GDPR-inspired IoT Ontology enabling Semantic Interoperability, Federation of Deployments and Privacy-Preserving Applications

Rachit Agarwal  
MiMove  
Inria-Paris, Paris, France  
rachit.agarwal@inria.fr

Tarek Elsaleh  
Centre for Vision, Speech and Signal Processing  
University of Surrey, Guildford, United Kingdom  
t.elsaleh@surrey.ac.uk

Elias Tragos  
Insight Centre for Data Analytics, University College Dublin, Dublin, Ireland  
elias.tragos@insight-centre.org

Abstract—Testing and experimentation are crucial for promoting innovation and building systems that can evolve to meet high levels of service quality. IoT data that belong to users and from which their personal information can be inferred are frequently shared in the background of IoT systems with third parties for experimentation and building quality services. This sharing raises privacy concerns, as in most cases, the data are gathered and shared without the user’s knowledge or explicit consent. With the introduction of GDPR, IoT systems and experimentation platforms that federate data from different deployments, testbeds, and data providers must be privacy-preserving. The wide adoption of IoT applications in scenarios ranging from smart cities to Industry 4.0 has raised concerns for the privacy of users’ data collected using IoT devices. Inspired by the GDPR requirements, we propose an IoT ontology built using available standards that enhances privacy, enables semantic interoperability between IoT deployments, and supports the development of privacy-preserving experimental IoT applications. We also propose recommendations on how to efficiently use our ontology within a IoT testbed and federating platforms. Our ontology is validated for different quality assessment criteria using standard validation tools. We focus on “experimentation” without loss of generality because it covers scenarios from both research and industry that are directly linked with innovation.

Index Terms—Semantic Interoperability, Privacy, IoT, Best Practices, Testbeds, Experimentation

I. INTRODUCTION

IoT experimentation assumes that a testbed/data provider provides the collected data to third parties (i.e., researchers) for running experiments. This requires sharing the data with external parties that are not the data owners. Thereby raising privacy concerns if external parties mishandle the data. The latest European Union (EU) directive for data protection (GDPR - General Data Protection Regulation [1]) sets strict guidelines on the collection and processing of user data. Consider a user participating in a crowd-sourcing based environmental monitoring testbed by installing an application on their smartphone. A third party having access to the data may be able to interpret the user’s information such as location at any given time.

Thus, one important aspect of the provided data is its privacy sensitivity and the context in which IoT devices capture it.

For a multi-domain experimentation facility, it is important to provide a flexible means to configure the level of data privacy, depending on the association with real-world entities, and the purpose of data processing. With the introduction of GDPR, a data model that reflects the concepts defined by these laws is essential. This requires modelling the roles of different actors, the consent mechanism, and the data owner’s permissions on what third parties can consume.

Past activities in IoT experimentation were focused on developing the fundamental technologies to facilitate the provision of experimentation services on top of shared testbed(s) and mostly neglected privacy. IoT Data marketplace platforms such as Big-IoT [2] provide interoperability but do not focus on data privacy. A federation platform such as FIESTA-IoT [3] is semantically enabled and provides a module for user authorisation, but it does not consider data privacy.

The lack of focus on data privacy from IoT experimentation platforms motivates us to propose a privacy-enhanced ontology specifically designed for IoT experimentation (but easily generalised for any IoT system). This includes main privacy concepts (as extracted by GDPR and ISO/IEC 29100 [4]) and helps achieve interoperability between testbeds by targeting horizontal silos of IoT. Privacy in IoT experimentation projects is usually addressed using signed agreements with users, getting their consent to gather and use data for experimentation purposes. This may cause issues when a user wants to change or revoke her consent, which requires communication with the testbed owners for requesting these changes. This is usually a timely and manual process, during which the users’ data are still getting gathered, stored, and (re-)used. Our motivation is to provide an ontology that allows users to dynamically alter their policies and consent in near “real-time,” so that any changes can immediately affect. Also, the users can create advanced and personalised policies and testbed owners will be able to provide new applications, having a process to request the consent of users. The key contributions of this work are:

- Ontology alignment: We follow the methodology proposed by Noy et al. [5] and align our ontology to use standard
and most popular ontologies such as SSN\textsuperscript{1} to easily achieve semantic interoperability between IoT testbeds.

- **Addition of privacy-related concepts**: as per GDPR and ISO guidelines, we include essential concepts required for enabling privacy-aware experimentation.

- **Ontology usage recommendation**: We provide recommendations for testbeds/data providers and federation platforms on efficiently using our ontology. We further provide application scenarios where our ontology fits best.

