Interoperability of Security-Enabled Internet of Things

ARTICLE *in* WIRELESS PERSONAL COMMUNICATIONS · DECEMBER 2011

Impact Factor: 0.98 · DOI: 10.1007/s11277-011-0384-6

3 AUTHORS, INCLUDING:

Josef Noll
University of Oslo
88 PUBLICATIONS **343** CITATIONS

Wireless Pers Commun (2011) 61:567–586
DOI 10.1007/s11277-011-0384-6

Interoperability of Security-Enabled Internet of Things

Sarfraz Alam · Mohammad M. R. Chowdhury · Josef Noll

Published online: 30 August 2011

Abstract The future Internet will embrace the intelligence of Web 3.0 and the omnipresence of every day connected objects. The later was envisioned as the Internet of Things. Security and interoperability concerns are hindering the service innovations using the Internet of Things. This paper addresses secure access provision to Internet of Things-enabled services and interoperability of security attributes between different administrative domains. In this paper we proposed a layered architecture of Internet of Things framework where a semantically enhanced overlay interlink the other layers and facilitate secure access provision to Internet of Things-enabled services. The main element of semantic overlay is security reasoning through ontologies and semantic rules. Finally the interoperability of security aspect is addressed through ontology and a machine-to-machine platform. This paper provides implementation details of security reasoning and the interoperability aspects and discusses crucial challenges in these areas.

Keywords Internet of Things · Ontology · Security · Semantics · Reasoning

S. Alam · M. M. R. Chowdhury · J. Noll
University of Oslo, Kjeller, Norway

S. Alam · M. M. R. Chowdhury (✉)
UNIK, Kjeller, Norway
e-mail: mohammad@unik.no

S. Alam
e-mail: sarfraz@unik.no

J. Noll
Center for Wireless Innovation Norway, Oslo, Norway
e-mail: josef@unik.no

1 Introduction

The Web 2.0 facilitates user-centric collaboration. In Web 3.0, Internet becomes intelligent as the Web itself understands the meaning of its contents. However, the future Internet is no longer limited to connecting people and services, it envisioned to connect every things that were not historically connected. Within this vision Internet of Things (IoT) creates a new paradigm of Internet that is intelligent and omnipresent.

Through IoT new digital ecosystem creates new business opportunities for retail, logistics, food, health, energy, smart home, and transportation sectors. For instance, IBM utilized the IoT for Norwegian Sea oil platforms by implementing a service, which gathers real-time information from the bottom of Sea so that a better decision can be made in order to drill down to the Sea. The miniaturization of devices, increase of computational power, and reduction of energy consumption support this new ecosystem driven by IoT. However, coupling of intelligence with omnipresence is not a trivial job. Objects ranging from powerful nodes to tiny sensors constitute the 'Things' in Internet of Things.

Offering innovative services using intelligent and omnipresent IoT is hindered due to several technical and non-technical challenges such as reliable integration of scores of 'Things'; scalability of systems; security, privacy and interoperability concerns. In this paper, we are going to limit our focus on security and interoperability concerns. Among the security challenges such as ensuring confidentiality, integrity, availability and access control, we only focus secure access provision to IoT-enabled services. Within the scope of interoperability of security, this paper will address how different security attributes and constraints lying in different administrative domains will work together to secure an integrated operation. The concepts and results presented in this are the outcome of the research conducted in an ongoing European project, pSHIELD [1].

In this regard we propose a functional architecture of the IoT framework that incorporates secure access provision. We implemented several components of the functional architecture using the semantic technologies. As a whole, the paper makes the following key contributions:

- A functional architecture of IoT framework is going to be introduced.
- In the architecture, a semantic overlay (on top of 'Things') is proposed to facilitate the intelligence in IoT.
- Ontologies are designed to contrive partly the semantic overlay.
- A rule-based service access mechanism is proposed.
- Interoperability of security is going to be addressed through ontology and machine-to-machine (M2M) technology.

The rest of the paper is structured as follows: Sect. 2 provides motivation of including semantic overlay layer into the IoT framework, Sect. 3 presents the Interoperable Rail Information System (IRIS) scenario that introduces the secure access and interoperability of security aspects, Sect. 4 explains the detailed security requirements of an IoT environment, Sect. 5 introduces the conceptual view and functional architecture of the proposed IoT framework, the implementation details of part of the proposed architecture and the interoperability of security aspect will be presented in Sect. 6, the next section will discuss several related works, and introduce the challenges and future works in relation to the the areas presented in this paper. The paper will conclude with a summary of the key contributions of the research.

2 Motivation

This section provides a brief discussion on the motivation of incorporating an overlay layer and semantic enhancement of such layer in the IoT framework.

2.1 Overlay

Sensors sensing the environments constitute one of the key elements of the Internet of Things. In most cases sensors are resource constrained devices having no or limited processing capability. Most of them are only capable of retrieving information from the environment. In order to provide services, we need to derive some decisions based on these retrieved information and pre-defined logics. For example, In case of secure access we need to compute access authorization decisions based on complex constraints. Instead of hardcoded decisions, we need dynamic update of decisions. Nowadays human intervention is not desirable for these decision making processes. It requires automated reasoning which is defined as the process of deriving new facts based on predefined knowledge. Such reasoning process involves extensive information processing and computation of data and logics. Hence reasoning requires structured knowledge about the devices and sensors, sensor networks, and sensor data.

To realize these, on top of physical IoT environment we need an overlay that contains a model to describe these structured knowledge and a reasoning process.

2.2 Semantic Enhancement

From a technological point of view, semantics mean the explicit interpretation of domain knowledge to make machine processing more intelligent, adaptive and efficient. The data along with their interpretation are critical for decision making and planning. In this regard, semantic technologies that include standards, methodologies and tools act as the enabler. Berners-Lee et al. [2] first envisioned the promise of the Semantic Web technologies in order to make the meaning of data useful. Semantic technologies make use of the formal meaning of data and information for deriving new knowledge from the known facts. Semantic technologies can satisfy the requirements described in previous section through the following capabilities:

- machine understandable knowledge description
- machine understandable logic description
- automated reasoning

3 The Interoperable Rail Information System (IRIS)

This section will introduce a specific scenario (IRIS) around which we are going to propose an architecture and implement a prototype. The scenario envisioned a continuous monitoring of trains and railway infrastructure. The scenario puts forward the interoperability of security requirement and this section will explain this issue.

