# A semantic security framework and context-aware role-based access control ontology for Smart Spaces

Shohreh Hosseinzadeh and Seppo Virtanen
Department of Information Technology
University of Turku, Finland
{shohos, seppo.virtanen}@utu.fi

Natalia Díaz-Rodríguez and Johan Lilius
Department of Information Technologies
Åbo Akademi University, Turku, Finland
ndiaz@decsai.ugr.es, jolilius@abo.fi

## ABSTRACT

Smart Spaces are composed of heterogeneous sensors and devices that collect and share information. This information may contain personal information of the users. Thus, securing the data and preserving the privacy are of paramount importance. In this paper, we propose techniques for information security and privacy protection for Smart Spaces based on the Smart-M3 platform. We propose a) a security framework, and b) a context-aware role-based access control scheme. We model our access control scheme using ontological techniques and Web Ontology Language (OWL), and implement it via CLIPS rules. To evaluate the efficiency of our access control scheme, we measure the time it takes to check the access rights of the access requests. The results demonstrate that the highest response time is approximately 0.2 seconds in a set of 100000 triples. We conclude that the proposed access control scheme produces low overhead and is therefore, an efficient approach for Smart Spaces.

## CCS Concepts

•**Security and privacy** → **Access control;**

## Keywords

Smart Space, Smart-M3, computer security, privacy, e-Health, semantic web

## 1. INTRODUCTION

Smart Spaces (SS) are types of Wireless Sensor Networks (WSN), that are composed of wearable and embedded sensors and devices that co-operate with each other in order to make people's lives smarter and more comfortable. They also have shown to be successful in efficient energy consumption. The use of Smart Space technology has drawn a lot of attention in health and well-being, for instance for health monitoring and remote rehabilitation [9]. Information sharing makes the cooperation of the devices feasible, which on the other hand, raises privacy concerns. Hence, the collected data from the user/environment needs to be protected.

We have developed a Smart-M3-based home gateway that stores and processes the information gathered from the sensors, in order to automate the user's actions. An example of the services the gateway can provide is the virtual remote rehabilitation monitoring with the help of depth sensors such as Kinect [9]. The Smart-M3[1] is a Multi-device, Multi-domain, and Multi-vendor platform that was originally developed at Nokia Research Center in 2009. It is an inter-operable approach based on principles of the Semantic Web (SW) [6]. Smart-M3 platform is composed of two types of components: *Semantic Information Brokers (SIB)* and *Knowledge Processors (KP)*. SIB acts as the back bone to the space and shares the information between KPs. KPs are agents in the Smart Space including devices, sensors, people, and other data providers that can affect, produce and consume the information on the space [21]. The communication between SIB and KPs is implemented with Smart Space Access protocol (SSAP). It allows the KP to join and leave, query, write and delete data, subscribe and unsubscribe to the space [15]. There are several benefits known for Smart-M3 that motivate projects to take advantages of it. The most remarkable feature is the publish/subscribe paradigm implemented on top of the RDF store. The SW uses Resource Description Framework (RDF) to store and represent data in a machine understandable manner [12]. This feature enables the subscribers to be notified automatically any time a change happens to the subscribed resource.

In this paper we provide solutions to protect the information and privacy of the users in Smart Spaces based on the Smart-M3 platform. The proposed solutions include a) a security framework and b) an access control scheme. The security framework is composed of various components collaborating together to support different aspects of security, such as authentication, authorization and access control. The proposed access control scheme is Context-Aware Role Based Access Control (CARBAC). It controls the access of the users to the system in accordance to their role in the system and the current context information. We model our access control scheme via ontological modeling techniques (ontologies are known to be one of the best ways to present knowledge), and implement it via C Language Integrated Production System (CLIPS) rules (CLIPS is a tool based on C language used for developing expert systems). In the end, we evaluate the overhead that the proposed access con-

[1]Smart-M3 Project: http://sourceforge.net/projects/smart-m3/



*SBD'16, July 01 2016, San Francisco, CA, USA*



DOI: http://dx.doi.org/10.1145/2928294.2928300

trol introduces to the system. The remainder of the paper is structured as follows. Section 2 studies the related work. Section 3 introduces the proposed security framework and discusses the proposed access control scheme in detail. In Section 4 we evaluate the proposed approach and analyze the results. Concluding remarks are given in section 5.