Note that we do not use the ontologies in totality, rather use and import only those concepts and properties that are needed to support our requirements. Without which, our ontology would be (a) bulky where most of the concepts and properties are useless and potentially replicated, (b) testbeds might use different (with the same meaning) concepts coming from different ontologies, thereby losing the semantic interoperability. Additionally, the proposed ontology does not intend to cover all GDPR requirements related to the full data life-cycle. Our main focus is on data gathering and sharing, providing entities that cover the phases of when (or not) user’s data should be gathered and to which other users/applications these data should be shared. Data retention, processing and deletion are not directly covered by our ontology, but could be indirectly addressed via the privacy policies.

II. RELATED WORK

An ontology that targets horizontal silos of IoT must address 4W1H (What, When, Where, Who, How) related competency questions [6]. However, most IoT ontologies address what, where, and when related competency questions. They fail to address the who part. In [6], the authors identified that no ontology is comprehensive enough in the IoT domain to address 4W1H. Popular ontologies such as SSN, IoT-lite\textsuperscript{2}, and oneM2M\textsuperscript{3} lack privacy and access control concepts. SSN further lacks concepts such as service, unit of measurement, and location. OneM2M provides an abstract model for IoT devices. While IoT-lite lacks concepts about observations. With many surveys on IoT ontologies already available [6], [7], we refrain from detailing them in this work. Instead, we focus on recent privacy-related IoT ontologies.

Several ontologies such as [8]–[11] target access control and privacy. However, they do not focus on consent for accessing particular data, a key requirement in GDPR. Nonetheless, developers can reuse privacy and access control concepts from these ontologies. Fatema et al. [12] proposed a consent ontology targeting the competency question of who is allowed or denied activity on the data. In another similar work, Pandit et al. [13] proposed provenance and consent ontology. These ontologies are very early work and require much attention to be complete, concise, and consistent.

In [14], the authors proposed IoT-Priv that extends IoT-lite ontology to include who aspects and target privacy. Thus, IoT-Priv focuses only on resource storage and access policies. It neglects observations taken by the resources. Moreover, there is no notion of a data controller. It also does not follow the conventions proposed by GDPR. LloPy [15] and PrOnto [16] are privacy-related ontologies with a legal focus. They are not optimised for advanced access control nor consider the dynamic context of an IoT environment. Many past EU IoT-related projects (such as IoT-A [17]) did not include privacy in their ontologies. To the best of our knowledge, no ontology enables privacy, interoperability, federation, and experimentation in IoT. The closest one being the FIESTA-IoT ontology\textsuperscript{4} [18]. Using concepts from previously well-defined ontologies, it defines a taxonomy for the different IoT devices, phenomena, and measurement units adopted across multiple IoT domains, but misses the privacy-related concepts.

III. PRIVACY REQUIREMENTS

Privacy requirements are usually extracted from the privacy principles proposed in ISO 29100 [4]. For IoT, there have been several attempts to adapt the ISO requirements, but for our ontology, we adopt the eight privacy requirements as defined by the EU project RERUM [19], [20], as an adaptation of the ISO principles combined with the requirements of GDPR:

1) **Consent and choice**: IoT applications should only collect data when a user gives explicit consent and allow the user to disable the collection of personal information or withdraw consent at a later stage.

2) **Purpose legitimacy and specification**: The collection and processing of personal data should only be allowed for a specific and legitimate purpose. Post-processing of collected data for a different purpose should be forbidden.

3) **Collection limitation**: The collected data should be adequate and relevant for the purpose and not excessive.

4) **Data minimisation**: is closely related to collection limitations and requires only the minimum amount of necessary data to be collected and processed.

5) **Accuracy and quality**: collected data must be accurate and up-to-date, deleting or rectifying false/incorrect data.

6) **Notice and access**: users should be notified when their data are collected, and allowed to access/delete their data.

7) **Individual participation and transparency**: users should have complete control over their data, know who has accessed/accessing it. Users should also have the ability to start/stop the data collection process at any given time.

8) **Accountability**: this ensures whenever there is a privacy breach, the IoT system should be able to hold the responsible person accountable for that breach.

