Semantic Knowledge and Privacy in the Physical Web

Prajit Kumar Das, Abhay Kashyap, Gurpreet Singh,
Cynthia Matuszek, Tim Finin and Anupam Joshi

University of Maryland, Baltimore County, Baltimore, Maryland, USA
prajit1@umbc.edu, abhay1@umbc.edu, gurpree1@umbc.edu,
cmat@umbc.edu, finin@umbc.edu, joshi@umbc.edu

Abstract. In the past few years, the Internet of Things has started to become a reality; however, its growth has been hampered by privacy and security concerns. One promising approach is to use Semantic Web technologies to mitigate privacy concerns in an informed, flexible way. We present CARLTON, a framework for managing data privacy for entities in a Physical Web deployment using Semantic Web technologies. CARLTON uses context-sensitive privacy policies to protect privacy of organizational and personnel data. We provide use case scenarios where natural language queries for data are handled by the system, and show how privacy policies may be used to manage data privacy in such scenarios, based on an ontology of concepts that can be used as rule antecedents in customizable privacy policies.

Keywords: physical web, internet of things, privacy, context sensitive policies

1 Introduction

In the past few years, the Internet of Things (IoT) has started to become a reality. Advances like affordable Bluetooth Low Energy devices, the Physical Web protocol, and ubiquitous devices such as smart-phones and smart-watches make it possible for people to communicate with devices and places in the world around them usefully and intuitively. Unfortunately, the lack of good privacy and security solutions for ubiquitous sensors and interconnected devices is a continuing concern. Existing approaches to managing privacy tend to assume a closed system, in which known users with known needs can be matched to predefined policies; the open, mobile nature of the IoT requires more semantically informed solutions. It is a system where devices need to advertise and discover others in their vicinity, interoperate with them in a given context, and publish and honor their privacy and security constraints.

The UN specialized agency responsible for issues concerning information and communication technologies have defined the Internet of Things as a global infrastructure for the information society, enabling advanced services by interconnecting (physical and virtual) things based on existing and evolving interoperable information and communication technologies [18]. The Physical Web, a concept first introduced by Google, is a concretization of IoT. As per Google's definition, "The Physical Web is an open approach to enable quick and seamless interactions with physical objects and locations." The Physical Web works by broadcasting small packets of data or URLs using the Bluetooth Low Energy (BLE) protocol and beacons.

Over the past few years, we have observed a rise in usage of IoT devices in various domains. However, the IoT domain has also been criticized for privacy and security issues that plague these devices. One major cause for concern stems from a lack of interoperability among manufacturers and the devices they create. Most IoT systems today are built in a bottom-up manner—that is, deployed

sensors talk to custom and proprietary gateways, which in turn expose proprietary services; at the highest level manufacturers provide intelligent applications based on data from those services. This vertical architecture, which is controlled by manufacturers, hampers horizontal interoperability [6] across manufacturers. Meanwhile, a typical IoT deployment consists of a heterogeneous collection of devices. As a result, it becomes difficult to create data privacy and security solutions.

Semantic Web technology uses the general Web architecture, e.g., URIs to access data, with such access potentially controlled based on rules defined in a knowledge base (KB). Semantic Web has been used to define access control rules in various domains, e.g., mobile applications, smart meeting rooms, RDF triple stores, Social Media, etc. [22,2,8,11,4,27,3] In this paper, we present the CARLTON framework, which allows us to manage data privacy for entities in a Physical Web deployment using Semantic Web technologies. We also present part of the CARLTON ontology (see Figure 2), which allows policies to be defined based on entity relationships.

Our proof-of-concept study involves a deployment at our organization, where we study a set of use cases pertaining to queries about persons, places and events associated with the organization. Our goal is to protect the privacy of organizational and personnel data. We examine cases where a user's data privacy is managed by their personal privacy policies. We also study scenarios which handle data privacy in the absence of a personal privacy policy, or in case of queries about places and events. Privacy policies in our framework are context-sensitive; as such, they are dependent on *user context*, where a user is either (1): a person whose data is represented in the system; or (2): a person who is querying the system. In the following sections we will describe the CARLTON system and some initial scenarios, followed by a discussion of ways in which a similar architecture could support more sophisticated privacy reasoning.

2 Related Work

Role Based Access Control (RBAC) [9] and Attribute Based Access Control (ABAC) [23] are two most popular access control models, that have been used to achieve the goal of managing access control, in various domains. In the mobile domain, Ghosh et. al. [11] used a semantically rich context model to manage data flow among applications and filter them at a deeper granularity than it was possible using available security mechanisms on smart-phones. The CR$\hat{e}$PE system [4] was one of the earliest known ABAC model implementations using the XACML standard [12] for fine-grained context-related policy enforcement on smart-phones. CR$\hat{e}$PE didn't use Semantic Web but it followed the ABAC model. CARLTON applies techniques learned in mobile domain to IoT domain and uses context-sensitive policies to define access to organizational and personnel data. Our access control model is an ABAC model where the attributes defining data access are 'requesting user' context, 'requested entity' context and a query.