## 2. RELATED WORK

In this section, we survey some of the existing security frameworks, access control schemes and ontologies that are proposed in the field, and elaborate the novelty of our scheme.

Consec [3] is another context-aware security framework developed for SS. It is discussed that the context information may carry sensitive data of users, so it is essential to ensure the legitimacy of both the context provider (sensor) and the context receiver (application). To this end, both parties authenticate themselves to each other, using Kerberos Protocol [17]. The integrity of the communication is guaranteed using hash functions and digital signature, and the confidentiality of the communication is guaranteed via symmetric key cryptography. The context-aware access control proposed for Smart-M3 in [13] is a combination of Role-base Access Control (RBAC) and Attribute-based Access Control (ABAC). It is ABAC, since the context information (the state of an entity or an activity) is used to give a trust level to users, and assign roles to users based on the trust values. It is RBAC, since access permissions are assigned based on the given roles. There are access control models that are proposed for different domains, which are not suitable for dynamic environments such as Smart Spaces. The access control model proposed in [11] is meant for web services and may not be applicable to an SS. The roles are activated by the access control system based on the context information. Users and context information do not influence the role selection. Another access control model proposed for web services is [24], in which environment roles and the pre-defined roles are taken into account for decision making. Unlike any of the previously discussed related approaches, our access control model is context aware, role-based, designed for SS, and preserves the privacy of the users through their preferences. We model our access control scheme using ontological modeling techniques; they are known as successful techniques in representing knowledge and useful for automatic data reasoning and inferencing. To this aim, we studied the existing access control ontologies proposed for context-aware, dynamic and distributed systems. By comparing them, we concluded that none of these ontologies tackles all our security requirements. Thus, we propose an access control ontology (Section 3.1). Table 1 summarizes some of the existing access control ontologies and compares them with ours. The approaches are compared with respect to: a) the access control model they present, b) context-awareness, c) support for the rules, d) the domains they are proposed for, e) privacy protection support, and f) access control at triple level. According to Table 1, we can claim that our ontology is unique by being context-aware, rule-based, proposed for Smart Spaces and being able to control the users' privacy and access at triple level.

In [22] resources have a set of context conditions. A requester by fulfilling those conditions can access the resource. In [14] patients give permission for using their information, and in accordance to these permissions (preferences), the access control rules are defined. CoBrA [7] is a set of ontologies that build a context-aware system together. Privacy protection ontology is one of the ontologies in CoBrA. The privacy of the users is protected via the rules defined by them. In [1] various ontologies are designed to sketch different security related aspects (credentials, security mechanisms, privacy). Access to the resources is controlled by pre-defined privacy policies. OPO [20] is a lightweight vocabulary generated according to the users' privacy preferences to control access over RDF data (any user has its own Access Control List (ACL) that defines the access privileges to the data). A requester, in order to gain access to an object, is required to satisfy a set of attributes that specify who is granted access. In [16] the proposed access control model is based on users' behavior, context information, and historical data (the pattern of past actions of the user). The behavior of the users is tracked to uniquely identify them to provide more customizable services. In [4] all users are issued a set of credentials by a central issuer. When the user requests to access a resource, it receives a set of policies. The policies specify the requirements for gaining the access. In this model, the privacy is supported via anonymous credentials (accessing a resource without revealing their identity). ROWLBACK [10] studies the relation between OWL and RBAC. OWL has successfully been used for expressing the authorized policies. Our work has been inspired by them, to model our CARBAC via OWL. SitBAC knowledge framework [19] is a Situation Based Access Control model that is proposed to control the access over Electronic Health Records (EHRs). In this model, the access is controlled according to circumstances that match a pre-defined pattern. The access control policies are specified in accordance with the possible situations. Privacy in this system is protected via access control scheme. Proteus [23] is a context-centric policy model based on semantic technologies. In this approach instead of associating the access control rules with the subjects, they are associated with the context (i.e., the information that describes the situation of an element). An entity is able to perform an action, if the current context of the environment matches the required context of the requester.

