ciTIzen-centric DAta pLatform (TIDAL): Sharing Distributed Personal Data in a Privacy-Preserving Manner for Health Research

Chang Sun [a,*], Marc Gallofré Ocaña [b], Johan van Soest [c,d] and Michel Dumontier [a]

[a] *Institute of Data Science, Faculty of Science and Engineering, Maastricht University, The Netherlands*
E-mails: chang.sun@maastrichtuniversity.nl, michel.dumontier@maastrichtuniversity.nl
[b] *Department of Information Science and Media Studies, University of Bergen, Norway*
E-mail: marc.gallofre@uib.no
[c] *Brightlands Institute of Smart Society (BISS), Faculty of Science and Engineering, Maastricht University, The Netherlands*
[d] *Department of Radiation Oncology (Maastro), GROW School for Oncology and Reproduction, Maastricht University Medical Centre+, The Netherlands*
E-mail: j.vansoest@maastrichtuniversity.nl

Abstract. Developing personal data sharing tools and standards in conformity with data protection regulations is essential to empower citizens to control and share their health data with authorized parties for any purpose they approve. This can be, among others, for primary use in healthcare, or secondary use for research to improve human health and well-being. Ensuring that citizens are able to make fine-grained decisions about how their personal health data can be used and shared will significantly encourage citizens to participate in more health-related research. In this paper, we propose a ciTIzen-centric DatA pLatform (TIDAL) to give individuals ownership of their own data, and connect them with researchers to donate the use of their personal data for research while being in control of the entire data life cycle, including data access, storage and analysis. We recognize that most existing technologies focus on one particular aspect such as personal data storage, or suffer from executing data analysis over a large number of participants, or face challenges of low data quality and insufficient data interoperability. To address these challenges, the TIDAL platform integrates a set of components for requesting subsets of RDF (Resource Description Framework) data stored in personal data vaults based on SOcial LInked Data (Solid) technology and analyzing them in a privacy-preserving manner. We demonstrate the feasibility and efficiency of the TIDAL platform by conducting a set of simulation experiments using three different pod providers (*Inrupt*, *Solidcommunity*, Self-hosted Server). On each pod provider, we evaluated the performance of TIDAL by querying and analyzing personal health data with varying scales of participants and configurations. The reasonable total time consumption and a linear correlation between the number of pods and variables on all pod providers show the feasibility and potential to implement and use the TIDAL platform in practice. TIDAL facilitates individuals to access their personal data in a fine-grained manner and to make their own decision on their data. Researchers are able to reach out to individuals and send them digital consent directly for using personal data for health-related research. TIDAL can play an important role to connect citizens, researchers, and data organizations to increase the trust placed by citizens in the processing of personal data.

Keywords: Health Data, Personal Data, Linked Data, Data Governance, Decentralisation, Solid, Citizen Science, Data Privacy, Policies, Privacy-Preserving Analytics, GDPR

*Corresponding author. E-mail: chang.sun@maastrichtuniversity.nl.

1. Introduction

Giving individuals more control over who can access their personal data for what purpose and making the data available using transparent and privacy-preserving methodologies will significantly encourage their engagement in healthcare research [1, 2]. To improve evidence-based healthcare research and empower healthcare authorities to optimize the accessibility and effectiveness of the healthcare services, we need sufficient personal health data [3]. However, personal health data is largely collected and managed by various healthcare providers. In Europe, many citizens have limited electronic access to their health data, often scattered among healthcare service providers [4]. As a result, citizens have limited control over their data, or need to control their data at various locations.

Since the General Data Protection Regulation (GDPR) has been released, more data rights and information privacy are valued and provided to European Union citizens. However, there is no mature technology and standards that enable individuals to fully exercise their data rights in a simple way [5]. The public consultation on the European strategy for data showed that almost 88% of all respondents (806 contributors) would like to have more access and control over the data they generate [6]. A large proportion of the respondents would be willing to share their data, especially for health-related research, but a majority of them considered that there are no sufficient tools and mechanisms to "donate" the use of their data. For example, at present, if individuals are willing to contribute their health data to help chronic disease research, they need to look for an ongoing research study that is recruiting new participants and has requirements that are applicable. Meanwhile, individuals need to trust and be willing to share their data with this research study. However, sharing personal data often raises concerns about privacy, security, ownership, and accountability. Examples of these concerns are: who will have access to the data and study results, how the individuals can change/revoke the permissions to (fully or partially) access the data, and whether the data is used for other purposes.

In this study, we propose a new citizen-centric data platform (called TIDAL) that gives individuals fine-grained access to their data and ensures that citizen-controlled data are processed in a predefined manner. We designed a prototype as a proof-of-concept following an exploratory technology development process in light of our experience in the development of a privacy-preserving distributed data analysis infrastructure in the previous studies [7–10]. TIDAL consists of an integrated set of components for requesting subsets of data stored in personal data vaults using Solid technologies [11] and analyzing them in a privacy-preserving manner. Solid, which stands for SOcial LInked Data, is a set of technologies that facilitates users to create decentralized applications using Linked Data and W3C standards and protocols. We evaluated the performance of TIDAL by executing simulation experiments on various sizes of simulation data using three different pod providers (*Inrupt*. *Solidcommunity*, Self-hosted server).

Our long-term objective is to engage individuals to "donate" the use of their personal data for health-related research with maximal control over data access, storage, and analysis. To achieve this objective, this study takes one step to address the technical feasibility, efficiency, and scalability of a Solid-based application for health-related research that makes use of user-focused data access control, digital consent, and privacy-preserving data analysis. We believe the TIDAL platform will enhance citizen science, increase the trust placed by individuals, and the transparency of the processing of their personal data.

We summarize the main contributions of this paper:

1. developing an open source citizen-centric data platform that facilitates individuals to store and manage their data using a personal data vault technology (Solid) and provide direct consent to health-related research;
2. integrating Data Privacy Vocabulary [12, 13] to structure personal data requests as digital consents in TIDAL to meet the requirements of GDPR;
3. enhancing the interoperability of personal data use by formulating data requests into RDF format with integration of vocabulary services and standards;
4. integrating privacy-preserving data analysis algorithms using parameters and configurations promised in the data request and only the results are sent to the researchers; and
5. demonstrating the feasibility and efficiency of TIDAL in different experimental settings.

The article below is structured as follows: section 2 introduces the recent related work and remaining challenges. Section 3 describes the technologies we applied in the TIDAL platform. Section 4 presents the architecture of TIDAL, and demonstrates how it works for researchers and participants. Section 5 describes the experimental setup

and results of TIDAL in different user scenarios. Section 6 discusses our discovery and limitations of the current version of TIDAL. Finally, Section 7 outlines the conclusions and future work.

2. Related work

Researchers and companies have developed several personal data vault solutions with different emphases and features to enable individuals to manage the access, share, and use of their data. We identified the following projects and tools that have been applied in practice and provided a comparison table including detailed information and additional resources in the supplementary material[1]. DEecentralised Citizen Owned Data Ecosystems (DECODE) [14], MyHealthMyData (MHMD) [15] and OwnYourData [16] are based on distributed-ledger technologies to provide traceable and transparent data-access control. MIDATA [17] and MedMij [18] are national programs in Switzerland and Netherlands that provide citizens with new data ecosystems to use their medical data for healthcare services and research. Digi.me [19] are CozyCloud [20] are commercial products providing mobile applications and cloud services to share personal data. The Hub of All Things (HAT) [21] – a foundation –, MyDex [22] – a community interest company –, and openPDS [23] – a research project – utilize Personal Data Store (PDS) [24] technologies to provide users with servers to store and share their personal data and execute on-device computations.

2.1. Existing personal data vault technologies

DECODE and MHMD were both funded by the European Union's Horizon 2020 research and innovation programme [25]. DECODE enables individuals to keep personal information private or share it for the public good using peer-to-peer networks and blockchain technologies. This ecosystem, from an operating system to an interactive dashboard, has been developed and piloted in Barcelona and Amsterdam. However, it focuses on individual control over data sharing rather than data processing and analysis. Individuals can specify "smart rules" for their personal data to pre-define under what conditions the data can be used. To guarantee privacy-preserving transactions in the Blockchain, DECODE uses the Coconut selective disclosure credential system [26]. Since DECODE relies on its own operating system and tools, it lacks the interoperability and extensibility that would be required for data mobility across healthcare systems and national borders.

MHMD [15] is another Blockchain-based solution that connects organizations and individuals to make anonymised data available for open research. It enables individuals to provide dynamic consent for different types of potential data usage and monitor the usage. Similar to the DECODE "smart rules", the MHMD consent determines under what conditions the data can be used. MHMD supports data analysis algorithms combined with secure multi-party computation and asymmetric encryption for preserving privacy. However, individuals' data is still hosted at organizations (e.g., hospitals), which are the only ones empowered to give permission to researchers requesting data. OwnYourData [16], developed by a non-profit organization, is another personal data management product that uses Blockchain technology to make data immutable. OwnYourData stores users' personal data and provides the insights from it.

MIDATA [17], a nonprofit cooperative in Switzerland, operates a data platform that enables Swiss citizens to selectively share their data with medical research and clinical studies. MIDATA shares the same limitations as DECODE on the interoperability and extensibility of their data ecosystem. MedMij [18] is established as a standard in the Netherlands for the secure exchange of health data between Dutch residents and healthcare providers. MedMij, serving as a high-level guideline, proposes a set of information standards to structure health data from different sources and standardize data exchange. However, MedMij does not yet include researchers in the network nor facilitates citizens to voluntarily share their health data for research studies or any other purposes.

