE App, which in turn communicates with all devices that have a copy of the resource in their Trusted Execution Environment via the Pull-in Oracle. The Pull-in Oracle, then, requests evidence that the usage policies are being adhered to. The Push-out Oracle is subsequently required by the DE App to send the pieces of evidence gathered from the various trusted applications (for instance, Alice's Trusted Application) to the Pod Manager that initiated the policy monitoring process.

V. Discussion

In the following, we expand the discussion of our decentralized usage control architecture, paying special attention to the properties of privacy, security, integrateability and affordability.

*1) **Privacy**:* According to the Solid protocol, data owners decide which entities (authenticated or unauthenticated) can access their resources via Access Control Lists (ACLs). This requirement has a significant impact on privacy and data confidentiality, as the need to subscribe to terms and conditions specified by applications is eliminated. However, Solid principles entail that data are kept in specific user-trusted datastores. Such a design choice can represent a limitation for computationally intensive web applications, which are forced to retrieve data from several data sources.

The establishment of usage control through Blockchain applications and Trusted Execution Environments allows data owners to keep control over their data (even after data consumers have obtained copies thereof) and further supports the Solid principle of data ownership. At the same time, Trusted Execution Environments facilitate compliant data storage and resource usage by implementing usage policy enforcement and related obligations. After the resource retrieval, Trusted Applications benefit from locally stored data (as long as the Usage Policy permit it) without the need to constantly communicate with Solid Pods, which leads to significant improvements in latency and scalability.

The most critical issue regarding confidentiality relates to the blockchain metadata, which are publicly exposed in most cases. Public blockchains offer public ledgers that are fully readable by every node of the network. In our setting, this availability implies that all users can read usage policies and resource locations. Although making this information public can be desirable on occasions, data owners might request that only authorized parties (e.g., those with access to the decryption key) can access this information. Typically, approaches that achieve this objective in a blockchain context are based on encryption [17], [18].

*2) **Security**:* In a decentralized web environment, the lack of a central authority increases the chances that malicious users make unauthorized use of data and metadata managed by the infrastructure. However, the integrity of user data residing in Pods is already guaranteed by the Solid protocol through access control policies. The blockchain's consensus algorithm and its distributed nature protect the stored metadata (resource locations and usage policies) from unauthorized modifications, making this information tamper-proof. Moreover, methods through which the state of smart contracts is changed can be invoked only by signing transactions with auditable digital signatures. The Trusted Execution Environment provides a separate environment for code execution and data storage. Studies such as the one conducted by Sabt et al. [14] already showed the effectiveness of these technologies in preventing the execution of malicious code from the operating system's machine, which could compromise the integrity of resources and usage policies stored inside the Trusted Execution Environment. Interactions between the various components that could lead to the modification of resources or usage policies are managed via off-chain and on-chain oracle components, which are able to enact secure information exchange between the blockchain and outer parties [19].

The availability of the DE app is preserved by the distributed nature of the blockchain. If an attack succeeds in bringing down one of the nodes, the blockchain ecosystem can continue to operate by relying on the rest of the nodes. However, both the Solid Pods and the Trusted Execution Environments hosted on user devices need to adopt best practices in terms of hardware and software security to guarantee communication with the blockchain platform.

*3) **Integrateability**:* One of the requirements that steer our design process is the need to easily integrate our architecture with the existing Solid ecosystem, so that pod data management functionality could be extended to cater to usage control. Pods interact with blockchain applications via a plug-in module, which enables subscription, usage policy specification, and resource indexing. On the client side, an additional requirement on the hardware is set by the fact that `Trusted Execution Environments` rely on separation kernel methodologies through hardware support [14]. However, this kind of technology is supported via various extensions to existing operating systems.

*4) **Affordability**:* Public blockchains use the tamper-proof register feature to define cryptocurrencies whose transactions are stored in blocks. The execution of on-chain code requires that cryptocurrencies are spent, depending on the computational effort required by the run of the code.

Resorting to a public blockchain, users of our infrastructure would make a payment to interact with the blockchain metadata through transactions. The market scenario can justify the costs involved in our architecture. A subscription-based business model could offer an incentive mechanism that allows users to overcome the sharing costs and earn a remuneration upon access to their data. Therefore, blockchain applications provide an easy way to guarantee a market profit redistribution to users, proportionately to the accesses granted to their data. The economic incentives are out of scope for this paper and pave the path for future work.

VI. Conclusions

Motivated by the need to ensure that data consumers adhere to usage restrictions specified by Solid data owners, we proposed a decentralized usage control web architecture that extends existing Solid access control mechanisms to cater to usage control. The effectiveness of the proposed architecture is demonstrated with the help of a motivating use case scenario in the context of data markets. Additionally, we examined the proposed architecture from privacy, security, integrateability, and affordability perspectives.

Future work includes the integration of a policy language that can be used to specific usage policies at different levels of granularity. We are also interested in the study and design of economic mechanisms supporting the data market adoption. The proposed architecture generalizes the blockchain concept, although a wide variety of technologies are currently available. Following the comparative methodology proposed in [20], we plan to instantiate a specific blockchain technology that meets the technological requirements evidenced by our decentralized usage control scenario [21]. An analogous analysis will be applied to the multitude of trusted execution environment technologies available, including Intel SGX [22], TrustZone [23] and OpenTEE [24]. The instantiation process will allow us to evaluate the architecture from the perspectives of performance, scalability, and robustness.