3.1 Scenario Description

The purpose of this scenario is twofold *(i)* detecting any unusual condition such as high temperature of components, vibrations and unexpected movement, and *(ii)* transferring and making available such information to different actors (i.e., train operator, rail infrastructure

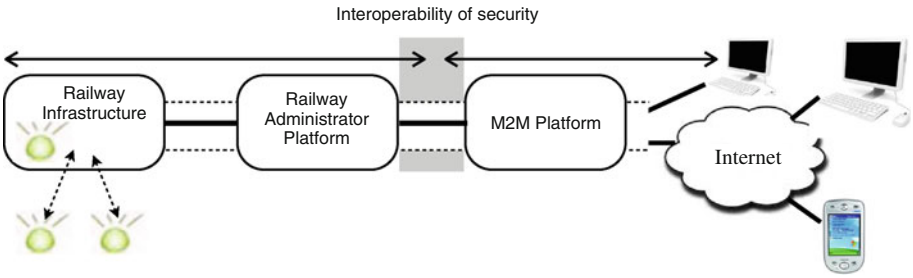

Fig. 1 Infrastructure scenario, collecting and distributing sensor information

owner, consumer) involved in the rail system both automatically and in a request/response demand-based passive mode. The train is equipped with several heterogeneous computing devices such as sensors, actuators, GPS receiver, and gateway-embedded computer for detection of such conditions. These devices interact using heterogeneous protocols for sensing the information in their vicinity and send it to the gateway. As an intelligent device, the gateway figures out any irregularity, and it sends the details to the smart train operator. If the irregularity information is related with rail infrastructure, then the infrastructure owner and provider will also be interested to know the information. The gateway sends this information to all concerning actors, but they also need monitoring and periodically checking about the condition of the train and the rail infrastructure. The gateway embedded-computer is geared with the proposed IoT virtualization framework that exposes the train sensors as services for enabling on-demand remote monitoring application.

Figure 1 outlines the main elements of the IRIS scenario where the third party service providers can access the sensor data collected from the railway infrastructures. It is required to ensure the secure access and interoperability of security when they are transferred from one administrative domain (the Railway) to another administrative (any Service Provider). In this paper, we are concerned with the following two aspects:

– access to sensors and sensor data
– interoperable security between different administrative domains

In order to facility sensor integration and interoperability, this work makes use of the standardized machine-to-machine (M2M) technology as suggested by ETSI.

3.2 Cell-Based M2M Standardisation

According to the definition from [3], the concept of M2M denotes a communication mechanism between two or more machines in a network. The main goal of M2M is to make the devices communicate with each other without human interaction, the so-called automatic network of smart devices. The idea behind the M2M technology is to enable the flow of information between different devices, so that the device can deliver the observations it has performed on the site to the end users (at remote location).

The European Telecommunications Standards Institute (ETSI) has proposed an architectural standard, TS 102.690 that can be used for any infrastructure based on the M2M concept [4]. The main focus is on generation of the eco-system of device manufacturers, service providers and trusted third parties.

Details are defined in the technical standards document TS 102 690, which a.o. describes authentication and authorization of applications through the Network Security Capability

(NSEC). The suggested solution is based on a hierarchical set of cryptographic keys, starting with a root key K_R being used to create service keys K_S and allowing for one application key K_A per application [4]. The main application environment heads towards an M2M device or gateway being able to negotiate the K_R, supporting mobile-based installations.

3.3 Interoperability of Security

Even though the TS 102 690 supports root, session and application keys K_R, K_S, and K_A, the current implementation is based on username/password to achieve access to the sensor data. Such a limitation does not support role-based, context-based or content-based access, as necessary for the integrated operation as indicated in our scenario. In addition when multiple administrative domains are integrated, varying definition of roles, contexts and contents across the domains are real concerns for integrated operation. As for example, during the integrated operation the following questions may arise:

- how the system would recognize two different roles from different administrative domains that literally possess the same access rights?
- how the system would make distinction between the security levels defined using different notions at different administrative domains?

These two situations will be revisited in the Sect. 6 for further clarification. When multiple stakeholders are involved in operation, the security attributes and constraints need to be interoperable across the border. We envision an extension of the current TS 102 690 that can deal with this requirement. Our implementation is based on the functional architecture of Fig. 3, using the semantic overlay for the interoperability of security attributes. Based on a definition of the content of a sensor, e.g. temperature higher than anticipated range, a service notifier is activated allowing service access for operators with corresponding access rights. A second example includes the position of the train in case of normal and delayed operation. Delay margins are characterized in the semantic overlay to allow for distributions as *on-time* or *delayed between 3 and 5 min*, which is information available for everyone, and *30 s* delay as information for the train control at the control center.

4 Handling Security in IoT

This section outlines the conventional security requirements for an IoT environment and how security operation can be externalized. The interoperability of security becomes a crucial requirement for integrated operation and the issue has been explained in the previous section.

4.1 Conventional Security Requirements for IoT

We anticipate that security of IoT will soon become a challenging task as IoT paradigm will bridge the physical world with future internet. The increasing complexity of the system will multiply the number of security challenges. All IoT services need to satisfy some basic security properties. However, additional security requirements for a specific IoT service may depend on specific applications and contexts.

4.1.1 Confidentiality

IoT services may contain sensitive data; therefore, IoT connected objects data should be kept confident. Confidentiality can be achieved through encryption. Different existing symmetric

and asymmetric encryption schemes can be leveraged to ensure confidentiality. However, selection of a particular type of encryption is highly application and device capability dependent. To exemplify, consider a smart home environment that maintains the information about the owner activity at the home. The owner will never welcome that anybody who comes to his home will read the data just by viewing the activity monitoring device.

4.1.2 Integrity

IoT services exchange critical data with other services and also with the third parties (e.g., authorities, service providers, control centers etc.), which put forward stringent demand that sensed, stored and transmitted data must not be tampered either maliciously or accidentally. Integrity protection of sensor data is crucial for designing reliable and dependable IoT applications. This is ensured with message authentication codes (MAC) using one way hash functions. The selection of MAC technique again depends on application and device capabilities. Consider the example of smart home that is connected with the smart grid. The smart grid provider deployed an electricity consumption monitoring service in order to produce electric bill. The provider never wants that the consumption data can be tampered during transmission.