Our access control model is a Context Aware Role Based Access Control (CARBAC) scheme. In this model, roles are assigned to the users by the administrator when they register in the system. At run-time, the security rules are executed to grant/deny access (based on the user's role and the context information). Privacy is protected via privacy rules. Our modeled ontology aggregates all the aspects that were missing in the existing access control ontologies, by managing the security and privacy issues related to SS.

## 3. A SEMANTIC SECURITY FRAMEWORK

In a Smart Space with the main focus on health care and well-being, it is highly significant to protect the privacy and confidentiality of patients' medical and personal data from unauthorized access while stored or transmitted. It is even more crucial and difficult to administrate the information and physical security in ubiquitous environments with numbers of participants continuously joining and leaving the space. Currently, most of the security proposals for the SS are limited and coarse-grained, that is they control the access to the whole data store, for example in the existing RDF repositories (such as Sesame and Jena). However, we need to have fine-grained access control at triple level. Although fine-grained access controlling comes with costs and

Table 1: Comparison of access control ontologies and their Smart Space (SS) domains

| Reference & Access control model | Context aware | Rule-based | Domain | Privacy control | Triple level control |
|---|---|---|---|---|---|
| [22] Context-based | ✓ | ✓ | Pervasive Computing Environments | × | × |
| [14] Privacy-centric | × | ✓ | Heterogeneous administrative medical domains | ✓ | ✓ |
| [7] CoBrA, No access control ontologies | ✓ | ✓ | Context-aware systems and SS | ✓ | × |
| OWL-S Services[1] Ontology-based | × | ✓ | Semantic Web services | ✓ | × |
| [20] OPO Access Control List (ACL) | × | ✓ | Linked Data | ✓ | ✓ |
| [16] User Behavior and Capability Based Control Access | ✓ | ✓ | Smart Spaces | ✓ | × |
| [4] Credential-based | ✓ | ✓ | XACML and SAML-based systems | ✓ | × |
| [19] SitBAC | ✓ | ✓ | Smart Spaces | ✓ | × |
| [23] Proteus, Context-centric | ✓ | ✓ | Pervasive Environments | × | × |
| [10] ROWLBACK | × | ✓ | Dynamic Environments | × | × |
| This work: CARBAC | ✓ | ✓ | Health and well-being SS | ✓ | ✓ |

overhead (more computation and higher response time), it is acceptable when dealing with highly sensitive information.

We propose a security framework that supports the following security objectives: authentication, authorization and access control. The proposed security framework (Figure 1) comes on top of an RDF repository to check authenticity, authority, and access right of the requester before accessing the data repository. Our framework is inspired by some existing design principals [2]. The framework works as follows: when a request comes (1), first the authentication engine in (2) assures that the requester is authentic. If positive, the request is forwarded (3) to Access Control Engine in (4). There, the access control rules are executed, and it is checked whether the requester has the right to perform the requested action. The access log (6) keeps a record of the recent accesses. Finally in (8), the result (accessible triples for the requester) is retrieved from the repository (7) and sent to the user.