Digi.me [19] and CozyCloud [20] deliver commercial products to give people control of their data when using web or mobile applications, but both host data centralized on their own cloud servers. Similarly, MyDex [22] and HAT [21] offer PDS as cloud-hosted servers to store personal data and connect them to other applications and services on the Web or mobile. Different from the previous tools that host the PDS in their own servers, openPDS [23]

[1]Existing personal data management tools: https://doi.org/10.6084/m9.figshare.19111508

allows users to self-host the PDS and use it as a service. OpenPDS also applies the SafeAnswers framework which executes the queries inside the PDS rather than sending anonymized data and returns and aggregates results from more than one PDS. It allows users to manage data access and monitor data usage. However, SafeAnswers presents a computational challenge for complex data analysis and does not consider the scenario of conducting research studies in large populations.

2.2. Remaining challenges

The existing solutions often focus on one particular aspect, such as personal data storage or overview, data access control, or data sharing with healthcare providers. To the best of our knowledge, there is no platform that enables individuals to connect with researchers to donate use of their personal data for research while being in control of the whole data life cycle including data access, storage, and analysis. Only a few tools support personal data analysis over a number of participants. These tools face challenges such as the data permissions are specified by the data organizations rather than individuals and the analysis algorithms are relatively simple. We also see an urgent need for more investments in data quality and interoperability to improve the feasibility and sustainability of personal data management platforms [2]. Therefore, we propose TIDAL to fill the gaps that we have identified from the existing works.

3. Background

3.1. Solid - Decentralized data management

Solid (SOcial LInked Data) is a decentralized data management platform based on W3C standards, Resource Description Framework (RDF), and Semantic Web technologies, initiated by Tim Berners-Lee [27–29]. Rather than tech giants storing and controlling personal data from their users, Solid technologies enable users to store and manage their data independently from the applications so that users can retain sovereignty over their data. Solid is composed of three core components - the data pod (i.e., where the data is stored), the application (i.e., the services that users can use and grant access to), and providers (i.e., where the pod and application are technically hosted).

Each Solid user is assigned with a WebID[2] as a unique global ID for identification and authentication. Solid data pods are web-based storage services and databases where various types of data can be stored such as RDF triples, free text, images, videos, or even webpages. However, Solid is featured by its capability to parse and serialize structured data using RDF in syntaxes like Turtle and JSON-LD. Data in Solid pods can be accessed and managed using a decentralized authentication[3] and Web Access Control (WAC)[4] mechanism [27] that is a decentralized cross-domain access control system. WAC in Solid provides pod owners with a fine-grained access control for every single data element in their data pod by granting other Solid users and applications the permissions to read, modify, and write the stored data elements. The Access Control List [30] and Access Control Policy [5] are utilized in Solid to describe the different operations on the target data elements in the pods.

Solid applications are developed on top of the aforementioned technology stack using the Solid Protocol [31]. Most applications are developed for web or mobile platforms. Users can grant and revoke permissions of reading, writing, appending, and removing data elements to both Solid applications and other users at any time. Solid allows multiple applications to access and reuse the same data from a pod, thereby potentially minimizing data duplication and staleness. Solid pods can be hosted on public servers by pod providers which play a similar role as the cloud storage providers. Solid pods can also be self-hosted on personal servers, and migrated from pod providers to self-hosted. A single Solid user can own more than one data pod which is hosted by one or multiple pod providers. In this case, the user's data can be distributed across different pods but all controlled by the same user and linked to the

[2] WebID: https://w3id.org/wiki/WebID
[3] Solid OpenID Connect (Solid-OIDC) specification: https://solid.github.io/solid-oidc/
[4] Web Access Control: https://solid.github.io/web-access-control-spec
[5] Access Control Policy: https://solid.github.io/authorization-panel/acp-specification/index.html

user's WebID. Users are able to select and change their pod providers at any time based on providers' geographical locations, responsibilities, different degrees of privacy protection and legislation. Thus, Solid presents a distributed scenario that challenges the communication between Solid applications and data pods, but provides fine-grained data control to users.

3.2. Personal Health Train - Distributed data analysis initiative

The Personal Health Train (PHT) initiative was designed for healthcare innovators and researchers to access heterogeneous data sources and learn from distributed data in a privacy-preserving manner [32], [10]. The essence of this approach is to transfer the research questions and analysis algorithms (from researchers) to data rather than centralizing data. Only the analysis results are sent back to the researchers.

The PHT technology has been developed and implemented in several real-life use cases in the healthcare domain. In our previous studies, we have developed the PHT infrastructure to address horizontally and vertically partitioned data[6] problems [9], [8], [33], [34]. In this study, we further extend the PHT infrastructure from the level of information control by organizations to information control by individuals themselves.

4. Overview and implementation of TIDAL

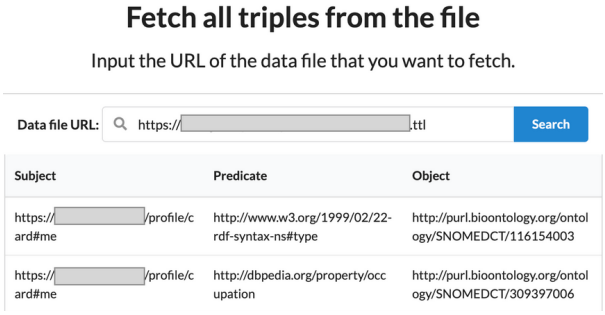
The primary use case of TIDAL is for researchers (data requesters) who want to analyze personal data and participants (data subjects) who are willing to share their data for research. TIDAL, as a web application, facilitates researchers to publish research participation requests as digital consent to request personal health data from potential participants. TIDAL also enables participants to approve or decline requests, grant or withdraw permission to use their personal data by interacting with their Solid data pods. In this section, we will present the overview and implementation of TIDAL by describing a use case between two types of users - the participant and the researcher.

Participants and researchers need Solid accounts and data pods to be authenticated and logged in TIDAL using their valid WebIDs and credentials. They can create a Solid account and request a data pod from any public pod providers [35] in a simple way or host a pod themselves if they have sufficient technical knowledge [36]. When logging in TIDAL for the first time, users need to grant TIDAL "Read" and "Write" permissions to their pods. With these permissions, TIDAL can query existing data elements in RDF format (Fig. 1a), create new data elements or files (Fig. 1b), and modify or delete data elements in a simple way. For example, to add a new data element to the pod, the user only needs to provide a path to the data file (e.g., *https://username.podprovider.com/private/newFile.ttl*), the URI of the data element (e.g., *https://schema.org/name*), and data value (e.g., "TIDAL"). The new data element is represented in the form of *subject – predicate – object*, where the subject is an automatically generated URI (e.g., https://username.podprovider.com/private/newFile.ttl#123456789), while the URI of the data element and new data value are added as the predicate and the object respectively. Then, this information is converted to RDF triples and sent to the data file in the user's pod by TIDAL. General users are not required to have much knowledge of technical specifications of Solid and semantic web technologies. But it is preferable to know basis of linked data such as what URIs and RDF triples are. Users who have knowledge and experience in these technologies can do the same actions directly in their Solid pods.

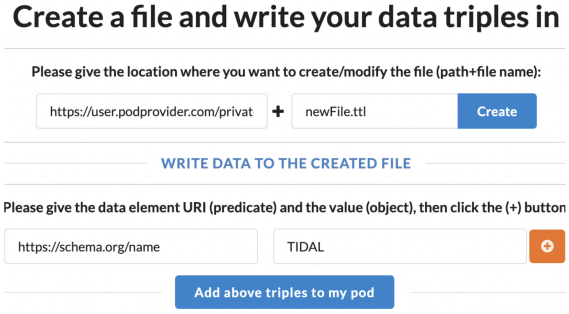
The permissions that are granted to TIDAL can be changed anytime by the users in the access control setting of their pods. It is important to note that the operations on the data can only be performed after users have successfully authenticated and logged in TIDAL. TIDAL does not take any actions on behalf of users or without interacting with users. For user who has multiple Solid pods, TIDAL enables the user to operate (read, write, modify, and delete) the data from all the pods that are linked to the user's WebID.

TIDAL authenticates and interacts with Solid pods with a Javascript package - solid-node-client (V2.1.10) [37]. Solid-node-client enables pod owners to access their pods, create or modify data in their pods, and grant or revoke the permissions from an individual data file level to the entire pod level via a Solid web application. To store,

[6]Horizontally partitioned data is that particular data from different individuals are distributed over multiple data sources, while vertically partitioned data represents different data about a particular individual are distributed over multiple sources.



(a) An example of query personal data in a RDF format from a Solid pod using TIDAL ("Fetch" page).



(b) An example of creating a new data file and writing data element to a Solid pod using TIDAL ("Create" page).

Fig. 1. Examples of querying and creating data in Solid pod using TIDAL.

parse, and query RDF data from Solid pods, TIDAL uses the rdflib.js (V2.2.19) [38] and tripledoc (V4.4.0) [39] library. Similar libraries such as solid/query-ldflex can also be used to access data in Solid pods through LDflex expressions [40].

4.1. Researcher publishes participation request

To publish a participation request, the researcher is required to register as a "researcher role" on TIDAL by providing basic information such as job position, affiliation, and research topics. The researcher is issued a public-private key pair from TIDAL that will be used to verify the identity of the researcher and the integrity of the request. The public key is stored at the TIDAL, whereas the private key is privately stored only in the researcher's pod. When the researcher publishes a new request, the request file is assigned to a uniform resource identifier (URI) by Solid. The URI of the request is stored in the researcher's pod and TIDAL's server, but the original request file is only stored in the researcher's Solid pod. The content of the request is automatically signed by the researcher's private key and the signature is stored in TIDAL's server. Any changes to the request will cause a verification failure when the request is executed to retrieve participants' data. TIDAL uses the Ed25519 algorithm [41, 42], a high-speed and high-security signature scheme, for public-key signature encryption. Ed25519 is an implementation of the Elliptic Curve Digital Signature Algorithm (EdDSA) using SHA-512 (SHA-2) and Curve25519 with Twisted Edwards Curve [43]. It has been widely used in protocols such as TLS 1.3 and SSH [44]. TIDAL uses Ed25519 from the TweetNaCl (V1.0.3) package [45], a port of the Networking and Cryptography library [46] to Javascript.