4.1.3 Availability

Our envisaged IoT environment may comprise of sensor node hosted services. Therefore, it is extremely important that these IoT services be available from anywhere at any time in order to provide information (i.e., measured data, sensor alarm, etc.) continuously. There is no single security protocol that can satisfy this property. However, different pragmatic measures can be taken to ensure the availability. For example, in the aforementioned smart home if the attacker knows the consumption monitoring service, he can launch the denial-of-service (DoS) attack by just trying to send false service requests and the sensor nodes are incapable of handling huge number of requests due to resource limitations. Since any transmission (i.e., receiving or sending) consume power, the node will eventually run out of its battery.

In addition to these traditional security properties we also identify the following properties that need to be addressed by any IoT environment.

4.1.4 Authentication

It refers to the means used for the verification of one's identity. In IoT context, mutual authentication is required because IoT data is used in different decision making and actuating processes. Therefore, both the service provider and service consumer needs to be assured that the service is access by authentic user and service is offered by an authentic source. Furthermore, strong authentication mechanism needs to be deployed in order to prevent impersonation. Enforcing any authentication mechanism requires to register user identities and resource limitation of IoT objects poses stringent constraints to enable any authentication technique.

4.1.5 Authorization

It refers to the means of expressing the access polices that explicitly assign certain permissions to subjects. The IoT environment needs to provide fine-grained, re-useable, dynamic, easy to use polices defining and updating mechanism. Thereby, it is imperative to externalize

the policy definition and enforcement mechanism of IoT services. Furthermore, the resource limitation of IoT sensor node restricts to employ such mechanism.

4.1.6 Access Control

This is an enforcement mechanism that allows only authorized users' access to the resources. The enforcement is usually based on access control decisions. Since, IoT is becoming omnipresent, privacy issue has become a real concern. For instance, consider the example of smart home that has smart power metering as IoT services and without a proper access control mechanism it could not only lead to disclosure of electricity usage pattern but it could also help adversary to deduce user related information such as when the user is at home, at office or traveling. Even it is possible to infer about the user activity (i.e., watching TV, sleeping, etc.) and home appliance present in the home. Therefore, it is extremely important to disclose users data only to authorized parties.

4.1.7 Trustworthiness

Many applications which are sensitive in nature such as safety critical services, health care services need to assess trustworthiness of several entities involved. From IoT application perspective, assessing trustworthiness of sensors and sensor data is important. Malicious sensor nodes and erroneous or non-trustworthy sensor data can lead to a disaster in a safety critical situation. Untrusted sensor data may come from a trusted sensor node. Non-trustworthy behavior may have two reasons: intentional misbehavior and unintentional errors. It might be easier to ensure trustworthiness of Internet of Things by including trustworthiness assessment feature than by hardening the security of nodes and data through physical measures.

4.1.8 Auditing

The auditing keeps track of the user's interaction with the system. The IoT environments need to know when their services are accessed, who is making the service request, when the request is happening. This information will not only help in managing the security but also in evaluating security risk. In case of security breach, such information may help in identifying the security hole exist in the system. Maintaining an audit-trail in IoT services is a challenging task.

4.2 Security Proxy Model

The authorized access to IoT services or data becomes the pivotal issues, which was considered to be trivial in the past due to very low number of attacks. These contemporary security models are based on firewall and SSL that could only provide point-to-point security, which does not fit in a IoT.

This paper proposes a security proxy model for IoT services by leveraging SOA [5] approach to satisfy the requirements outlined in previous section. Figure 2 depicts the overall security proxy approach. The proxy consists of Audit and Policy Enforcement Point (PEP) that is connected to a Policy Decision Point (PDP) and an Identity manager (IdM). The Audit is responsible for managing the logs of service calls-out and maintains the history of service interaction. The proxy follows the principal of reverse proxy but we tailored it to the special characteristics of the IoT. The security proxy acts as a service level gateway in front of IoT services. The service consumers access IoT service through the security proxy. In this case,

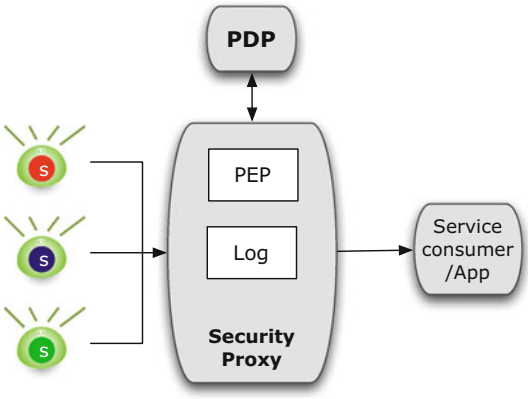

Fig. 2 Security proxy model

the security proxy plays a role of edge-oriented policy enforcement point, which uses a policy decision point (PDP) to get access decisions.

Among the security requirements described above, we are going to address the access authorization and access control aspects in this paper. We exploit semantic rules to express the access authorization constraints to infer access decisions. We named this process security reasoning which will be described in Sect. 6.

5 From Concepts to Architecture

In light of the scenario and the security focus of the paper, this section will propose a conceptual view of the Internet of Things framework. The understanding of the concepts will then help introducing a functional architecture of Internet of Things.

5.1 The Conceptual View of IoT Framework

The high level view of the proposed IoT framework is illustrated in Fig. 3. The framework targets the user-centric IoT cloud, consisting of different classes of devices. We classify these devices in nano, micro and personal nodes based on their capabilities (see Sect. 6 for classification). The main driver for the virtualization framework is to ease the application development process for IoT. The framework helps the application developers to focus on the application logic by minimizing the development efforts related to connectivity, monitoring and security. The proposed framework combines technologies from SOA [5] and semantic web to address the dynamics to a real-time IoT cloud.

The core idea is to provide the semantic description of node types, capabilities of an IoT cloud and expose nodes capabilities in the form of web services. This will not only integrate the IoT with service-world but it will also allow third party applications to query about the data resided in the IoT cloud.