Using policy based security is not a new technique. Kagal et. al. [22] used distributed policy management as an alternative to traditional authentication and access control schemes. Rei, a policy language described in OWL and modeled on deontic concepts of permissions, prohibitions, obligations and dispensations [20,22], have used Semantic Web technologies to express what an entity can/cannot do and what it should/should not do. In Rei, credentials and entity properties like user, agent, etc are associated with access privileges. This allowed Rei to describe a large variety of policies ranging from security policies to conversation and behavior policies. The Rein framework [21] which builds on Rei and is based on N3 rules and uses a CWM reasoning engine for distributed reasoning. In CARLTON we define data sharing policies that determines behavior of the Physical Web-Mobile Agent interaction.

KAoS [28] relies on a DAML description-logic-based ontology of the computational environment, application context, and the policies. The KAoS system was capable of supporting runtime policy changes and was extensible to a variety of platforms. In ROWLBAC [10], the Web Ontology Language (OWL) [1] was used to support the standard RBAC model and extending OWL constructs used to model ABAC. All of these systems are using some Semantic Web technology for their implementation. In our work, we use Semantic Web technology through an ontology to define a hierarchical context model for a user and a requester an rules are defined using the Semantic Web Rule Language [17] to determine access control decisions. In short, we are achieving the goal of access control in a different domain with techniques that have been proven to work in other domains. We do modify the system design, if and when required, to suite our needs for the IoT domain and particular use cases we handle,

The hierarchical notion of context defined in this paper is an extension of our previous work [19,29] where the part_of relationship was defined for stating that a location is subsumed by another bigger location. We have extended this notion to Identity context as explained in Subsection 3.2.

A model for context-sensitive policy based access control for IoT was presented in our previous work [5], where we proposed a system design that achieves such a goal for a generic IoT environment. We capture access control policies using the ABAC model represented in OWL. We used a vehicular IoT use case for describing our policies in that work. CARLTON in contrast doesn't yet have a well-defined mechanism for capturing policy modification, as defined in the other system. However, CARLTON handles far more complex use cases with respect to the contextual granularity of queries made to the system.

Finally, we take a look at indirect vs obvious privacy implications. Ma et. al. [25] showed in their study how a relatively small amount of 'side' information can lead to significant privacy concerns. We try to address such issues with the current best available solution of writing a few general rules to make our system usable. A detailed discussion on this can be found in the Section 4.1.

3 System Overview

The prototype CARLTON application targeted in this work is a pervasive information system that uses beacons (low powered, battery efficient devices that broadcast content over Bluetooth), fixed-position kiosks, mobile devices, and a NLP engine to respond to natural language queries. Beacons are deployed in key locations in the department, like offices, classrooms, and labs. The application is intended to provide 'Help Desk'-style information about an academic department, allowing people to ask questions and get spoken responses, sometimes augmented with an appropriate display.

One of our goals is to design a system that could be installed by other university departments and customized by providing a database of key information about its people and places. Both kiosks and mobile devices use beacons to know their locations, using the Nearby Messages API [16]. This maximizes portability and enables use of user location context when interpreting and responding to questions, which allows responses to be physically informed. Table 1 shows a few examples of supported question types, including possible privacy concerns; Section 5 gives more examples of the kinds of questions CARLTON is intended to address. Further details about context are inferred using an ontology and external sources of data like a user's calendar if it is shared with the system.

The CARLTON system architecture is shown in Figure 1. The system is invoked by a user by asking queries as either natural language text or speech through our kiosk or our CARLTON Android app. Spoken language is converted to text using a SpeechRecognizer service in the Android app, that is part of Android SDK[14] or using the Cloud Speech API [15] in the kiosk. Text, whether

	User	Query	Target	Location	Additional Context	Response
1	Faculty member	"Is this room booked from 2PM-3PM?"	Conference room	ITE conference room	User identity; room calendar	"No; shall I book it for 2PM-3PM?"
2	Student					"No. See the front desk for room reservations."
3	Staff member	"Is Dr. Joshi here?"	Dr. A Joshi	Main department office	User identity; target disambiguation; target location	"The chairman is not in the office right now."
4	Student in class		Dr. K Joshi	Hall outside faculty office		"No. Would you like to hear her office hours?"
5	Visitor	"Where is Dr. Tim Finin's office?"	Dr. T Finin	ITE building third floor	Building office map; target disambiguation	"Please see the front desk in ITE 325 for directions."
6				Admin building	Target disambiguation	"Please see the publicly available campus directory or the information desk."
7	Student				User identity; campus directory; target disambiguation	"His office is in the ITE building."