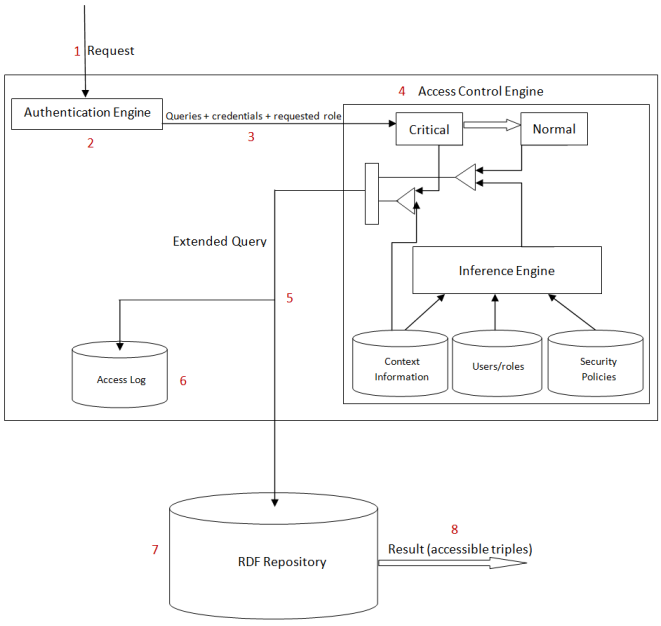


**Figure 1: Security framework architecture**

## 3.1 Access control scheme

There are different mechanisms used to control the access over resources. Among all Role Based Access Control (RBAC) is known as the most suitable one for health care information systems. The proposed access control scheme is a Context-Aware Role Based Access Control (CARBAC), which is fairly similar to [13] in the way that both are combinations of RBAC and ABAC. What makes them different is that in [13], the roles are conceded to individuals dynamically in accordance with the context; however, in our system, roles are allocated statically to the users at the time of registering in the system by the central administrator, and assigned dynamically at the time of logging into the system. For more flexibility, users may have several roles, one of which is activated for them when logging in the system. The proposed access control scheme is called Context-Aware RBAC, because it has the characteristics of RBAC, and is context aware. It is RBAC, since users get permissions/prohibitions for accessing a resource, according to their roles in the system. It is context aware, since in three different points context information affects the access control. These point are: a) when the context information defines that the situation is critical, b) when the context information affects what role is assigned to a user, and c) when the context information affects how the rules are executed. The effect of context information on access control is discussed in 4.2. in detail. The other attribute that differentiates our proposed scheme from traditional RBAC is that, for privacy protection purposes, users of the system have the authority to specify access control rules according to their preferences.

## 3.2 Access control ontology and rules

To represent and classify knowledge, there are two different approaches: data-driven and knowledge-driven. Data-driven methods rely on machine learning and statistical approaches. Although they have shown to be accurate in many domains, they are not appropriate in dynamically changing environments, such as smart Spaces. For such environments, knowledge-driven techniques are preferably used; for instance, rule-based systems and ontological approaches [8].

In terms of modeling context information, several models are available: object oriented, graphical, logic based, key-value and ontology-based models [5]. In the proposed architecture, the ontology based model is chosen due to its

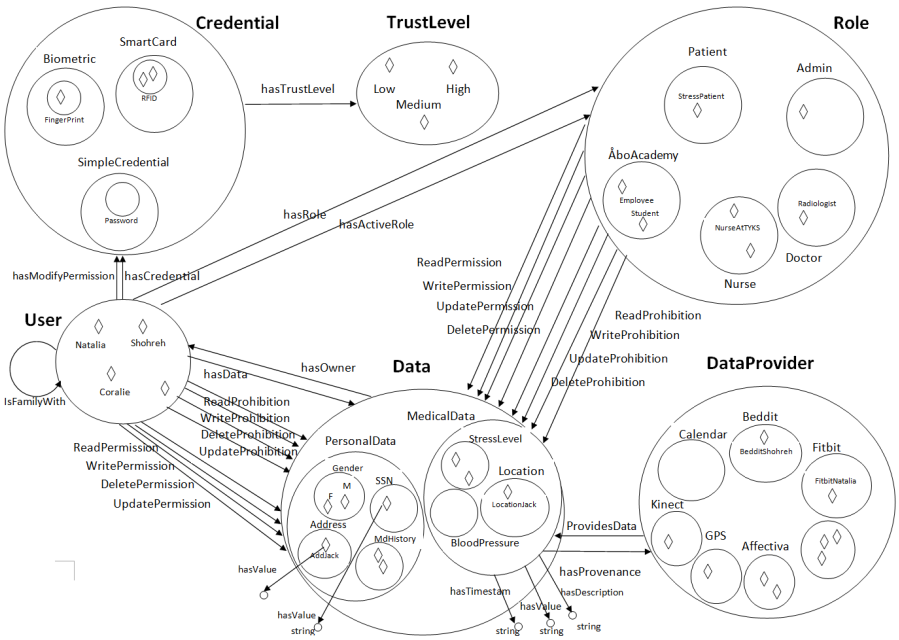


**Figure 2: Access control ontology. Users choose credentials to login to the system. Then, they activate a role to get rights for accessing data. The data is provided by data providers that belong to the users.**