5.2 Functional Architecture

The functional architecture of the proposed IoT framework is illustrated in Fig. 4. The architecture is composed of four layers: (i) the Communication and Real-world Access Layer, (ii) the Semantic Overlay Layer, (iii) the Service Virtualization Layer, and (iv) the Application

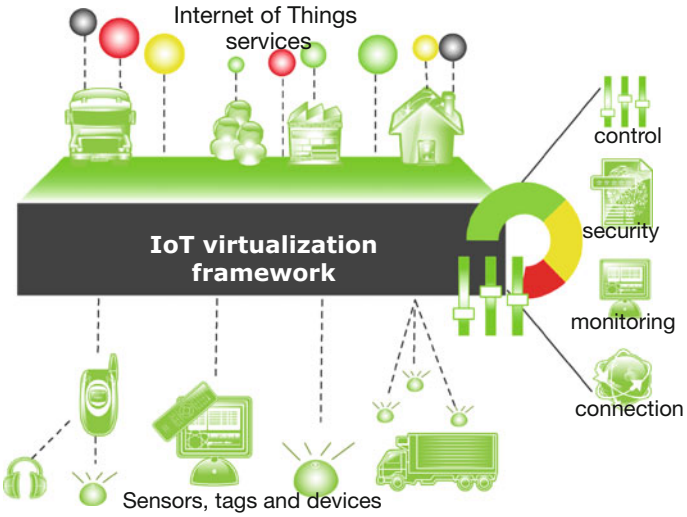

Fig. 3 The conceptual view of IoT framework

Layer. The framework follows the separation of concerns [6] principals, where each layer is responsible for different concerns. Here, we will provide the overview of each layer of the proposed Internet of Things architecture.

5.2.1 The Communication and Real-World Access Layer

The layer provides an interface with an underlying IoT cloud. It implies an adapter oriented approach to address the technological diversity regarding nodes and communication mechanisms. The layer provides different adapters to communicate with different type of nodes for example Sun SPOT adapter to communicate with Sun SPOT nodes. The layer perform number of tasks such that discovering nodes, receiving events from nodes and dispatching them to upper layers both for making sense of the events and sending them to their subscriber, and invoking services hosted on the nodes.

5.2.2 The Semantic Overlay Layer

The semantic overlay layer acts as both the integrator and the interface between different layers. It provides the semantic model of an underlying IoT cloud by maintaining IoT ontology, sensor ontology, event ontology and service access polices. The layer also performs number of tasks such as facilitating create, read, update and delete (CRUD) operations on the semantic model, and translating SensorML [7] description into OWL description.

5.2.3 The Service Virtualization Layer

The service virtualization layer provides web service interface for the functional aspects of the nodes in an IoT cloud. The layer perform various tasks such as translating virtual service into web service definition, generating micro-formats of available web service, publishing

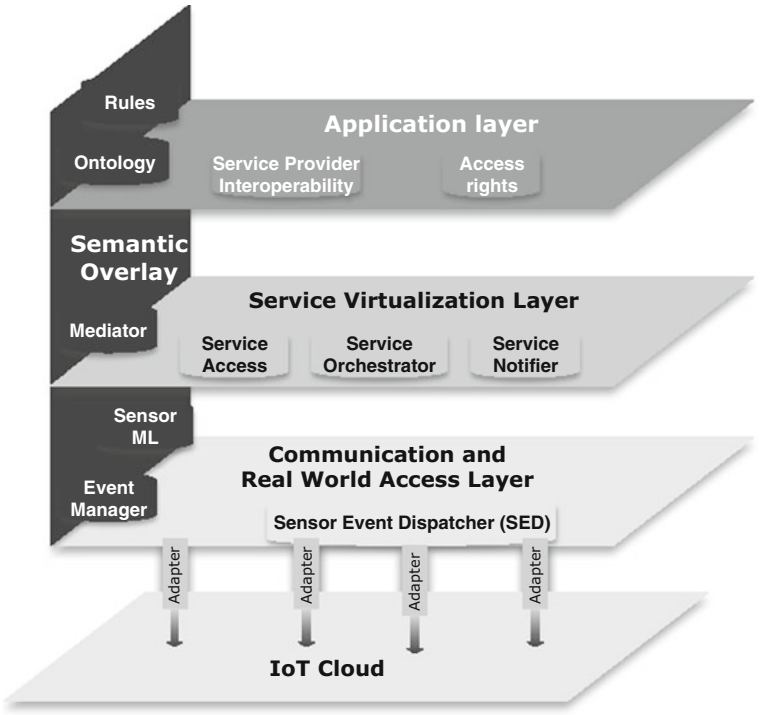

Fig. 4 Functional architecture

services both in service registries and social network sites, and notifying subscribers about the IoT cloud events.

5.2.4 The Application Layer

The application layer contains the real applications created using the data, semantics of data and application logics. Resolving the interoperability issues between different service provider's platforms is the functionality of this layer.

6 Implementation

This section presents the implementation details of the security reasoning and interoperability functionalities. The security reasoning module would be located in the semantic overlay layer of the functional architecture (Fig. 4). From the use case scenario description (see Fig. 1), this module may lie either on-board the wagon or at the railway administrator platform.

6.1 Security Reasoning

The security reasoning functionality derives access authorization decisions to IoT-enabled services. It requires a formal knowledge base of the whole domain containing sensor, sensor