Figure 2 shows the request form for publishing a participation request. The request form includes fields to specify the following requested field (RF):

RF 1: the purpose of the research where researchers clearly indicate the purpose of processing personal data in their research. Researchers can select one or more from a list of data processing purposes described in DPV such as `dpv:EnforceSecurity`, `dpv:ResearchAndDevelopment`. These elements will be described and stored using their URIs (e.g., *https://w3id.org/dpv#EnforceSecurity*in the request form in the researcher's Solid pod.

RF 2: description of the specific purpose where researchers elaborate the purpose with more details in human readable text. Researchers can fill in the answers in free text such as *"Learn association between the status of Type 2 diabetes and patients' dietary patterns using linear regression"*.

RF 3: the category of requested data elements where researchers indicate which category of personal data best describes the requested data elements. Researchers can select one or more from a list of personal data categories described in DPV such as `dpv:Health` or `dpv:Income` and store them in the researcher's pod.

RF 4: the data elements where researchers indicate the data elements (URI) that are requested from the participants. Researchers can fill in one or more URIs of the requested data elements. Researchers can also search for existing URIs from existing ontologies and select those for the requested data elements. For

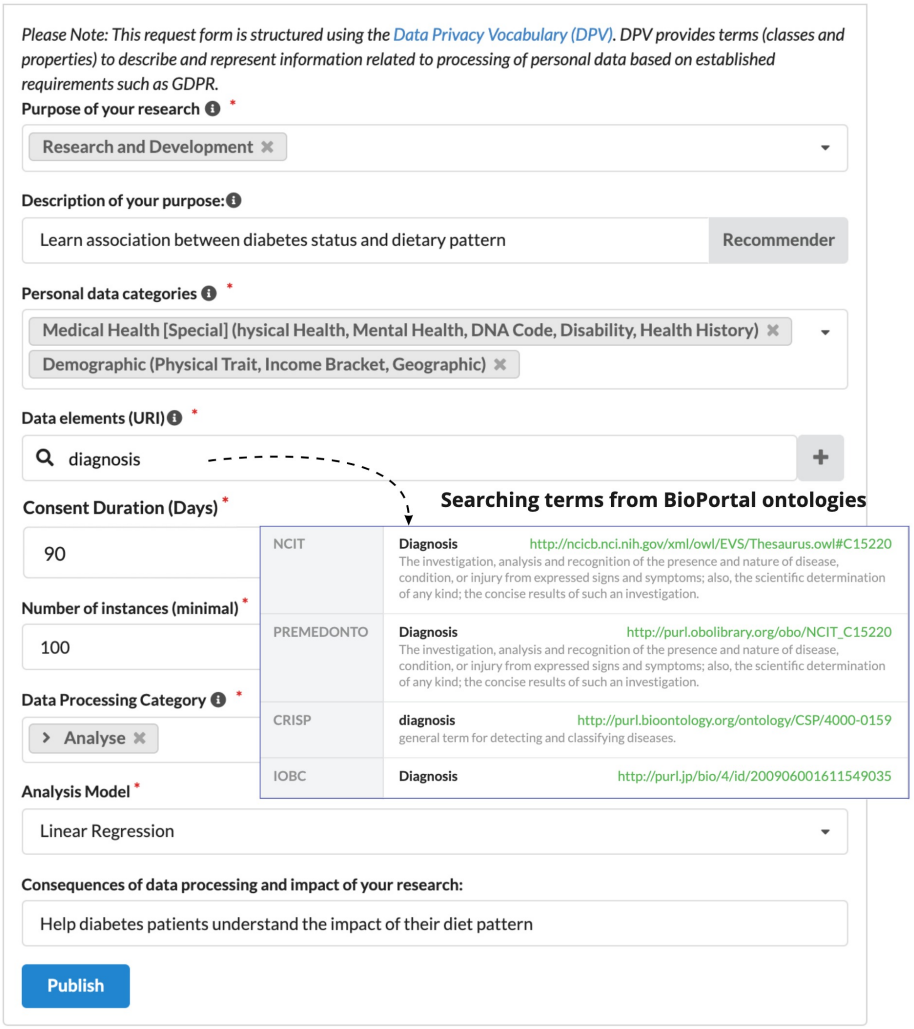


Fig. 2. Participation request form on TIDAL.

example, instead of requesting the "*Age*" in plain text, researchers can set the URI of Age in SNOMED (`SNOMEDCT:397669002`) as requested data element in the form. In this paper, we assume the personal health data of all the participants is structured and stored in the same way in their data pods. The data models that are used to structure and store the data are beyond the scope of this study. The limitation and future work regarding this point is described in the Discussion section.

RF 5: the consent duration (days) where researchers indicate a future duration of the consent, which means how long the consent will last. TIDAL uses this information to calculate an exact expiry date when the consent will be no longer valid. Researchers can only give the number of days as answer in this field such as *90*. Guidelines, references, and good practices can be provided on TIDAL in the future for how long a consent duration should be.

RF 6: the number of individuals who agree to participate in the study where the researchers specify a minimal number of participants required to initiate data processing. Researchers can only give integer numbers as the answer in this field such as *1000*.)

RF 7: the categories of data processing where researchers indicate which category or a chain of data processing will be performed on the requested data. Researchers select one or more from a list of data processing categories described in DPV such as `dpv:Copy`, `dpv:Anonymise` and `dpv:Analyse`.

RF 8: the methods or algorithms in data processing where researchers specify how the requested data will be processed. Researchers select one or more from a list of predefined algorithms such as *Linear regression*.

RF 9: the impact and consequences where researchers communicate the possible impact and consequences of data processing to the participants in terms of influence, change, or effect on citizens or society. Researchers answer in text that is human-readable and understandable for the general public.

To improve the interoperability of requested data elements, we have integrated the BioPortal API [47] in TIDAL to help researchers use standardized ontologies and terminologies for specific information elements. Bioportal is the most comprehensive repository of biomedical ontologies that includes more than 800 ontologies. TIDAL supports researchers to search the existing biomedical ontologies and terminologies provided by Bioportal and to apply them to the requested data elements. For example, instead of using "*diagnosis*" as a requested data element, researchers can look for the terms from well-established ontologies such as "*http://purl.obolibrary.org/obo/NCIT_C152*" or "*http://ncicb.nci.nih.gov/xml/owl/EVS/Thesaurus.owl#C15220*", by searching the keyword "*diagnosis*" in Data Elements (URI) in the request form.

After the researcher completes and publishes a participation request, TIDAL structures the request form as a RDF turtle file and represents it as a digital consent using Data Privacy Vocabulary (DPV-V0.7) and Schema.org vocabulary. Several ontologies and vocabularies that use semantic technology to implement and manage consent for data privacy and protection purpose have been studied by [48, 49]. Given the legal focus of TIDAL, we applied the DPV which specifically captures the relevant concepts of data processing in relation to EU GDPR. Each published participation request is labeled as a `dpv:PersonalDataHandling` and `schema:AskAction` and stored only in the researcher's Solid pod (Step 2 in Fig. 3). Meanwhile, the URI of the request is recorded in an index file on TIDAL. When the participants look for the ongoing requests on TIDAL, TIDAL reads the URIs of the published requests in the index file, retrieves the information from the request files from researchers' pods, and presents the information to the participants (see section 4.2). The request form is designed to be specific and structured as a digital informed consent. Complying with the GDPR requirements on consents, the request form describes the identifiers of the requester (researcher) and controller (trusted party – a certificated organization compliant with GDPR that executes the requests and analyses), what data elements in which personal data categories will be processed in what time frame, how the requested data will be processed for what purposes, and the possible risks and consequences of data processing such as for participants in relation to automated decision making. The overall schema of an example participation request is illustrated in Fig. 4a and the stored RDF format of the example is shown in Listing 1.

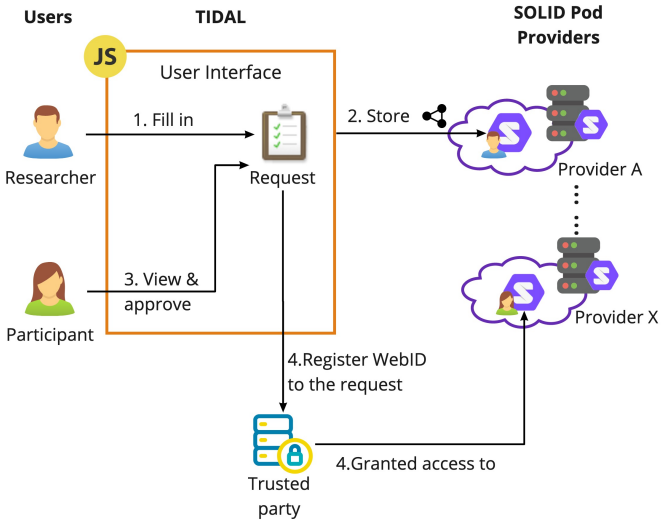
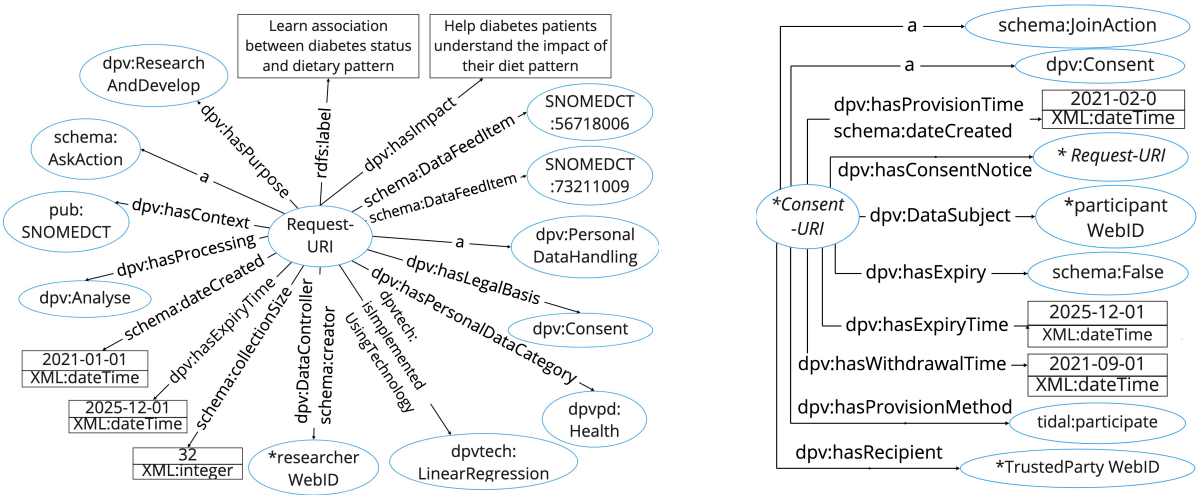


Fig. 3. Interaction between researchers and participants on TIDAL. The researcher publishes the request and store it in his Solid pod. The participant views and approves the request on TIDAL. The participation record is stored in the participant's pod and the trusted party.