advantages: a) it is flexible, expressive and generic; b) it is the most favorable method to model context information; c) we can take advantage of ontology reasoning and automatic code generation [18].

Figure 2 depicts the access control ontology proposed in this paper. Users choose credentials (with different trust levels) to authenticate themselves to the system. Users in the system can have several roles, but can activate one role at a time. By activating roles, users get rights to access the data. The rights are permissions/prohibitions to perform the four actions: read, update, delete, and query data. As discussed, in order to support the privacy of the users, they are able to define some access rights to have more restrictions on their own data. In Figure 2, there are relations between the classes, *User* and *Data* to define if a subject (User) is permitted/prohibited to perform action on a specific object (Data). These relations are similar to the relations between the roles and the data entities. The data is provided by data providers (captured by the devices/sensors), or users (their personal data).

Our OWL ontology was modelled with Protégé 4.2 editor and is available online[2]. An ontology is composed of various components, including *Classes*, *Object Properties*, *Data Properties* and *Individuals*. Individuals (represented by diamonds in Protégé) are the objects we are interested in. Classes (represented by circles) are the abstract concept used to group individuals of certain type. Individuals are connected to each other via properties, which are classified as object properties mapping two individuals together, and data properties mapping an individual to a data value. Several characteristics are defined for properties; The following are examples of various types of properties in our ontology: a) Functional Property: there is only one single individual corresponding to the given property. For instance, in *(LocationShohreh hasDataProvider GPSShohreh), hasDataProvider* is a functional property because there is only one value (an individual, *GPSShohreh*) to be related to the individual *LocationShohreh*.
b) Transitive Property: If individual I1 is associated with individual I2 and I2 is associated with individual I3, we can deduce that I1 and I3 are linked via property P. For instance: If *(Shohreh hasRole StressPatient)* and *(StressPatient hasReadPermission stressLevel-Shohreh)* then we can deduce that *(Shohreh hasReadPermission stressLevelShohreh)*.
c) Symmetric Property: If individual I1 is linked to individual I2 via a symmetric property, then we can claim that I2 is linked to I1 via the same property. For example, if *(Coralie isFamilyWith Shohreh)*, then *(Shohreh isFamilyWith Coralie)*.
d) Anti-Symmetric Property: If individual I1 is related to individual I2 via anti-symmetric property P, then we cannot say that I2 is linked to I1 via the same property P. For example, if *(Natalia hasCredential fingerprintNatalia)*, then we cannot say that *(fingerprintNatalia hasCredential Natalia)*.
e) Reflexive Property: A reflexive property links an individual I to itself. As an example, in the triple *(Shohreh knows Shohreh)*, the property P = *knows* is a reflexive property where I = *Shohreh*.

In our system, the access control policies are expressed via rules. At run-time, the rules are executed in order to decide whether an agent is permitted or prohibited to perform an action. For writing the access control rules, we used CLIPS version 6.24. CLIPS itself is developed in C language, and has been used in improvement of expert system technology. Our access control rules include two sets of rules: the rules designed by the administrator, and the rules defined by the user for privacy protection purposes. Users may determine particular individuals to be permitted/prohibited to access their data. The user defined rules have higher priority than the admin rules. For instance, the user *Jack* with the role of

[2]https://github.com/NataliaDiaz/AccessControlOntology

*Doctor*, by default, has all access permissions (read, write, update and delete) over the users' medical history. Another user, *Maria*, with the role of *Patient*, declares that *Jack* should only be permitted to *Read* her medical history and be banned from other actions. As a result, *Jack* is only eligible to *Read* the medical history of Maria. Moreover, in our access control rules we have designated the negative-negative strategy. It means that in cases of "conflict" (when a triple comes in the scope of both positive and negative permissions) and "not-defined" (no access right defined) we consider prohibition of an action.