IV. ONTOLOGY AND RELATED DESCRIPTION

Following the methodology in [5], we update the ontology defined in [18] to include concepts enabling interoperability, federation, and experimentation and comply with the new version of SSN ontology. Our motivation to update [18] is to ease the migration of testbeds that use FIESTA-IoT ontology and ease new semantically enabled testbeds to federate with

\textsuperscript{1}SSN: http://www.w3.org/2003/01/llw
\textsuperscript{2}IoT-lite: http://purl.oclc.org/NET/UNIS/fiware/iot-lite#.
\textsuperscript{3}OneM2M: https://tinyurl.com/tmnn24y
\textsuperscript{4}FIESTA-IoT: http://purl.org/iot/ontology/tega-iot#
Fig. 1: Modified version of Ontology defined in [18]. For clarity, we only show concepts related to Sensors. Note that sosa:FeatureOfInterest is replicated to show the connections clearly. They should be considered the same.

other testbeds. The ontology in [18] is modelled using two subgraphs (i) one contains essential concepts for defining a resource (resource graph), while (ii) another defines the observations taken by a resource (observation graph). Similar to [18], we adopt all the resource and observation-based concepts and properties in the new proposed ontology as well from well-known ontologies such as SSN, SSN-Systems5, SOSA6, IoT-Lite, M3-lite7, QU8, Geo9, Geo-sparql10, Schema.org11, and DUL12. We use owl:equivalentClass to link with popular ontologies such as oneM2M. This enables a testbed that is already semantically enabled to federate with those that would use our ontology. Figure 1 shows the proposed changes in detail. The specific concepts we used are colour encoded. A unique colour depicts the ontology from which a particular concept is used. Nonetheless, we mark the changes in the concepts with a red dot and the changes in the properties with a red colour. We kept the core idea of reusing concepts intact and tried to perform minimal yet required changes.

In Figure 1, concepts belonging to the resource graph are marked with a brown dot, while those belonging to the observation graph are marked with a blue dot. A multi-coloured dot depicts that a particular concept belongs to more than one subgraph. The changes worth highlighting are:

- We update M3-lite to reflect the new version of SSN. We move object and data properties defined in m3-lite to our ontology and use this prefix to denote them. This makes M3-lite clean. We rename it to iot-taxonomy-lite13 to account for new subclasses. We use iot-taxonomy prefix for it.
- iot-taxonomy:QualityOfObservation: defines the quality of a sensor determines and thus the quality of the observations taken by the sensor. The quality may be low due to calibration issues. While, once resolved, the quality can change to high. Hence, we define iot-taxonomy:Poor, iot-taxonomy:Fair, and iot-taxonomy:Good as the subclasses of iot-taxonomy:QualityOfObservation. Although QUDT14 ontology presents another way of such representation, we refrain from using it currently.
- We use sosa:Actuator to define an Actuating device. Previously, it was defined using iot-lite:ActuatingDevice. We also

5SSN-Systems: http://www.w3.org/ns/ssn/ssn-systems
6SOSA: http://www.w3.org/ns/sosa/
7M3-lite: http://purl.org/iot/vocab/m3-lite#
8QU: http://purl.org/NET/ssnx/quat# 
9Geo: http://www.w3.org/2003/01/geo/wgs84_pos# 
10Geo-sparql: http://www.ogc.org/ont/geo-sparql
11Schema.org: https://schema.org
12DUL: http://www.loa.istc.cnr.it/ontologies/DUL/owl#
13iot-taxonomy-lite: http://purl.org/iot/vocab/iot-taxonomy-lite#
14QUDT: http://qudt.org/1.1/schema/qudt#
add `sosa:Actuation` and `sosa:ActuatableProperty` to account for the measurement made using `sosa:Actuators` and the phenomena they act on, respectively.

- **sosa:FeatureOfInterest**: defines “the thing whose property is being estimated.” For example, `sosa:FeatureOfInterest` can be a room and a building. For now, we add only a few subclasses of `sosa:FeatureOfInterest` and make them available as a part of `IoT-Taxonomy`. A `sosa:FeatureOfInterest` is also associated with `iot-lite:Service`.

- **ssn-system:SystemProperty** is “the observable characteristic that represents the System’s ability to operate its primary purpose.” For `ssn-system:SystemProperty`, we reuse its `Accuracy`, `Frequency`, `Latency`, `Precision`, `Resolution`, and `Response time` subclasses. We associate these subclasses with three different data properties: (i) `dul:hasDataValue` - used for those system properties that do not have a range of values, (ii) `schema:minValue`, and (iii) `schema:maxValue` - used for those system properties that do have a range of values. We associate the above-mentioned data properties with the range `xsd:int` and `xsd:double`.

We include privacy-related concepts in our ontology to address the requirement of privacy and call this new subgraph a **privacy graph** (see Figure 2). For the privacy graph, we again follow the seven-step methodology [5]. Here we identify the scope of the ontology asking competency questions:

- What resources a user has permission to discover?
- Who has permission to discover resources and observations?
- Who provides the consent for access?
- Who owns a particular resource or observation?
- What purpose a user has in discovering a type of data?
- Who controls the data collection process?