(a) Schema of an example participation request.

(b) Schema of an example participation statement.

Fig. 4. Schema of the participation request (from researchers) and participation statement (from participants).

```
@prefix : <http://exampleresearcher.solidprovider.com/public/request.ttl#>.
@prefix schema: <https://schema.org/>.
@prefix exre: <http://exampleresearcher.solidprovider.com/profile/card#>.
@prefix dpv: <http://w3id.org/dpv#>.
@prefix dpvpd: <http://w3id.org/dpv/dpv-pd#>.
@prefix dpvtech: <http://w3id.org/dpv/dpv-tech#>.
@prefix SNOMEDCT: <http://purl.bioontology.org/ontology/SNOMEDCT/>.

:161964062096710764675982245664
   a schema:AskAction, dpv:PersonalDataHandling;
   dpv:hasLegalBasis dpv:Consent;
   rdfs:label "Learn association between diabetes status and dietary pattern";
   schema:collectionSize 32;
   schema:creator exre:me;
   schema:DataFeedItem SNOMEDCT:10396001, SNOMEDCT:230125005, SNOMEDCT:56718006, SNOMEDCT:73211009;
   schema:dateCreated "2021-01-18T00:00:00Z"^^XML:dateTime;
   dpvtech:isImplementedUsingTechnology dpvtech:LinearRegression;
   dpv:hasImpact "Help diabetes patients understand the impact of their diet pattern";
   dpv:hasContext SNOMEDCT;
   dpv:hasDataController exre:me;
   dpv:hasExpiryTime "2021-12-31T00:00:00Z"^^XML:dateTime;
   dpvpd:hasPersonalData dpvpd:Health;
   dpv:hasProcessing dpv:Analyse;
   dpv:hasPurpose dpv:ResearchAndDevelopment.
```

Listing 1: An example of generated RDF triples from the request form stored in researcher's Solid pod

4.2. *Participant views and approves requests*

When a participant wants to view the ongoing data request, TIDAL queries RDF data from all ongoing data requests and displays them on the TIDAL Participation Web page (Fig. 5). To do so, TIDAL (1) retrieves all published requests from the index file; (2) filters the ones that are in the valid period (i.e., before the expiration date of the request); (3) checks if requested data elements exist in the participant's data pod; and, (4) and displays the data requests to the participant in a human readable manner in a card view. Each card is linked to its original request file in RDF format from the researcher's pod (Listing 1). We assume that participants have their personal health data (e.g., medical records, medications, lifestyle and behavior data) structured and stored using RDF in their own Solid pods. Figure 5 shows an example view of the published participation requests on TIDAL from a participant's perspective. Research purpose, personal data category, data processing category, and data elements are linked to the URIs of the terms.

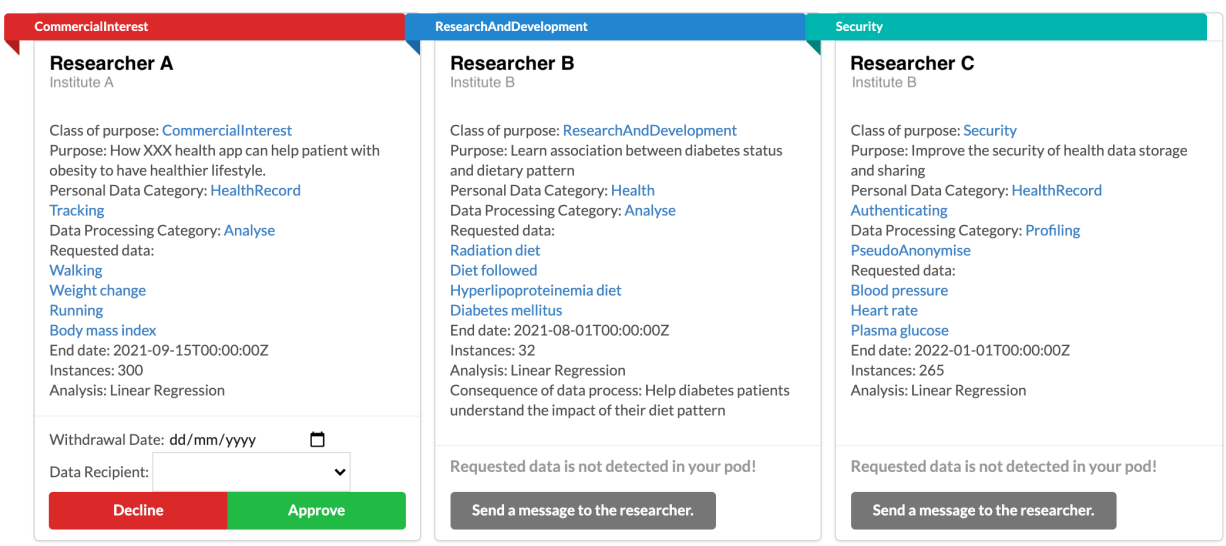


Fig. 5. An example of viewing published participation requests on TIDAL.

If TIDAL detects the requested data elements in the participant's pod, the card displays the "Approve" and "Decline" options, and the participant can voluntarily join the data request by setting up a preferable withdrawal time (earlier than the request expiry date) and selecting the party they trust to process their personal data. TIDAL generates an instance adhering to the `schema:JoinAction` and `dpv:Consent` in RDF format describing which request (URI) has been approved at what time and until when this approval is valid. The statement, acting as an approval consent (`dpv:Consent`), is structured by using DPV [50] and stored in a private folder in the participant's Solid pod. Figure 4b and Listing 2 shows the schema of an example of participation and generated consent statements.

```
@prefix : <http://exampleParticipant.solidProvider.com/private/participation#>.
@prefix part: <https://exampleparticipant.solidprovider.com/profile/card#>.
@prefix req: <https://exampleresearcher.solidprovider.com/public/request.ttl#>.
@prefix extp: <http://examplertrustedparty.solidprovider.net/profile/card#>
@prefix app: <https://examplesolidapp.com/

:1619704126629529965754215 5198
   a schema:JoinAction, dpv:Consent;
   schema:dateCreated "2021-02-18T00:00:00Z"^^XML:dateTime;
   dpv:DataSubject part:me;
   dpv:hasConsentNotice req:161964062096710764675982245664;
   dpv:hasExpiry schema:false;
   dpv:hasExpiryTime "2021-12-31T00:00:00Z"^^XML:dateTime;
   dpv:hasProvisionMethod app:participate;
   dpv:hasProvisionTime "2021-02-18T00:00:00Z"^^XML:dateTime;
   dpv:hasWithdrawalTime "2021-09-18T00:00:00Z"^^XML:dateTime;
   dpv:hasRecipient extp:me.
```

Listing 2: An example of generated participation statements in a RDF format in the participant's Solid pod.

When the participant approves the request, two key actions will happen. First, the participant gives the trusted (or authenticated) party access to the requested data elements in the pod. Second, the participant's WebID will be registered at the trusted party under the corresponding participation request URI (Step 4 in Fig. 3). Meanwhile, TIDAL generates logging information in the participant's pod including at what time the access has been granted, to whom (WebID), to what data elements, for what data request (request ID), and the valid period of the permission. The logging is readable by the participants, but not editable by anyone. Until now, data elements have not been accessed and retrieved by any parties.

If TIDAL fails to detect the requested data elements in the pod, the card displays the "Send an anonymous message to requester" option and the participant cannot participate in the research. It is possible that the participant does not have the requested data, or that the researcher and the participant use different standards or ontologies to describe the same data element. In this case, the participant can send messages to the researcher anonymously on TIDAL to report this issue.

4.3. Data retrieval and analysis execution

To process the request, the following conditions need to be satisfied: (1) the request is in the inclusion period, and (2) the number of participants exceeds the minimum number set in the request. When the request meets both conditions, the researcher can communicate with the trusted party on TIDAL to trigger the data retrieval and analysis. The trusted party hosts the data analysis component including verifying the request, querying data from the participants' pods, and executing the predefined analysis algorithms. The data analysis component was built using Javascript and Docker Containers. Docker Container has similar resource isolation and allocation benefits to virtual machines, creating temporary and secure sandboxes. We used the node-docker-api package (V1.1.22) [51] in a combination of the solid-node-client and rdflib.js libraries to access Solid pods from a Docker container.

Figure 6 shows the workflow of data retrieval and analysis after the researcher triggers the execution of a request. TIDAL will first generate and send a `schema:ActivateAction` message (Listing 3) to the trusted party. TIDAL retrieves the request file from the researcher's pod, parses it, and verifies it using the public key. The data must specify the docker image identifiers (`dpvtech:isImplementedUsingTechnology`), requested data elements (`schema:DataFeedItem`), valid period of the request (`dpv:hasExpiryTime`) and other input parameters for the trusted party to retrieve the Docker image from the central repository (e.g., Dockerhub) and execute the analysis. The main input for the analysis includes the requested data elements from each participant's pod, defined targeted feature, and hyper-parameters for the analysis model. An example code for an linear regression analysis in a Docker image can be found in *https://github.com/sunchang0124/TIDAL.*. The queried data elements from participants is converted to a tabular data format (e.g., Python Pandas DataFrame format) to be loaded and processed by the analysis models. Only the analysis results such as classification accuracy, precision, recall, regression coefficient can be output and sent to the researcher's pod. Finally, TIDAL can manage multiple data retrieval and analysis request from researchers simultaneously.