Table 1: Several examples of queries handled by CARLTON, showing the different forms of context that can affect the answer. *Column 1:* the CARLTON user. A "visitor" is a user who has provided no identifiable information. *Column 2:* the user's query. *Column 3:* the system's determination of the query target, based on context. *Column 4:* the position of the user, as detected by proximity to BLE beacons. *Column 5:* additional contextual information, beyond location, that helps determine the query response and allowable levels of information. *Column 6:* CARLTON's response. Target disambiguation, privacy management, and access control are all managed by applying policy rules to information in the knowledge base.

from the speech to text system or entered directly, is then processed using the Stanford CoreNLP suite of tools [26] to do POS tagging, parsing and identify entities and relations. A simple system takes this output and attempts to map it into one of the requests, questions or assertions that our system can handle. The responses from our system are always in text form but if the query was made using speech through the CARLTON Android app, we use Android SDK's TextToSpeech service [13] to speak the response back to the user.

Context is retrieved from an ontology as well as from external sources such as location. Users have the option of identifying themselves; users who lack credentials, or who do not identify themselves, will potentially receive more limited data, depending on privacy policies. A reasoning engine uses inferred context, the user's speech act and defined privacy policy as input, to reason over, and infer the granularity at which data or descriptions will be shared in the response. After the response is generated, a natural language query response is generated and presented, either as text or speech. Table 1 shows some examples of queries, context, and responses.

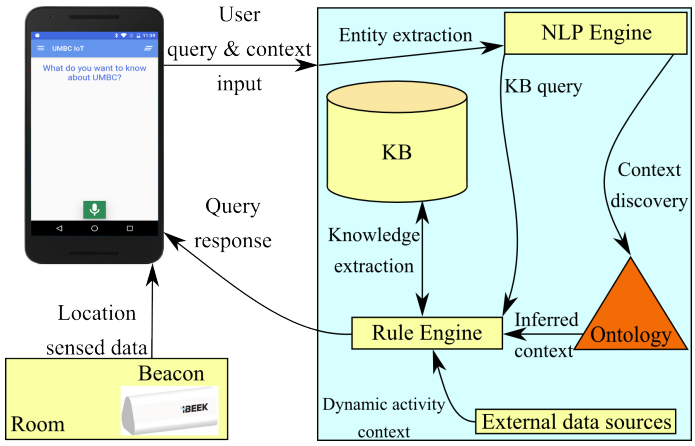

Fig. 1: System architecture for CARLTON. The system is invoked by natural language queries from a user (either text or speech) and mapped into one of the requests, questions or assertions that the system can handle. Context is retrieved from an ontology as well as from external sources such as location. After results are retrieved and checked against existing rules, a natural language query response is presented.

3.1 Rules

CARLTON uses context-sensitive privacy policy rules defined in the Semantic Web Rule Language (SWRL) [17] to manage the privacy of data. The abstract syntax for SWRL rules follow the Extended Backus-Naur Form (EBNF) notation which, while useful for XML and RDF serializations, isn't particularly easy to read. For readability, we use the following informal format: `antecedent` $\Rightarrow$ `consequent`. Antecedent(s) must hold for a consequent to apply. Multiple antecedents in a rule are defined as a conjunctions of atoms. The consequent atom states whether the access is allowed or denied.

Antecedents in our rule specification consist of the *context of a requesting entity*, along with the *entity type* that is being requested, and may also include context for the entity *whose data is requested*.

A more abstract representation may be considered as a triple (U, C, Q) which contains: U, that represents requested user's context, that is the user whose data is represented in the system. C is requesting user's context, that is the user who is querying the system. Q is the query that is received by the system. The following is an example policy rule with instantiated values, based on our ontology, which demonstrates how information is controlled by our system. Imagine a scenario where Professor Xavier shares information about his present precise location with a person, if they are member of his lab and he supervises them, given he is not speaking at a talk at the moment. The first 8 predicates in this policy rule represent the requester student's context. The next 8 relate to the requested party's context. The query is about location request and response from the system is to share the location.

Example 1.

```
@prefix crltn:<https://www.ebiquity.org/ontologies/carlton/0.1>.
@prefix swrlb:<http://www.w3.org/2003/11/swrlb>.
crltn:student(?requester) ∧
( crltn:supervises("Xavier",?requester) ∨
  (crltn:affiliatedWith(?requester,?labName) ∧
  crltn:leads("Xavier",?labName)) ) ∧
crltn:hasCurrentLocation(?requester,?aBldgLocation) ∧
crltn:room(?aBldgLocation) ∧
```

```
crltn:sitsIn("Xavier",aBldgLocation) ∧
crltn:currentTime(?currTime) ∧
swrlb:Exists(?anEvent) ∧
crltn:speakingAt("Xavier",?anEvent) ∧
( (crltn:startTime(?anEvent,?eventStartTime) ∧
  swrlb:greaterThan(?eventStartTime,?currTime)) ∨
  (crltn:endTime(?anEvent,?eventEndTime) ∧
  swrlb:greaterThan(?currTime,?eventEndTime)) ) ∧
crltn:hasCurrentLocation("Xavier",?aLocation) ∧
crltn:Location(?aLocation) ∧
crltn:requestLocation("Xavier")
⟹
shareLocation(?aLocation)
```

On the other hand, if a student is simply affiliated with the same department as he is, Professor Xavier might share a generic location using the privacy policy rule shown below.

