A Decentralized Personal Data Store based on Ethereum: Towards GDPR Compliance

M.Alessi, A.Camillò, E.Giangreco, M.Matera, S.Pino, and D.Storelli

***Abstract*—Personal data sharing with service providers represents an unavoidable risk, due to issues like: improper data treatment, lack of users' awareness to whom they are sharing with, wrong or excessive data sharing from end users who ignore that they are exposing personal information. But sharing personal information, in the IoT era forces us to consider not only personal data, but also personal devices sharing. It becomes fundamental to consider users' awareness and centrality in the act of sharing, and resilience towards malicious third parties, which are problems that blockchain technologies are fit to solve. In order to make decentralized solutions usable for real, there is another challenge, the not simple compliance with the General Data Protection Regulation (GDPR), the European Authority has provided, in order to implement protection of sensitive data in each EU member. Such regulation protects sensible data throughout certification mechanisms (according to Art. 42 GDPR), which is a mandatory requirement for any service which may come in contact with sensitive data. The current paper offers a contribution, showing that the decentralized approach for personal data sharing, may be compliant not only with the requirement of users' centrality but also with GDPR, representing a novelty for IoT-ready personal data sharing management systems based on a distributed environment. This is possible by embedding the consent mechanism described by GDPR, within a real decentralized prototype developed to share personal data and devices. We present our approach and an architectural blueprint which evolves the prototype.**

***Index terms*—Security and Privacy for Iot, Privacy Challenges, Personal Data Storage, Blockchain application, GDPR, Profile management.**

I. INTRODUCTION

Breaks in data protection and security are a real problem, while data are shared everywhere at every minute, it becomes more and more unclear what exactly may happen to these often personal and sensitive data. Improper behaviors, wrong or excessive data sharing malicious data usage from subjects offering services over internet, often lead to data breaches, which may be imputable not only to people's fault, but above all to lack of awareness. Sharing personal contents on websites or social networks, is one of those actions which ofter lead to loss of control over personal data.

Manuscript received January 17, 2019; revised March 5, 2019. Date of publication April 15, 2019. Date of current version June 3, 2019.

Authors are with the Engineering Ingegneria Informatica S.p.a, Italy (e-mails: {marco.alessi}, {alessio.camillo}, {enza.giangreco}, {marco.matera}, {stefano.pino}, {davide.storelli} @eng.it).

Digital Object Identifier (DOI): 10.24138/jcomss.v15i2.696

Management, modification or cancellation of already shared information may sometime result impossible or lead to lawsuits[1].

Collections of users observed online behaviors often lead to highly valuable and huge data profiles, and the subjects of those profiles could have no clue these data exist, processed by some third party that the subject has never even heard of. The problem is so real, that from 25 May 2018 in all EU member states, the newest General Data Protection Regulation (GDPR) has become applied[1]. The General data protection regulation (GDPR), which entered into force in the european union in 2016 and into application in 2018, is the latest development in the european union's ongoing efforts to protect the personal data of its citizens. The users have to be informed and aware about the data they share, and for what purpose. According to the GDPR, the procession of personal data by any party requires either the consent of the data subject or a legal basis (Art. 6 GDPR). The fundamental vehicle for ensuring the lawful access to personal data, and ensure personal data protection as well, is the consent of the data subject (Art. 16 section 1 of the Treaty on the Functioning of the European Union, TFEU). And yet efforts must be made in order to make people have control over their personal data - what they are sharing and with whom, through a technological applications fully GDPR compliant. The answer to loss of control over personal data, is a solution which may handle sensible information, and the way information is disclosed towards third party services, in a way that the data subject is the solely owner and manager of the process of sharing. A Personal Data Management system is the optimal solution, which is not a novel concept, but it has evolved in time, even considering the modern IoT personal devices within the "sensible data" class[4]. Building such service on top of distributed environment represents a challenge: while on one hand it offers the opportunity to structure a technological tool to manage personal data, capable to embed the "privacy by design" paradigm, on the other hand there are still many issues to be addressed in order to meet a full GDPR compliance[18]. But the applicability of the Blockchain and how it could move closer to GDPR compliance is largely proposed and foreseen in many fields[18][5][6][7], even if it has not yet reached a common approach . This shows that exist a strong understanding around the Blockchain, and how it has the potential to handle sensible information, even providing suggestions for compliance from a legal perspective[8]. Nonetheless distributed technologies have the potential to

practically empower users with effective control over their personal data: give them ultimate awareness, manage the difficult event of "consent", provide insight in the processing of personal data sharing and without any central authority, which may have malicious intents. To be usable in real settings, such IoT-ready decentralized personal data management system must solve the apparent difficulty of the Blockchain to be GDPR certified.. The aim of the paper is to extend the functionalities offered by the PDS prototype we have developed so far[4], presenting as element of novelty a technological blueprint of how an IoT-ready Decentralized Personal Data Store may implement the users' "consent" action, as described in the GDPR, in a distributed Blockchain enabled environment. To provide a full advancement of the state of the art of IoT-ready personal data stores, we focus our work trying to solve decentralization issues with GDPR compliance, analyzing each one in order to propose, where possible, a solution embedded in our technical blueprint.

The rest of the paper is organized as follows. Section II gives insights about the problem of Personal Data Store and recent regulations. Section III presents how distributed technogies, at the state of the art, have tackled the problem of GDPR compliance. Section IV presents our decentralized Personal Data Store, and how it has been realized so far. In section V an evaluation of the proposed prototype is given in its basic scenarios developed to test the functionalities, finally evaluating its limits towards the new European regulation. In section VI we present our approach in order to evolve our decentralized PDS and meet the GDPR's requirement, thus a theoretical roadmap and an architecture is provided. Finally, section VII concludes with final outcomes and future work.