Given such competency questions, we identify the need to include consent [12] and provenance [13] concepts. Therefore, we reuse concepts from `Consent`15 (we use prefix `con`) and `GDPRtEXT`16 (we use prefix `gdprtext`) ontologies. Nonetheless, these ontologies do not fulfill all the requirements and are not standardized. Upon exhaustive search, we did not find ontologies that we could reuse to fulfill all of our requirements (cf. Section III). Thus, we had to define several object properties. We use the prefix `priv` to denote our properties. We model the privacy graph using two subgraphs: `consentGraph` and `userPermissionsGraph`. The `userPermissionsGraph` stores the user permissions to discover either resources or observations along with the permissible actions and their purpose. While, the `consentGraph` contains information on who gave the permissions, who is the owner of the resources/observations, and who controls the data (usually a testbed owner). These privacy-related subgraphs are depicted in Figure 2. The concepts related to `consentGraph` are marked with a brown dot, while those in the `userPermissionsGraph` are marked with a blue dot. Next, we describe these concepts and properties in detail.

- **The privacy graph** focuses on the discovery of `ssn:System`, `iot-lite:Service`, `sosa:Observation`, and `sosa:Actuation`.

---

15 Consent: http://purl.org/adaptcentre/openscience/ontologies/consent#
16 GDPRtEXT: https://w3id.org/GDPRtEXT#
ject property con:permission_given_for_activity has domain con:Permission and range con:Action to help identify the set of activities that are granted to a con:AllowedParty (addressing Privacy Req#4, #7, and Req#8) and for what purpose (con:Purpose), addressing Privacy Req#2 and #3. Each con:Action has a purpose. con:activity_has_purpose has domain con:Action and range con:Purpose to help provide such information. A con:Purpose can be either free text (xsd:string) or from a predefined set (for example: iot-taxonomy:KnowSensorsInTheArea).

- The consentGraph contains information about who can access resources and observations, addressing Privacy Req#1 and #6. The root of consentGraph is the con:ConsentingParty, the person who has given the consent (con:Consent) and also owns (priv:owns) the resources and observations. priv:owns has an inverse property called priv:ownedBy. The con:ConsentingParty can priv:owns multiple resources and observations while a resource or an observation is priv:ownedBy by exactly one con:Consenting-Party. con:given_consent gives con:ConsentingParty (domain) the con:Consent (range) and has the cardinality is ∈ W. Users can monitor the quality of their data using the accuracy entity, addressing Privacy Req#5.

- gdprtext:Controller is the “legal person, public authority, agency or other body which, alone or jointly with others, determines the purpose and means of the processing of personal data.” The administrator of the testbed controls the testbed and the related data.

V. BEST PRACTICES FOLLOWED AND RECOMMENDATIONS TO TRIPLE-STORE CREATORS AND DATA PROVIDERS

We list the best practices we followed to build our ontology and recommendations to testbeds and triple-stores that use it.

A. Best practice followed to create the ontology

Our ontology follows the methodology described in [5]. For IoT systems, our ontology further is engineered to follow recommendations described in [21]. Other than those that we have defined, most of the concepts and properties are taken from ontologies that already follow the best practices and are available in LOV [22]. Further:

- Several formats such as RDF/XML, JSON-LD, Turtle, and Ntriples of our ontology are published. Each concept and property fulfils a set of metadata that includes rdfs:labels and rdfs:comments. We maintain the online accessibility and the permanent URL of the ontology. The ontology web documentation is generated using WIDOCO [23]. It reflects detailed documentation of the concepts, properties, how-to, and related material.

- We aim to standardise the ontology. As a step towards standardisation, the ontology is listed in LOV.

- Testbeds/data providers can request an update in our ontology or taxonomy facilitating integration in the federated triple-store. They should use our GitHub issue tracker. For completeness, each concept or property that needs to be added should consist of a name (rdfs:label), a clear description (rdfs:comment), and the parent (if any). Further, for properties, additionally, should consist of the domain, the range, and the cardinality. If the concept or the property is already defined in some other ontology, rdfs:isDefinedBy or rdfs:seeAlso should be provided.

B. Recommendations to federating triple-stores and testbed/data providers

The interactions of the consentGraph with the resource and the observation graph is proportional to the number of resources and observations. This increases communication and storage overheads.