Example 2.

```
@prefix crltn:<https://www.ebiquity.org/ontologies/carlton/0.1>.
@prefix swrlb:<http://www.w3.org/2003/11/swrlb>.
crltn:student(?requester) ∧
crltn:affiliatedWith(?requester,?deptName) ∧
crltn:affiliatedWith("Xavier",?deptName) ∧
crltn:hasCurrentLocation("Xavier",?aLocation) ∧
crltn:Location(?aLocation) ∧
crltn:partOf(?aLocation,?city) ∧
crltn:City(?city) ∧
crltn:requestLocation("Xavier")
⟹
shareLocation(?city)
```

3.2 User Context Specification

Context has been defined by Dey and Abowd [7] as:

> "[...] any information that can be used to characterize the situation of an entity (i.e., identity, location, activity, time). An entity is a person, place, object or events that is considered relevant to the interaction between a user and application, including the user and applications themselves."

In our system, user context is considered from a perspective of both who is querying the system and who the query is about. Both of these user contexts include location, identity and activity information. For the use cases discussed in this paper we have not considered temporal context. We use the Physical Web to sense user location using beacons. This information is then sent to a back-end knowledge base that uses a context ontology and an inference engine to determine user location context. The *identity context* is defined by voluntary user sign-in and authentication. The *activity context* is linked to the identity of a user in our system, as activity context is derived from the user's calendar data. This provides both static and dynamic context to evaluate against data privacy rules.

An interesting feature of our system is that we wish to protect the privacy of the user querying the system, as well as the information being queried. Identification does have privacy repercussions, and therefore we have designed our system to be capable of working without identifying a user. When context information is not available, or if only generic context information is available (e.g., 'the user is a student'), we evaluate data privacy rules accordingly by using rules with more generic contextual antecedents. We are able to do this using our ontology, which allows location and activity generalization by using `partOf` relationships among location entities and activity hierarchies. The generalization technique described here is an extension of our work in data privacy in the mobile domain [19,29,11]. We use the `owl:sameAs` property to incorporate certain classes and properties from the Platys [19] and Place [29] ontologies. These ontologies have been previously used for defining user location and activity context in a hierarchical manner.

This technique helps us preserve privacy by sharing data with varying levels of granularity, as specified by policies. It involves replacing a value with a less specific but semantically consistent value. As an example of location generalization, a user might define a policy stating that "My building level location can be shared with my colleagues," which allows partial location sharing with anyone who is identified as a 'colleague', based on rules written in our system. In this case, if an exact room location of a user is known, the ontology allows us to generalize to a building using the "Part Of" transitive property. In our ontology `Location` is a super class of the `Room`, `Building`, `City`, `State` and `Country` classes. The various sub-classes are used to denote different levels of abstractions for the location.

Activity generalization is similar. Consider a case where we have a hierarchy of work activities:

$$\texttt{working} \xleftarrow{\texttt{partOf}} \texttt{meeting} \xleftarrow{\texttt{partOf}} \texttt{department meeting} \xleftarrow{\texttt{partOf}} \texttt{lab meeting}$$

This hierarchy lets a user define policies like "Share with team members when I am in a team meeting" and "Share with team non-members when I am in a meeting." This is especially useful if we consider that a team meeting might be accessible only to team members, as it allows obfuscating the data to just a "meeting" when the party querying the data is allowed some access but not access to confidential team data.

We look at identity based generalization in this paper as an extension of the generalization technique. Our ontology contains a `Person` class with a property `affiliatedWith` that defines if a requester is "affiliated with" a university, department or research group. If the user is not affiliated with any instances of these classes that match the affiliation of an entity whose data was requested, we generalize the response accordingly. This allows us to define rules like "Share with non-members of the organization only my city level location info".