II. MOTIVATION

A Personal Data Store (PDS), or called personal data vault or locker, is a service allowing an individual store, manage, and deploy their key personal data in a highly secure and structured way[4][10][11]. Each user has not only control over his/her data: users have ownership, thus they can decide what services may access personal data store, and eventually what kind of data can or cannot be retrieved by those services. A general description of Personal Data Management systems has been given by Bus and Ngyuen[12]: they divide the objectives of a general PDS into three levels: infrastructure, data management, and user interaction.

• The infrastructure level has two main objectives: ensure integrity and confidentiality of data. That includes supporting all appropriate techniques like encryption, logging, monitoring, authentication and identification of protocols. For example, reliability and acceptability of an infrastructure can be verified through mechanisms of marketing; regular certified checks or forms of supervision over part of the infrastructure

• Data management level ensures safe and effective data control including permissions management mechanisms, communication policies, data auditing capabilities, etc. The most common approach to data reliability management is to create a contract between the user and the data controller by "giving responsibility" to the latter and making him/her aware of the fact if the required permissions are ignored.

• User Interaction is defined as "the element that enables end users to have a significant interaction with service providers, regarding permissions and policies associated with the use of their personal data".

The previous issues suggest that each PDS should offer simple and intuitive tools to control context-dependent data sharing, which all rely on trustworthy underlying data management and infrastructure layers. Many Personal Data Stores have been developed in order to solve the issues of personal data management, thus proving the importance of the matter.

Various data sharing approaches are possible: a user could decide to share some raw data, some aggregate data, or just a representative model of behavior. In addition, the data shared could be strictly related to the user or could be anonymized in order to assure privacy. Within the same PDS, users may be eligible not only to share their personal data to a particular subset of users, but also to access other users' data. Large scale development of PDSs and data control systems, is a long discussed subject in literature, and nowadays, more than ever, it is an unsolved and important issue.

Personal data stores may be divided into three main categories: centralized, decentralized and hybrid. The centralized PDS often make use of a central authority which is entitled to manage not only the service, but also the trust between users and services, and eventually acting as an intermediary if trust or legal issues arise. On the contrary, decentralized PDSs lack of a central authority, thus they have to implement mechanisms for regulating trust and data exchange. Hybrid approaches are also possible, where the management and trust is divided amongst users and a small number of reliable authorities. In the following section we describe various examples of PDSs which have been analyzed for both categories.

We have analyzed various examples of Persona Data Stores, available in literature, and how decentralization has been leveraged in handling personal data[4]. Some of the examples taken into account are MyDex[12] IRMA[14], OpenPDS[15], OnenameBITNATION[17]. Examples showed that, while decentralization was fundamental in increasing security, awareness and user inclusion in the process of data sharing, those lacked of a common general approach for handling personal data disclosure and sharing, together with IoT-device sharing. Moreover, the fact of dealing with personal data, a modern PDS has to face issues related to the European GDPR regulation, and more difficult than that, there are specific issues when trying to make a decentralized technology work under such legal conditions. The tensions between the GDPR[18] and blockchain revolve mainly around three issues:

- The identification and obligations of data controllers and processors. While there are many situations where data controllers and data processors can be identified and comply with their obligations, there are also cases where it is difficult, and perhaps impossible, to identify a data controller, particularly when blockchain transactions are written by the data subjects themselves.
- The anonymisation of personal data. There are intense debates, and currently no consensus, on what it takes to anonymise personal data to the point where the resulting output can potentially be stored in a blockchain network. to take one example, the hashing of data cannot be considered to be an anonymisation technique in many situations, and yet there are cases where the use of hashing

to generate unique digital signatures of data that is stored off-chain, is potentially conceivable on a blockchain.

- The exercise of some data subject rights, like the right to rectify or withdraw. We note that if personal data is recorded in a blockchain network, it may be difficult to rectify or remove it. Defining what can be considered erasure in the context of blockchains is under discussion. We may enlist various fundamental rights, which have to be deeply evaluated within a distributed system:
    - Right to be forgotten. One of the main challenges for blockchain developers to comply with the right in Art. 17 GDPR[6]: the right to erase the personal data wherever those may be stored.
    - Right to rectify. The data subject has the right to obtain rectification of inaccurate personal data concerning him or her, without undue delay, in Art. 16 GDPR.
    - Right to withdraw. Even more important than giving consent, the data subject have the right to withdraw his or her consent at any time, as seen in Art. 7 GDPR.

In this paper we present the experimentation on our decentralized Personal Data Store which has been originated by Servify project. Servify is the first project of SI-Lab for the development of skills and technological tools in User – Driven ICT – based Service Innovation[19]. Such research took us to develop the current Decentralized Identity Manager (DIM). In particular, the focus of the current work is to take into account the previous GDPR requirements, then to evolve the DIM, showing how a decentralized PDS, , may have a decentralized architecture and a process more GDPR compliant.

III. State of the Art

In this paragraph we describe the main projects about identity systems based on blockchain that we have taken into account, which have considered in their developing the problem of GDPR. We can distinguish among public permissionless, public permissioned and private permissioned. Uport[22] is one of the main identity systems for the decentralized web based on a public permissionless blockchain, Ethereum, that enable the creation and management of user digital identities, and is very similar to what we proposed. The ecosystem provides ways to make and request private and public claims to an identity, embedding a selective disclosure approach. Uport features are implemented following open standards such as JWTs and DIDs (Decentralized Identifiers), and trying to keep privacy and security of user data as the main focus. The GDPR compliance is more difficult to realize in this case, since that Uport is based on a public permissionless blockchain. To solve this issue Uport approach is to store on-chain only DIDs, which is a random string of characters that tells nothing about a user: it is a random public address that bind a user with other entities in order to interact with them. Actually, the personal data of the user is stored encrypted on IPFS, the distributed file system.