- Currently, one cannot store information regarding how the permissions are granted or denied using our ontology. Our focus is only ‘who’ are granted permissions. The triple-store should provide the functionality to update the consentGraph and the userPermissionGraph with relevant information when a new resource or new observation is added. Further, the functionality should also enable the update of the userPermissionGraph when a new permission is granted. The relevant information should be deleted from the graph when a permission is denied. Also, our ontology does not store any information about how a user gets registered or any other information about the user. It only contains the URIs of users that are con:AllowedParty or on:ConsentingParty. A triple-stores should build its own user management functionality.

- The privacy graph should have restricted access. No user except the triple-store admin should be allowed to query it.

- A testbed should respect the inverse properties and cardinalities of the object properties. If the inverse property exists, they should provide triples to support both links. Moreover, if a testbed wants to publish triples related to some feature not supported currently by the ontology, they can use iot-lite:Metadata. However, this should not be abused.

- IoT resources should first be inserted into the resource graph. Then privacy-related information should be updated. Then only testbeds should publish observations in the observation graph. If a resource or sos:ObservableProperty is not granted any permission, then they are only accessible to the con:ConsentingParty and the gdprtext:Controller.

VI. WORKFLOW, ANNOTATIONS, AND QUERYING

The workflow for annotation and access to resources and observations would primarily depend on the system design, but as a guideline, we propose the following approach. At the initial stage, the data provider should register resources it wants to make available for experimentation and populate the resource graph. Post this process, the data provider should create instances for gdprtext:Controller and con:ConsentingParty using the registration module, unless this is set by default by a prior data provider. User access to any resource or observations belonging to the data provider...

17 LOV: https://lov.linkeddata.es/dataset/lov/
18 FIESTA-Priv: http://purl.org/iot/ontology/fiesta-priv#19 FIESTA-Priv GitHub: https://github.com/ragarwa2/fiesta-priv
present in the triple-store is denied by default, unless a policy is pre-set. The policy would primarily be based on the con:Purpose and con:Action, due to their significance with GDPR compliance. Assume, con:AllowedParty are users that are registered. The data provider creates a con:Permission that would link all the current and any future con:AllowedParty with a resource if the policy requirements are met. This type of process can be viewed as “passive” consent. If a policy is not pre-set, then a user (con:AllowedParty) requests for access to resources or observations triggering a request mechanism for an “active” consent usually within a set period before expiration. This would involve the platform notifying all the con:ConsentingPartys. Whenever a con:ConsentingParty grants access to their resources or observations, a new con:Consent and con:Permission is instantiated and the usersPermissionGraph is updated accordingly.

Once data is available in a triple-store, users/experimenters can write SPARQL queries depending on their needs using the SPARQL endpoint provided by the triple-store. As the users/experimenters only need to access the resource and observation graphs, they should only write queries that target either the resource, the observation graph or both. Consider an experimenter who has registered his interest for discovering available sensors (iot-taxonomy:DiscoverSensors) in a given location for knowledge purpose (iot-taxonomy:KnowSensorsIntheArea) with the platform and the platform has stored this information. The userPermissionsGraph is only updated after a particular interest/request of the user is granted permission. The user then queries to get data associated with all the sensors available in a region bounded between -50 and +50 degrees longitude and between 0 and 100 degrees latitude. The triple-store then augments the query internally to include the privacy-related concepts depending on the registered interest of the user. On the other hand, the triple-store can also first execute the user query and then filter the results based on the privacy graph (a time-consuming alternative). The ontology documentation also provides a SPARQL query for knowing all resources a particular user can discover in a particular region.

VII. Ontology Validation

We perform a qualitative evaluation of the ontology, validating its conformance (example: lexical, syntactic, and semantic aspects) [24] and quality (example: accuracy, completeness, conciseness, adaptability, clarity, computational efficiency, and consistency) [25]. We also comment on the reusability and modularity aspects. This work does not provide any usage-specific evaluation as we do not focus on building a platform.

For structural aspects, we validate our ontology using tools such as Ontology Pitfall Scanner20 (OOPS), TripleChecker21, and Vapour22. Among the above tools, OOPS also provides us with an evaluation of completeness, conciseness, clarity, and consistency. An OOPS report on the validation of our ontology is available on the ontology document page. The tool identifies that our ontology is concise. For completeness, it reports that our ontology has unconnected elements. This is true in the case of iot-lite:Metadata for which domain is deliberately kept open. It further reports that some properties are missing domain and range. This is irrelevant as we have used schema:domainIncludes and schema:rangeIncludes instead of domain and range. These annotations are not checked by the tool. The tool also reports some inverse relations other than those defined. However, these reported inverse relations are not valid. Thereby making our ontology consistent. The only clarity issue raised is due to the different naming conventions in the ontology because Consent ontology uses a different naming convention. Other than OOPS report, we also provide the TripleChecker23 and Vapour24 reports.