3.3 Semantic Knowledge

The CARLTON ontology defines entity relationships between persons, locations and events in the organizational structure. We are able to make inferences of the form described in Section 5 based on the entity relationship properties defined in the ontology. See Figure 2 for details of the ontology. The four primary classes defined in our ontology include `Person`, `Location`, `Organization` and `Event`. A `Person` in our ontology might be a `Faculty`, `Staff`, `Student` or `Visitor` to the organization. `Organization` is further broken into `University`, `Department`, `ResearchGroup` classes. `Event` have the sub-classes `Talk`, `Meeting` and `Course`. For the `Location` class `partOf` relationship allows hierarchical location context definition. The same applies to the `Organization` sub-classes. The `Person` class allows us to infer affiliation with an `Organization`. A `Visitor` is a `Person` in our

ontology and might have a affiliation too, with an `Organization` but the instance of a visitor's `Organization` would not match `Organization` information for `Faculty`, `Staff` or `Student` from a particular `University` that the `Visitor` does not belong to. We are able to infer `Location` of an `Event` through the relationship `heldAt` between `Event` and `Room` classes. Obviously a `Talk` will be held in a `ConferenceRoom`, a `Course` will be taught in a `Classroom` etc. where the rooms are types of the `Room` class and therefore using the `heldAt` relationship we are able to infer triples that state such a relationship. Similar inference maybe drawn for affiliation of a `Faculty`, `Staff` or `Student` with a `Department` or `ResearchGroup`.

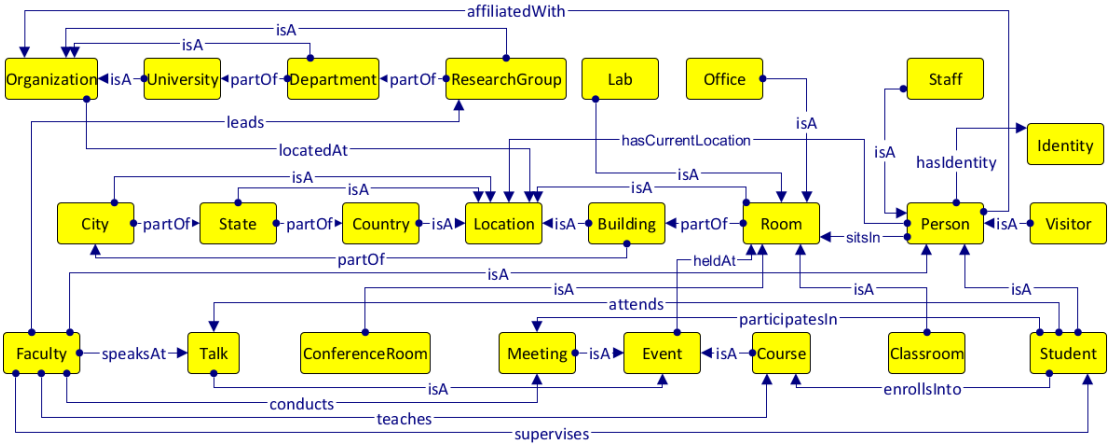

Fig. 2: Part of the CARLTON prototype's ontology, showing classes and properties for some events, places, people and relations appropriate for an academic environment. The ontology will be expanded and integrated with other LOD vocabularies in the future.

4 Privacy Implications

The system described in this paper takes steps towards fulfilling the promise of the Internet of Things: it allows easy access to appropriately contextualized knowledge, making it possible for users to find appropriate information quickly and easily while reducing administrative overhead. Consistent with that, however, it also showcases some of the privacy concerns associated with precisely those capabilities. In this section, we look at some of the ways in which privacy comes into play in such a system, and discuss how using a semantically informed knowledge framework allows for informed, appropriate behavior.

4.1 Privacy Stakeholders

One consideration when discussing privacy concerns is the correct identification of *stakeholders,* individuals or groups whose privacy may be compromised. Some stakeholders are easily identified, while others may be more subtle. The same is true of the possible impact of different kinds of information access, that is, privacy failures may range from obvious to indirect. We consider three possible stakeholders and, for each, discuss selected privacy concerns.

The most obvious stakeholder group is *query targets:* individuals about whom queries are posed. Information about query targets may be very direct, as in location, or more general, such as working at a particular research lab. In addition, this information varies in consistency across time: location presumably changes frequently, whereas membership in the `Faculty` group is usually permanent. Broadly speaking, query targets can be assumed to have some (but not complete) control over their own information; as an example, the department may make information available about professors' office hours, but not otherwise provide information from calendars, while an individual may choose to make their meeting times available to their own students.

Less obviously, target privacy may also apply to entire organizations or subgroups. While an academic department may be relatively public, some types of information may still be protected. As an example, what research groups exist within the department is likely to be public information, but the physical location of labs engaged in controversial research may not be. Similarly, a research group may or may not choose to make their group meeting time public, depending on member preference and research topic.

The final stakeholder whose privacy must be considered is the CARLTON user. The pattern of a user's queries may reveal unwarranted information—for example, a professor might look askance on a student continually asking about lab meeting times, even if the student is punctual; if a faculty member consistently asks about the chair's location and then manages to "just miss" him or her, active avoidance might be deduced. Even a user who does not identify him or herself may be de-anonymizable after relatively few queries. [25] In most cases, this may be addressed as treating queries as strictly transient unless the user engages in an action that makes changes to the world (e.g., reserving a conference room); however, this disallows logging of the sort that is commonly used for debugging and system optimization.