In order to cope with GDPR, many projects based on public permissioned blockchains are emerging. One example is provided by the Alastria[23] spanish consortium, that propose a national public permissioned blockchain infrastructure in order to enable the creation and provisioning of services with legal effectiveness in the Spanish country scope and respecting the European regulation. The proposed infrastructure is open to all institutions, SMEs and large enterprises, which are the entities responsible of running network nodes. Actually, the Alastria is based on Quorum, the Ethereum permissioned fork developed by JPMorgan that enable higher performance and scalability with respect to public permissionless blockchain infrastructures (e.g. Ethereum). The main focus of the consortium is the creation of a Self-Sovereign Identity blockchain-based named “Alastria ID”, GDPR compliant, that gives users complete control over their personal data. Alastria ID architecture allows on one hand authorities and entities to provide attestations to users, on the other users can share claims of this data with services they want to use. The identity framework of Alastria is based upon a modified version of uPort: the blockchain store only records evidences (hashes) while data is stored encrypted on IPFS.

Another project is led by Sovrin[24], a non-profit foundation, that is developing a self-sovereign identity network. It uses a public permissioned blockchain, based on the Hyperledger Indy project, which consists of nodes located around the world hosted and managed by a group of trusted organizations, called Stewards, who have agreed with the Sovrin Trust Framework. which provide GDRP compliance. Sovrin implements Privacy by design by providing pseudonimity by default, peer-to-peer private agents and selective disclosure of personal data. In Sovrin every user has multiple DIDs (Decentralized Identifier), one for each relation between a user and a third-party, and for each DID there is a corresponding private agent from which the identity owner can exchange verifiable claims and data with third parties through an encrypted peer-to-peer private channel[25]. The ledger store only pseudonymous identifiers, public keys and agent addresses, providing in this way the compliance with the European regulation. So, the data storage is responsibility of the private agent, and not of the ledger itself. Moreover, Sovrin implements selective disclosure through a cryptographic technique known as a zero-knowledge proof (ZKP) that enable the verification of claims without the necessity to read explicit data (e.g. a bartender can verify that a user is old enough to drink without knowing the birth date).

Private permissioned blockchains are used primarly for financial trading and supply chain management, as in the case of We.trade[26] and Tradelens[27]. All nodes in a private blockchain, both validating and participating, are approved by a consortium or by an organization and for this reason the GDPR compliance is very easy to reach. However, this context doesn’t fit well with a self-sovereign identity system, since that its fundamental goal is to manage and validate identity of every citizen in order to create relations/transactions with other entities: data on the blockchain need to be readable by everyone and not only by a small group of entities. Indeed, we didn’t find any project of decentralized identity based on private permissioned blockchain.

IV. Our Decentralized PDS - The Decentralized Identity Manager

The Decentralized Identity Manager component is developed in order to manage both a user's identity and user data. The most important feature of the component is to provide user profile data, both static and dynamic, necessary for the contextual personalization of pervasive services. On the one hand, users can exploit the component to create, edit and delete profile information, indended as personal data or devices; on the other hand, the services take advantage of information within this component, on user's permission, to provide a better contextualized experience. Decentralized because the first design purpose of the component is to let the users own their data, without considering a centralized authority or third-party services that the users have to trust. In recent years, centralized organizations, public and private, accumulate large amounts of sensitive and personal information, sometimes leading to tremendous scandals[28]. With this in mind, we have intended to implement the user identity and its management, having the data subject as the sole owner of his/her personal data. The user profile, on which the current component is based, derives from a multidimensional model. In order to conceptually represent the user model, it was decided to adopt schema.org[29] model. This choice is motivated by the fact that the ontology in question is one of the most widespread at this time. Among the ontology representation formats studied, it was decided to use the JSON-LD format, a choice conditioned by the fact that user information must also be exchanged between web services. Using this format, information is expressed more clearly and concisely and can also be human readable, which cannot be said for other markup languages. In addition there are numerous tools that allow to quickly verify the semantic correctness of the file.

A. The User Profile Schema

The model used as basement for user profile was derived by schema.org. The main schema.org entities considered were: *Person*, the starting point to describe the user and all its dimensions; *Product*, representation of the devices owned by the user; *Place* representation of places; *Action*, actions performed by a user (e.g. travel or driving). Such entities were the starting point to describe the user context. Since there was no specific schema.org entity who could describe the context, it was decided to provide context information by using the *Person*, *Action*, *Place*, *Product* entities respectively. Action identified a user's action, referenced through a relationship with the Person entity (for example, moving from one place to another or physical activity in a specific time frame).

This action was characterized by one or more execution places, described by the Place entity, and may involve one or more devices, described by the Product entity. Even in the case of a user's interests, schema.org did not provide supporting entities and attributes. A new *interests* attribute of the Person entity was therefore defined, as well as a new InterestTag entity, derived from the Thing entity. This included two attributes: name and weight. The first identified the name of the object of interest to the user. The second attribute identified a weight, i.e. how much the specific interest is strong for the user. A partial schematization of the proposed model is shown in Fig. 1. The entities and attributes in green are the extensions proposed to schema.org.