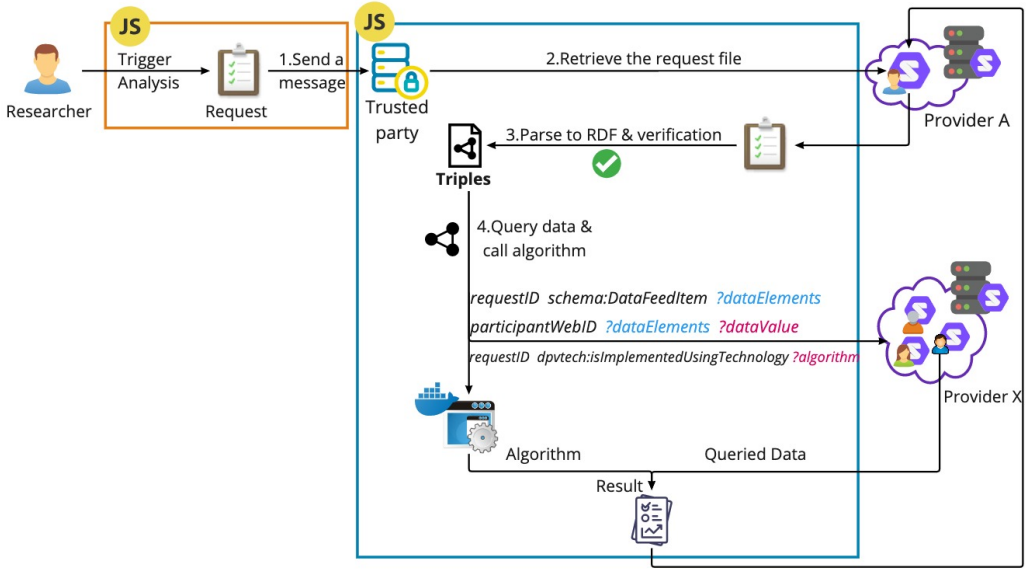


Fig. 6. Workflow of data retrieval and analysis triggered by the researcher.

```
@prefix : <http://examplertrustedparty.solidprovider.net/inbox/triggermessage#>.
@prefix req: <http://exampleresearcher.solidprovider.com/public/request.ttl#>.
@prefix exre: <http://exampleresearcher.solidprovider.com/profile/card#>.

:160622932739325095672093710975
   schema:actionStatus schema:ActivateAction;
   schema:creator exre:me;
   schema:dateCreated "2021-04-20T09:37:57.499Z"^^XML:dateTime;
   schema:target req:161964062096710764675982245664.
```

Listing 3: An example of generated trigger message (Activate Action) sent by the researcher.

If the integrity of the request is verified, the trusted party fetches the requested data elements from each participant's pod (adhering to participation constraints such as participation time period) without storing their identifiers (i.e., WebIDs). This process includes fetching and parsing the entire RDF data file from participant's pod, and querying the requested data elements. The participant's WebID as the subject and the URI of the requested data element as the predicate are used to query the value of the data element as the object. To explain it using a simple example, we take one triple about the participant's age <solid:participant01 SNOMEDCT:39766900227^^xsd:int> from a participant's data pod (Listing 4). The value of Age is retrieved by doing a query *solid:participant01 SNOMEDCT:397669002* ***?age*** using solid-node-client and rdflib.js libraries (the pseudo code of querying published requests and data elements from participants' Solid pods is provided in the Appendix A). When any data are being retrieved from the participants, TIDAL writes logging records in participants pods. These logging records identify what data elements are extracted, by whom (WebID), at what time, for which data request (request ID), and whether the analysis is executed. The queried data is then fed into the data analysis model, which is pre-defined in the Docker image. Finally, the results of the analysis will be generated automatically and sent back to the researcher's Solid pod. All information received and created by the trusted party, such as queried data and intermediate results, is destroyed.

```
@prefix : <https://exampleparticipant.solidprovider.com/profile/card#>.
@prefix SNOMEDCT: <http://purl.bioontology.org/ontology/SNOMEDCT/>.

:me a SNOMEDCT:116154003; # Patient
  SNOMEDCT:397669002 "27"^^xsd:int; # Age
  SNOMEDCT:50373000 "165"^^xsd:int; # Height
  SNOMEDCT:726527001 "55"^^xsd:int; # Weight
  SNOMEDCT:263495000 SNOMEDCT:248152002; # Gender, Female
  SNOMEDCT:271649006 "110"; # Systolic blood pressure
  SNOMEDCT:271650006 "90"; # Diastolic blood pressure
  SNOMEDCT:405751000 SNOMEDCT:44054006. # Type 2 diabetes
```

Listing 4: An example of the RDF data file in a participant's Solid pod.

5. Experiments and results

To evaluate the performance of TIDAL, we designed experiments with three following objectives: 1) to prove TIDAL can interact with an amount of pods that are hosted by different pod providers, 2) to prove TIDAL can retrieve an amount of data elements from these pods and conduct data analysis under reasonable time costs, 3) to prove the feasibility and report efficiency of each step in the workflow of TIDAL.

5.1. Experiment setting

At the time of implementation of TIDAL (December 2020), there were two public Solid pod providers: *Inrupt* and *Solidcommunity*. We experimented TIDAL using these two public pods providers and one self-hosted server. The self-hosted server was operated on the Data Science Research Infrastructure at Maastricht University[7]. Each

[7]Data Science Research Infrastructure (DSRI): https://maastrichtu-ids.github.io/dsri-documentation/

pod provider hosts 256 Solid pods, corresponding to 256 participants. Each participant has a data file containing 128 generated variables and values structured by SNOMED CT [52] vocabularies in RDF/turtle format in their Solid pods. A simplified data example is presented in Listing 4.

Using each pod provider, we conducted a set of experiments with varying scales of participants and configurations. We started with requesting 4 variables from 4 to 256 participants, and ended with requesting 128 variables from 4 to 256 participants. The experiments focused on the steps after the researcher collected enough responses from the participants and triggers the analysis. The execution time has been measured from:

1. querying the data request URI;
2. querying signature and verification key of data request;
3. verifying the signature to ensure the data request has not been modified;
4. (if the verification succeeds) querying the content of data request and WebIDs of participants; and
5. querying RDF data from all participants' pods.

The web interface of TIDAL was developed using the Semantic User Interface Framework (V2.4.2) [53] with responsive and scalable layout. We tested the web interface in recent versions of Safari, Chrome, and Firefox. Data retrieval and analysis are performed on a 2.3 GHz PC using Dual-Core Intel Core i5 with 16GB RAM and 500GB hard disk running MacOS 10.15.7. To run the simulation experiment, we created 256 Solid pods, generated and stored simulation data in each pod, and granted permission to the requests in an automatic way. The open-source code are published at: *https://github.com/sunchang0124/TIDAL*.

5.2. Results

Figure 7 shows how TIDAL scales for querying and analyzing data from individual pods as we increase the number of variables from 4 to 128 and the number of pods from 4 to 256 hosted by *Inrupt*, *Solidcommunity*, and the self-hosted server. The servers from the pod providers respond to a limited number of requests at one time. Considering the scalability, we enable TIDAL to parallelly and concurrently access participants' pods using a concurrent asynchronous function in Javascript (with NodeJS and Express package) to send HTTP requests. TIDAL only queries the required data elements from Solid pods of 64 participants simultaneously. Once a task gets finished, a new task is scheduled in the execution queue. We ran each experiment 10 times and presented the average time of the 10 experiments to avoid possible network latency fluctuations.

Figure 7 shows that the total time costs in querying 4 and 8 pods is approximately 4 to 5 seconds with a negligible increase as the number of variables increases. When we query data from a large number of pods, the time costs in fetching data from participants' pods becomes substantial. It rises linearly when we increase the number of pods using all pod providers. In the case of querying data from 256 pods, a gradual increase in time costs is observed as the number of variables increases. In all experiments, the time costs of the first 4 execution steps are constant and independent of how many variables and pods are required because they query information from a fixed number of pods from researchers or trusted parties.

Figure 8 shows the total time cost when querying the number of variables from 4 to 128 and the number of pods from 4 to 256 on three pod providers. From the experiments on all pod providers, the total time cost linearly scales when the number of pods is increased. The more variables are queried from each pod, the steeper the increase in time cost is presented. On the contrary, the *Inrupt* server has a more stable rate and the least time consumption than the other two pod providers when querying data from more than 64 pods.