Table 1 OWL building blocks used to design the ontology

Ontology element	Ontology building block	Explanation
Class	*owl:subClassOf*	$SC\,(C_1) \subseteq SC\,(C_2)$, the semantic scope of C_1 is narrower than that of C_2
	owl:disjointWith	$SC\,(C_1) \cap SC\,(C_2) = 0$, each *owl:disjointWith* statement asserts that C_1 and C_2 have no individuals in common
	owl:unionOf	The semantic scope of C_3 is $SC\,(C_1) \cup SC\,(C_2)$
	owl:intersectionOf	The semantic scope of C_3 is $SC\,(C_1) \cap SC\,(C_2)$
	owl:complementOf	$SC\,(C_1) \neq SC\,(C_2)$, the semantic scope of C_1 is the complement of the scope C_2
	owl:equivalentClass	$SC\,(C_1) = SC\,(C_2)$, the semantic scope of C_1 is equal to the scope C_2
Property	Property	$P(i_1, i_2)$ states that i_1 relates with i_2 through the property P. i_2 can be a numeric value in case of datatype property
	owl:symmetricProperty	If P relates i_1 & i_2 then P also relates i_2 & i_1 and can be represented as $P \equiv P^-$, where P is the inverse property of P^-
Instance	Instance	$\{i_1, i_2, \ldots, i_n\} : SC(C_1)$, instances $i_1, i_2, \ldots, i_n$ belong to class C_1
	owl:sameAs	$\{i_1\} = \{i_2\}$, two instances i_1, i_2 are stated as the same
	owl:differentFrom	$\{i_1\} \neq \{i_2\}$, two instances , i_1, i_2 are stated as different from each other

data, users and user attributes such as role. Semantic rule specifies the access authorization constraints and the execution of rules will generate the authorization decisions.

6.1.1 Formal Knowledge Base

A knowledge base is a repository of information about a particular domain of interests. Among the two different types of knowledge base: human-readable knowledge base and machine-readable knowledge base [8], this research used the later one because of its machine understandability. For example, the sensor node descriptions (e.g. identifiers, capabilities) and the extracted data and relevant semantics will be formally represented in the knowledge base. A typical knowledge base consists of concepts, properties, and instances. We encoded the knowledge base using the ontology [9] and to be more specific we used the Web Ontology Language (OWL) [10]. The ontology is a set of classes C, properties P and instances i. The key concepts of the domain are defined through classes. In ontology a property establishes the relationship between the two instances. A property belongs to a domain and has a range. Syntactically, a domain links a property to a class and range links a property to either a class or a data range [10]. From an instance point of view, a property relates instances from the domain with the instances from the range. The real actors of a practical use case scenario (e.g. individuals) are defined through instances and they belong to the classes. At the beginning we started with a very simple ontology and we used the building blocks of OWL defined in Table 1 to design the ontology. We assume that the semantic scope (SC) of a concept (class) C_i is represented as $SC(C_i)$.

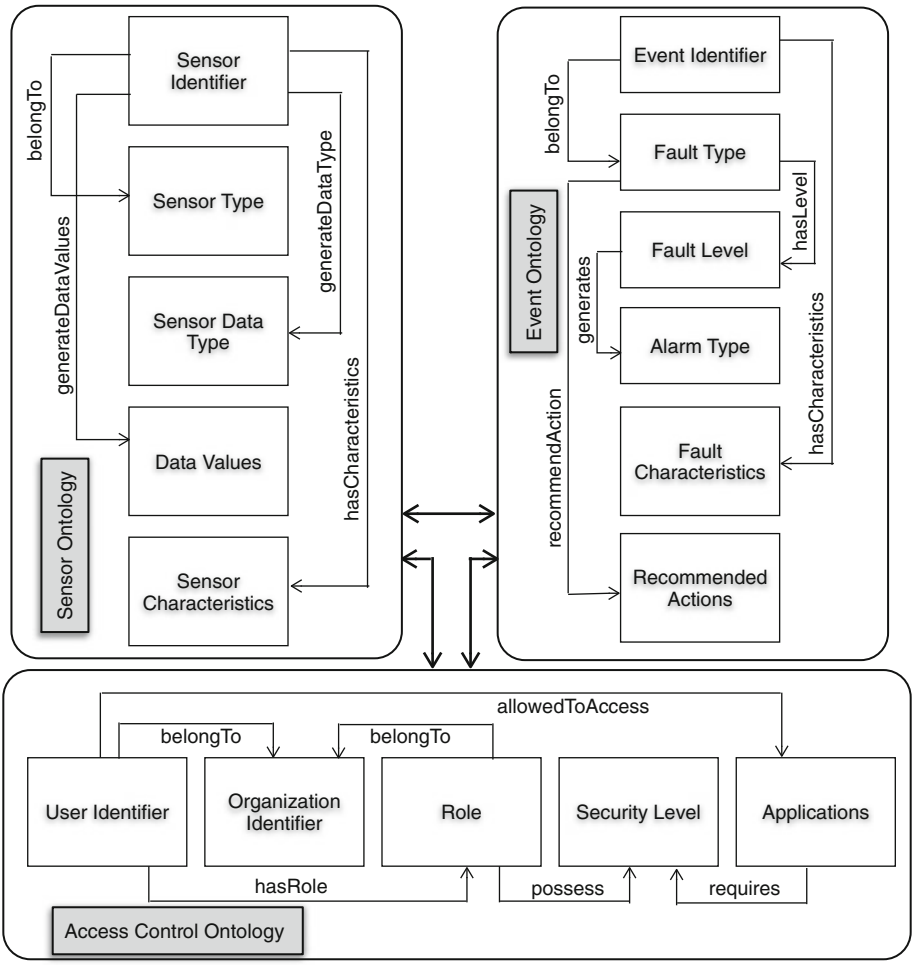

Fig. 5 Overview of the ontologies through classes and properties

In the initial implementation, the complete knowledge base is divided into three different ontologies: Sensor Ontology, Event Ontology, and Access Control Ontology. Figure 5 illustrates these ontologies through main classes and properties. These three ontologies are interlinked and evaluation of some rules (discussed in the next section) requires retrieving of elements from multiple ontologies. The instances of the ontologies are derived from actors of the IRIS scenario described in Sect. 3.

Sensor Ontology: It describes the sensors and the retrieved data by the sensors.

Event Ontology: It describes faults and their characteristics. Most of the instances of these classes are derived from the Sensor Ontology using certain policies.

Access Control Ontology: It describes the actors involved in secure access provisioning.

relatesTo property links *Sensor Data Type* (domain) to *Application* (range). Similarly relatesTo links Alarm Type and Recommended Action to Application. This is how these three ontologies are interlinked through a property relatesTo. In this paper, we used Protégé Ontology editor platform [11] to design these ontologies.