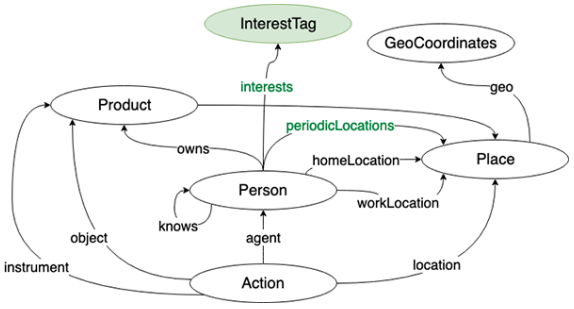

Fig. 1 Ontology schema defined for Decentralized Identity Manager

The Context attribute specified the chosen vocabulary. In the specific case this was equivalent to http://schema.org. This information allowed services and applications that receive such a file, to identify the ontology used for the description of the entity. The structure was divided into two public and private blocks, useful for the service that implemented the user identity to manage the privacy of information. In the public section were inserted all the attributes with a level of public privacy. In the private section, the list of all attributes with a private privacy level was inserted. The level of privacy was of great importance in this deployment, because it defined whether to make information visible to external services (public) or visible only to the owner (private). The attributes in the private section were be encrypted and visible only to the user who owns the profile. The *shares* attribute, not present in schema.org and introduced in this model, indicated all the applications with which the user had decided to share his/her profile information, thus making the model ready to build up the history of services to which data have been shared. The application name was specified thanks to the *service* attribute. The *owns* attribute was suitable to include the devices of a user (eg smartphones, computers, tablets) coded with the Product type. This entity included information about owned devices (device APIs, unique IDs, etc...), useful to optimally address devices by services. *homeLocation* and *workLocation* respectively identified the geographical location of the user's home and work site. We introduced a new attribute: *periodicLocations*. This had the purpose to describe the places frequently visited by the user (e.g. gym, supermarket, etc.). The social relations of the user were described by the attribute *knows*, and their preferences, as previously stated, by the *interests* attribute.

B. Technological Requirements

We have implemented a distributed approach to personal data sharing, taking into account technologies which could support the lack of a "central authority"[4].To do so we have chosen Ethereum[20] to handle a distributed environment for the personal data sharing process and IPFS[21] as distributed storage.

C. Deployment of the DIM Prototype

Writing on the blockchain involved a cost in terms of ether[4]. The greater the amount of information that needed to be saved, the greater the number of ether required. To save needless costs, we introduced IPFS as a "distributed storage", where to save the profile information of each user. In this way, we mantained relatively minimal the information had to be saved on the blockchain (low impact on the monetary ether cost), Only a limited amount of information was stored on

Ethereum for each user: Username, Ethereum address, Hash of the profile file. The username was the unique username in the whole system chosen by the user during registration. This username was also associated with an Ethereum (personal) address generated when the Decentralized Identity Manager was used for the first time. The hash of the profile file, is a string generated dynamically by IPFS, which could allow to reach the file and access the information content. Username, Ethereum address and hash of the profile file were stored in a smartcontract that had the role of "identity log" of end-users. For prototype validation, two different deployments of the same DIM prototype were istantiated: a desktop and a smartphone version. The first architecture involved the use of any user PC, on which the Ethereum blockchain and the IPFS daemon were executed. In this case, the user interface of the application was accessible via a browser. Ultimately, the PC became an IPFS node, with the ability to read or write user profile information. The second architecture, seen as an evolution of the first, was intented as a mobile application, able to read or modify the user profile, integrate the user's personal data with the data coming from the most important social networks and share personal information with external services which make explicit request. This type of architecture involved the use of an external Ethereum public node for reading and writing on the blockchain and a local node which could assure the connection with the IPFS network. The components and deployment diagrams in Fig. 2 and Fig. 3 describe how the mobile application works. In this context, the user interface and the backend service were all included in the mobile application. The choice of Android operating system was made for many reasons: its large use, and because there were libraries allowing to use both IPFS and Ethereum. Android IpfsDroid[32] library was meant to store and share decentralized files, and the Ethereum Lightwallet library[35] allowed to send transactions to a remote node, and to create an ethereum account by providing a passphrase. A Lightwallet library and an IPFS daemon were installed on each device.

Fig. 2 Components of the Decentralized Identity Manager

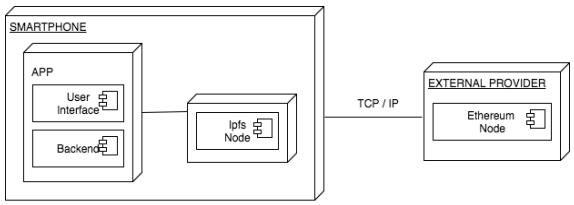

Fig. 3 Smartphone deployment of the DIM prototype

The Lightwallet library communicated with an Ethereum node which could be self-hosted by the user or provided by third party entities. This could bring to numerous advantages, above all in terms of performance, given that the device did not have to manage the memory required for the download of all the blocks which make up the blockchain. Encryption took effect on two levels: the first is at Ethereum level, where the users took advantage of a pair of asymmetric keys which identifies an Ethereum account, the second was at IPFS level with one symmetric key in order to encrypt profile data. Ethereum private key give authorization in modifying information registered on the Ethereum smart contract. Each time the profile file got modified and stored on IPFS, a new IPFS hash related to the profile was generated and then saved on the smart contract. Private profile data were encrypted on the same file with the symmetric key.

A native Android app has been realized, where the user could delete the entire profile, add/edit information, or delete a subset of it. In the homepage of the profile application three main buttons allowed to: share personal information with external services, integrate personal data with information from Facebook and Twitter social networks and finally a QR code scan function to manage user authentication from external applications. In the main interface, it was possible to enter the profile management section, by selecting the "About me" button. In this section, for each information field, using the button next to the textbox, users can specify the privacy level of the attribute that can be public or private. If the user decided to make certain information private, the latter were saved in the JSON file and encrypted using a personal symmetric encryption key. We made available personal data insertion from Social Network too, where the user could decide which information import from Facebook and Twitter, by selecting the respective checkbox along with the privacy level (private/public) of that information. It is possible to choose new data, that users desire to import from social networks, by clicking the "Integrate" button. Each change of personal information resulted in a modification of the associated JSON file.

V. Testbed and Limits

The validation phase has proven 3 main functionalities of the prototype: (i) explicit personal data insertion within the profile, (ii) intrinsic personal data extraction from social networks towards the profile, and (iii) personal data sharing with a requesting service.

- *Explicit personal data insertion* happens when users interact with a specific interface exposed by the component: the user fills the profile with his/her personal data (such as gender, name, age, devices owned, interests, etc...).
- *Personal data sharing* takes place when a service sends a request for personal data to the decentralized identity manager: the user will receive such request notification and then he/she can accept or refuse the data sharing. If the data sharing is accepted, the user application will send requested data to the service. The insertion of personal data in the profile file can optionally be performed on a second moment, after service request. In fact, when a service requests certain missing data in order to provide its outcomes, the user receive not only sharing request, but he/she will be also asked to insert such missing information.