6. Discussion

We have demonstrated and tested a ciTIzen-centric DatA pLatform (TIDAL) using an increasing number of requested data elements retrieved from an increasing number of Solid pods. From the performance evaluation of TIDAL, the execution time shows a linear correlation between the number of pods and the number of variables. The process expends the most of the time in querying data from all the participants. However, it only requires an average of 40 seconds to query 128 variables from 256 participants' Solid pods. For a limited set of participants, this can be

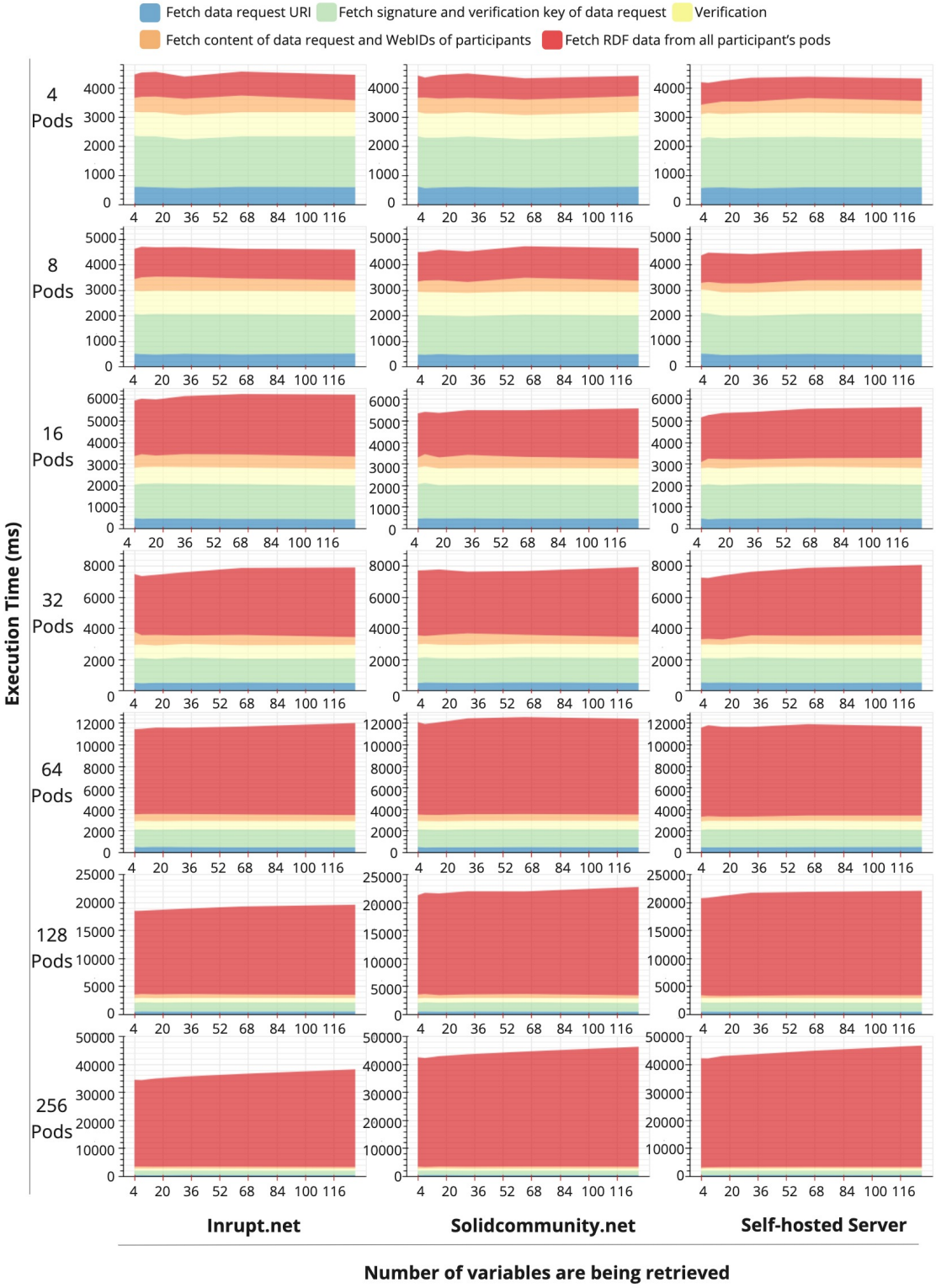


Fig. 7. Time costs in each execution steps in querying and analyzing data from Solid pods with increasing the number of variables and pods hosted by *Inrupt*, *Solidcommunity*, and self-hosted server respectively.

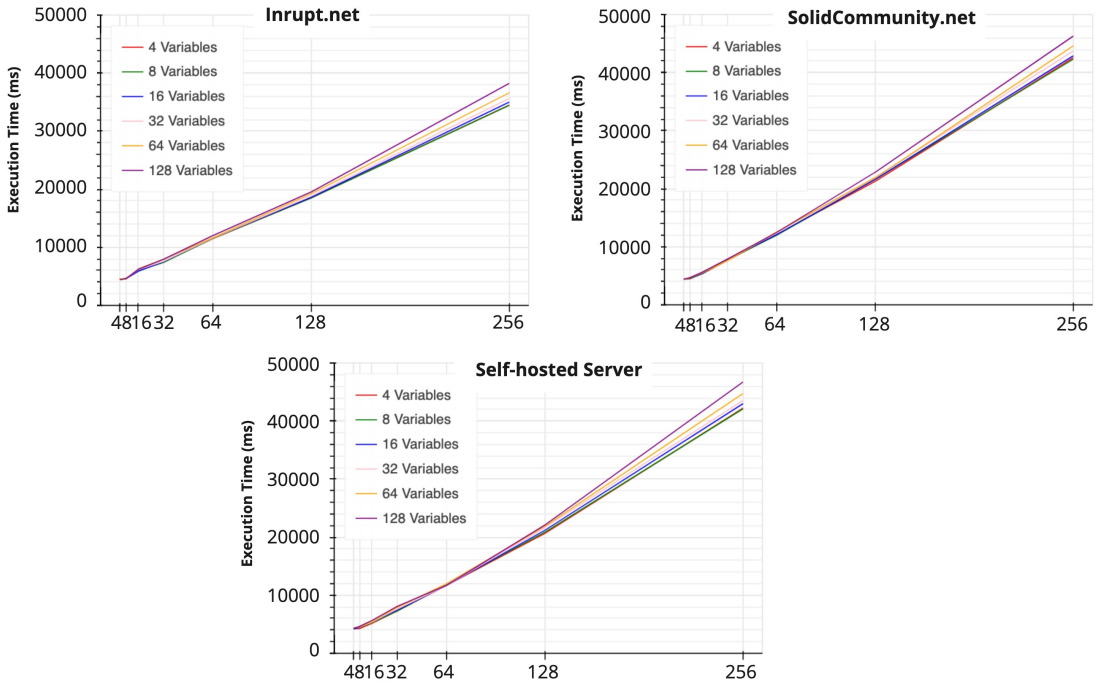


Fig. 8. Total time costs in querying the increasing number of variables and pods on three pod providers Inrupt, Solidcommunity, and self-hosted server respectively.

considered as an acceptable time for a batch process for use cases that do not demand instant results. In the future, we will improve the workflow and reduce processing time.

When querying data from enormous pods, the number of variables being queried influences the total querying time (Fig. 8). A possible solution to improve the query performance would be the provision of SPARQL support on Solid pods which is missing in the current Solid specification. A SPARQL endpoint would facilitate the execution of complex queries on pods instead of retrieving full RDF files and post-processing them on the client side to extract the requested data elements, decreasing applications performance. The increase in time of the "Fetch RDF data from all participants pods" (Fig. 7) and execution (Fig. 8) is also influenced by the limited number of simultaneous requests handled by the Solid server. Additionally, the processing capability of the experimental hardware also created a bottleneck in querying and analyzing data processes. Therefore, for a practical application we advise the allocation of sufficient computational resources at key architectural locations to reduce the potential bottle-neck when querying and analyzing data from a large number of participants' pods.

TIDAL supports users to store and request personal data in a structured RDF format using well-established ontologies and terminologies by integrating the Bioportal API. The data structured with ontologies and terminologies help users linking their data from multiple sources to enrich and improve the quality of data in their pods. The structured data are human and machine readable, provide language neutrality, unambiguous definitions, and clear relationships. TIDAL works on a decentralized network where people can choose to store their data in different and multiple pods, or self-host the pod individually. Self-hosted data pods in our platform do not differ from other scenarios in terms of granting access, managing data (creating, modifying, deleting), donating their data to some research studies. Furthermore, the data protection laws such as EU General Data Protection Regulation (GDPR) require a number of information requirements about consent for processing personal data. To integrate those requirements into digital consent, TIDAL used the Data Privacy Vocabulary in the participation request to describe and represent information related to requesting and processing of personal data. Data protection laws grant data subjects (participants) the right to withdraw or modify their data anytime they want. On TIDAL, these rights are respected. After the participants approve the data request, they can still update the data elements or withdraw the

approval decision anytime via TIDAL or from their Solid pods directly. On TIDAL, for example, the participants can go to "Create" page and give the data file path (e.g., https://username.podprovider.net/private/medicalrecord.ttl) as the subject in the data triple, the URI of the data element as the predicate, and the new data value as the object. The analysis is conducted on the (latest) updated value of the data element or without the data elements which have been withdrawn. This process can also enhance reproducibility in research, as researchers can expand and scale their research, both in participants and in future long-term effects / follow-up studies.

The data of the participants are queried and analyzed only at the trusted party. The trusted party can be a separate, independent entity in comparison to the researcher, Solid provider and/or participant. Researchers can only formulate the request, define the algorithm parameters and receive the final results of the analysis but never have access to the data. However, if the Solid provider hosts the trusted party, this trusted party can become a node in a Personal Health Train (PHT) or Federated Learning (FL) infrastructure. In such an infrastructure, the research question travels to the data rather than data being transported to the research question. PHT or FL methodologies connect multiple distributed data sources (e.g., hospitals, clinics) and enable researchers to send analysis models to each data source (e.g. Solid providers) and get the final learning results. In addition to strengthening the binary connection between data sources and researchers, TIDAL emphasizes on engaging individuals in health research and connecting them with both researchers and data sources. This is currently still missing in most PHT/FL implementations.

Our work has to be seen in light of some potential limitations. First, we assume participants have their personal data structured, uploaded, and stored in their own Solid pods. In practice, people who do not have enough knowledge about the data or the technologies will face challenges to structure and store a large amount of data correctly. To tackle this challenge, one solution can be that TIDAL provides well-structured data models and adds functions to convert different formats of data files (e.g., CSV, XML) into RDF turtle file and upload converted RDF triples to Solid pods for participants. Another solution can be encouraging data collectors such as hospitals or pharmacies to help participants structure their own data. For example, if the patients' medical records have been structured and linked with some international terminologies by the hospital, then the hospital can request to store the structured medical records data to patients' Solid pods directly. This is connected to other research topics of data modeling and data conversion for personal health data in Solid pods which is beyond the scope of the current study.