As mentioned earlier, in our access control scheme, context information affects the execution of the access control rules at three points:

1. The impact of the context information in reporting the patient's medical status: If some pre-defined emergency conditions are satisfied, e.g., the user's heart beat rises unexpectedly, the status will be reported as "critical". To define the critical situation, different information is collected from different sources. For instance, when the user's sensor shows a high heart rate, the space should check from the user's calendar whether he/she is in a gym, or resting at home. Then it can decide whether the rise in the heart rate is due to a normal activity or if it is a critical situation. In a critical situation, no control for the access is checked. This eliminates the required time for authorization and access control for caregivers assisting the patient.

2. The impact of the context information in role allocation: At this point, some run-time parameters take part in decision making. For instance, The context of location and time may be taken into account in deciding which role and, consequently, which restrictions the user may have. For example: a doctor is restricted to only read the medical history of the patients after office time or outside the hospital:
*(triple (Shohreh, hasRole, Doctor))*
*(triple (Shohreh, hasData, LocationShohreh))*
*(triple (LocationShohreh, hasValue, TrainStation))* $\Rightarrow$
*(assert(triple(Shohreh,roleHasReadPermissionOnData, ?MedicalHistory)))*

# 4. ACCESS CONTROL EVALUATION

The case study focuses on an eHealth information system, in which there is a range of personal and medical information of the users stored on the SIB. Users take the available roles, e.g., patient, doctor, and patient's family member. In accordance with the roles, they are given privileges to access the data on the SIB. We perform an experiment to measure the required time to check the access for four different types of requests: write new data to the SIB, read, delete and update the data, for different data set sizes. The size of the SIB grows exponentially by increasing the number of the triples (from 1 to 100000). From the result of the response times, we can have an estimation on the overhead that our access control scheme brings to the system.

The experiment is done on an Intel Core 2 Duo CPU E8500 @ 3.16GHz, on a virtual machine with a RAM of 1.5 GB, and Ubuntu 12.04 LTS running Python M3 library.

We evaluated the overhead of the access control scheme by measuring the time it takes to check the access right, when a request is received. The experiment is done for four different operations supported by SSAP protocol (read, write, delete and update a triple). We repeat the experiment for datasets with different numbers of triples, ranging from 1 to 100000. The result of the experiment is presented in Table 2. According to the table, the highest response times belong to the requests for update and write in the 100000-triple dataset.

Table 2: Access control times for different sizes of SIB per operation in seconds

| #Triples | Read | Write | Delete | Update |
|---|---|---|---|---|
| 1 | 0.00471 | 0.00621 | 0.00422 | 0.00738 |
| 10 | 0.00407 | 0.00582 | 0.00439 | 0.00607 |
| 100 | 0.00387 | 0.00750 | 0.00408 | 0.00680 |
| 1000 | 0.00539 | 0.00905 | 0.00496 | 0.00909 |
| 10000 | 0.01908 | 0.03672 | 0.02008 | 0.03145 |
| 100000 | 0.10668 | 0.20454 | 0.10594 | 0.19553 |

The result is also illustrated as a plot in Figure 3. The response times to check the access rights for four different actions are displayed in distinct colors. As seen in the figure, with the enlargement in the size of the SIB, the execution time grows gradually for each plot. Throughout the whole experiment, the graphs related to the response time of the *Write Request* and *Update Request* have approximately the same scheme. Their similar functions at triple level explain this behavior. Moreover, these two requests hold higher response times than the other two. The *Delete Request* appears to be the fastest request to be checked in execution time because in this operation no value is returned.