To complete the validation of our ontology, we execute the Hermit reasoner with the hope to identify any entailments that we did not intend. We apply reasoning to the ontology structure, the relationships among classes and among properties. As there are no instances, we do not reason instances. For our ontology, other than direct relations, the reasoner inferred results mainly using rules such as:

- (rdfs11) if A ⊆ B and B ⊆ C then A ⊆ C
- if A ⊆ B and B ⊆ C then A ⊆ C
- if A ⊆ B and B ⊆ C- then A is a ⊆ C-
- if A ⊆ B- and B ⊆ C then A ⊆ C-

Note that here we only show the rules for classes. Similar results were obtained for objectProperties as well. Nonetheless, we notice no issues25. The reasoner also detected that our ontology is ‘satisfiable.’ For completeness, we also executed Pallet Reasoner and found similar results.

Our ontology is partially modular. It reuses most of the concepts from the existing ontologies. We defined the concepts and properties within the ontology because we did not find related concepts in the existing ontologies.

According to the ontology metrics, our ontology has 3133 Axioms, 645 Logical Axioms, 608 Declaration Axioms, 520 Classes, 46 Object Properties, 13 Data Properties, 31 Annotation Properties, 10 Inverse Object Properties, and 3 Equivalent properties. Among these, we only defined 10 object properties.

VIII. Use Cases

We now demonstrate different use-cases for our ontology.

A. Selective Privacy for Participants

Testbeds and Living Labs are often closely associated with people. Data captured is often multi-variate because IoT devices associated with a testbed usually monitor multiple phenomena simultaneously. A testbed may comprise devices installed in a room that measure temperature, humidity, sound, and light. Participants might not feel comfortable sharing sound level data or device state (i.e. lights or air-conditioning)

20OOPS: http://oops.linkeddata.es
21TripleChecker: http://graphite.ecs.soton.ac.uk/checker/
22Vapour: http://linkeddata.uriburner.com:8000/vapour
23FIESTA-Priv TripleChecker: https://tinyurl.com/qwbnj87
24FIESTA-Priv Vapour: https://tinyurl.com/75fbv5
25The results of Hermit reasoner are available at http://smart-ics.ee.surrey.ac.uk/ontology/fiesta-priv/HermitResults.owl
around them since they might reveal user presence. Thus, the associated testbed must take the user’s acceptance and ultimately consent, into serious consideration and restrict access to these data on the level mandated by the participant. This could be based on user roles, affiliation, or locality. In this case, the `con:ConsentingParty` would not give `con:Consent`, and thus the `con:Permission` to the `sosa:ObservableProperty` would be denied. Based on this, the testbed provider in `priv:control` as the `gdprtext:Controller`, will enforce denial of access to consumers who are not an `con:AllowedParty`.

In cases where devices are carried by humans or are mobile, such as mobile crowdsourcing, an `con:Actions` relating to discovering sensors or observations can be restricted based on spatial and/or temporal rules (restricted to certain locations and times of the day). Consider a living lab deployed in a workplace where sensory data is captured from an employee’s smartphone managed by an employer. The employee might not want his employer to know what he does or where he goes outside of working hours. Here the employee (`con:ConsentingParty`) can restrict the discovery of `sosa:ObservableProperties` (`con:Action` being `iot-taxonomy:GetWorkplaceObservations`) captured during work hours only to the employer (`con:AllowedParty`).

B. Selective Data Sharing for Providers

IoT-enabled platforms have evolved from collecting data from homogeneous to heterogeneous sources. This involves integrating systems from multiple devices where data from associated devices are sent to the testbed, stored, analysed, and monitored. Here, a data exchange point can be established by having a central or federated repository to pull data from or a data broker where experimenters can subscribe to data of their interest. Some providers/testbeds might want to restrict exposing their data to experimenters based on the purpose (`con:Purpose`) of data consumption, information governance compliance, or explicitly declared interest submitted by the experimenter, and those identified as direct competitors. Data providers (`con:ConsentingParty`) can only allow `con:AllowedParty` for the data access. From the consumer perspective, users with different roles would have access to the data. For example, in the context of healthcare trial systems, users can be clinical monitoring staff, general practitioners, society support members, technicians or researchers. End users such as patients or carers participating in a trial can restrict which roles are allowed access to their data.