References

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

[2] X. Xu, I. Weber, and M. Staples, *Architecture for Blockchain Applications*. 2019.

[3] G. Ayoade, V. Karande, L. Khan, and K. Hamlen, "Decentralized IoT data management using blockchain and trusted execution environment," in *IRI*, pp. 15–22, 2018.

[4] M. Zhaofeng, W. Lingyun, W. Xiaochang, W. Zhen, and Z. Weizhe, "Blockchain-enabled decentralized trust management and secure usage control of iot big data," *IEEE Internet Things*, vol. 7, no. 5, pp. 4000–4015, 2020.

[5] M. Y. Khan, M. F. Zuhairi, T. A. Syed, T. G. Alghamdi, and J. A. Marmolejo-Saucedo, "An extended access control model for permissioned blockchain frameworks," *Wirel. Networks*, vol. 26, no. 7, pp. 4943–4954, 2020.

[6] J. Park and R. Sandhu, "The uconabc usage control model," *ACM transactions on information and system security (TISSEC)*, vol. 7, no. 1, pp. 128–174, 2004.

[7] Y. Xiao, N. Zhang, J. Li, W. Lou, and Y. T. Hou, "Privacyguard: Enforcing private data usage control with blockchain and attested off-chain contract execution," in *Computer Security – ESORICS 2020* (L. Chen, N. Li, K. Liang, and S. Schneider, eds.), pp. 610–629, 2020.

[8] M. Ramachandran, N. Chowdhury, A. Third, J. Domingue, K. Quick, and M. Bachler, "Towards complete decentralised verification of data with confidentiality: Different ways to connect solid pods and blockchain," in *Companion Proceedings of the Web Conference 2020*, p. 645–649, 2020.

[9] T. Cai, Z. Yang, W. Chen, Z. Zheng, and Y. Yu, "A blockchain-assisted trust access authentication system for solid," *IEEE Access*, 2020.

[10] H. Becker, H. Vu, A. Katzenbach, C. H. Braun, and T. Käfer, "Monetising resources on a solid pod using blockchain transactions," in *The Semantic Web: ESWC 2021 Satellite Events*, pp. 49–53, 2021.

[11] G. Havur, M. Vander Sande, and S. Kirrane, "Greater control and transparency in personal data processing," in *ICISSP*, 2020.

[12] D. Basile, V. Goretti, C. Di Ciccio, and S. Kirrane, "Enhancing blockchain-based processes with decentralized oracles," in *BPM (Blockchain and RPA Forum)*, pp. 102–118, 2021.

[13] A. V. Sambra, E. Mansour, S. Hawke, M. Zereba, N. Greco, A. Ghanem, D. Zagidulin, A. Aboulnaga, and T. Berners-Lee, "Solid: a platform for decentralized social applications based on linked data," 2016.

[14] M. Sabt, M. Achemlal, and A. Bouabdallah, "Trusted execution environment: What it is, and what it is not," in *TrustCom/BigDataSE/ISPA*, pp. 57–64, 2015.

[15] W. Cai, Z. Wang, J. B. Ernst, Z. Hong, C. Feng, and V. C. Leung, "Decentralized applications: The blockchain-empowered software system," *IEEE Access*, vol. 6, pp. 53019–53033, 2018.

[16] R. Mühlberger, S. Bachhofner, E. C. Ferrer, C. Di Ciccio, I. Weber, M. Wöhrer, and U. Zdun, "Foundational oracle patterns: Connecting blockchain to the off-chain world," in *BPM (Blockchain and RPA Forum)*, pp. 35–51, 2020.

[17] J. Pan, S. Paul, and R. Jain, "A survey of the research on future internet architectures," *IEEE Commun. Mag.*, vol. 49, no. 7, pp. 26–36, 2011.

[18] E. Marangone, C. Di Ciccio, and I. Weber, "Fine-grained data access control for collaborative process execution on blockchain," in *BPM (Blockchain and RPA Forum)*, pp. 51–67, 2022.

[19] H. Al-Breiki, M. H. U. Rehman, K. Salah, and D. Svetinovic, "Trustworthy blockchain oracles: Review, comparison, and open research challenges," *IEEE Access*, vol. 8, pp. 85675–85685, 2020.

[20] D. Basile, I. D'Adamo, V. Goretti, and P. Rosa, "Digitalizing circular economy through blockchains: The blockchain circular economy index," *J. Ind. Prod. Eng.*, vol. 40, no. 4, pp. 233–245, 2023.

[21] D. Basile, C. Di Ciccio, V. Goretti, and S. Kirrane, "Blockchain based resource governance for decentralized web environments," *Frontiers in Blockchain*, vol. 6, p. 1141909, May 2023.

[22] V. Costan and S. Devadas, "Intel SGX explained," *Cryptology ePrint Archive*, 2016.

[23] S. Pinto and N. Santos, "Demystifying arm trustzone: A comprehensive survey," *ACM computing surveys (CSUR)*.

[24] B. McGillion, T. Dettenborn, T. Nyman, and N. Asokan, "Open-tee–an open virtual trusted execution environment," 2015.