- *Implicit personal data extraction* happens when personal data is automatically retrieved. Such personal data can consist in: location extracted from GPS device position, prototype usage (obtained by observing the user's interactions with the services, with other users or with devices.) or from an interest mining component to which the prototype interfaces. Thanks to the Interest Mining component, the users' interests and preferences are processed automatically from those contents users release on social networks (Facebook or Twitter).

In Fig. 4 the profile sharing information section is illustrated, with an external service, which makes explicit request for personal data. In particular, the user is obliged to provide the mandatory attributes requested by the service, but may decide not to share some optional attributes, going to select/deselect the respective checkboxes. When that user accepts to share his/her - at least mandatory - personal data, then a special URL (provided by the requesting service) is invoked. Towards this URL, the DIM will provide the required data to the service. Fig. 5 shows the user's interests extracted from Twitter through an "Interests Mining" procedure developed in a dedicated module.

In order to prove the usability and effectiveness of the solution the *mobile prototype* (see Deployment of the DIM Prototype) had been installed, together with the IPFS daemon, on a Android Smartphone.

Fig. 4 Sharing Data with external services

Fig. 5 Interests Section

As previously stated, the mobile prototype was equipped with a Lightwallet, which was interfaced with an Ethereum node on a remote virtual machine, reachable through HTTP calls. Two scenarios were considered: a basic *identification* scenario meant to give access to the SERVIFY service eco-system, a *"device sharing"* scenario, where a service requested user's devices in order to offer its outcomes following a pervasive metaphor. While in the identification scenario, the solution was exploited in order to retrieve the unique userID through his/her devices acknowledgement, and so to identify user , in the other "device sharing" scenario were created, different services requested access to user's devices in order to convey information through these. Considered scenarios are (i) Hotel room booking and (ii) Continous chat[4]. In such scenarios (identification and device sharing) the services asked for user's personal information: in the identification scenario, it was required to share personal ID, in the device sharing (both hotel room booking and continous chat), it was required to share device information (see The User Profile Schema). The requests for personal data were made at the same manner, with a special code, generated each time the service makes request. the service had to include a special Javascript library: servify_qrcode.js. All the pervasive services took advantage of the library by generating a QR code with the following information taken as input: (i) the attributes that the service required from the user (mandatory and optionals), (ii) the URI where the service wanted to receive the corresponding data. The resulting QR code was shared with the user's DIM. Once the QR code was available, the user had to frame such QR code with the QR code scanning functionality of the prototype. In this manner the DIM was able to decode what required personal information was requested and the URI to which such information had to be retrieved. We tested our scenarios in a laboratory setting, while tests in open, real settings, with users related to academic and civic world would require more attention to GDPR compliance. Looking at GDPR compliance, the proposed architecture has some limits with respect to those main issues of the blockchain and GDPR . We shall list the arisen issues in details.

Primarly, the identification of a data controller, the figure that is in charge of controlling the data. As reported by the report of the European Union Blockchain Observatory and Forum, there is an ongoing debate about how to identify the data controller in a public permissionless blockchain. The report states that if users submit their own personal data for their own personal use they are likely to fall under the household exemption of the GDPR and may not be considered data controllers.

The pseudonymization requirement may be another issue in this version of the DIM. This is because of the presence, not only over the blockchain, but also within the distributed IPFS storage, of encrypted and plain personal data. It is unavoidable to have data which may identify a real subject, but a new advancement is required in order to have only encrypted data and keys over the system.

We believe that the current deployment of the system does not suffer from the right to rectify issue. In fact, the data subject is the solely owner of his/her data, then the user is always capable to update the owned data and see those updated soon after.

We consider that the main challenge is represented by the “right-to-be-forgotten” or right to erasure, that is when an individual asks an organization that has their data to remove that data. This feature is unfeasible for a blockchain since every transaction/data on the blockchain will remain there forever, unless the chain is destroyed. Moreover, this is an issue also for IPFS. Indeed, it is not possible to force deletion of a file from the IPFS network. After a user put a file on the IPFS network, anyone can get that file on his local storage. Even if the owner of the file remove it from his node, this action will not be reflected on other nodes. As opposed to the public permissionless blockchain, a distributed file system such as IPFS can be theoretically extended with a legal framework that force deletion of data from every node if the owner request it. But this is almost unreachable since that the hypothetical legal framework should be adopted globally, being IPFS a global public network.

VI. Proposed Solution

We have started from the issues raised from our decentralized implementation in the previous work. Then we have focused our research on how to surpass the technological limitations and meet the requirements provided by the GDPR regulation.

For what concerns the data controller, the GDPR is focused on administrators but not on Peer-2-Peer-networks. By addressing mainly, the controller as the target of the duties of the GDPR – defining him as "the natural or legal person which, determines the purposes and means of the processing of personal data" (Art. 4 subsection 7) the regulation takes into account mainly entities which have the ability to actively control the data-flow of an IT-system. This is not the case of Blockchain-technologies. While, in permissioned blockchains, the entity who manages the key infrastructure has the potential to determine the purpose and means of the service, making them the controller, in permissionless blockchains there is no obvious controller: all the miners concurr in the process but are not concerned with the (personal) content of the distributed ledgers: programmers lose their influence after the blockchain is set into motion. As a result, only each individual node is, legally, in control[33]. It would be helpful to consider a (new) category of joint controllers in Art. 26 GDPR, which may apply, if the nodes "jointly determine the purposes and means of processing." We believe that, for the purpose of the current work, we may consider all the nodes within the blockchain, equally responsible for the purpose and means of services, for it is not up to us, as researchers, architects and developers, to find a practical interpretation for the regulation.

As stated in the previous paragraph, we may assert that the current architecture does not suffer from the right to rectify issue, as the existing prototype did not. The user is already capable of change and update the owned personal data, without the help of any other entity.