Furthermore, the current version of TIDAL presents every published data request that is in the valid period to all participants. In this case, participants receive some data requests that are not relevant to them. As researchers do not know which participants have the relevant data for their research, they are unable to send the data request to the target cohort instead of the general public. Therefore, to improve TIDAL, we invest in generating privacy-preserving metadata for each solid pod. The privacy-preserving metadata is supposed to describe sufficient information about one pod but not reveal any sensitive information. One of the potential solutions is to employ Bloom Filter which is a probabilistic data structure for efficient set membership querying [54]. Bloom filter tests whether the participant has the requested data elements in their data pods and returns two possible answers: "probably in the pod" or "definitely not in the pod". With this method, we can prevent participants in a specific study (e.g. for psychological disorders) from being identified that they are diagnosed with a specific disease or disorder. Another approach is that TIDAL asks participants to indicate their preference on the type of research and data request. For example, if the participant is only interested in diabetes research, then TIDAL will only present data requests that are related to diabetes research in order to decrease the complexity of using TIDAL for general users.

7. Conclusion and future work

In this paper, we presented a novel citizen-centric data platform (called TIDAL) to give individuals fine-grained access to their data and facilitate health research. The TIDAL platform not only collects data and manages digital consents, but also structures data requests with integrating the vocabulary services and standards such as the Data Privacy Vocabulary. The data requests are used as a digital consent and provide the algorithm parameters and model configuration for the predefined data analyses. The analyses are executed in an automatic manner which ensures the data to be exactly analyzed as promised by the researchers in their data requests. Finally, only the analysis results are sent to the researchers. We demonstrated the feasibility and efficiency of TIDAL by running a set of simulation experiments using different numbers of variables and Solid pods hosted on three different providers

(*Inrupt*, *Solidcommunity* and a self-hosted server). TIDAL is not only limited to health research, it can be used in other fields such as social sciences (e.g., demographic and anthropology studies), economic and finance studies, political, marketing, and education research.

To improve the user experience, we intend to recruit a group of users to assess the human interaction of TIDAL and collect their feedback. In the future, we will evaluate TIDAL in a real-life use case with real participants and health researchers. We will evaluate how usable the request form is for researchers and how long it will take researchers to complete the entire request form. Meanwhile, we will also investigate how understandable the data request cards are for general participants, and how easy they feel to approve and withdraw the permissions.

The current version of TIDAL allows researchers to perform only a predefined set of analysis models. More analysis models will be designed in future work for complex analysis and experiments according to the researchers' scientific questions. Researchers can apply the needed model and tune the parameters rather than coding or modifying the entire model. The risk of hacking or data leakage in the analysis process can be minimized. It is also possible for researchers to register a new analysis algorithm by themselves by pushing the analysis Docker image to Docker hub so that TIDAL access and pull the image. We require these Docker images to be public so that everyone can view and check the algorithms. However, how to check, control, authenticate, and authorize these Docker images for analysis is still ongoing research. Another future work can be considered is to improve the logging process. The logging files in the current version of TIDAL store the data access records in participantss' Solid pods when the participants grant permission or anyone access to their data. Next, we intend to investigate in applying Blockchain technologies for handling loggings in a more transparent and secure manner. Several studies have developed tools integrating Solid and Blockchain [55], [56].

Furthermore, the current version of TIDAL only handles static data. In the further development, we consider extending TIDAL to also handle streams of RDF data (RDF triples or graphs with temporal annotations) or real-time data processing [57]. For example, TIDAL users can synchronize their health or fitness data from their wearable devices such as mobile phones or fitness watches to their Solid pods. These data are first converted to RDF stream data and stored in the users' pods. Then, we consider integrating with RDF stream processing engines in TIDAL to handle the long-standing query, which is continuously executed, over RDF stream data from the distributed Solid data pods.

Acknowledgements

Financial support for this study was provided by a grant from the Dutch National Research Agenda (NWA; project number: NWA.1418.20.006). We would like to thank our past colleagues (Federico Igne, Gianmarco Spinaci, Glenda Amaral, Kabul Kurniawan), and our tutor (Dr. John Domingue) from the International Semantic Web Summer School 2019 (ISWS 2019) for brainstorming and generating the idea. We gratefully acknowledge the time and effort devoted by Dr. Andre Dekker and Dr. Leonard Wee for their feedback and suggestions to help us construct the platform in the early developmnent stage. We thank Vincent Emonet and Tim Hendriks for their technical support for the platform. We appreciate Harshvardhan J. Pandit for his generous comments and very helpful discussion during revision of the manuscript. Finally, special thanks are given to the reviewers (Dimitrios Karapiperis, Vassilis Kilintzis, and Pavlos Fafalios) for their valuable comments and feedback to improve this study.

Appendix A. Examples (pseudo code) of retrieving request files and data from Solid pods

Pseudo-code 1 Retrieve valid requests and display on TIDAL

```
1: procedure RETRIEVE A PUBLISHED REQUEST
2:     PublishedRequestslist ← a list of URIs of published requests from TIDAL index file
3:
4:     for PublishedRequestslist do
5:         RequestRDF ← await fetchDocument(PublishedRequestslist)  ▷ Fetch and parse RDF triples from pods
6:         ExpiryDate ← RequestRDF.getObject(schema.endDate))
7:                 ▷ Query ExpiryDate (Object in the triple) from <RequestURI schema:endDate ?ExpiryDate>
8:
9:         if ExpiryDate > today then
10:            Display RequestRDF
```

Pseudo-code 2 Accessing data elements from participants using TIDAL

```
1: After the request is successfully verified:
2: procedure RETRIEVING REQUESTED DATA FROM A SOLID POD
3:     RequestURI ← a request with enough participants and being triggered by the researcher
4:     ParticipantWebIDs ← a list of WebIDs from participants who approved the data request
5:
6:     RequestRDF ← await fetchDocument(RequestURI)                    ▷ Fetch and parse the request file
7:     RequestedDataElementURI ← RequestRDF.getObject(schema.DataFeedItem)    ▷ URIs of data elements
8:
9:     for ParticipantWebIDs do
10:        for RequestedDataElement do
11:            AllDataTriples ← await fetchDocument(ParticipantDataFile)    ▷ Fetch and parse the data file
12:            RequestedDataValue ← AllDataTriples.getObject(RequestedDataElementURI)
13:                    ▷ Query the data value (an object in a triple: e.g., <SomeWebID schema:name ?name>
```

References

[1] J. Chen, C.D. Mullins, P. Novak and S.B. Thomas, Personalized strategies to activate and empower patients in health care and reduce health disparities, *Health Education & Behavior* **43**(1) (2016), 25–34. doi:10.1177/1090198115579415.

[2] T. Hulsen, Sharing is caring—data sharing initiatives in healthcare, *International journal of environmental research and public health* **17**(9) (2020), 3046.

[3] European Commission, White paper: A European strategy for data, Technical Report, COM(2020) 66 final, European Commission, 2020. https://eur-lex.europa.eu/legal-content/EN/TXT/?qid=1593073685620&uri=CELEX/3A52020DC0066.

[4] European Commission, White paper: on enabling the digital transformation of health and care in the Digital Single Market; empowering citizens and building a healthier society, Technical Report, COM(2018) 233 final, European Commission, 2018. https://ec.europa.eu/digital-single-market/en/news/communication-enabling-digital-transformation-health-and-care-digital-single-market-empowering.

[5] European Parliament, Understanding EU data protection policy, Technical Report, European Commission, 2020. https://www.europarl.europa.eu/RegData/etudes/BRIE/2020/651923/EPRS_BRI(2020)651923_EN.pdf.

[6] European Commission, Summary report of the public consultation on the European strategy for data, Technical Report, COM(2018) 233 final, European Commission, 2020. https://ec.europa.eu/digital-single-market/en/news/summary-report-public-consultation-european-strategy-data.

[7] Dutch Techcentre for Life Sciences (DTL), Personal Health Train, https://www.dtls.nl/fair-data/personal-health-train/, Access on 12-8-2021.

[8] C. Sun, L. Ippel, J. Van Soest, B. Wouters, A. Malic, O. Adekunle, B. van den Berg, O. Mussmann, A. Koster, C. van der Kallen et al., A Privacy-Preserving Infrastructure for Analyzing Personal Health Data in a Vertically Partitioned Scenario., in: *MedInfo*, 2019, pp. 373–377.

[9] J. Van Soest, C. Sun, O. Mussmann, M. Puts, B. van den Berg, A. Malic, C. van Oppen, D. Townend, A. Dekker and M. Dumontier, Using the Personal Health Train for Automated and Privacy-Preserving Analytics on Vertically Partitioned Data., in: *MIE*, 2018, pp. 581–585.