6.1.2 Semantic Rules

The policy encodes definite course of actions to determine present and future decisions. Through policies the middleware layer provides the reasoning support aiming for decision making. The reasoning process derives new facts (the decisions) based on the constraints defined on top of the pre-defined facts (the knowledge base). The policy in the middleware layer is composed of constraints that control the courses of actions in decision making process. Sometime such process only handles the composition of sensors and sensor data. Then the policy will simply be responsible for the discovery of right sensors and sensor data. The applications in the upper layer and the requirements of the applications govern the design and specification of the policies. Policies are specified manually under the umbrella of each application or multiple applications. The policies and the knowledge base work closely so that the elements of the constraints can come from the knowledge base and multiple constraints can be added in conjunction within a policy. We implemented the policies using the Semantic Web Rule Language (SWRL) [12] and the Semantic Web Query–Enhanced Web Rule Language (SQWRL) [13].

We want to create a semantic rule that would in practice allow only specific *Role* group members within the service provider administrative domain to access an application such as monitoring. The underlying features of monitoring application are for example establishing communication with the sensors, retrieve and transfer data across multiple domain etc. The complete detail of such application is beyond the scope of this paper. However, the implementation of establishing communication to sensors has been demonstrated in the Sect. 6.2 of this paper.

At the initial phase of implementation, we are concerned with ensuring access authorization provision to authenticated service. The logical explanation of rule to *generate decisions on access authorization provision* is:

If User Identifier belongs to specific Organization (e.g. JBV)
and User Identifier belongs to a specific Role Group,
and the Role Group possesses a certain Security Level,
and the Monitoring application requires the same Security Level,
then that particular User is allowed to access Monitoring application.

In this case, we assume that not all Supervisor has the same security level. This can be represented through the following semantic rule using the SWRL syntax:

$$belongsTo\,(UserIdentifer, JBV) \wedge hasRole\,(UserIdentifer, Supervisor) \wedge$$
$$possess\,(Supervisor, Level4) \wedge requires\,(Monitoring, Level4) \Rightarrow$$
$$allowedToAccess\,(UserIdentifer, Monitoring)$$

6.1.3 Interoperability Through Ontology

We planned to resolve the interoperability problem using an ontology and a M2M platform. The ontology in this case makes the keyword mapping. In order to clarify the interoperability aspect, we take into consideration the following two situations:

Different Role Group: This situation refers to the scenario where different organizations maintain their Roles/Responsibilities in a different way. Across the organizational boundary Role names can be different, meaning and responsibilities can be different, and even the hierarchies can also vary. For example, role hierarchies of the Railway Administration (Fig. 1) can be as follows:

$Supervisor \Rightarrow GroupLeader \Rightarrow Engineer \Rightarrow Technical \Rightarrow FieldWorker$
When a user belongs to an Express Mail Service Provider (that is using the freight train to carry goods) is willing to access the Monitoring application, it is necessary to map the role of this user to the equivalent role of the Railway Administration. While mapping we always have to consider the corresponding security levels the roles possess. If the user is a Supervisor of the Service Provider and we assume that he should possess the security level similar to what an Engineer of the Railway Administration possesses. The mapping (inside the mapping ontology) was done using *owl:equivalentClass* constructs (described in Table 1) between these two Role Groups. This can be represented as follows:

$SC\left(Supervisor_{OrganizationA}\right) = SC\left(Engineer_{OrganizationB}\right)$, where Role Groups are subclasses of the class $Role$.

Different Security Level: This situation points to the case when different organizations maintain their security level in a different way. In order to process an access request coming from the Express Mail Service Provider, it is required to map the security level of their users to that of the users at the Railway Administration. Referring to the situation when the above rule (Sect. 6.1.2) is applied, we map security level (Level 1) of the Supervisor of the Service Provider to that (Level 4) of an Engineer of the Railway Administration. The mapping was done using *owl:sameAs* constructs between these two Security Level instances. This can be represented as follows:

$\left\{Level1_{OrganizationA}\right\} = \left\{Level4_{OrganizationB}\right\}$, where Security Levels were defined as instances of the ontology.

6.1.4 Rule Execution Environment

In this paper, policies are formal description of constraints. The constraints are represented through a set of rules. The evaluation of rules will derive the decisions. In this regard, the implementation used a rule execution engine. The policies are executed using Jess rule engine [14] and the results represent the decisions. As the rules are built on top of the OWL knowledge base, SWRLJess bridge (a java class) allows the rule engine to interact with the knowledge base and SWRL-SQWRL rules.

6.2 Sensor Integration to M2M Platform

We are still in the early phase of integrating the proposed security reasoning into the prototype which requires significant enhancement of the current M2M platform. This paper only demonstrates sensor integration using a legacy M2M platform. This section provides details of the implementation using SunSPOT sensors being integrated into the Telenor Object's M2M platform.

6.2.1 Classification of Sensor Nodes

While implementing the prototype, we classified different nodes ('Things' in IoT) as nano, micro and personal nodes according to the increasing order of capabilities. Many of such nodes can act like a sensor. An overview of each type of node and a description of specific nodes we used for the prototype are given as follows. Table 2 summarizes the capabilities of the different sensor nodes.

Table 2 Capabilities of different types of sensor nodes

Capability	Nano node	Micro node	Personal node
Hardware	Limited	Limited	Extendable
Processing	No processing power just sense & send	Limited process power (e.g., compare two values)	Considerable processing power (e.g. decision making)
Network access	Through gateway	Through gateway	Direct access
Communication interface	Fixed or wireless	Fixed or wireless	Fixed or wireless or USB-like
Example nodes	GPS sensor	Sun SPOT	Personal computer or Linux embedded system

Nano node: It is a sensor with lowest capabilities. It senses the surrounding environment and sends data to other nodes. It has no processing power. A typical example of nano node is a GPS sensor.
Micro node: In terms of capabilities micro node is more powerful than a nano node. It contains limited processing power to perform some basic tasks such as comparison between two values. This type of sensor nodes are programmable. But communication with micro nodes is only possible through gateways.
In the proposed prototype we used Sun SPOT sensors as micro nodes. The main units are SPOT devices with embedded sensors and base station. Each Sun SPOT has a so-called eSPOT with battery, while the base station is not equipped with battery and must be powered from the host computer via an USB cable. The main hardware components of a SPOT senor platform are as follows:

- 180 MHz for 32-bit ARM920T core processor with 512K RAM and 4M Flash, runs on Squawk Virtual Machine (VM)
- 2,4 GHz based IEEE 802.15.4 radio which is integrated in the antenna
- Integrated Sensors: temperature sensor, accelerometer sensor, light sensor
- I/O pins (analog and digital)
- 3.7 V battery (720 mAh)
- USB interface for connecting to a host computer

Squawk is a highly portable Java VM which can run without an operating system. It allows multiple applications running on the same VM. Squawk supports connectivity with mobile phones.
Personal node: This type of nodes support high processing power and direct communication with the nodes. Such nodes can manipulate complex ontologies and rules. We used Linux-based embedded system as our personal node in the prototype.
Our embedded system is a VIA EPIA N700 Nano-ITX board which is integrated with a VIA VX800 media system processor, an all-in-one chipset solution. The VIA EPIA N700 is equipped with a power-efficient 1.5 GHz VIA C7, supports up to 2 GB of DDR2 system memory and includes two onboard SATA connectors, USB 2.0, COM and Gigabit LAN ports. Expansion includes a Mini-PCI slot with an IDE port, additional COM and USB ports and PS/2 support available through pin-headers. The implementation uses Ubuntu Linux Kernel 2.6.32-24-generic and Java runtime environment (JRE) 1.6 for development

on this embedded system. In the proposed prototype, this system hosts an application facilitates the communication between SPOT sensors and the M2M platform.

6.2.2 M2M Platform

For our implementation, we used Shepherd™, an M2M platform from Telenor Objects, Norway which is an instance of ETSI TS 102 690. It is a platform for interoperability and integration that supports communication between connected devices (e.g.,nano and micro sensors) and makes them accessible from anywhere at anytime. Any pluggable component can be connected in Shepherd platform as a connected object. The platform offers number of services including:

- Service Management for monitoring, device configuration, SLAs, and supporting.
- Service Enabler has a specific API that allows further access to other modules.
- Message Engine handles and secures the process of message flow, including capturing, processing, routing and storage of data in an environment.
- Notification services that inform about the status of devices and applications.
- Device library consists of interfaces for tools and services recognition.

The Shepherd offers two methods for establishing connection:

- HTTP Connection API
- The Connected Objects Operating System (COOS) which is a Java based open source tool.

In our scenario we use the http connection API for access to the information from the railway infrastructure.

6.2.3 Prototyping the Sensor Integration

The devices used in the implementation are shown in Fig. 6. Figure 7 illustrates the system overview of the implemented prototype. It shows the establishment of an intended two-way communication between Sun SPOT sensors and its base station, and also two-way communication between the embedded Linux system (where host application was installed) and the Shepherd Platform. An instance of proposed framework in the form of Host application has been developed, that performs broadcasts every 15 s. While the spot application will detect the broadcasts every 30 s. But it does not transmit the sensory data to the base station after 1 min has passed since the last envoy. When the data arrives, it will also be stored in-memory cache. At the same time the Host application sends out a request to Shepherd for receiving the data. The connection is opened until the application has received confirmation of receipt from the Shepherd. However, the data to be sent to Shepherd only happens in every 5 min.

7 Discussion

This section introduces several other works that are closely related to the concepts and architecture of the proposed Internet of Things framework. However, implementing a complete prototype is quite challenging. In this context, this section discusses some of these challenges.

Fig. 6 The devices used in the implementation

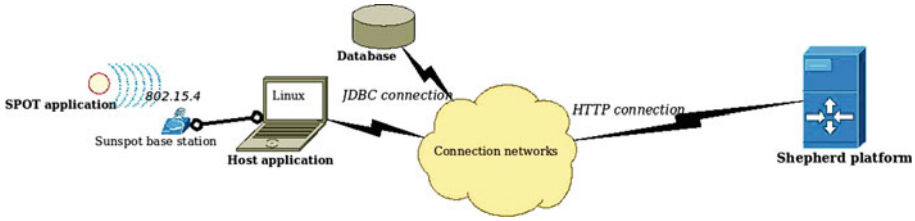

Fig. 7 Integration of sun SPOT sensors with Shepherd platform

7.1 Related Work

IoT is tightly coupled with the sensor technologies because of their sensing, actuating and communication capabilities. The sensor technologies have become pivotal to influence the physical objects in the creation and usage of future services. Besides convenience such as the Nike+iPod iPhone application [15], impact is expected for energy saving, i.e. energy grids, health or security services. Monitoring the home and informing neighbors in case of accidents and alarms might be such an application. Here context- and content-awareness is important. Depending on 'what happened' and 'who is available' will trigger the reasoner to decide who has to be informed. Such a social network based reasoning is subject to further research.

Light weight semantics make the information machine readable that facilitates export of knowledge by software agents and automated machine processing. Nowadays tiny nodes may even have processing capability or memory enough for interpreting light weight semantic models. Light weight semantics would facilitate the information processing to some extent at the IoT could and put little bit less burden on semantic overlay layer (see Fig. 4).

One of the most popular examples of light weight semantics are Microformats [16] that use existing XHTML techniques. RDFa [17] is another notable light weight semantics technology that allow semantic markup to be included within XHTML. RDFa is more powerful than Microformats as it can include powerful expressive ontologies. Ostermaier et al. [18]

proposed a sensor microformats using HTML syntax. Otherwise there is no standard and acceptable format of light weight semantics for sensors.

Scalability of semantic enhancement is real concern considering the sheer size of IoT environment. Here we are talking about not only thousands of sensors but also lot of data from them. The complexity and size of the ontologies affect the execution of rules. When rule needs to evaluate scores of relationships in a big ontology, it may take considerable amount of time to compute any decision. One of our earlier papers [19] analysed this aspect and concluded that deriving decisions using real-time semantic reasoning may take considerable time due to bigger ontology and complex rules.

7.2 Challenges

While ETSI defines through the TS 102 690 the interoperability between operators, and specifically the key exchange to authorize sensors or sensor gateways, the standard lacks the semantic overlay. As pointed out in the implementation section, and defined in Fig. 4, the semantic overlay can act as a mediator between sensor, service, and access layers. The overlay may contain the following entities:

SensorML [7] is aiming at providing a semantic description of the sensor, allowing for specification of the output format and other sensor characteristics.
Sensor Models to describe the "normal" behavior of a sensor, allowing for classifications of non-standard deviations.
Sensor Deployment represent the information on where and how sensors are deployed. Such deployment information will allow for monitoring of the environment, and calculations of dependencies.
Service Access ontologies to describe the content-aware distribution of sensor data, in conjunction with the **Service Notifier**.
Interoperability Information for federation of sensor data, foreseen for privacy or trust-based distribution.