To achieve the goal of a decentralized PDS application, compliant to the GDPR, it is necessary to improve the previous software architecture adding some modules which allow to store personal data in a centralized cloud environment and preserving business logic in a decentralized environment. In this way, the owner of the data can, in any time, delete the access to his data to all or some of the services in a selective mode without leaving any copy distributed on other nodes of the network (right-to-be-forgotten).

We improve the previous software architecture to meet the requirements of pseudonymization, the right to withdraw and the right to be forgotten. The architecture we are going to propose, as seen in Fig. 6, is composed by three components:

- User mobile app
- Ethereum node
- Cloud Storage System

We based part of our new solution on some ideas from MyData architecture, which is a model for human-centered personal data management and processing. In particular, we introduced two MyData concepts, Links and Consents, in order to provide a trackable and privacy-oriented data sharing between user and services. MyData[34] reference architecture presented in this set of specifications is a human centric approach to liberate the potential of personal data and to facilitate its controlled flow from multiple data sources to applications and services. It responds on a practical and technical level to individuals growing demand for control over their own digital identity and to organizations need to fulfill the requirements of tightening data protection regulation, especially on digital, dynamic consents. Architecture takes an attempt to provide individuals and service providers a rigid framework for consent and data authorisation management and service registration via a standard and interoperable mechanism.

To better explain the current approach, consider the following example: a user exploits the DIM app on his mobile device and he needs to share some personal data like name, surname and age to a third-party service called Health@Home (H@H). The first step is to create a Link between DIM and H@H.

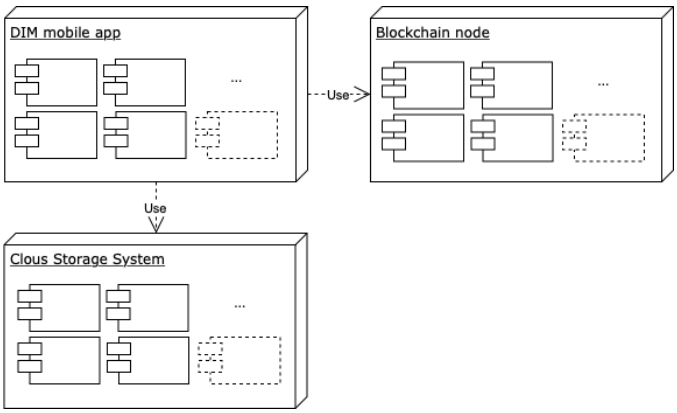

Fig. 6 The proposed DIM architecture

The Link is going to be created using a specific smart contract on Ethereum and it contains:

- Surrogated ID: this special id represents the user data owner, along all those transactions between DIM and H@H. This ID is unique for each user-service couple.
- Notification URL: a URL used to send notifications to H@H service, namely the url where the service is reachable.

After the successful creation of this Link and the correct storing on the blockchain, the DIM takes care of creating a Consent item that contains (i) the attributes type, (ii) the terms and (iii) the access duration to the user's data. In addition, it also contains its status (Active, Disabled or Withdrawn), a public key of H@H and a symmetric key encrypted using H@H public key. This Consent is stored on blockchain too.

As soon as the Consent is created and active, the DIM carries out the subsequent operations:

1. it creates a new file with user data needed by H@H;
2. it creates a hashcode of this file using SHA-256 algorithm and encrypts such hash using a symmetric key previously created in Consent phase;
3. it encrypts the file created at step 1 using H@H public key;
4. it saves the encrypted hashcode, produced in step 2, over the blockcahin;
5. it saves the file created on step 1 on a cloud storage system and retrives its url;
6. it sends the url retrieved at step 5 to the NotificationUrl of H@H service.

Now H@H is able to retrieve the requested user data and it is able to check if this is genuine by comparing the hashcode previously stored over the blockchain. By saving the hashcode of data over the blockchain, we assure to every subject involved in the process that (i) data is incorrupted, (ii) that owners of such data are certified, and that (iii) the subjects involved in the consent (user and H@H) are always capable of

retrieving in any moment the list of previous transactions and data exchanged between DIM and H@H.

This approach moves further the results obtained by our previous prototype by enhancing the "privacy by design" paradigm, following the requirements provided by GDPR regulation. In particular:

Pseudonymisation: within our model, we store over the Blockchain only hashcodes and encryption keys. In such sense, the data stored over the blockchain may be considered pseudonymous, namely if someone has the possibility to combine it with other available information and can thus identify a person (Recital 26 GDPR). Following the previous statement, we may consider the data stored over the blockchain as pseudonymous data at the state of the art, since malicious attacks would need enormous time and computing power (Recital 26 GDPR) in order to decrypt personal and sensitive data.

Right to withdraw: the user is always able to change the status of any Consent, he or she owns, in any moment. By accessing to the DIM mobile application, the user can update the status of each Consent. The application will update the Consent item over the blockchain, adding the latest status value. When the user withdraw a Consent, the system will delete data referenced by it from the cloud storage.

Right to be forgotten: this architecture allows users to delete any file containing their data. There are no distributed nodes, like in the previous structure, replicating the data. A single point of storage make it possible to delete the link between the blockchain and the data storage. Over the blockchain only hashcodes reside, which are encrypted with symmetric keys. This is not a definitive solution, but makes it more difficult to apply reverse engineering to restore data from the encrypted hash. After the deletion of data, the DIM application takes care of notify the connected services about the unavailability of it.