[10] A. Jochems, T.M. Deist, J. Van Soest, M. Eble, P. Bulens, P. Coucke, W. Dries, P. Lambin and A. Dekker, Distributed learning: developing a predictive model based on data from multiple hospitals without data leaving the hospital–a real life proof of concept, *Radiotherapy and Oncology* **121**(3) (2016), 459–467.
[11] E. Mansour, A.V. Sambra, S. Hawke, M. Zereba, S. Capadisli, A. Ghanem, A. Aboulnaga and T. Berners-Lee, A demonstration of the solid platform for social web applications, in: *Proceedings of the 25th International Conference Companion on World Wide Web*, International World Wide Web Conferences Steering Committee, 2016, pp. 223–226. doi:10.1145/2872518.2890529.
[12] H.J. Pandit, A. Polleres, B. Bos, R. Brennan, B. Bruegger, F.J. Ekaputra, J.D. Fernández, R.G. Hamed, E. Kiesling, M. Lizar, E. Schlehahn, S. Steyskal and R. Wenning, Creating a Vocabulary for Data Privacy, in: *On the Move to Meaningful Internet Systems: OTM 2019 Conferences*, Springer International Publishing, 2019, pp. 714–730. ISBN 978-3-030-33246-4.
[13] A. Polleres, B. Esteves, B. Bos, B. Bruegger, E. Kiesling, E. Schlehahn, D. Hickey, F.J. Ekaputra, G.P. Krog, H.J. Pandit, J.D. Fernández, J. Flake, M. Lizar, P. Ryan, P. Bonatti, R.G. Hamed, R. Wenning, R. Brennan and S. Steyskal, Data Privacy Vocabulary (DPV) - version 0.2, Accessed on 12-08-2021.
[14] T. Symons and T. Bass, Me, my data and I: The future of the personal data economy, Technical Report, DECODE (DEecentralised Citizen Owned Data Ecosystems), 2017.
[15] M. Koscina, D. Manset, C. Negri and O. Perez, Enabling Trust in Healthcare Data Exchange with a Federated Blockchain-Based Architecture, in: *IEEE/WIC/ACM International Conference on Web Intelligence - Companion Volume*, WI '19 Companion, Association for Computing Machinery, New York, NY, USA, 2019, pp. 231–237–. ISBN 9781450369886. doi:10.1145/3358695.3360897.
[16] OwnYourData, Own Your Data, Accessed on 12-08-2021. https://www.ownyourdata.eu/.
[17] MIDATA Cooperative, MIDATA: My Data - Our Health, Accessed on 12-08-2021. https://www.midata.coop/en/home/.
[18] MedMij, MedMij: Personal health data in the palm of your hand, Accessed on 12-08-2021. https://www.medmij.nl/en.
[19] DIGI.ME LTD., Digi.me, Access on 12-08-2021. https://digi.me.
[20] C. Cloud, Cozy Cloud, Access on 12-08-2021. https://cozy.io/.
[21] HAT Community Foundation and Dataswift, Hub of All Things (HAT), Access on 06-10-2021. https://www.hubofallthings.com/.
[22] Mydex CIC, MyDex, Access on 06-10-2021. https://mydex.org.
[23] Y.-A. de Montjoye, E. Shmueli, S.S. Wang and A.S. Pentland, openPDS: Protecting the Privacy of Metadata through SafeAnswers, *PLOS ONE* **9**(7) (2014). doi:10.1371/journal.pone.0098790.
[24] H. Janssen, J. Cobbe, C. Norval and J. Singh, Decentralised data processing: Personal data stores and the gdpr, *International Data Privacy Law* **10**(4) (2020), 356–384. doi:10.1093/idpl/ipaa016.
[25] European Commission, The European Union's Horizon 2020 research and innovation programme, Access on 12-08-2021. https://ec.europa.eu/programmes/horizon2020/.
[26] A. Sonnino, M. Al-Bassam, S. Bano and G. Danezis, Coconut: Threshold Issuance Selective Disclosure Credentials with Applications to Distributed Ledgers, in: *Network and Distributed Systems Security (NDSS) Symposium 2019*, 2019. doi:10.14722/ndss.2019.23272.
[27] 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, *MIT CSAIL & Qatar Computing Research Institute, Tech. Rep.* (2016). http://emansour.com/research/lusail/solid_protocols.pdf.
[28] Solid, Solid: Your data, your choice, Access on 12-08-2021. https://solidproject.org/.
[29] S. Lohr, He Created the Web. Now He's Out to Remake the Digital World., *The New York Times* (2021). https://www.nytimes.com/2021/01/10/technology/tim-berners-lee-privacy-internet.html.
[30] F. Giunchiglia, R. Zhang and B. Crispo, Ontology driven community access control, Technical Report, University of Trento, 2008. http://ceur-ws.org/Vol-447/paper3.pdf.
[31] S. Capadisli, T. Berners-Lee, R. Verborgh and K. Kjernsmo, Solid Protocol - Version 0.9.0, Access on 20-07-2022.
[32] T.M. Deist, F.J. Dankers, P. Ojha, M.S. Marshall, T. Janssen, C. Faivre-Finn, C. Masciocchi, V. Valentini, J. Wang, J. Chen et al., Distributed learning on 20 000+ lung cancer patients–The Personal Health Train, *Radiotherapy and Oncology* **144** (2020), 189–200.
[33] Z. Shi, I. Zhovannik, A. Traverso, F.J. Dankers, T.M. Deist, P. Kalendralis, R. Monshouwer, J. Bussink, R. Fijten, H.J. Aerts et al., Distributed radiomics as a signature validation study using the Personal Health Train infrastructure, *Scientific data* **6**(1) (2019), 1–8. doi:10.1038/s41597-019-0241-0.
[34] O. Beyan, A. Choudhury, J. van Soest, O. Kohlbacher, L. Zimmermann, H. Stenzhorn, M.R. Karim, M. Dumontier, S. Decker, L.O.B. da Silva Santos et al., Distributed analytics on sensitive medical data: The Personal Health Train, *Data Intelligence* **2**(1–2) (2020), 96–107.
[35] Solid team, Get a Pod from a Pod Provider, https://solidproject.org/users/get-a-pod, Access on 8-8-2022.
[36] Solid team, Running your own Solid server, https://solidproject.org//self-hosting/css, Access on 8-8-2022.
[37] Solid, Solid access to Pods, local file systems, and other backends via nodejs, Access on 12-08-2021. https://www.npmjs.com/package/solid-node-client.
[38] LinkedData team, Javascript RDF library for browsers and Node.js, Access on 28-02-2021. https://github.com/linkeddata/rdflib.js/.
[39] Inrupt, Tripledoc - The easiest way to get started writing Solid apps, Access on 28-02-2021. https://vincenttunru.gitlab.io/tripledoc/.
[40] R. Verborgh and R. Taelman, LDflex: A Read/Write Linked Data Abstraction for Front-End Web Developers, in: *International Semantic Web Conference*, Springer, 2020, pp. 193–211.
[41] D.J. Bernstein, N. Duif, T. Lange, P. Schwabe and B.-Y. Yang, High-speed high-security signatures, *Journal of cryptographic engineering* **2**(2) (2012), 77–89. doi:10.1007/s13389-012-0027-1.
[42] D.J. Bernstein, S. Josefsson, T. Lange, P. Schwabe and B.-Y. Yang, EdDSA for more curves, *Cryptology ePrint Archive* (2015). https://eprint.iacr.org/2015/677.

[43] S. Josefsson and I. Liusvaara, Edwards-curve digital signature algorithm (eddsa), in: *Internet Research Task Force, Crypto Forum Research Group, RFC*, Vol. 8032, 2017, pp. 257–260. https://www.rfc-editor.org/rfc/pdfrfc/rfc8032.txt.pdf.
[44] J. Brendel, C. Cremers, D. Jackson and M. Zhao, The provable security of ed25519: theory and practice, *IEEE Security & Privacy* (2021).
[45] D. Chestnykh, D. Mandiri and AndSDev, TweetNaCl.js - a port of TweetNaCl / NaCl to JavaScript, Access on 28-02-2021. https://www.npmjs.com/package/tweetnacl.
[46] D.J. Bernstein, T. Lange and P. Schwabe, NaCl: Networking and Cryptography library, 2016, Access on 28-02-2021. http://nacl.cr.yp.to/.
[47] N. Noy, N. Shah, P. Whetzel, B. Dai, M. Dorf, N. Griffith, C. Jonquet, D. Rubin, M. Storey, C. Chute and M. Musen, BioPortal: Ontologies and integrated data resources at the click of a mouse, *Nucleic Acids Research* **37**(SUPPL. 2) (2009), W170–W173, Funding Information: National Center for Biomedical Ontology, under roadmap-initiative from the National Institutes of Health [grant U54 HG004028]. Funding for open access charge: National Institutes of Health [grant U54 HG004028]. doi:10.1093/nar/gkp440.
[48] A. Kurteva, T.R. Chhetri, H.J. Pandit and A. Fensel, Consent through the lens of semantics: State of the art survey and best practices, *Semantic Web* (2021), 1–27. https://content.iospress.com/articles/semantic-web/sw210438.
[49] B. Esteves and V. Rodríguez-Doncel, Analysis of ontologies and policy languages to represent information flows in GDPR, *Semantic Web* (2022), 1–35. https://content.iospress.com/articles/semantic-web/sw223009.
[50] H.J. Pandit and B. Esteves, Enhancing Data Use Ontology (DUO) for Health-Data Sharing by Extending it with ODRL and DPV, *Semantic Web* (2022), Under review. http://www.semantic-web-journal.net/content/enhancing-data-use-ontology-duo-health-data-sharing-extending-it-odrl-and-dpv.
[51] A.C. Berrini, Docker Remote API driver for node.js, Access on 28-02-2021. https://www.npmjs.com/package/node-docker-api.
[52] K. Donnelly, SNOMED-CT: The advanced terminology and coding system for eHealth, *Studies in health technology and informatics* **121** (2006), 279. https://pubmed.ncbi.nlm.nih.gov/17095826/.
[53] Semantic-UI, Semantic - a UI framework designed for theming, 2013, Access on 20-02-2021. https://semantic-ui.com/.
[54] B.H. Bloom, Space/Time Trade-Offs in Hash Coding with Allowable Errors, *Commun. ACM* **13**(7) (1970), 422–426–. doi:10.1145/362686.362692.
[55] A. Third and J. Domingue, Decentralised Verification Technologies and the Web, in: *Media, Technology and Education in a Post-Truth Society*, Emerald Publishing Limited, 2021. doi:10.1108/978-1-80043-906-120211018.
[56] M. Eisenstadt, M. Ramachandran, N. Chowdhury, A. Third and J. Domingue, COVID-19 antibody test/vaccination certification: there's an app for that, *IEEE Open Journal of Engineering in Medicine and Biology* **1** (2020), 148–155. doi:10.1109/OJEMB.2020.2999214.
[57] S. Sakr, M. Wylot, R. Mutharaju, D. Le Phuoc and I. Fundulaki, Processing of RDF Stream Data, in: *Linked Data*, Springer, 2018, pp. 85–108.