4.2 Semantically Informed Access and Risks

Having a formal, semantic representation of the underpinnings of information to be accessed and actions to be taken allows for the formal representation of policies as rules. CARLTON combines information about users, contexts, and query targets as antecedents to rules that allow users to express complex sets of constraints on information that can be shared. Information can be controlled based on the user or the query target (as shown in Example 2), the user's location (as shown in Table 1), or more generally—for example, the department might write default rules that prohibit the sharing of information about peoples' locations. Such a rule should include an antecedent specifying that it applies only if there is no more permissive target-specific rule.

Semantic hierarchies also make it possible to provide "fall-upward" information. Example 1 shows a case in which only the `City` of the query target is explicitly provided, rather than the specific location. However, because rules can be expressed with arbitrary complexity, such rules can be specified implicitly in both antecedent and consequent. A generalized policy might, for example, say "For members of the academic community, give a summary of my daily schedule; otherwise, provide no information." Such a policy would chain with more general rules about affiliation and summarization; this is explored further in Section 5 below.

One notable category of privacy failures in a rule-based system is combinatorial failures, that is, failures in which multiple pieces of allowed information lead to a conclusion that is not intended to be public. This is a concern for overlapping rules with varying privacy implications. As an example, Professor Grey does not allow anyone to see her location or calendar. Dr. Xavier does not allow anyone to see his location, but allows his students to see his calendar, meaning that they can find

out that he is in a meeting with Dr. Grey and Dr. Hank McCoy. Dr. McCoy does not allow access to his calendar, but makes his location available to anyone. With a small number of queries, Dr. Xavier's students can find the location and current calendar event of all three professors.

This is a difficult problem to address. Determining whether any combination of rules will allow for unintended access would be excessively restrictive. More importantly, it is isomorphic to a complete truth maintenance problem, and not tractable for a knowledge base of reasonable size. While it is possible to write additional rules that explicitly restrict information access in addition to explicitly exposing it, this solution is not logically complete and is, additionally, prone to logical conflicts unless rules occur in a strict precedence ordering. The current best available solution is to write relatively few, fairly general rules, and consider the output in different cases carefully; while unsatisfying, this is currently the best case for achieving a usable system.

5 Use Case Scenarios

We are developing and testing the CARLTON prototype in an academic Computer Science department. A number of beacons have been deployed in our University's Information Technology and Engineering (ITE) building's front office suite, offices of faculty members, labs and conference rooms. CARLTON is capable of handling a variety of natural language queries, e.g., "Who is Foggy Nelson?", "Where is Ben Ulrich's office?", "What does Matt Murdock teach?", etc. Since we have installed our system in an academic organization, we look at use cases that make sense in such a setting. We will look at use cases in which personal data privacy policies have been set up by Professor Jean Grey.

Use Case 1: Share specific information with people who are NEARBY and HAVE A RELATIONSHIP with the query target. Prof. Grey prefers certain private information to be shared with people she knows who are near her office. She shares her private calendar data with students from a course she is teaching and with people she advises. This may be defined using the rule: "Share my calendar information with people I teach or supervise if they are near my office." Such a query would have relevance in case a student is standing in front of her office and wants to know when she will be available for a meeting.

Use Case 2: Share only summary information with people who are FAR AWAY and HAVE A RELATIONSHIP with the query target. Prof. Grey does not want her students to skip lab meetings irrespective of whether she is busy in another meeting or not. Therefore, she sets up a policy stating "Share only summary information from my calendar with people I supervise if they are far away from my office." According to this policy her students will only get to know that Prof. Grey is available today and has some meetings unless they are already in the building.

Use Case 3: Share only publicly available data with UNKNOWN PEOPLE. CARLTON allows Prof. Grey to block anyone who is not part of her group or school from getting any information from her calendar or any other source that is private to her group or school. This is a special use case which not only takes care of data privacy for the professor but it also takes care of privacy implications from a requester's perspective. We explain this further in Section 6.

Use Case 4: Share information about organization's entities with people who are NEARBY and if they BELONG TO the group or organization. Now we look at privacy of a physical or virtual entity of the organization. The entity in this scenario maybe a meeting room, a lab or an event. Prof.

Grey has a lab called PhoenixForce. PhoenixForce owns a meeting room denoted as ITE-1, two labs named ITE-4 and ITE-5, and organizes an event called FallWelcomeBack. Only researchers from her lab are allowed to book the meeting room, get access to the lab, or join the event. These require that the requester has information about when the meeting room is available, whether lab has empty seats, and where the event is. CARLTON is thus able to use privacy policies like "Share meeting room availability with members of PhoenixForce," "Share lab empty seat count for ITE-4 if user is near ITE-4 and AND a member of PhoenixForce," and finally "Share event location with members of PhoenixForce if they are on campus."