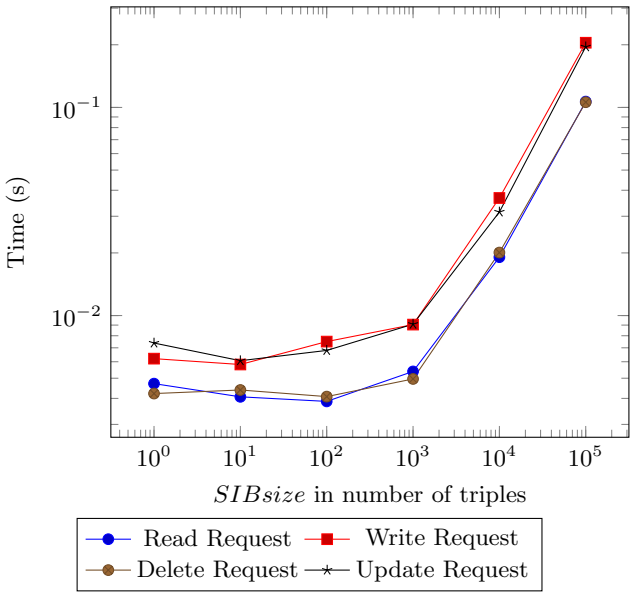


**Figure 3: Average time for different access control requests and different sizes of the SIB**

# 5. CONCLUSION

In this paper, we proposed a security framework and an access control scheme for Smart-M3 based spaces. We implemented the scheme via CLIPS rules and evaluated it by measuring the response time of controlling an access request. The attained results showed that the response times related to write and update requests had approximately the

same value in the whole experiment, because of their similar functionality at triple level. While these two requests had the slowest response times, delete was the fastest request to check the access for. We concluded that our access control scheme introduces only a small overhead to the system. The highest response time was 0.2 seconds (to check the access right for a write or an update request. This small overhead is acceptable considering the importance of the security and privacy that the approach provides for the system.

Our future work will include the implementation of security alerts using the publish/subscribe mechanism supported by the Smart-M3. Security alerts can be implemented for two different situations: a) the critical situation in the user's health status, and b) access to the data that the user has labeled as highly sensitive.

# 6. REFERENCES


[1] OWL for Services: http://www.ai.sri.com/daml/services/owl-s/security.html.

[2] F. Abel, J. L. De Coi, N. Henze, A. W. Koesling, D. Krause, and D. Olmedilla. Enabling advanced and context-dependent access control in RDF stores. volume 4825 of *Lecture Notes in Computer Science*, pages 1–14. Springer, 2007.

[3] S. Al-Rabiaah and J. Al-Muhtadi. ConSec: Context-Aware Security Framework for Smart Spaces. In *Innovative Mobile and Internet Services in Ubiquitous Computing (IMIS), Sixth International Conference on*, pages 580–584, Palermo, 2012. IEEE.

[4] C. A. Ardagna, S. De Capitani di Vimercati, G. Neven, S. Paraboschi, F.-S. Preiss, P. Samarati, and M. Verdicchio. Enabling privacy-preserving credential-based access control with XACML and SAML. In *Computer and Information Technology (CIT), 2010 IEEE 10th International Conference on*, pages 1090–1095, Bradford, United Kingdom, 2010.

[5] M. Baldauf, S. Dustdar, and F. Rosenberg. A survey on context-aware systems. *International Journal of Ad Hoc and Ubiquitous Computing*, 2(4):263–277, 2007.

[6] T. Berners-Lee, J. Hendler, O. Lassila, et al. The semantic web. *Scientific american*, 284(5):28–37, 2001.

[7] H. Chen, T. Finin, and A. Joshi. An ontology for context-aware pervasive computing environments. *The Knowledge Engineering Review*, 18(03):197–207, 2003.

[8] N. Díaz Rodríguez, M. Cuéllar, J. Lilius, and M. Delgado Calvo-Flores. A survey on ontologies for human behavior recognition. *ACM Computing Surveys (CSUR)*, 46(4):43, 2014.

[9] N. Díaz-Rodríguez, R. Wikström, J. Lilius, M. P. Cuéllar, and M. D. C. Flores. Understanding Movement and Interaction: An Ontology for Kinect-Based 3D Depth Sensors. In *Ubiquitous Computing and Ambient Intelligence. Context Awareness and Context-Driven Interaction*, pages 254–261. Springer International Publishing, 2013.