While the semantic representation opens for interoperability, the challenge of reasoning on low-power sensors and devices is not solved. Our suggestion is to work into the direction of light-weight semantics, including both *(i)* light weight descriptions, *(ii)* local reasoning and *(iii)* extractions of rules to be executed on sensors or gateways.

The last area of future research addresses the scalability of semantic enhancements. Mediation of ontologies and distributed reasoning, including trust-based access to confidential sensor, context and personal information, is required for privacy protection of the future IoT.

8 Conclusion

The Internet of Things-enabled services may require integration of multiple administrative domains. One domain may host the devices and enable access to devices and information, whereas another domain may make use of the information for designing innovative services. Each domain contains its own security attributes and constraints. To facilitate secure access to services, devices and data in such integrated operation scenario, ensuring interoperability of security is a challenging task. In order to address this challenge, this paper proposed an architecture and presented implementation details of parts of the architecture involving real devices and platform. As it is a challenging task to provide a complete solution, the paper also identified several challenges and possible future works.

References

1. pilot SHIELD (2011). Pilot embedded Systems arcHItecturE for multi-Layer Dependable solutions [online]. available: http://www.pshield.eu.
2. Tim Berners-Lee, J. H., & Lassila, O. (2001). *The semantic web*. New York: Scientific American Magazine.
3. Walter, K. D. (2009). Implementing M2M applications via GPRS, EDGE and UMTS. White paper—M2M Alliance.
4. ETSI: Ts 102.690 machine-to-machine communications (M2M); M2M functional architecture (Stable draft, June 2011).
5. Bell, M. (2008). *Service-oriented modeling (SOA): Service analysis, design, and architecture*. London: Wiley.
6. Mili, H., Elkharraz, E., & Mcheick, H. (2004). Understanding separation of concerns. *Proceedings of the Workshop on Early Aspects—Aspect Oriented Software Development (AOSD)*.
7. Sensor model language (SensorML). (2005). [online] available: http://www.opengeospatial.org/standards/sensorml.
8. Akerkar, R. & Sajja, P. (2010) *Knowledge-based systems*. Jones & Barlett Publishers: MA, USA.
9. John Davies, R. S., & Warren, P. (2006). *Semantic web technologies: Trends and research in ontology-based systems*. London: Wiley.
10. Michael, K., Smith, C. W., & Mcguinness, D. L. (2004). Owl web ontology language guide (W3C Recommendation, February 2004).
11. The protégé ontology editor. [online] (Available: http://protege.stanford.edu/ [accessed on 01. March 2011]).
12. Horrocks, I., Patel-Schneider, P. F., Boley, H., Tabet, S., Grosof, B., & Dean, M. (2004). SWRL: A semantic web rule language combining OWL and RuleML. W3C Member Submission.
13. OConnor, M., & Das, A. (2009). SQWRL: A query language for OWL. *Proceedings of OWL: Experiences and Directions (OWLED), the fifth International Workshop.*
14. Friedman-Hill, E. Jess, the rule engine for the java platform [online]. available: http://www.jessrules.com/.
15. Nike + running shoes and a nike + ipod sport kit or sensor. [online] (Available: http://www.apple.com/ipod/nike/).
16. Khare, R. (2006). Microformats: The next (small) thing on the semantic web? *Internet Computing, 10*(1), 68–75.
17. Adida, B., Birbeck, M. (2008). RDFa primer: Bridging the human and data webs [online]. available: http://www.w3.org/TR/xhtml-rdfa-primer/.
18. Ostermaier, B., Romer, K., Mattern, F., Fahrmair, M., & Kellerer, W. (2010). A real-time search engine for the Web of Things. *Proceedings of Internet of Things 2010*, pp. 1–8.
19. Chowdhury, M. M. R., Alam, S., & Noll, J. (2009). Secure connected home: Where the semantic technologies meet the device community. *Journal of Information Assurance and Security (JIAS), 4*(5), 390–402.

Author Biographies

Sarfraz Alam is a Ph.D. researcher at UNIK-University Graduate Center in Kjeller, Norway. He received his M.Sc. degree in the area of Information Security from The Royal Institute of Technology (KTH), Sweden. His research area covers Service Oriented Internet of Things, Semantic Services, SOA security and Wireless Sensor Networks.

Mohammad M. R. Chowdhury is currently working as a postdoctoral fellow at UNIK-University Graduate Center, Norway. He received the Ph.D. degree from the Department of Informatics, University of Oslo in 2009 in the area of information and system security. Prior to that he received the M.Sc. degree in Telecommunication Engineering from Helsinki University of Technology (HUT). His current research interests include next generation networks and services, access control, identity management, personalized service access, and Internet of Things. Dr. Chowdhury has been involved in research projects funded by the Norwegian Research Council and the European Union. He has published more than 35 scientific articles in journals, books and international academic conferences. He contributed in several international conferences as technical program committee member, session chair and reviewer.

Josef Noll is professor at the University of Oslo in the area of Mobile Services. His group ConnectedLife concentrates on the working areas mobile-based trust and authentication, personalised and context-aware service provisioning, and service continuation in 5G systems. He is also Chief Technologist in Movation, Norway's open innovation company for mobile services. He is co-founder and steering board member of the Center for Wireless Innovation Norway and Mobile Monday Norway, the Norway section of the worldwide community for nerds and professionals in mobile services. Previously he was Senior Advisor at Telenor R&I in the Products and Markets group, and project leader of Eurescom's 'Broadband services in the Intelligent Home' and use-case leader in the EU FP6 'Adaptive Services Grid (ASG)' projects, and has initiated a.o. the EU's 6th FP ePerSpace and several Eurescom projects. In 2008 he received the IARIA fellow award. He is editorial board member of four International Journals, as well as reviewer and evaluator for several national and European projects and programs.