C. Applicability in IoT Platforms and Existing IoT Ontologies

Besides testbeds/data providers, our ontology finds applicability on IoT platforms that support federation, semantic interoperability, experimentation, and other existing well-known IoT ontologies. The FIESTA-IoT ontology [18] has been successfully integrated with IoT Cloud Platforms such as FIESTA-IoT [3] and used to federate at least 11 testbeds [26] supporting more than 23 experiments. These testbeds and experiments have different application domains (smart cities, crowdsensing, smart agriculture, smart building, smart energy, and smart sea) and were selected based on their usefulness, complementarity, sustainability, and competence. With such a success and data released to experimenters, compliance with GDPR is must. Thus, integrating our ontology with the FIESTA-IoT Platform would provide an edge to FIESTA-IoT. The same applies to other IoT Platforms such as Wise-IoT [27] and ACTIVAGE [28] that adopt the same ontology.

Several IoT ontologies that lack privacy concepts can adopt the privacy graph as a whole or the object properties defined therein to make themselves privacy enabled. As an example, well-known ontologies those miss privacy concepts such as SSN, IoT-lite, OneM2M, and SAREF [26] including its extensions can leverage from our defined privacy graph.

IX. DISCUSSION AND CONCLUSION

In a hyper-connected world with devices continuously monitoring and assisting person’s everyday life, keeping life private is not easy. IoT systems should be fully compliant with legal requirements and built under the concept of privacy by default [1]. In both real-world implementation and experimentation, a well-defined ontology providing the required privacy-enhancing features to protect the users’ data is mandatory.

Having this in mind, we enhance existing well-defined ontologies (such as in [18]), adding the necessary components to address the main requirements of GDPR mapped to the specifics of an IoT system. Note that the proposed ontology does not claim to address all the GDPR requirements, especially not the ones related to data processing, retention, or deletion, but instead focus on the privacy requirements for data gathering and sharing. The ontology addresses the GDPR requirements (cf. Section III) by adding a new privacy subgraph, which includes key concepts such as consent, permission, data subject, purpose, and links these concepts with IoT services and data controllers. The main goal of this graph is to ensure that any well-defined request/purpose for data gets the data subject’s permission (if the data are sensitive) who has full control and view over who has access to his data.

Fig. 3: Circles represent different ontologies used. Connections between different ontologies use object properties from stated ontologies. Privacy-related concepts are shown in red.

26SAREF: http://saref.linkeddata.es
We expect that such an ontology will be highly adopted by IoT experimentation projects and IoT services that aim to protect user privacy. Since this is a high-level view of a privacy-enhanced IoT ontology, in the future, we would like to include concepts related to conditions or context (such as in [29]) in which a particular observation is captured. Further, we would like to add concepts that address more fine-grained scenarios that might not be fully captured by this version.

ACKNOWLEDGMENT

All authors have contributed equally. Authors would like to thank FIESTA-IoT project members and the anonymous reviewers for their feedback.

REFERENCES

[1] Council of the European Union and European Parliament, “Regulation (EU) 2016/679 of the European Parliament and of the Council of 27 April 2016 on the protection of natural persons with regard to the processing of personal data and on the free movement of such data, and repealing Directive 95/46/EC (GDPR),” 2016.

[2] M. Marjani et al., “Big IoT Data Analytics: Architecture, Opportunities, and Open Research Challenges,” IEEE Access, vol. 5, 2017.

[3] J. Lanza et al., “Experimentation as a service over semantically interoperable internet of things testbeds,” IEEE Access, vol. 6, pp. 51607–51625, September 2018.

[4] ISO/IEC 29100:2011 Information technology – Security techniques -Privacy framework,” 2011.

[5] N. Noy and D. McGuinness, “Ontology Development 101: A Guide to Creating Your First Ontology,” tech. rep., Knowledge Systems Laboratory, Stanford, 2001.

[6] G. Bajaj, R. Agarwal, P. Singh, N. Georgantas, and V. Issarny, “4W1H in IoT Semantics,” IEEE Access, vol. 6, pp. 65488–65506, October 2018.

[7] G. Bajaj, R. Agarwal, P. Singh, N. Georgantas, and V. Issarny, “A study of existing Ontologies in the IoT-domain,” 2017.

[8] M. Abomhara and G. Katsi, “Security and privacy in the Internet of Things: Current status and open issues,” in Inter. Conf. on Privacy and Security in Mobile Systems, pp. 1–8, IEEE, May 2014.

[9] M. Imran-Daud, D. Sánchez, and A. Viejo, “Ontology-based access control management: Two use cases,” in 8th Inter. Conf. on Agents and Artificial Intelligence, pp. 244–249, SciTePress, February 2016.