Use Case 5: Share public information about an organization's entities with UNKNOWN PEOPLE. In this use case CARLTON unaware of a requester's identity, and therefore shares only publicly available event information, public lab info and public meeting room data—for example, that PhoenixForce does Robotics research and meeting room ITE325 is used for dissertation defenses.

These use cases are intended to showcase how data privacy for an individual or organization's entities are managed using Physical Web context discovery and by using privacy policies defined in Semantic Web technologies.

6 Trusting the Physical Web

Although our query client includes a login option, it is not necessary for a user to log in in order to query our system. One may observe that our use cases include scenarios where a user's identity is unknown. In these cases we do share less information but this acts as a feature of our system. If a user logs into our app then they have authenticated themselves to our system and we may authorize access to entities as per the privacy policies that have been defined. However, this means that the system constantly knows who the user is and where they go and what they ask etc. This may be considered as a violation of privacy of the requester.

In case of the physical web we may be able to determine where a particular user is at all times, given enough number of beacons are used to ensure organization-wide coverage. We have thus created a privacy issue for users requesting data from our system, as a side effect of actually trying to manage data privacy for individuals and entities of the organization. How do we address this issue? We have already alluded to a potential restriction on our system in Use Cases 3 and 5 that will allow us to reduce some of the privacy concerns. By not making the login step mandatory, we have reduced these privacy concerns. Queries are still responded to in this case but we generalize our response based on the affiliation information that we might have.

Is it possible to authenticate a user without knowing the actual identity of the user? Yes, we have implemented Authentication without Identification using Zero-Knowledge Proof as described by [24]. This allows us to respond to user with a higher level of data granularity. For example all group related data maybe exposed to a person who has been authenticated to be part of the group. It's worth noting that we will not be able to respond to queries where identity context match has to be an exact match. For example if Prof. Grey queries the system about "When is my meeting with Prof. Xavier scheduled?".

The CARLTON mobile app does not require any permission other than:

- `android.permission.INTERNET`
- `android.permission.ACCESS_FINE_LOCATION`
- `android.permission.BLUETOOTH`

These permissions are required to talk to the beacons over the BLE protocol and to obtain beacon data and call back-end server for query responses over the internet. The location access is required by the Nearby Messages API [16] which provides us with all the data associated with a beacon. No identity based permissions are required by our app. Neither do we take device id information. The Nearby Messages API doesn't require user authentication for sharing data. As a result, the system is designed to not share any identity information with the Google servers, in order to address privacy concerns with regard to explicit associations being made between user identity and user location by Google.

7 Conclusions and Future Work

In this paper, we have presented CARLTON, a framework for managing data privacy for entities in a Physical Web deployment using Semantic Web technologies. Our system used context-sensitive privacy policies to protect privacy of organizational and personnel data. We have provided few use case scenarios where natural language queries for data are handled by CARLTON. We have explained how privacy policies may be used to manage data privacy in such scenarios, based on an ontology of concepts that can be used as rule antecedents in customizable privacy policies.

Although we have presented a few use cases in this paper, the number of scenarios are small relative to fully functional, in-real-life system. The purpose of use cases that we have presented, was to demonstrate the utility of CARLTON. However, many more scenarios are possible and we intend to explore them all eventually. A challenge that we constantly faced during this project was managing the beacons when they are in close proximity. This problem was solved in a trivial manner by reducing the signal strengths of the beacons. However, this still doesn't handle conflict resolution in presence of multiple beacons in close-proximity. Handling, such issues could be a challenging goal to pursue, in the future.

8 Acknowledgment

We gratefully acknowledge the support of the Google Internet of Things (IoT) Technology Research Award pilot, which provided the hardware used for this research.

References

1. Bechhofer, S.: Owl: Web ontology language. In: Encyclopedia of Database Systems, pp. 2008–2009. Springer (2009)
2. Chen, H., Finin, T., Joshi, A., Kagal, L., Perich, F., Chakraborty, D.: Intelligent agents meet the semantic web in smart spaces. IEEE Internet Computing 8(6), 69–79 (2004)
3. Cheng, Y., Park, J., Sandhu, R.: A user-to-user relationship-based access control model for online social networks. In: IFIP Annual Conference on Data and Applications Security and Privacy. pp. 8–24. Springer (2012)
4. Conti, M., Crispo, B., Fernandes, E., Zhauniarovich, Y.: Crêpe: A system for enforcing fine-grained context-related policies on android. IEEE Transactions on Information Forensics and Security 7(5), 1426–1438 (2012)
5. Das, P.K., Narayanan, S., Sharma, N.K., Joshi, A., Joshi, K., Finin, T.: Context-sensitive policy based security in internet of things. In: 2016 IEEE International Conference on Smart Computing (SMARTCOMP) (2016)