VII. Conclusions and Future Work

In this paper we have presented a particular deployment of a Personal Data Store, namely the Decentralized Identity Manager, and the experimentations going within our research lab. It is decentralized, in the sense of a technological framework which aims at solving the problem of personal data storing, protection and privacy, but without any central authority which may take advantage of collecting sensitive data. The solution has been conceptualized, having in mind the challenges of the data sharing in the modern era: the need to share data and personal devices while assuring the right to protect personal data related to data subjects. To assure device and data sharing, our prototype extended the concept of PDS in different ways: using a dedicated ontology and a special distributed architecture. It takes advantage of a special profile schema, specially deployed to model personal information, made up not only by data about the user, but also by devices owned by him/her. In this way we provide a solution, ready to face the challenges of IoT era: such as device sharing. The developed prototype takes advantage of distributed technologies: IPFS and Ethereum, both assuring the absence of centralized authorities, thus avoiding the perils of a central entity, who could illegally exploit personal information. The distributed architecture has been useful in various ways. This leveraged data security by taking advantage of two different pairs of secret keys: a pair of public/private encryption keys offered by Ethereum platform, plus a pair of symmetric keys to encrypt profile stored on IPFS, when users decide to set private some of their data. This choice has also enhanced the ownership of personal data to users who legally own them. But a distributed PDS has to face numerous challenges to comply with the GDPR. Two main scenarios were outlined for the purpose to exploit functionalities of the prototype. The first one is the identification scenario, where the user gets identified within the SERVIFY eco-system, exchanging with service his/her unique ID. The second scenario is the device sharing, where the user shares his/her devices in order to receive a pervasive service. In particular three main pervasive services were presented. Starting from such scenarios we have tested the DIM prototype against the main issues related to blockchain technologies and GDPR: the identification of data controllers, the anonymisation of personal data, the right to rectify, withdraw and the right to be forgotten. Then we have presented our approach, in order to solve the aforementioned issues and evolve the current DIM prototype towards GDPR compliance. For the purpose of the current work we believe that the nodes of the blockchain may be considereded equally responsible for the data threatment. Moreover we consider that the right to rectify did not affect our prototype as will not affect its evolution. The new architecture we have proposed moves further the DIM prototype to better fit the GDPR issues of the right to withdraw and to be forgotten. The new concept of consent, previously absent in our deployment, is embedded in such architecture, being stored with asymmetric public/private keys specially generated. The consent ties the service, the data subject and the data together, as long as the data subject confirms consent. On the other hand, sensible data is stored and encrypted with a different key, in a separated module: a cloud storage system. The data subject will be able, thanks to this module, to exploit a single point of access in order to modify or permanently delete data he or she owns.

As future work, we plan the development of a new version of the DIM prototype, with the new Cloud storage module and the consent structure. Then we plan to test the prototype in a larger real setting, where research professionals with other users related to the academic and civic world may provide fundamental feedback for the current project.

Another important prototype evolution is the necessity to make it work in a bigger scenario, external to the Servify ecosystem, considering different types of pervasive, IoT services, or other third-party services. For the same reason we envision the importance of the transition from Ethereum testnet (the blockchain we use at the moment) towards Ethereum mainnet. Other possible future features involve the possibility to manage one's own Ethereum identity through a user friendly interface, directly from the mobile application.

An issue which is left to be solved in a future release, is the Ethereum dependence on RSA cryptography with asymmetric keying. This introduces a vulnerability problem due to the fact that a quantum computer, thanks to the Shor factoring algorithm[36] can be used to violate key encryption. To solve the aforementioned problem, a group of Guardtime scientists[37] have realized the KSI Technology Stack standard able to solve the aforementioned problem and, consequently, our prototype could be made safer by adopting it.

The opportunities that distributed technologies may offer in terms of data protection are clear, but there are issues raised by GDPR compliance, which are currently being discussed not only by IT specialists, but also Law experted within the European Union. From a technological perspective, the ideas in this paper might serve as a contribution to substantiate the

principle of Privacy by Design (Art. 25 GDPR) for the practical use of blockchain technology within Personal Data Store systems. By doing so we may achieve the goal of data protection in the pervasive services and data sharing era, assuring that the ownership of personal data to people who really own them: ours.

REFERENCES

[1] https://www.marketwatch.com/press-release/class-action-lawsuits-against-facebook-consolidated-creating-one-of-the-largest-data-privacy-lawsuits-2018-08-23] Accessed on 01 March 2019

[2] https://ec.europa.eu/info/law/law-topic/data-protection/data-protection-eu_en Accessed on 19 March 2018

[3] M. Levin, "Designing Multi Device Experiences". O'Reilly Media. February 2014

[4] Alessi, M., Camillo, A., Giangreco, E., Matera, M., Pino, S., & Storelli, D. (2018). Make Users Own Their Data: A Decentralized Personal Data Store Prototype Based on Ethereum and IPFS. *2018 3rd International Conference on Smart and Sustainable Technologies (SpliTech)*, 1-7.

[5] C. Marcelo, P. Jurcys, G. Kousiouris, "*Smart Contracts and Smart Disclosure: Coding a GDPR Compliance Framework",* SSRN Electronic Journal · January 2018. doi: 10.2139/ssrn.3121658

[6] Wirth, Christian; Kolain, Michael (2018): Privacy by BlockChain Design: A Blockchain-enabled GDPR-compliant Approach for Handling Personal Data. In: W. Prinz & P. Hoschka (Eds.), Proceedings of the 1st ERCIM Blockchain Workshop 2018, Reports of the European Society for Socially Embedded Technologies. ISNN 2510-2591. DOI: http://dx.doi.org/10.18420/blockchain2018_03.

[7] X. Zheng, R. R. Mukkamala, R. Vatrapu and J. Ordieres-Mere, "Blockchain-based Personal Health Data Sharing System Using Cloud Storage," *2018 IEEE 20th International Conference on e-Health Networking, Applications and Services (Healthcom)*, Ostrava, 2018, pp. 1-6.doi: 10.1109/HealthCom.2018.8531125

[8] J. Bacon, J. D. Michels, C. Millard & J. Singh. "Blockchain Demystified: A Technical and Legal Introduction to Distributed and Centralised Ledgers", 25 RICH. J.L. & TECH., no. 1, 2018

[9] M. Mun, S. Hao, N. Mishra, K. Shilton, J. Burke, D. Estrin, M. Hansen, and R. Govindan, "Personal data vaults," Proceedings of the 6th International COnference on - Co-NEXT '10, 2010. doi:10.1145/1921168.1921191.