[10] S. Kirrane, A. Mileo, and S. Decker, “Access control and the resource description framework: A survey,” Semantic Web, vol. 8, no. 2, 2017.

[11] S. Mouliswaran, C. Kumar, and C. Chandrasekar, “Inter-domain role based access control using ontology,” in Inter. Conf. on Advances in Computing, Communications and Informatics, IEEE, 2015.

[12] K. Fatema, E. Hadziselimovic, H. Pandit, C. Debruyne, D. Lewis, and D. O’ Sullivan, “Compliance through Informed Consent: Semantic Based Consent Permission and Data Management Model,” in PrivOnt@ISWC, CEUR-WS, October 2017.

[13] H. Pandit and D. Lewis, “Modelling Provenance for GDPR Compliance using Linked Open Data Vocabularies,” in PrivOnt@ISWC, CEUR-WS, October 2017.

[14] M. Arruda and R. Bulcão Neto, “Toward a Lightweight Ontology for Privacy Protection in IoT,” in 34th Symposium on Applied Computing, SAC’19, pp. 880–888, ACM, 2019.

[15] F. Loukil, C. Ghedira-Guegan, K. Boukadi, and A. Benharkat, “LIoPY: A legal compliant ontology to preserve privacy for the Internet of Things,” in 42nd Annual Computer Software and Applications Conference, vol. 2, pp. 701–706, IEEE, 2018.

[16] M. Palmirani, M. Martoni, A. Rossi, C. Bartolini, and L. Robaldo, “Pronto: Privacy ontology for legal reasoning,” in Electronic Government and the Information Systems Perspective (A. Kó and E. Francesconi, eds.), pp. 139–152, Springer, 2018.

[17] A. Bassi, M. Bauer, M. Fiedler, and R. V. Kranenburg, Enabling things to talk. Springer Berlin Heidelberg, 2013.

[18] R. Agarwal et al., “Unified Ontology to Enable Interoperability and Federation of Testbeds,” in 3rd World Forum on Internet of Things, pp. 70–75, IEEE, December 2016.

[19] J. Cuellar (eds.), “Deliverable D2.2: System Requirements and Smart Objects Model,” EU-FP7-Smartcities-HERUM consortium, 2014.

[20] R. Pohls, H. Staudemeyer (eds.), “Deliverable D3.2: Privacy enhancing techniques in the Smart City applications,” tech. rep., EU-FP7-Smartcities-HERUM consortium, 2015.

[21] A. Gyrard, M. Serrano, and G. Atemezing, “Semantic web methodologies, best practices and ontology engineering applied to Internet of Things,” in 2nd World Forum on Internet of Things, pp. 412–417, IEEE, December 2015.

[22] P. Vandenbussche, G. Atemezing, M. Poveda-Villalón, and B. Vatant, “Linked open vocabularies (lov): a gateway to reusable semantic vocabularies on the web,” Semantic Web, vol. 8, no. 3, pp. 437–452, 2017.

[23] D. Garjio, “WIDOCO: A wizard for documenting ontologies,” in ISWC, pp. 94–102, Springer, 2017.

[24] S. Datta, C. Bonnet, H. Baga, M. Zhao, and F. Le Gall, “Approach for Semantic Interoperability Testing in Internet of Things,” in Global Internet of Things Summit, pp. 1–6, IEEE, June 2018.

[25] J. Raad and C. Cruz, “A survey on ontology evaluation methods,” in 7th Inter. Joint Conf. on Knowledge Discovery, Knowledge Engineering and Knowledge Management, pp. 179–186, SciTePress, 2015.

[26] L. Sanchez et al., “Federation of internet of things testbeds for the realization of a semantically-enabled multi-domain data marketplace,” Sensors, vol. 18, pp. 1–32, October 2018.

[27] E. Kovacs, M. Bauer, J. Kim, J. Yun, F. Le Gall, and M. Zhao, “Standards-based worldwide Semantic Interoperability for IoT,” IEEE Communications Magazine, vol. 54, no. 12, pp. 40–46, 2016.

[28] C. Karabir, G. Dafoulas, A. Ballis, and O. Raptis, “ACTIV AGE 2018: A survey on ontology evaluation methods,” in 7th Inter. Joint Conf. on Knowledge Discovery, Knowledge Engineering and Knowledge Management, pp. 179–186, SciTePress, 2015.

[29] R. Agarwal, S. Chopra, V. Christophides, N. Georgantas, and V. Issarny, “Detecting Mobile Crowdsensing Context in the Wild,” 29th Inter. Conf. on Mobile Data Management, pp. 170–175, IEEE, June 2019.