6. Desai, P., Sheth, A., Anantharam, P.: Semantic gateway as a service architecture for iot interoperability. In: 2015 IEEE International Conference on Mobile Services. pp. 313–319 (June 2015)
7. Dey, A.K., Abowd, G.D.: Towards a better understanding of context and context-awareness. In: First Int. symposium on Handheld and Ubiquitous Computing (HUC) (1999)
8. Dietzold, S., Auer, S.: Access control on rdf triple stores from a semantic wiki perspective. In: ESWC Workshop on Scripting for the Semantic Web. Citeseer (2006)
9. Ferraiolo, D.F., Kuhn, D.R.: Role-based access controls. arXiv preprint arXiv:0903.2171 (2009)
10. Finin, T., Joshi, A., Kagal, L., Niu, J., Sandhu, R., Winsborough, W., Thuraisingham, B.: Rowlbac: Representing role based access control in owl. In: Proceedings of the 13th ACM Symposium on Access Control Models and Technologies. pp. 73–82. SACMAT '08, ACM, New York, NY, USA (2008), http://doi.acm.org/10.1145/1377836.1377849
11. Ghosh, D., Joshi, A., Finin, T., Jagtap, P.: Privacy control in smart phones using semantically rich reasoning and context modeling. In: Security and Privacy Workshops (SPW), 2012 IEEE Symposium on. pp. 82–85. IEEE (2012)
12. Godik, S., Anderson, A., Parducci, B., Humenn, P., Vajjhala, S.: Oasis extensible access control 2 markup language (xacml) 3. Tech. rep., Tech. rep., OASIS (2002)
13. Google: Text to speech (September 2009), https://developer.android.com/reference/android/speech/tts/TextToSpeech.html
14. Google: Speech recognizer (May 2010), https://developer.android.com/reference/android/speech/SpeechRecognizer.html
15. Google: Cloud speech api (May 2016), https://cloud.google.com/speech/
16. Google: Google nearby messages api (May 2016), https://developers.google.com/nearby/messages/overview
17. Horrocks, I., Patel-Schneider, P.F., Boley, H., Tabet, S., Grosof, B., Dean, M.: SWRL: A semantic web rule language combining OWL and RuleML. W3c member submission, World Wide Web Consortium (2004)
18. ITU-T: Series y: Global information infrastructure, internet protocol aspects and next-generation networks. ITU-T Recommendation Y (2012)
19. Jagtap, P., Joshi, A., Finin, T., Zavala, L.: Preserving privacy in context-aware systems. In: Semantic Computing (ICSC), 2011 Fifth IEEE International Conference on. pp. 149–153. IEEE (2011)
20. Kagal, L., Finin, T., Joshi, A.: A policy language for a pervasive computing environment. In: Policies for Distributed Systems and Networks, 2003. Proceedings. POLICY 2003. IEEE 4th International Workshop on. pp. 63–74 (June 2003)
21. Kagal, L., Berners-Lee, T.: Rein: Where policies meet rules in the semantic web. Computer Science and Artificial ... (2005), http://groups.csail.mit.edu/dig/2005/05/rein/rein-paper.pdf
22. Kagal, L., Finin, T., Joshi, A.: A policy based approach to security for the semantic web. In: International Semantic Web Conference. pp. 402–418. Springer (2003)
23. Kuhn, D.R., Coyne, E.J., Weil, T.R.: Adding attributes to role-based access control. Computer 43(6), 79–81 (June 2010)
24. Lysyanskaya, A.: Authentication without identification. IEEE Security and Privacy 5(3), 69–71 (May 2007), http://dx.doi.org/10.1109/MSP.2007.52
25. Ma, C.Y., Yau, D.K., Yip, N.K., Rao, N.S.: Privacy vulnerability of published anonymous mobility traces. IEEE/ACM Transactions on Networking 21(3), 720–733 (2013)
26. Manning, C.D., Surdeanu, M., Bauer, J., Finkel, J.R., Bethard, S., McClosky, D.: The stanford corenlp natural language processing toolkit. In: ACL (System Demonstrations). pp. 55–60 (2014)
27. Sun, L., Wang, H., Yong, J., Wu, G.: Semantic access control for cloud computing based on e-healthcare. In: Computer Supported Cooperative Work in Design (CSCWD), 2012 IEEE 16th International Conference on. pp. 512–518. IEEE (2012)
28. Uszok, A., Bradshaw, J.M., Jeffers, R.: KAoS: A Policy and Domain Services Framework for Grid Computing and Semantic Web Services. Trust Management –Lecture Notes in Computer Science 2995/2004, 16–26 (2004)

29. Zavala, L., Dharurkar, R., Jagtap, P., Finin, T., Joshi, A.: Mobile, collaborative, context-aware systems. In: Proc. AAAI Workshop on Activity Context Representation: Techniques and Languages, AAAI. AAAI Press (2011)