[10] T. Kirkham, S. Ravet, S.Winfield, and S. Kellom¨aki, "A personal data store for an Internet of Subjects," in Proceedings of the International Conference on Information Society, i-Society 2011, pp. 92–97, June 2011.

[11] I. Drago, M. Mellia, M. M. Munaf`o, A. Sperotto, R. Sadre, and A. Pras, "Inside dropbox: understanding personal cloud storage services," in Proceedings of the ACM Internet Measurement Conference (IMC '12), pp 481-494, Novembrer 2012. doi:10.1145/2398776.2398827.

[12] Bus, J. and Nguyen, M.-H. C. Personal data management a structured discussion. pages 270-288. 2013. *doi:10.3233/978-1-61499-295-0-270*

[13] https://mydex.org/ Accessed on 19 March 2018

[14] https://www.irmacard.org/ Accessed on 19 March 2018

[15] http://openpds.media.mit.edu/ Accessed on 19 March 2018

[16] http://onename.com Accessed on 19 March 2018

[17] https://bitnation.co/join-the-team/ Accessed on 19 March 2018

[18] T. Lyons, L. Courcelas, K. Timsit. "Blockchain and GDPR thematic report". The European Union Blockchain Observatory and Forum. 16 October 2018.

[19] http://www.silab-sicilia.it/ Accessed on 19 March 2018

[20] http://ethereum-project.org Accessed on 19 March 2018

[21] http://ipfs.io/ Accessed on 19 March 2018

[22] https://www.uport.me Accessed on 9 January 2019

[23] https://alastria.io Accessed on 9 January 2019

[24] https://sovrin.org/ Accessed on 9 January 2019

[25] https://sovrin.org/wp-content/uploads/2018/03/Sovrin-Protocol-and-Token-White-Paper.pdf

[26] https://we-trade.com Accessed on 9 January 2019

[27] https://www.tradelens.com/ Accessed on 9 January 2019

[28] https://www.theguardian.com/world/2013/jun/08/nsa-prism-server-collection-facebook-google Accessed on 19 March 2018

[29] https://schema.org/docs/schemas.html Accessed on 19 March 2018

[30] https://www.ethereum.org/ Accessed on 19 March 2018

[31] https://github.com/ethereum/go-ethereum/wiki Accessed 19 March 2018

[32] https://github.com/ligi/IPFSDroid Accessed on 19 March 2018

[33] M. Martini, Q. Weinzierl, "Die Blockchain-Technologie und das Recht auf Vergessenwerden. Neue Zeitschrift für Verwaltungsrecht" (NVwZ) (2017), 1251 – 1259.

[34] hiit.github.io/mydata-stack/ Accessed on 9 January 2019

[35] http://lightwallet.io Accessed on 19 March 2018

[36] J Buchmann, and E Dahmen, "Post-Quantum Cryptography, DJ Bernstein", eds. Springer Berlin Heidelberg, 2009. doi:10.1007/978-3-540-88702-7

[37] https://guardtime.com/technology Accessed on 19 March 2018

M. Alessi is Director of Innovative Paradigms for Public Administration Unit at Engineering Ingegneria Informatica S.p.A R&D Lab: Management of European and Italian E-government Project. He received the B.E. degree in computer engineering from the University of Palermo, Italy, in 2000.Since 2010 he is Head of R&D Unit in Engineering, where he carries out his principal research topics: Public sector new delivery models Open Service Innovation tecniques and models, Distributed network for personal data management, IoT for Public Service improvements, Open Data tools.

A. Camillò is a researcher of the Open Public Service Innovation Unit of the of R&D Office at Engineering Ingegneria Informatica since 2015. He graduated in Management Engineering from the University of Salento in 2014, with a long experience in complex systems design and development, like energy monitoring using IoT and social platforms. His research interests mainly focus on Open Innovation and Open service Innovation, Open Government, Sentiment and Social Network analysis and new models of engagement, gamification.

E. Giangreco head of Innovation for Public Administration Group (part of Open Public Service Innovation Unit). She is a researcher in Engineering Ingegneria Informatica since 2005. She graduated in Management Engineering from the University of Salento in 2005. Her relevant expertise and experiences are related to Service Oriented Architecture, Business Process management, Enterprise Architecture, Rule Based Systems, Semantic Web. Her research interests mainly focus on processes innovation that include tools and techniques of gamification, opening data and processes, ideas life cycle management, sentiment and social networks analysis at local level, Innovative Personal Data management and new models of user-service interaction.

M. Matera is a researcher of the Open Public Service Innovation Unit of the R&D Office at Engineering Ingegneria Informatica since 2015. He graduated in Computer Engineering from the University of Salento in 2015. His main research interests focus on Open Innovation and Open service Innovation, with experience in Distributed Ledger Technologies, Blockchains and Decentralized networks, Artificial Intelligence systems, IoT, Social Network Analysis and Sentiment Analysis algorithms.

S. Pino was born in Treviglio, Italy, in 1978. He received the B.E. degree in computer engineering from the University of Salento, Lecce, Italy, in 2010.He is a researcher in Engineering Ingegneria Informatica since 2015, in Open Public Service Innovation Unit, where he contributes in leveraging complex systems, social platforms and prototypes with behavioral studies, engagement models, gamification tools and novel HCI paradigms. His principle areas of research interest are Open Innovation, HCI, intuitive interaction, engagement techniques, gamification and Internet of Things. He is actually working on 3d GIS modeling and visualization, and advanced interfaces between users and IoT.

D. Storelli is a Researcher of the Open Public Service Innovation Unit of the of R&D Office at Engineering Ingegneria Informatica since 2012. He graduated in Computer Engineering from the University of Salento in 2006. He currently has the technical coordination of some research projects related to the innovation services for the Public Administration. His research interests mainly focus on data-driven innovation, with special focus on urban-scale data management and personal data management.