[10] T. Finin, A. Joshi, L. Kagal, J. Niu, R. Sandhu, W. Winsborough, and B. Thuraisingham. Rowlbac: Representing role based access control in owl. In *Proceedings of the 13th ACM Symposium on Access Control Models and Technologies*, SACMAT '08, pages 73–82, New York, NY, USA, 2008. ACM.

[11] S. Haibo and H. Fan. A context-aware role-based access control model for web services. In *IEEE ICEBE*, pages 220–223, 2005.

[12] J. Hebeler, M. Fisher, R. Blace, and A. Perez-Lopez. *Semantic web programming*. Wiley, J. & Sons, Indianapolis, Indiana, 2011.

[13] A. Kashevnik and N. Teslya. Context-Aware Access Control Model for Smart-M3 Platform. 2013.

[14] A. Khan and I. McKillop. Privacy-centric access control for distributed heterogeneous medical information systems. In *Healthcare Informatics (ICHI), 2013 IEEE International Conference on*, pages 297–306, Philadelphia, PA, 2013. IEEE.

[15] D. G. Korzun, S. I. Balandin, and A. V. Gurtov. Deployment of Smart Spaces in Internet of Things: Overview of the Design Challenges. In *Lecture Notes in Computer Science 8121: 48–59, 2013*.

[16] A. Mhamed, M. Zerkouk, A. Husseini, B. Messabih, and B. Hassan. Towards a context aware modeling of trust and access control based on the user behavior and capabilities. In J. Biswas, H. Kobayashi, L. Wong, B. Abdulrazak, and M. Mokhtari, editors, *Inclusive Society: Health and Wellbeing in the Community, and Care at Home*, volume 7910 of *Lecture Notes in Computer Science*, pages 69–76. Springer Berlin Heidelberg, 2013.

[17] S. P. Miller, B. C. Neuman, J. I. Schiller, and S. J. H. Kerberos authentication and authorization system. In *Project Athena Technical Plan*, Cambridge, USA, 1987. Massachusetts Institute of Technology (MIT).

[18] M. Mohsin Saleemi, N. Díaz Rodríguez, J. Lilius, and I. Porres. A Framework for Context-Aware Applications for Smart Spaces. In *Applications and the Internet (SAINT), 2011 IEEE/IPSJ 11th International Symposium on*, Munich, Bavaria.

[19] M. Peleg, D. Beimel, D. Dori, and Y. Denekamp. Situation-based access control: Privacy management via modeling of patient data access scenarios. *Journal of Biomedical Informatics*, 41(6):1028 – 1040, 2008.

[20] O. Sacco and A. Passant. A privacy preference ontology for linked data. In *Linked Data on the Web Workshop at the World Wide Web Conference*, 2011.

[21] J. Suomalainen. Flexible security deployment in smart spaces. In *Lecture Notes in Computer Science*, volume 7096, pages 34–43. Springer, 2012.

[22] A. Toninelli, R. Montanari, L. Kagal, and O. Lassila. A semantic context-aware access control framework for secure collaborations in pervasive computing environments. In I. Cruz, S. Decker, D. Allemang, C. Preist, D. Schwabe, P. Mika, M. Uschold, and L. Aroyo, editors, *The Semantic Web - ISWC 2006*, volume 4273 of *Lecture Notes in Computer Science*, pages 473–486. Springer Berlin Heidelberg, 2006.

[23] A. Toninelli, R. Montanari, L. Kagal, and O. Lassila. Proteus: A semantic context-aware adaptive policy model. In *Policies for Distributed Systems and Networks. POLICY '07. Eighth IEEE International Workshop on*, pages 129–140, Bologna, Italy, 2007.

[24] C. D. Wang, T. Li, and L. C. Feng. Context-aware environment-role-based access control model for web services. In *Multimedia and Ubiquitous Engineering. International Conference on*, pages 288–293, 2008.